OUKA
The University of Osaka
Institutional Knowledge Archive

Title	Tensor Network Methods for Quantum Computer Design: Simulation, Compilation, and Dequantization
Author(s)	真鍋, 秀隆
Citation	大阪大学, 2026, 博士論文
Version Type	VoR
URL	https://doi.org/10.18910/105535
rights	
Note	

The University of Osaka Institutional Knowledge Archive : OUKA

https://ir.library.osaka-u.ac.jp/

The University of Osaka

Tensor Network Methods for Quantum Computer Design: Simulation, Compilation, and Dequantization

HIDETAKA MANABE

MARCH 2026

Tensor Network Methods for Quantum Computer Design: Simulation, Compilation, and Dequantization

A dissertation submitted to
THE GRADUATE SCHOOL OF ENGINEERING SCIENCE
The University of Osaka
in partial fulfillment of the requirements for the degree of
DOCTOR OF PHILOSOPHY IN ENGINEERING

BY

HIDETAKA MANABE

MARCH 2026

Abstract

Quantum computers have attracted significant attention for their potential to perform information processing tasks that are intractable on classical computers. In recent years, substantial progress has been made in the development of quantum computers, and devices with hundreds of qubits have been experimentally realized under various architectures. However, the development of practically useful quantum computers requires far more than simply increasing the number of qubits. It is essential to reduce resource overhead through software-level approaches across multiple layers of the quantum computing stack, including architecture design, circuit optimization, and quantum algorithm development.

A fundamental challenge in these layers is that quantum computation is defined on large-scale quantum many-body systems. Analyzing the behavior of such systems is inherently difficult, both analytically and via classical simulation. Consequently, it is often unclear whether proposed architectures and algorithms will perform efficiently in practice, which poses a significant obstacle to the analysis and design of a practical quantum computer.

In this thesis, we address this challenge by employing tensor networks. A tensor network is a mathematical framework for representing a high-rank tensor as a network of low-rank tensors, thereby reducing memory and computational cost. Tensor network methods provide both strong theoretical foundations and powerful numerical tools. For example, the ground states of certain quantum many-body systems are proved to be efficiently represented by tensor network states and can be obtained numerically using variational optimization, which is one of the most successful numerical methods in condensed matter physics. Because of their broad applicability, tensor networks are also used in a wide range of fields, including high-energy physics and data science.

Since quantum algorithms and architectures are defined on quantum many-body systems, tensor network techniques can be naturally applied to their analysis. By representing quantum states as tensor network states and leveraging numerical methods developed in computational physics, we can significantly reduce the classical simulation cost required to analyze quantum computations for systems with low entanglement. This enables a detailed investigation of computational behavior and provides concrete guidance for improving their design.

Crucially, tensor networks offer more than a tool for state representation. The elementary operations of quantum computation are all local linear maps and can therefore be expressed as local tensors. From this perspective, quantum

algorithms themselves can be viewed as tensor networks. This observation leads to a new paradigm in which tensor networks serve not merely as numerical tools for simulation, but as a language for describing quantum algorithms.

Building on these perspectives, this thesis explores the use of tensor network methods for analyzing and designing quantum architecture and algorithms across multiple layers of the quantum computing stack. We apply tensor network methods to the analysis of quantum error-correcting codes, circuit compilation, and the assessment of quantum advantage via dequantization.

First, we investigated the performance of quantum error correcting codes under realistic coherent leakage noise. By employing matrix product states, one type of tensor network ansatz, we numerically executed efficient classical simulation for the repetition code and thin surface code for systems consisting of hundreds of qubits and gates, which are intractable for state-vector simulations. These results highlight that tensor network-based circuit simulation provides an indispensable framework for the theoretical analysis of quantum error correction, including the comparison of leakage-removal strategies.

Second, we proposed a scalable compilation method for multivariate normal distribution state preparation circuits based on tensor networks, in particular tree tensor networks. By analyzing the structure of multivariate normal distributions, we provided theoretical guarantees for distributions whose correlations decay exponentially in one dimension. For more general distributions, we introduced a heuristic circuit optimization method that utilizes state-of-the-art tensor network numerical techniques such as tensor cross interpolation and automatic structure optimization. We numerically verified that this approach achieves circuit cost reductions by factors of 10 to 1000 for several benchmark models.

Finally, we developed a tensor network formulation of dequantized algorithms for ground state energy estimation. We constructed a practical dequantized algorithm that is executable within realistic computational resources by representing Chebyshev vectors as tensor network states and consequently reflects the physical structure, such as entanglement and locality, in the computational cost. This formulation not only enables the execution of dequantized algorithms with theoretical guarantees, but also makes it possible to quantitatively investigate the boundary between classical and quantum computational regimes through numerical experiments.

Throughout these studies, we adopt tensor networks not merely as a tool for simulating quantum systems, but as a framework for describing and designing quantum computation at different layers. By doing so, it becomes possible to analyze and design quantum architectures, circuits, and algorithms by combining the theoretical guarantees of tensor networks with their powerful numerical techniques. We demonstrate that tensor networks constitute an essential tool for the analysis and design of quantum architectures and algorithms toward the practical realization of quantum computers.

Contents

1 Introduction **1**

1.1 Motivation: Software-level approaches for practical quantum computing . . . 1
1.2 Tensor networks . . . 3
1.3 Tensor networks and quantum computation . . . 4
1.4 Goal of this thesis . . . 5
1.5 Overview of this thesis . . . 7

2 Preliminaries **9**

2.1 Quantum Computation . . . 9
2.1.1 Quantum states . . . 9
2.1.2 Quantum dynamics . . . 10
2.1.3 Quantum measurement . . . 10
2.2 Tensor Networks . . . 11
2.2.1 Diagrammatic notation . . . 11
2.2.2 Matrix Product States . . . 12
2.2.3 Canonical form of Matrix Product States . . . 13
2.2.4 Bond truncation via canonical form . . . 13
2.2.5 Entanglement and Matrix Product States . . . 14
2.2.6 From tensor networks to quantum circuits . . . 15
2.2.7 Tree tensor networks . . . 15
2.2.8 Quantum circuit simulation with Matrix Product States . . . 16

3 Tensor Networks for the Simulation of QEC Codes under Leakage Errors **19**

3.1 Introduction . . . 19
3.2 Leakage noise models . . . 21
3.2.1 Single-qubit gate errors . . . 22
3.2.2 Controlled-phase gate errors . . . 22
3.2.3 Measurement . . . 23
3.2.4 Idling noise . . . 24
3.3 Tensor network methods . . . 24
3.3.1 Matrix product states . . . 25
3.3.2 Quantum circuit simulation with MPS . . . 26

3.4 Error-correction simulations . . . 27
3.4.1 Error correcting codes . . . 27
3.4.2 Leakage removal strategies . . . 29
3.5 Results . . . 31
3.5.1 Bond dimension . . . 31
3.5.2 Exact vs twirling approximation . . . 33
3.5.3 Comparison of leakage processes and removal strategies . 36
3.6 Discussion . . . 39

4 Tensor Networks for the Compilation of State Preparation Circuits of Multivariate Normal Distributions 41
4.1 Introduction . . . 42
4.2 Tensor network methods . . . 43
4.2.1 Matrix product states and tree tensor networks . . . 43
4.2.2 Functional matrix product states and tree tensor network 44
4.2.3 Tensor cross interpolation . . . 45
4.2.4 Canonical form of tree tensor network . . . 45
4.3 Methods . . . 46
4.3.1 Fourier series approximation . . . 47
4.3.2 Representing Fourier coefficients using tree tensor network 48
4.3.3 Automatic structural optimization of the TTN . . . 55
4.3.4 Optimizing the quantum circuit by tree tensor network . 56
4.4 Results . . . 59
4.4.1 Multivariate normal distribution with low-rank structure 59
4.4.2 Multivariate normal distribution with exponentially decaying correlation coefficients . . . 60
4.4.3 Structural optimization with 1D tree structure . . . 61
4.4.4 State preparation of random normal distribution . . . 62
4.5 Discussion . . . 64

5 Tensor Network Formulation of Dequantized Algorithms for Ground State Energy Estimation 67
5.1 Introduction . . . 68
5.2 Preliminaries . . . 70
5.2.1 Tensor network . . . 70
5.2.2 Eigenvalue filtering . . . 72
5.3 Dequantized algorithm with tensor networks . . . 79
5.3.1 Additional assumptions on the problem . . . 79
5.3.2 Chebyshev polynomial expansion . . . 81
5.3.3 Complexity analysis via tensor networks . . . 82
5.4 Practical dequantized algorithm with tensor network approximation 84
5.4.1 Approximated Chebyshev vectors . . . 85
5.4.2 Approximation method: MPS and linear prediction . . . 86
5.5 Results . . . 89
5.5.1 Problem settings . . . 90
5.5.2 Numerical results . . . 92

5.6 Discussion 98

6 Conclusion **101**

Acknowledgements **107**

Abbreviations

CP	Completely Positive
CPTP	Completely Positive and Trace Preserving
FSL	Fourier Series Loader
FTQC	Fault-Tolerant Quantum Computer
GSEE	Ground State Energy Estimation
GTA	General Twirling Approximation
MLR	Multi-Level Reset
MPO	Matrix Product Operator
MPS	Matrix Product State
NISQ	Noisy Intermediate-Scale Quantum device
PEPS	Projected Entangled Pair States
QEC	Quantum Error Correction
QECC	Quantum Error Correcting Codes
QPE	Quantum Phase Estimation
QSVT	Quantum Singular Value Transformation
SVD	Singular Value Decomposition
TCI	Tensor Cross Interpolation
TFIM	Transverse-Field Ising Model
TTN	Tree Tensor Network

Chapter 1

Introduction

1.1 Motivation: Software-level approaches for practical quantum computing

Quantum computers have attracted significant attention due to their potential to perform information processing tasks that are intractable on classical computers. One representative example is factorization, which underpins public-key cryptography such as RSA [1] and can be solved exponentially faster using quantum computers than classical approaches [2]. Another important example is the simulation of quantum many-body systems [3, 4], which is a central problem in quantum chemistry and materials science [5–7]. In recent years, remarkable experimental progress has been achieved, and quantum devices with hundreds of qubits have been realized across a variety of hardware platforms, such as superconducting circuits [8–13], neutral atoms [14–16] and trapped ions [17–19].

To execute practically important quantum algorithms, however, quantum computers are expected to require tens of thousands to millions of physical qubits. At such scales, the challenge can no longer be understood simply as increasing the number of physical qubits. Rather, as in classical computing, it is essential to view a quantum computer through the abstraction of a quantum computing stack [20]. We illustrate an overview of the quantum computing stack in Fig. 1.1. A typical stack consists of quantum applications, quantum algorithms, quantum circuits, quantum error correction and physical qubits. Each layer represents a distinct level of abstraction, in which the underlying mechanism of the lower layer is hidden behind an interface. At the lowest level, the physical layer is responsible for controlling physical qubits. Primitive operations such as 1-qubit and 2-qubit gates and measurements are included in this layer. Building on these imperfect operations, the quantum error correction (QEC) layer encodes logical qubits and logical gates by applying QEC codes. Assuming the availability of reliable logical operations, the quantum circuit layer constructs quantum circuits as sequences of logical gates acting on logical qubits. On top of this, the quantum algorithm layer provides an abstract description

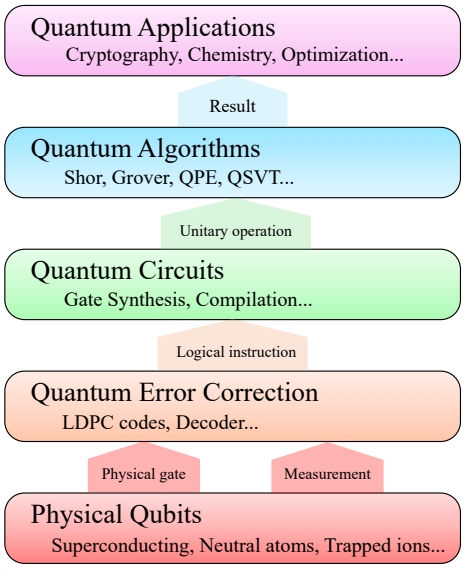


Figure 1.1: An overview of the quantum computing stack. The layers illustrate the abstraction from physical hardware to end-user applications. The vertical arrows represent the interfaces between the layers. The layer definitions constitute a modification of the architecture proposed in Ref. [20].

of quantum algorithms and produces outputs. Finally, the application layer interfaces with end users by specifying problem instances and interpreting the results returned by quantum algorithms. Such abstraction enables each layer to be analyzed and optimized independently, without requiring knowledge of the internal mechanism of the lower layers.

By optimizing the techniques and algorithms employed at each layer, the total resource requirements of the overall system can be systematically reduced. In particular, *software-level approaches* play a crucial role in realizing quantum computation. In this thesis, we refer to software-level approaches as methods that improve the performance and feasibility of quantum computation not by direct hardware improvements, such as increasing the number of qubits or improving gate fidelities, but by leveraging classical computation through algorithmic design, optimization and theoretical analysis. At the level of quantum algorithms, careful algorithmic design and analysis can dramatically reduce resource requirements. For example, a recent study of Shor's factoring algorithm has shown that the required number of qubits and gates can be reduced by several orders of magnitude [21, 22]. Similarly, in the context of ground state energy estimation, algorithmic advances based on quantum singular value transformation (QSVT) have significantly lowered the cost of achieving a target precision [23, 24]. At the quantum circuit level, software-based optimization plays

an important role. In particular, circuit compilation and optimization techniques are crucial for reducing gate counts and circuit depth [25–28]. Recent approaches based on decomposition, graph theory and machine learning have demonstrated that these highly efficient compilation methods can dramatically reduce the circuit cost [29–32]. At the level of quantum error correction, software and theoretical advances are among the most active and critical research frontiers in quantum computing. By designing quantum error correcting codes with higher code rates, the number of required physical qubits can be significantly decreased [33–35]. At the same time, the development of accurate and fast decoding algorithms is essential [36].

These examples demonstrate that software-level optimization across multiple layers of the quantum computing stack is indispensable for realizing practical quantum computation. However, such software-level improvements share a fundamental and common difficulty: all of these techniques are defined on large-scale quantum many-body systems. In general, the classical description of quantum many-body systems requires computational resources that scale exponentially with system size. As a consequence, analyzing or simulating the behavior of quantum software based on large-scale quantum many-body systems is intrinsically difficult, both analytically and numerically. For example, when simulating the dynamics of an n-qubit quantum state without any special structure, brute-force classical simulation is typically limited to around 30 qubits on a standard laptop PC, and to roughly 50 qubits even on state-of-the-art supercomputers [37]. In contrast, quantum algorithms expected to demonstrate quantum advantage necessarily operate on significantly larger numbers of qubits. As a result, understanding the behavior of quantum algorithms only at small system sizes is often insufficient to reliably predict their performance at the scales relevant for practical quantum computation. This scale gap makes it challenging to assess, *a priori*, how software-level optimizations will perform when deployed on realistic, large-scale quantum devices.

1.2 Tensor networks

To overcome the intrinsic difficulty associated with large-scale quantum many-body systems, this thesis adopts tensor networks [38–42] as its central analytical and computational framework. A tensor network is a framework for representing a high-dimensional tensor as a network of smaller tensors. An example of a tensor network is shown in Fig. 1.2. In a tensor network diagram, each node represents a tensor, and each edge represents an index of the tensor. Connecting two edges corresponds to summing over the associated indices, an operation known as tensor contraction. Through successive contractions, the network represents a single high-rank tensor.

Tensor networks have achieved remarkable success in the field of many-body physics. They provide an efficient representation for quantum states with low entanglement, which frequently arise in physically relevant systems. Using tensor networks, ground states of one-dimensional gapped systems can be represented

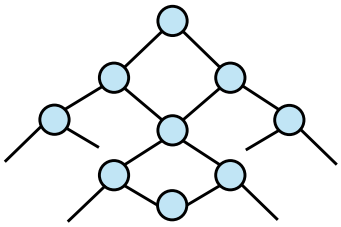


Figure 1.2: An example of a tensor network diagram. Each node represents a tensor, and each edge represents an index of the tensor. Detailed explanations are given in the preliminary section.

efficiently and computed accurately with numerical methods [39, 43, 44]. These approaches have been applied to both finite and infinite systems [45], as well as to higher-dimensional settings [46]. Beyond ground-state calculations, tensor network methods have been successfully applied to a wide range of problems, including excited states [47], spectral properties [48], and real-time quantum dynamics [49, 50]. Taken together, tensor networks are regarded as one of the state-of-the-art techniques for quantum many-body physics that can circumvent fundamental obstacles of the sign problem encountered in quantum Monte Carlo methods.

Beyond condensed matter physics, tensor networks have been utilized in a broad range of scientific fields. In quantum information theory, tensor networks provide a powerful framework for characterizing quantum phases of matter. In particular, the symmetries of local tensors enable a complete classification of one-dimensional gapped quantum systems [51–53]. In high-energy physics, tensor network ansatz has been employed as a model of the AdS/CFT correspondence [54–56]. Tensor networks offer an intuitive description of emergent spacetime geometry, and have led to deep connections between quantum error correction, holography, and black hole. More recently, tensor network techniques have also been explored in data science and machine learning [57, 58]. Tensor networks provide a systematic and interpretable framework for the compression of big data, allowing more efficient information processing.

In this way, tensor networks provide an efficient framework for compressing and representing high-dimensional tensor data by exploiting entanglement and correlation structures. Tensor networks combine rigorous theoretical guarantees with powerful numerical techniques, which has enabled their successful application across a wide range of scientific fields.

1.3 Tensor networks and quantum computation

Tensor network methods may provide a way to partially overcome the intrinsic difficulty discussed above. By representing quantum many-body states using

tensor networks, one can exploit the structure of entanglement and correlations in the system and execute more efficient classical simulations than with brute-force state-vector methods. As a result, the behavior of quantum algorithms and architectures can be investigated at system sizes well beyond tens of qubits, providing concrete insights into algorithmic performance.

However, the relationship between tensor networks and quantum computation goes beyond the representation of quantum states. Since the elementary operations of quantum computation, such as unitary gates and measurements, are linear maps, an entire quantum circuit can itself be regarded as a tensor network [59]. From this viewpoint, executing a quantum circuit corresponds to contracting the associated tensor network. A representative example is the analysis of quantum computational supremacy. In 2019, Google announced that its 53-qubit superconducting processor had achieved quantum computational supremacy by performing a random circuit sampling task in 200 seconds, which was estimated to require about 10,000 years on a classical computer [12]. However, it was later shown that this claim was undermined by a classical simulation based on tensor network methods, which reproduced the same task in 304 seconds on a supercomputer [60]. This result had a large impact as it showed that the classical simulation cost of the task had been significantly overestimated, and it was awarded the Gordon Bell Prize in 2021. In their work, the authors drastically reduced the simulation time by representing the entire quantum circuit as a tensor network and heuristically optimizing the contraction order of the network.

This line of research further highlighted the deep connection between quantum computation and tensor networks. In particular, it became widely recognized that tensor networks are not merely tools for representing quantum states, but can also serve as a framework for describing quantum circuits and quantum algorithms. As a consequence, many works in recent years have applied tensor network methods to quantum computation. Tensor networks are currently widely used as one of the most efficient methods for simulating intermediate-scale quantum circuits [61–65]. Furthermore, the diagrammatic and algebraic properties of tensor networks are utilized to design and optimize quantum circuits [66, 67]. In the context of quantum error correction, tensor networks provide a framework for constructing decoders that achieve near-optimal performance [13, 68, 69].

1.4 Goal of this thesis

The goal of this thesis is to demonstrate that tensor network methods can be utilized as a general and practical foundation for quantum computer design. Designing quantum computers requires the optimization of a wide range of algorithms, software, and protocols across the quantum computing stack. However, as discussed in the previous section, this task is accompanied by intrinsic computational difficulties arising from the fact that these algorithms are defined on quantum many-body systems. In this thesis, we address these challenges

by combining the theoretical framework of tensor networks with their powerful numerical techniques.

More concretely, we apply tensor network methods to several important problems that arise at different layers of the quantum computing stack, including the quantum error correction layer, the quantum circuit layer, and the quantum algorithm layer.

Performance analysis of quantum error correcting codes. At the level of quantum error correction, a central problem is to evaluate how well a QEC code can correct errors when realistic noise acts on physical qubits. Unless the noise model or gate set belongs to a special class, such analyses generally require computational resources that scale exponentially with the number of qubits. By employing tensor network-based simulation methods, we aim to overcome this computational barrier and numerically investigate the concrete performance of QEC codes in larger-scale qubit systems. This approach enables the systematic exploration of more efficient code constructions and the comparison of different noise-removal strategies.

Quantum circuit compilation. At the quantum circuit layer, one of the most important tasks is to compile a circuit that realizes a desired task into a more resource-efficient form. We represent target tasks using tensor networks and translate them into quantum circuits. This allows us to propose scalable and high-performance compilation methods with respect to the number of qubits.

Analysis of quantum advantage. At the level of quantum algorithms, it is crucial to identify which tasks can genuinely achieve quantum advantage, as this directly impacts the practical usefulness of future quantum computers. By using tensor networks as a language for describing quantum algorithms, we construct corresponding dequantized classical algorithms in a systematic manner. This enables a more natural and intuitive comparison between quantum and classical computational complexity.

Taken together, the central contribution of this thesis is to establish tensor networks as a unifying analytical and computational framework across multiple layers of the quantum computing stack. Although the problems addressed arise from fundamentally different settings and objectives, we show that they can all be formulated and analyzed within the same tensor network formalism. Crucially, the theoretical structure of tensor networks directly translates into rigorous algorithmic guarantees and scalability analyses, while their powerful numerical capabilities enable the treatment of problem sizes ranging from tens to hundreds of qubits. As a consequence, tensor network methods do not merely offer case-by-case improvements, but instead provide a coherent and essential framework for the design, analysis, and validation of future quantum computers.

1.5 Overview of this thesis

The rest of this thesis is organized as follows.

Chapter 2: Preliminaries
Chapter 2 presents the preliminaries required for this thesis. We introduce the basic concepts of quantum computation and review tensor networks. In particular, we summarize matrix product states (MPS), tree tensor networks (TTN), and numerical techniques that form the methodological foundation of the subsequent chapters.

Chapter 3: Tensor Networks for the Simulation of QEC Codes under Leakage Errors
Chapter 3 focuses on the classical simulation and analysis of QEC codes under realistic noise models. Using MPS, we investigate the efficient simulation of repetition codes and thin surface codes in the presence of coherent leakage noise. This chapter demonstrates that tensor network-based simulation enables the quantitative analysis of QEC codes beyond idealized Pauli noise assumptions and provides a scalable framework for benchmarking QEC circuits.

Chapter 4: Tensor Networks for the Compilation of State Preparation Circuits of Multivariate Normal Distributions
Chapter 4 addresses the problem of quantum circuit compilation for multivariate normal distribution state preparation. We propose a scalable compilation method based on TTN and analyze the structure of target quantum states from the perspective of correlations. By combining theoretical insights with advanced tensor network numerical techniques, we propose an efficient and high-performance compilation method and achieve substantial reductions in circuit cost for a variety of benchmark models.

Chapter 5: Tensor Network Formulation of Dequantized Algorithms for Ground State Energy Estimation
Chapter 5 develops a tensor network formulation of dequantized algorithms for ground state energy estimation. By representing Chebyshev vectors as tensor network states, we construct a practical dequantized algorithm that can be executed within realistic classical computational resources. This formulation enables a quantitative investigation of the boundary between classical and quantum computational regimes and provides a concrete tool for verifying quantum advantage through numerical experiments.

Chapter 6: Conclusion
Finally, Chapter 6 summarizes the results of this thesis and discusses possible directions for future research.

Chapter 2

Preliminaries

This chapter introduces the basic concepts and tools required for the remainder of this thesis. First, we review the fundamentals of quantum computation, including quantum states, quantum dynamics, and quantum measurements. Next, we introduce tensor networks, with a focus on Matrix Product States (MPS). We discuss how MPS naturally capture entanglement structure and how they can be used to simulate quantum circuits.

2.1 Quantum Computation

In this section, we briefly review the fundamental framework of quantum computation.

2.1.1 Quantum states

In quantum computation, the fundamental unit of information is the qubit. The state of a single qubit is described by a unit vector in a two-dimensional complex Hilbert space $\mathbb{C}^2$. We define the computational basis $\{|0\rangle, |1\rangle\}$ as the orthonormal basis of this space. A general pure state of a qubit is written as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle , \tag{2.1}$$

where $\alpha, \beta \in \mathbb{C}$ and $|\alpha|^2 + |\beta|^2 = 1$.

More generally, the state of an n-qubit system is described in the Hilbert space

$$\mathcal{H} = (\mathbb{C}^2)^{\otimes n}. \tag{2.2}$$

A pure state in this space can be expressed as

$$|\psi\rangle = \sum_{s_1,\ldots,s_n=0,1} c_{s_1\ldots s_n} |s_1 \ldots s_n\rangle . \tag{2.3}$$

Here, $s_i \in \{0, 1\}$ represents the state of the i-th qubit, and $|s_1 \dots s_n\rangle$ denotes the computational basis state of the n-qubit system. The complex coefficients $c_{s_1 \dots s_n}$ satisfy the normalization condition $\sum_{s_1,\dots,s_n} |c_{s_1 \dots s_n}|^2 = 1$.

So far, we have formulated quantum states in terms of qubits, which serve as the standard unit of quantum information. However, in certain physical implementations and theoretical settings, it is convenient to consider quantum systems with more than two internal levels. In such cases, quantum information is naturally described using d-level systems, referred to as qudits (or qutrits for $d = 3$). The local Hilbert space is given by $\mathbb{C}^d$, spanned by the orthonormal basis $\{|0\rangle, |1\rangle, \dots, |d-1\rangle\}$. For a system of n qudits, the total Hilbert space becomes $(\mathbb{C}^d)^{\otimes n}$, and the indices s_i run from 0 to $d-1$.

While pure states provide a complete description of isolated quantum systems, it is often convenient to use a more general representation known as the density operator. A density operator ρ describes a statistical ensemble of pure states $\{p_k, |\psi_k\rangle\}$ and is defined as

$$\rho = \sum_k p_k |\psi_k\rangle \langle\psi_k|, \qquad \sum_k p_k = 1. \tag{2.4}$$

The density operator satisfies $\rho \geq 0$ and $\mathrm{Tr}(\rho) = 1$. A quantum state is pure if and only if $\rho^2 = \rho$; otherwise, it is called a mixed state.

2.1.2 Quantum dynamics

The time evolution of a closed quantum system is described by a unitary operator. If a system is initially in a state ρ, its evolution under a unitary U is given by

$$\rho \mapsto U\rho U^\dagger. \tag{2.5}$$

In the circuit model of quantum computation, such unitary evolutions correspond to quantum gates.

Quantum systems often interact with their environment. The most general physical evolution of a quantum state is described by a quantum channel, which is a completely positive and trace-preserving (CPTP) linear map $\mathcal{E}$ acting on density operators:

$$\rho \mapsto \mathcal{E}(\rho). \tag{2.6}$$

Any quantum channel admits a Kraus representation

$$\mathcal{E}(\rho) = \sum_i K_i \rho K_i^\dagger, \qquad \sum_i K_i^\dagger K_i = I, \tag{2.7}$$

where $\{K_i\}$ are called Kraus operators. This representation is particularly useful for describing noise processes.

2.1.3 Quantum measurement

In general, quantum measurement processes are described by completely positive (CP) instruments. A CP instrument is a collection of completely positive

maps

$$\{\mathcal{M}_m\}_m,$$

where m labels the possible measurement outcomes. Each map $\mathcal{M}_m$ represents the state update associated with outcome m.

For an input quantum state ρ, the probability of obtaining outcome m is given by

$$p(m) = \mathrm{Tr}[\mathcal{M}_m(\rho)]\,, \tag{2.8}$$

and the corresponding post-measurement state is

$$\rho_m = \frac{\mathcal{M}_m(\rho)}{\mathrm{Tr}[\mathcal{M}_m(\rho)]}. \tag{2.9}$$

The overall measurement process is required to be trace-preserving. Therefore, $\sum_m \mathcal{M}_m$ must be a CPTP map.

2.2 Tensor Networks

Tensor networks provide a powerful framework for representing and manipulating high-rank tensors by decomposing them into networks of low-rank tensors. They have been extensively developed in condensed matter physics as efficient representations of quantum many-body states and operators, and are particularly well-suited for describing systems with limited entanglement. In this thesis, tensor networks serve as the central tool for analyzing, simulating, and designing quantum algorithms and architectures across multiple layers of the quantum computing stack.

2.2.1 Diagrammatic notation

A tensor is a multi-dimensional array of complex numbers. For example, a vector v_i is a rank-1 tensor, a matrix M_{ij} is a rank-2 tensor, and a general rank-k tensor is written as $T_{i_1 i_2 \ldots i_k}$. The total number of parameters in such a tensor scales exponentially with its rank, which makes direct manipulation infeasible for large quantum systems.

Tensor networks address this issue by representing a large tensor as a network of smaller tensors. Some examples of tensor networks are shown in Fig. 2.1. Each tensor is represented as a node, and each index is represented as an edge. An index shared by two tensors corresponds to a contraction over the corresponding index, while an open edge corresponds to an uncontracted (external) index.

In diagrammatic notation, a rank-k tensor is depicted as a node with k legs. For example, a matrix M_{ij} is represented as a node with two legs, and matrix multiplication corresponds to connecting the appropriate legs and contracting the connected indices. This graphical representation provides an intuitive and algebraically precise way to describe complex tensor contractions.

Given a tensor network, the value of the represented tensor is obtained by contracting all internal indices. The computational cost of this contraction depends strongly on the network structure and contraction order.

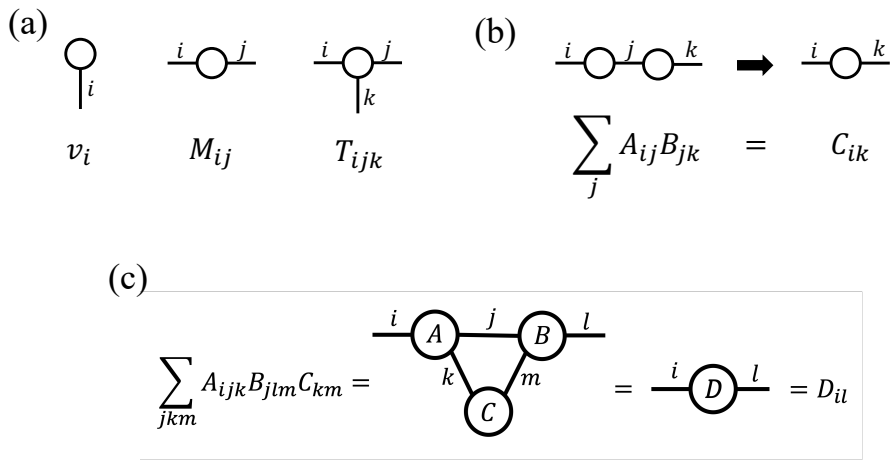


Figure 2.1: Examples of tensor network notation. (a) A vector v_i, a matrix M_{ij}, and a tensor T_{ijl} are expressed as a node with edges. (b) A fusion of two nodes corresponds to summing over the index. (c) Contracting all internal indices yields the result.

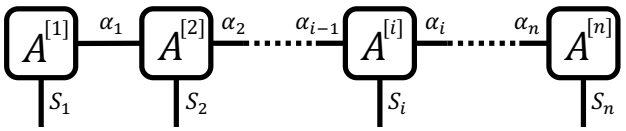


Figure 2.2: A diagrammatic representation of an MPS.

2.2.2 Matrix Product States

One of the most important tensor network ansatz is the Matrix Product State (MPS), which provides an efficient representation of one-dimensional quantum many-body states. Consider an n-qubit pure state

$$|\psi\rangle = \sum_{s_1,\ldots,s_n=0,1} c_{s_1\ldots s_n} |s_1 \ldots s_n\rangle . \tag{2.10}$$

In general, the coefficient tensor $c_{s_1\ldots s_n}$ contains 2^n complex parameters.

An MPS representation factorizes this tensor as

$$|\psi\rangle = \sum_{\{s_i\}} \sum_{\{\alpha_i\}} \left[A^{[1]s_1}_{\alpha_1} A^{[2]s_2}_{\alpha_1\alpha_2} \cdots A^{[n]s_n}_{\alpha_{n-1}} \right] |s_1 \cdots s_n\rangle , \tag{2.11}$$

where $A^{[k]}$ are local tensors associated with site k, and α_k are auxiliary indices called bond indices. The dimension of the bond index α_k is referred to as the

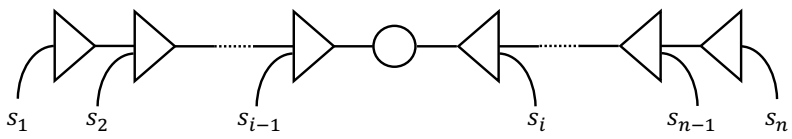


Figure 2.3: A canonical form of an MPS. A triangle tensor represents an isometry, which is a necessary condition for left-canonical and right-canonical forms. A center is chosen between site $i-1$ and i.

bond dimension χ_k. We illustrate a tensor network notation of an MPS in Fig. 2.2.

If all bond dimensions are bounded by a constant χ, the total number of parameters in the MPS scales as $O(n\chi^2)$, which is polynomial in the system size. This makes MPS an efficient representation for a large class of physically relevant states.

By analogy, this tensor network decomposition can be extended to represent operators, known as the Matrix Product Operator (MPO). An MPO consists of local tensors $A^{[k]}$ with two physical indices s_k and s'_k, corresponding to the output and input indices, respectively. An operator O is expressed as

$$O = \sum_{\{s_i\}} \sum_{\{s'_i\}} \sum_{\{\alpha_i\}} \left[A^{[1]s_1 s'_1}_{\alpha_1} A^{[2]s_2 s'_2}_{\alpha_1 \alpha_2} \cdots A^{[n]s_n s'_n}_{\alpha_{n-1}} \right] |s_1 \dots s_n\rangle \langle s'_1 \dots s'_n| . \tag{2.12}$$

2.2.3 Canonical form of Matrix Product States

An important property of MPS is the existence of canonical forms, which greatly simplify both theoretical analysis and numerical algorithms. An MPS is said to be in left-canonical form if the tensors satisfy

$$\sum_{s_k, \alpha_{k-1}} \left(A^{[k]s_k}_{\alpha_{k-1}\alpha_k} \right)^\dagger A^{[k]s_k}_{\alpha_{k-1}\alpha'_k} = \delta_{\alpha_k \alpha'_k}. \tag{2.13}$$

Similarly, an MPS is in right-canonical form if

$$\sum_{s_k, \alpha_k} A^{[k]s_k}_{\alpha_{k-1}\alpha_k} \left(A^{[k]s_k}_{\alpha'_{k-1}\alpha_k} \right)^\dagger = \delta_{\alpha_{k-1} \alpha'_{k-1}}. \tag{2.14}$$

An MPS can be brought into a mixed-canonical form, where tensors to the left of a chosen site are left-canonical, and tensors to the right are right-canonical. We illustrate a mixed canonical form of an MPS in Fig. 2.3. This form plays a crucial role in entanglement analysis and bond truncation. In this thesis, canonical form denotes the mixed canonical form.

2.2.4 Bond truncation via canonical form

The canonical form of an MPS provides a systematic and efficient way to truncate bond dimensions while controlling the approximation error. Consider an

MPS in mixed canonical form with respect to a bipartition between sites $1, \ldots, k$ and $k+1, \ldots, n$. In this case, the state can be written as

$$|\psi\rangle = \sum_{\alpha=1}^{\chi_k} \lambda_\alpha^{[k]} \, |\phi_\alpha^{[1:k]}\rangle \otimes |\phi_\alpha^{[k+1:n]}\rangle \,, \tag{2.15}$$

where $\{\lambda_\alpha^{[k]}\}$ are the Schmidt coefficients across the cut, and $\{|\phi_\alpha^{[1:k]}\rangle\}$ and $\{|\phi_\alpha^{[k+1:n]}\rangle\}$ form orthonormal sets. This expression is exactly the Schmidt decomposition of the quantum state with respect to the bipartition.

Bond truncation is performed by keeping only the largest χ_{trunc} Schmidt coefficients and discarding the rest. The truncated state is given by

$$|\psi_{\text{trunc}}\rangle = \sum_{\alpha=1}^{\chi_{\text{trunc}}} \lambda_\alpha^{[k]} \, |\phi_\alpha^{[1:k]}\rangle \otimes |\phi_\alpha^{[k+1:n]}\rangle \,. \tag{2.16}$$

The truncation error is quantified by the discarded weight

$$\epsilon_{\text{trunc}} = \sum_{\alpha > \chi_{\text{trunc}}} (\lambda_\alpha^{[k]})^2. \tag{2.17}$$

This procedure yields the optimal low-rank approximation in the sense of minimizing the 2-norm distance between the original and truncated states.

2.2.5 Entanglement and Matrix Product States

The efficiency of MPS representations is closely tied to entanglement properties of quantum many-body states. For a bipartition at bond k, the entanglement entropy is defined as

$$S_k = -\sum_\alpha (\lambda_\alpha^{[k]})^2 \log\left(\lambda_\alpha^{[k]}\right)^2, \tag{2.18}$$

where $\{\lambda_\alpha^{[k]}\}$ are the Schmidt coefficients. The entanglement entropy quantifies the amount of quantum correlations between the two subsystems.

If the bond dimension of an MPS is χ_k, the entanglement entropy across any bond is upper bounded by:

$$S_k \leq \log \chi_k. \tag{2.19}$$

This implies that MPS with a constant bond dimension can represent only quantum states whose entanglement is strongly constrained. Typical examples include ground states of one-dimensional gapped Hamiltonians and classical data with weak correlations. Conversely, when a quantum many-body system is in a low-entanglement regime, an MPS representation provides an efficient and accurate description.

2.2.6 From tensor networks to quantum circuits

As discussed in the introduction, tensor networks can be interpreted as a representation of quantum circuits. Each tensor can be viewed as a local linear map, and contractions correspond to the composition of such maps. From this perspective, a quantum circuit itself forms a tensor network, where quantum gates are tensors with input and output indices.

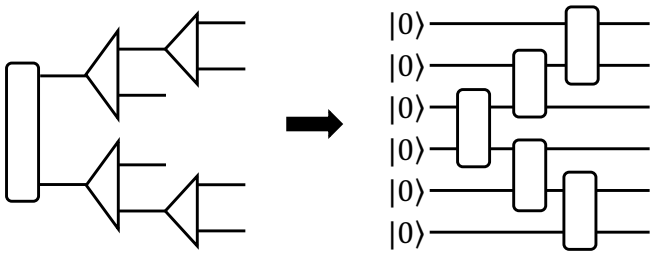


Figure 2.4: A conversion from a canonical form to a quantum circuit. An MPS is converted to a quantum circuit that generates the same quantum state as the original MPS. For simplicity, all bond dimensions are assumed to be 2, but the extension to larger bond dimensions is straightforward.

Conversely, an MPS can be converted into a quantum circuit that prepares the corresponding quantum state. This is achieved by sequentially applying the isometries obtained from the canonical form of the MPS. Specifically, each local tensor in mixed-canonical form defines an isometric map that can be implemented as a unitary operation by introducing ancillary qubits. We illustrate this procedure in Fig. 2.4. By applying these unitaries sequentially, one can prepare the MPS on a quantum computer. This provides a direct connection between tensor network complexity and quantum circuit complexity.

2.2.7 Tree tensor networks

Tree Tensor Networks (TTN) are a natural generalization of MPS, in which tensors are arranged in a tree-like structure rather than a one-dimensional chain, as shown in Fig. 2.5. TTN are particularly well-suited for representing quantum states with hierarchical or multi-scale entanglement structures. As in the case of MPS, canonical forms can be defined for TTN. These canonical forms enable optimal bond truncation and provide a systematic way to convert TTN representations into quantum circuits.

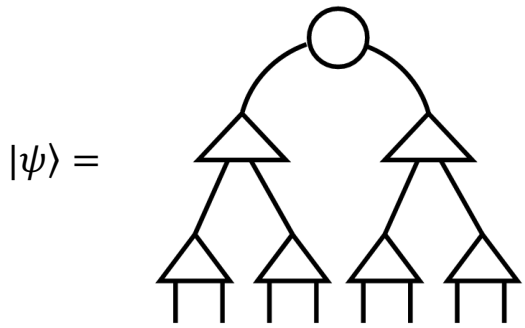


Figure 2.5: An example of a TTN. Each tensor is located on a tree graph.

2.2.8 Quantum circuit simulation with Matrix Product States

In this subsection, we briefly explain how quantum circuit dynamics can be efficiently simulated using MPS. This framework is particularly suitable for simulating noisy one-dimensional quantum circuits, such as those arising in quantum error correction protocols.

Single-Qubit Gates and CPTP Maps

A single-qubit gate or linear map M_i acting on site i corresponds to a local update of the MPS tensor,

$$A^{[i]s_i}_{\alpha_{i-1}\alpha_i} \leftarrow \sum_{s_i'} M^{s_i}_{s_i'} A^{[i]s_i'}_{\alpha_{i-1}\alpha_i}. \tag{2.20}$$

This update preserves the MPS structure and does not increase the bond dimension.

Single-qubit noise processes are modeled by CPTP maps. To simulate noisy dynamics, we sample Kraus operators $\{K_\mu\}$ according to

$$p_\mu = \mathrm{Tr}\big(K_\mu \left|\psi\right\rangle \left\langle\psi\right| K_\mu^\dagger\big), \tag{2.21}$$

which can be efficiently evaluated using the MPS representation. The sampled Kraus operator is then applied in the same manner as a single-qubit gate. Projective measurements can be simulated analogously.

Two-Qubit Gates as Matrix Product Operators

A two-qubit gate acting on neighboring sites i and $i+1$ is represented as an MPO,

$$M^{s_i s_{i+1}}_{s_i' s_{i+1}'} = \sum_\alpha M_\alpha^{[i]s_i s_i'} M_\alpha^{[i+1]s_{i+1}s_{i+1}'}, \tag{2.22}$$

which can be obtained by applying singular value decomposition (SVD) to the two-qubit gate tensor.

Two-qubit gates acting on non-neighboring qubits $i < j$ are handled by inserting identity tensors between sites i and j,

$$M^{[l]s_l s'_l}_{\alpha_{l-1}\alpha_l} = \delta_{\alpha_{l-1},\alpha_l}\delta_{s_l,s'_l} \quad (i < l < j), \tag{2.23}$$

thus forming an MPO that spans the intermediate region.

Canonical Form and Bond Truncation

Applying an MPO generally increases the bond dimension of the MPS. To control computational cost, we employ bond truncation based on the canonical form.

We first bring the MPS into a mixed-canonical form with the orthogonality center at site i. We then apply the MPO tensors from left to right (from site i to j), and after each local update we restore the canonical form by an SVD and truncate small singular values. At each step, two neighboring tensors are contracted and decomposed as

$$A^{[l,l+1]} = USV^\dagger, \tag{2.24}$$

and small singular values in S are truncated.

Finally, the MPS is restored to canonical form by absorbing U and $SV^\dagger$ into the neighboring tensors. This procedure is repeated until the gate application is completed.

Chapter 3

Tensor Networks for the Simulation of QEC Codes under Leakage Errors

Leakage errors, in which a qubit is excited to a level outside the qubit subspace, represent a significant obstacle in the development of robust quantum computers. In this chapter, we present a computationally efficient simulation methodology for studying leakage errors in quantum error correcting codes (QECCs) using tensor network methods, specifically Matrix Product States (MPS). Our approach enables the simulation of various leakage processes, including thermal noise and coherent errors, without approximations (such as the Pauli twirling approximation) that can lead to errors in the estimation of the logical error rate. We apply our method to two QECCs: the one-dimensional (1D) repetition code and a thin $3 \times d$ surface code. By leveraging the small amount of entanglement generated during the error correction process, we are able to study large systems, up to a few hundred qutrits, over many code cycles. We consider a realistic noise model of leakage relevant to superconducting qubits to evaluate code performance and a variety of leakage removal strategies. Our numerical results suggest that appropriate leakage removal is crucial, especially when the code distance is large. This chapter is based on [H. Manabe *et al.*, *New Journal of Physics* **27**, 114512 (2025)] with slight modifications to fit in the context.

3.1 Introduction

Overcoming the effect of noise is crucial for practical applications of quantum computers. A known method to suppress qubit errors is to use a quantum error correcting code (QECC) [13, 70–75], in which protected logical qubits are encoded into a larger number of physical qubits, and physical errors can be detected by appropriate measurements.

One type of error that is not typically corrected by QECCs is leakage error [76, 77], where a qubit escapes from its computational subspace. Recent experimental reports suggest that these errors can severely impact the performance of QECCs [75]. It is not effective to increase the code distance for mitigating this type of error because typical QEC codes are designed to reduce qubit-space errors, not leakage errors.

Furthermore, qubit control pulses are designed to achieve high performance when qubits are in the qubit space. Two-qubit operations acting on leaked qubits may have undesirable effects, such as moving surrounding qubits to leaked states. Thus, leakage errors are considered to be one of the major issues in scaling fault-tolerant quantum computing up to practical regimes. So far, massive efforts have been devoted to analyzing the effect of leakage errors and devising methods for their removal [75, 78–83].

Simulating the effect of leakage and evaluating the performance of leakage removal strategies accurately on large scale QECCs is challenging since leakage errors are not Pauli errors and therefore cannot be directly incorporated into standard stabilizer simulations [84]. Nevertheless, Refs. [13, 74, 75] used simplified leakage noise models, based on the Pauli twirl approximation, to treat leakage errors with the stabilizer formalism. In particular, they employed the General Twirling Approximation (GTA), a type of Pauli twirling, which replaces coherent leakage noise with stochastic incoherent errors. Yet, it is not clear how well these simplified models capture the performance of realistic leakage in QECCs, since such approximations discard the coherence between leaked state and qubit space and lead to significantly misrepresented performance for other types of noise [85, 86]. Therefore, there is a strong demand for developing scalable and accurate simulation methods to understand how leakage errors degrade the performance of QECCs and how they can be removed.

In this chapter, we use tensor-network methods [40, 41, 87] to efficiently simulate QEC codes under leakage errors without approximations that simplify the noise model. We utilize the Matrix Product State (MPS) tensor-network ansatz [88] for simulating the one-dimensional (1D) repetition code [89–92] and the thin $3 \times d$ surface code under leakage errors. This method allows efficient simulation of large numbers of qubits by exploiting the fact that the amount of bipartite entanglement generated by noisy repetitive stabilizer measurements in the QEC process is small enough to be accurately approximated by an MPS. We can simulate a few hundred qutrit systems, a scale unreachable even with the latest state-vector simulators [25, 26, 93, 94].

We have used our simulator to study a variety of leakage processes including coherent leakage and thermal noise. Our numerical results show that there can be a large discrepancy in performance between a coherent noise model and a stochastic approximation to the coherent noise model. These findings reveal that simplified metrics for leakage, such as leakage and seepage rates, are not entirely sufficient in capturing the real influence of leakage, and the ability of our method to capture general non-Pauli noise processes can be very useful for assessing the code performance accurately.

We have also used this method to compare the performance of different leak-

age removal strategies. We found that the effect of leakage removal strategies significantly depends on the details of the leakage process, with some differences becoming distinctly apparent only in regions with large code distances. These results underscore the necessity of simulating QECCs, particularly in large-scale quantum systems, in order to design qubits that are resistant to leakage. We expect our methods, which fulfill the requirements of both accuracy and scalability, will become indispensable tools for developing scalable quantum computing with practical quantum devices.

The chapter is structured as follows. We describe the leakage noise model settings in Sec. 3.2 and the tensor network simulation methods in Sec. 3.3. We formulate the problem settings of the QEC codes used in this chapter and review some leakage removal strategies in Sec. 3.4. In Sec. 3.5, we present the main numerical results on the MPS simulations of the 1D repetition code and the $3 \times d$ thin surface code and discuss the effect of leakage. We conclude the chapter by summarizing our contribution and future directions in Sec. 3.6.

3.2 Leakage noise models

We aim to provide a general simulation framework that can efficiently evaluate the performance of QECCs under leakage error models and compare the mitigation methods for them. In this section, we introduce several phenomenological leakage noise models to demonstrate our simulation framework.

Noise, including leakage noise, may occur during any operations such as gates, measurements, or when qubits are idle, and they are observed in a variety of qubit devices, such as superconducting circuits [13], neutral atoms [95] and trapped ions [96]. Their noise modeling highly depends on each qubit system and architecture. In this chapter, we focused on transmon superconducting qubits [8–11] as an example target of noise modeling. This is because they achieved the demonstration of surface codes for up to $d = 7$ and faced the problems of leakage errors [13, 74]. To realize fast gate operations, short control pulses are required. However, shortening the pulse duration inevitably broadens its frequency spectrum. Such fast driving increases the probability of transitions to higher energy levels due to the small anharmonicity.

To concisely demonstrate the effect of typical leakage errors on QECCs, we assume superconducting qubits as a qutrit system and employ the leakage noise models discussed below. While the specific noise descriptions may vary for other types of superconducting qubits or different qubit platforms such as trapped ions and neutral atoms, the fundamental mechanism of leakage originating from the presence of higher energy levels remains a common challenge. Given the tensor descriptions of the noise models and their Kraus decompositions, our framework can treat them in a unified manner, albeit potentially at a higher computational cost.

3.2.1 Single-qubit gate errors

A significant factor that incurs leakage in single-qubit operation is the use of short pulses for fast control [77]. The exact modeling of the leaking unitary operator is analytically difficult and is highly dependent on specific details of the architecture. To simplify the noise model, we make the following assumptions. 1) Rabi-oscillation in the qubit space and leakage states happen independently. 2) The oscillations between $|0\rangle \leftrightarrow |2\rangle$ and $|1\rangle \leftrightarrow |2\rangle$ can be treated independently. 3) The oscillation between leaked states can be modeled as far-detuned Rabi oscillation. The above assumptions are justified since the effect of the $|2\rangle$ state in a single-qubit gate is small, and other higher-order effects are expected to be negligible. According to the above assumptions, we account for the coherent transfer between the qubit space and the leaked state by fast control as follows.

$$U = U_z(\varphi_i) \cdot R_{02}(\theta, \lambda_i) \cdot R_{12}(\theta, \lambda_i), \tag{3.1}$$

where

$$R_{02}(\theta, \lambda_i) = U_{02}(\theta, \lambda_i) \oplus I_1, \tag{3.2}$$
$$R_{12}(\theta, \lambda_i) = U_{12}(\theta, \lambda_i) \oplus I_0, \tag{3.3}$$
$$U(\theta, \lambda_i) = e^{-i\theta/2} \exp(i\theta/2(\cos\lambda_i X + \sin\lambda_i Y)), \tag{3.4}$$

and

$$U_z(\varphi_i) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{i\varphi_i} \end{pmatrix}. \tag{3.5}$$

Here, θ denotes the strength of rotation noise, while λ_i and φ_i denote the relative phases of leakage rotations. While in experiment these phases are determined by the details of physical systems, in this chapter they are randomly chosen from $[0, 2\pi)$ for each qubit. I_0 and I_1 are identity operators on the 1D subspaces $\{|0\rangle\}$ and $\{|1\rangle\}$, respectively. We model the occurrence of the following unitary rotation after applying the CZ gate.

3.2.2 Controlled-phase gate errors

Two-qubit operations applied to leaked qubits can have undesired effects. We consider two effects from applying two-qubit gates acting on leaked qubits: undesired phase shifts, and leakage spreading. They have been experimentally observed in Ref. [75] and reported as major noise factors that increase logical error rates in Ref. [13]. To model undesired phase shifts, we define the noisy CZ

gate as follows:

$$\mathrm{CZ}_{\mathrm{noisy}} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & i & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & i & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}, \tag{3.6}$$

and we define the noisy CNOT gate as $(I \otimes H)\mathrm{CZ}_{\mathrm{noisy}}(I \otimes H)$ where

$$H = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \\ \frac{1}{\sqrt{2}} & \frac{-1}{\sqrt{2}} & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{3.7}$$

When both systems are in the qubit space, the noisy CZ and CNOT gates act as the ideal versions of these gates. However, a phase of i will be added when a control qubit is in the leaked space. Here, we chose phase i ($\pi/2$ phase shift) for CZ gates on $|2\rangle$ state since, according to Ref. [75], the observed phase shift for this state is intermediate of $|0\rangle$ state (0 phase shift) and $|1\rangle$ state (π phase shift).

Leakage spreading has also been observed in experiment [75], in which CZ gates on qubits in a leaked state cause neighboring qubits to leak. This phenomenon is understood as a composition of multiple transitions between states with higher levels [82]. In this chapter, we consider a simpler phenomenological model of leakage spreading consisting of a sequence of rotations about the Y axis in the subspaces $\{|02\rangle, |22\rangle\}, \{|12\rangle, |22\rangle\}, \{|20\rangle, |22\rangle\}, \{|21\rangle, |22\rangle\}$ with angle θ_{spread}. These four operations are applied after the CZ gate and cause a state in the qubit subspace to coherently rotate into the leakage subspace when the neighboring qubit is in the leaked state.

3.2.3 Measurement

Typical measurement schemes, such as dispersive readout in superconducting qubits, are not designed to discriminate leaked states from the qubit space. Thus, we assume that when a qutrit is in the leaked state, the measurement results are randomly classified as 0 or 1 with 1/2 probability. With this assumption, the action of measurements can be modeled using the following CP-instrument:

$$\mathcal{E}_0(\rho) = \Pi_0\rho\Pi_0 + p\Pi_2\rho\Pi_2, \tag{3.8}$$

$$\mathcal{E}_1(\rho) = \Pi_1\rho\Pi_1 + (1-p)\Pi_2\rho\Pi_2, \tag{3.9}$$

where $p = 0.5$ and $\Pi_i = |i\rangle\langle i|$.

Measurements can also be a significant contributor to leakage errors [97–99], and can easily be taken into account with a tensor network simulator (they can simply be absorbed into gate leakage before the measurement). However, for simplicity, in this chapter, we neglect measurement leakage and instead focus on the dominant sources of leakage, namely CZ and thermal leakage [13].

3.2.4 Idling noise

Even when qubits are idle, the thermal noise, i.e., amplitude damping and thermal excitation, may excite a qubit state to out-of-qubit states. This effect is typically not a dominant error source but is reported as a non-negligible contributor to leakage errors in TABLE S4 in Ref. [13]. For formulating the thermal noise model, for simplicity, we utilize the dynamics of harmonic oscillators connected to thermal baths written as:

$$\dot{\rho} = \gamma(N+1)(a\rho a^\dagger - \frac{1}{2}\{a^\dagger a, \rho\}) + \gamma N(a^\dagger \rho a - \frac{1}{2}\{aa^\dagger, \rho\}), \tag{3.10}$$

where $N = (e^{\hbar\omega/k_BT} - 1)^{-1}$, a and $a^\dagger$ are annihilation and creation operators, γ is the coupling strength to the bath, and $k_BT/\hbar\omega$ is the effective temperature relative to the photon energy. We calculate the CPTP-map with Kraus representation of the above model by cutting off the higher levels and integrating the above dynamics over a period of gate time τ. We apply this error map to all qubits at the beginning of each syndrome measurement round. In the simulation, we set the resonant frequency of qubit as 10GHz, which means $k_BT/\hbar\omega := \alpha T$ and $\alpha \sim 13.1\,\mathrm{K}^{-1}$. We assume $\tau = 1.0\,\mu\mathrm{s}$ and units of γ and T being MHz and mK, respectively for the numerical calculations of this chapter.

3.3 Tensor network methods

The noise models described in the previous section include non-Pauli processes such as control leakage, which cannot be efficiently simulated with a Clifford simulator. While it is possible to use twirling approximations [74] to convert a non-Pauli noise to a stochastic Pauli one, the approximate noise model is not guaranteed to faithfully reproduce the behavior of the original noise model.

In order to simulate general circuits, state-vector [37, 93, 100] and tensor-network methods [59–64, 101–105] are widely used. State-vector type simulations are straightforward to implement but require exponentially large memory capacity: for example, a few petabytes of memory for 30 qutrit systems.

Tensor-network methods, in contrast, are more complicated but can be used to simulate larger circuits than state-vector approaches. Broadly speaking, tensor-network methods for quantum circuit simulations can be classified into two types: 1) tensor-contraction type [59–63] and 2) ansatz type [64, 101–105]. For the tensor-contraction type, the entire quantum circuit is first converted into a tensor network, simplified, and a near-optimal contraction path is searched heuristically. On the other hand, the ansatz-type method expresses the quantum state during the execution of a quantum circuit by a tensor network with

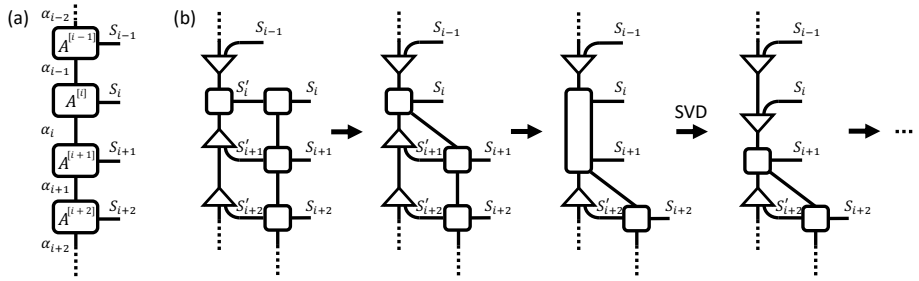


Figure 3.1: (a) The diagram notation of MPS, a simple tensor network with a 1D structure. (b) The procedure of canonical update. The triangular tensor represents an isometry from two indices (connected to the base of the triangle) to one (connected to the vertex of the triangle), and the square tensor represents the 'top tensor', which has no particular structure. When applying a two-qubit gate, the MPS is transformed into a canonical form where the top tensor is located at the ith position, where i is the first site on which the gate acts non-trivially, and the remaining tensors are isometries as pictured. The MPO representing the two-qubit gate is sequentially applied to the MPS from the site i. During the process, SVD is used to truncate the bond dimension and restore the state to the canonical MPS format.

a specific structure. It is known that many interesting quantum states can be efficiently represented using a tensor-network ansatz, which exploits their entanglement structure [40, 41, 46, 87, 88, 106]. In this chapter, we adopt ansatz-type approach and exploit the fact that qubits in many quantum error correcting codes, unlike in generic quantum circuits, have a specific entanglement structure that can be well captured with a tensor-network ansatz. This allows far more efficient simulation compared to tensor-contraction approaches designed for generic circuits. An example of this was seen in previous work [85, 107], where the simple tensor-network structure of the surface code was used to probe the effect of non-Pauli noise on error correction performance. However these simulations, unlike those in this chapter, were limited to the case with noiseless syndrome measurements.

In this section, we will briefly review the quantum circuit simulation method using Matrix Product States (MPS) [49, 50, 101, 103–105, 108], which we use to simulate QECCs.

3.3.1 Matrix product states

A Matrix Product State (MPS) [88] is a tensor network ansatz typically used to represent 1D pure states. We illustrate the diagram notation of MPS in Fig. 3.1(a), where nodes correspond to multi-dimensional arrays and each incident edge of a node corresponds to an index of that array. An edge connecting a pair of nodes corresponds to taking sums over corresponding indices. An MPS

can be written as

$$|\psi_{s_1,s_2,\cdots s_n}\rangle = \sum_{\{s_i\}} \sum_{\{\alpha_i\}} \left[A^{[1]s_1}_{\alpha_1} A^{[2]s_2}_{\alpha_1\alpha_2} \cdots A^{[n]s_n}_{\alpha_{n-1}} \right] |s_1 s_2 \cdots s_n\rangle \,, \tag{3.11}$$

where $A^{[i]}$ represent ith tensor in Fig. 3.1(a), and $\{s_i\}$ and $\{\alpha_i\}$ are referred to as physical and virtual indices, respectively. Each amplitude $\langle s_1 s_2 \cdots s_n | \psi_{s_1,s_2,\cdots s_n}\rangle$ is expressed by the product of n matrices $\{A^{[i]s_i}\}$.

The edges linking the nodes are called *bonds*. The dimension of these bonds is referred to as *bond dimension* χ. The expressibility of MPS is constrained by its 1D structure: an MPS with bond dimension χ can represent quantum states with bipartite entanglement entropy up to $\log_2 \chi$ with respect to any bipartition of the chain.

While MPS is a 1D ansatz, it is often applied to study two-dimensional (2D) systems by snaking the MPS through a 2D lattice e.g. [105, 109] . While the computational cost will typically grow exponentially with increasing the second dimension, the results of such methods are often competitive with other state-of-the-art methods in 2D [41, 47, 110].

3.3.2 Quantum circuit simulation with MPS

Here we briefly describe how the evolution of the many-qubit quantum state over many rounds of quantum error correction can be captured in the MPS formalism.

In the simulation, the state of the quantum system is represented as an MPS, and individual tensors in the MPS are modified according to the action of gates and measurements. Two-qubit gates cause the bond dimension of the MPS to increase. In order to prevent exponential growth in bond dimension, it is necessary to perform bond truncation, which we have achieved using a widely known method for MPS called *canonical update*. The diagram notation of the canonical update is illustrated in Fig. 3.1(b). The MPS is initially put in *canonical form*, where all tensors are isometric maps from two indices to one except for one tensor, called the 'top tensor'. Any MPS can be put in such a form, where the top tensor can be chosen to be at any position, by performing successive SVDs on individual tensors. When applying a gate, the top tensor is chosen to be the first site on which the gate acts non-trivially. The Matrix Product Operator (MPO) representing the gate is then applied, according to Fig. 3.1, followed by an SVD. By truncating small singular values at this stage, the action of the gate and bond truncation are both realized on the MPS.

The bond truncation reduces the bond dimension of an MPS to some chosen value χ. In general this will cause some loss in fidelity with the original state $|\psi\rangle$, however the fidelity can be improved systematically by choosing larger values of χ. The value of χ required to obtain a high fidelity will depend on the entanglement in $|\psi\rangle$. In the error correcting codes described below, we demonstrate that only a small χ (and therefore small computational cost) is required to represent the state with high fidelity.

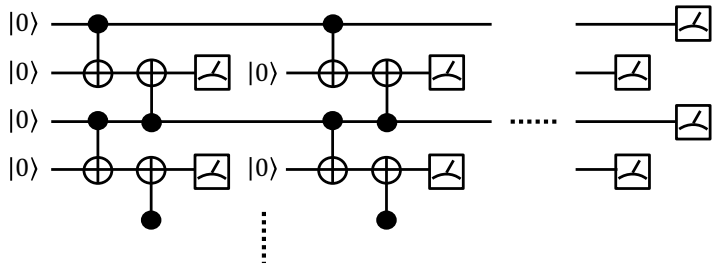


Figure 3.2: The quantum circuit for the 1D repetition code. Each check is executed by coupling two data qubits to a single ancilla qubit then measuring the ancilla, and repeated d times. At the end, to read out the encoded information, all data qubits are measured.

3.4 Error-correction simulations

In this section, we detail the simulation setup to investigate the impact of leakage on QEC codes. We focus on simulations of the 1D repetition code [89, 90, 92] and the thin surface code, which are regarded as simple to implement due to their low connectivity and have been shown theoretically to be highly efficient at correcting biased noise [111–115]. Their respective 1D and quasi-1D entanglement structure makes them ideal targets for the MPS method, described in the previous section. Extending these methods to codes with more complex connectivity in two or higher dimensions will likely require tensor network ansatz beyond MPS, which we leave to future work.

In addition to standard error correction methods, we also incorporate several leakage removal strategies into the simulation. This allows us to understand the effectiveness of these strategies in minimizing the effect of different types of leakage processes.

3.4.1 Error correcting codes

Quantum error correction (QEC) is a framework to protect quantum information from noise and decoherence. In this work, we specifically focus on stabilizer codes, which constitute a widely used class of QEC codes. We first encode the logical quantum information into a redundant state of multiple physical qubits, referred to as data qubits. To detect errors occurring during the process without disturbing the logical state, we execute the specific indirect measurements, known as stabilizer measurements, using ancilla qubits. Based on the outcomes of repeated syndrome measurements, we utilize the decoder to infer the error configuration in the process. For a detailed explanation of QEC and stabilizer formalism, we refer readers to Refs. [71, 116, 117].

The simplest example of a stabilizer code is the 1D repetition code. The quantum circuit of the 1D repetition code is shown in Fig. 3.2. The logical $|0\rangle$ and $|1\rangle$ states for the 1D repetition code are defined as $\{|0\cdots0\rangle, |1\cdots1\rangle\}$. We

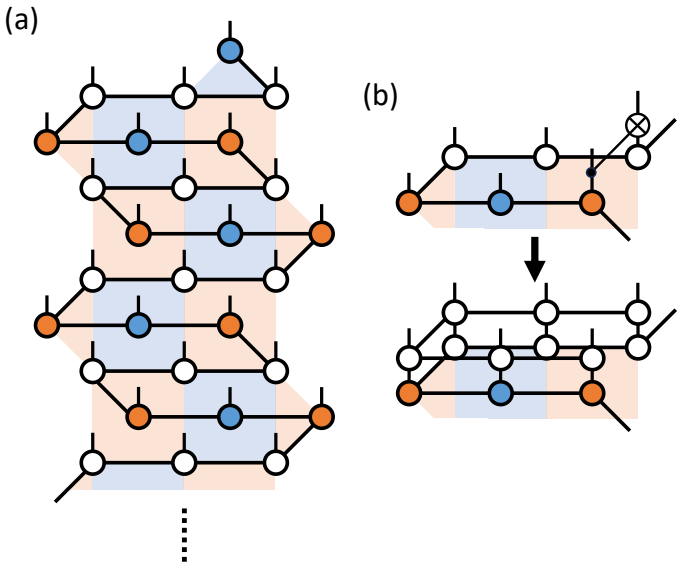


Figure 3.3: (a) The layout of the thin surface code and corresponding MPS. White vertices represent data qubits, and orange and blue vertices represent ancilla qubits for X-type and Z-type stabilizer measurements, respectively. The MPS is defined in a manner that connects data qubits and ancilla qubits in a snake-like style. (b) Two-qubit gate in the syndrome measurements is represented as MPO.

initialize an encoded logical $|0\rangle$ or $|1\rangle$ simply by preparing a product of d physical $|0\rangle$ or $|1\rangle$ states, respectively at data qubits. Then, syndrome measurements are performed d times. Finally, the encoded bit is decoded from the data qubits and syndrome outcomes using the minimum-weight perfect matching algorithm [90, 118]. Since noise is introduced to the quantum state, decoding does not always succeed. We define the probability of decoding failure as the logical error rate.

To address arbitrary errors, a QEC code must correct both X and Z errors. The surface code [33] is a candidate for this purpose due to its high error threshold and locality suitable for superconducting devices. As illustrated in the layout in Fig. 3.3(a), data qubits (white vertices) and ancilla qubits (orange and blue vertices) are arranged on a lattice. Z-type stabilizer measurements are defined on the blue plaquettes and detect X errors occurring on the adjacent data qubits. Similarly, X-type stabilizer measurements are defined on the orange plaquettes and detect Z errors occuring on the adjacent data qubits.

In this section, we conducted further simulations of the $3 \times d$ thin surface code, which has a quasi-1D structure. By thin surface code, we mean a surface code for which the size of the lattice can increase, while its width remains constant. The thin surface codes are considered useful when the noise is bi-

ased [111, 113, 115]. We can reduce the probability of logical Z errors (the vertical Z chain) by increasing d. Conversely, the minimum length of the X logical operator is fixed and, as a result, it is anticipated that the logical X error rate will increase with increasing d, rather than decrease, due to the larger number of possible horizontal error chains. From this point, we refer to d as the Z-distance, corresponding to the minimum number of physical Z errors needed to create a logical error.

In our simulation, both data qubits and ancillary qubits are included together in the tensor-network description of the state. For the 1D repetition code, the data qubits and ancilla qubits are alternately arranged in one dimension, constituting a $2d-1$ qubit system in total if the code distance is d. The logical $|0\rangle$ and $|1\rangle$ state, including ancilla qubits, is just a product state ($|00\cdots0\rangle$ or $|10101\cdots1\rangle$ respectively), which can be represented as MPS with $\chi = 1$. In the case of the $3 \times d$ surface code, there are $6d-1$ qubits in total, and we design the MPS structure in a snake-like manner as shown in Fig. 3.3(a). When the logical $|+\rangle$ state is encoded, the system is a low-entanglement state, which can also be efficiently represented by MPS with a maximum bond dimension of $\chi = 4$. For a 1D repetition code, the two-qubit gate is applied only to neighboring qubits. However, for a thin surface code where the MPS is snake-like, it is necessary to apply an MPO of up to length 6 as shown in Fig. 3.3(b), escalating the simulation cost.

If there is no noise, the logical state of both codes is a low-entangled state and can thus be efficiently described using MPS. When the noise contains Pauli and a certain simple type of incoherent leakage error, it can also be easily simulated using an extended Clifford simulator, such as Pauli+ [74]. When there is coherent noise, if the noise is sufficiently weak and the correlation between errors is small, it is expected that the required bond dimension to reach a certain fidelity will be sufficiently small. This is because the logical state itself is a low-entangled state, and the repetitive stabilizer measurements act to return the entire system to such a state (or a state where a local operator has been applied). This behavior is similar to the measurement-induced phase transitions in monitored quantum circuits [119, 120], but they differ in that we utilize the properties of specific error-correcting codes and noise models.

We should note that, besides the codes mentioned above, our method can also be applied to various larger QECCs. For the $5 \times d$ and $7 \times d$ surface codes, for example, only bond dimensions $\chi = 8, 16$, respectively, are needed at most to represent the logical states, and thus they can be efficiently simulated in low noise regions. However, since the length of the MPO for the 2-qubit gates is long ($10, 14$ respectively), the effect of noise spreads over numerous tensors, resulting in an increase in the required computational resources.

3.4.2 Leakage removal strategies

Leakage noise is known to harm the performance of quantum error correction. Therefore, several strategies have been devised to remove leakage errors [75]. In

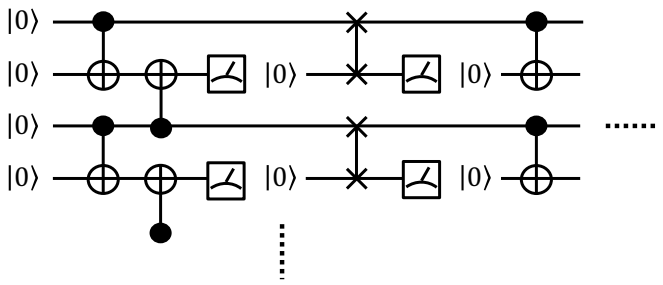


Figure 3.4: The quantum circuit of the DQLR process. After the stabilizer measurement, the LeakageISWAP gate is applied between pairs of data and ancillary qubits, then ancilla qubits are measured again.

this chapter, we simulate three approaches to compare and analyze the performance of these strategies.

No reset

No reset strategy does not offer any leakage removal operation after the measurements. In order to set the ancillary qubit to $|0\rangle$, we only apply the X gate if the outcome is 1. The quantum channel of this process can be represented as follows

$$\mathcal{M}_{\text{noreset}}(\rho) = (\langle 0|\,\rho\,|0\rangle + \langle 1|\,\rho\,|1\rangle)\,|0\rangle\langle 0| + \langle 2|\,\rho\,|2\rangle\ \ |2\rangle\langle 2|\,. \tag{3.12}$$

Multi-level reset

The *multi-level reset (MLR)* strategy involves applying a map to ancillary qubits (and not data qubits) after every syndrome measurement that converts all states including the leakage state perfectly to the $|0\rangle$ state. This can be expressed as the simple map $\mathcal{M}_{\text{MLR}}(\rho) = |0\rangle\langle 0|$.

DQLR

Although the MLR strategy can remove all leakage in ancillary qubits, it cannot eliminate leakage errors that occur in data qubits, which can cause serious time-correlated errors in QEC circuits. Therefore, in the *DQLR* strategy, we apply a *LeakageISWAP* gate [75] between the data qubits and ancillary qubits after performing MLR, as shown in Fig. 3.4. The LeakageISWAP gate executes an ISWAP gate that acts non-trivially on the $\{|11\rangle, |20\rangle\}$ subspace (rather than the usual $\{|10\rangle, |01\rangle\}$ subspace for ISWAP), and removes leakage in data qubits by converting the $|02\rangle$ state to $|11\rangle$. Finally, MLR is performed again on ancillary qubits. This method is designed to remove leakage from both data and ancilla

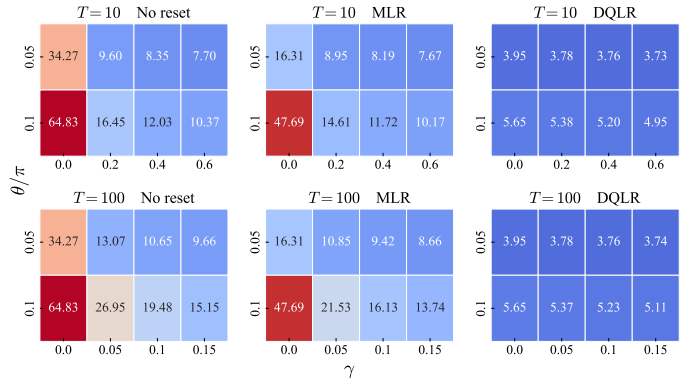


Figure 3.5: The average MPS bond dimension required to reach a truncation error of 10^{-6} when simulating the 1D repetition code for varying the temperature T, leakage rotation angle θ, coupling strength to bath γ, and the leakage removal strategy. The average MPS bond dimension is computed by collecting all the bond dimensions at the end of each round and taking their average. The code distance is $d = 99$, the number of rounds is also 99 and the angle of leakage spreading is fixed as $\theta_{\text{spread}} = 0.3\pi$.

qubits, but it also introduces additional implementation challenges and may increase the physical error rates.

3.5 Results

Here we present our simulation results for the codes and leakage removal strategies described in Sec. 3.4 that we have calculated using the MPS method described in Sec. 3.3. The tensor-network algorithms are implemented using the Python library TensorNetwork [121]. The numerical calculation runs on a machine with Intel Xeon Platinum 9242 system with 96 threads. Each data point on the plots is obtained by averaging ten thousand samples.

3.5.1 Bond dimension

In order for simulations with MPS to be efficient, it is essential that the bond dimension required for accurate simulation does not grow too quickly in time or with increasing system size. In our simulations, we adopt an approach where the bond dimension is chosen dynamically in order for the truncation error, which is the 2-norm of the vector of truncated singular values, of each truncation step to be below some threshold value. In all of our calculations, we have chosen this threshold to be 10^{-6} for the 1D repetition code and 10^{-4} for the thin surface code to balance both accuracy and computational efficiency. We have confirmed

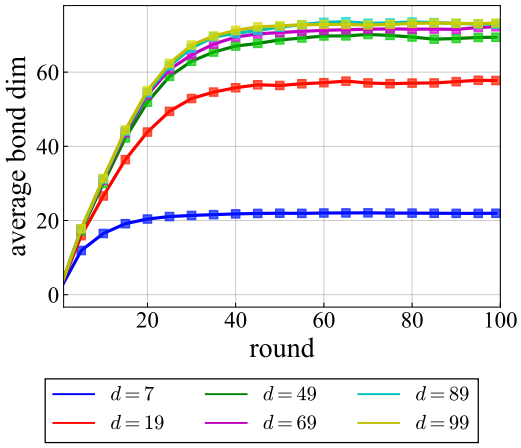


Figure 3.6: The bond dimension of the MPS at the end of each syndrome measurement round for the 1D repetition code simulation. The bond dimension is truncated at each step to ensure the truncation error does not exceed a threshold. The parameters are set as $T = 100$, $\theta = 0.1\pi$, $\gamma = 0.0$, $\theta_{\text{spread}} = 0.3\pi$ and no leakage removal strategy is applied, which requires the largest average bond dimensions in Fig. 3.5. We repeat syndrome measurements 99 times.

that these choices are sufficient to accurately calculate the logical error rate in the parameter regions studied.

We present the average bond dimension in the simulation of the 1D repetition code of $d = 99$ for the various noise parameters in Fig. 3.5. Note that the bond dimension can fluctuate for each trial due to the sampling of Kraus operators, and thus the averaged bond dimensions are generally non-integers. As shown in these figures, we find that, generally, a small bond dimension is sufficient for accurate simulations across all noise parameters and leakage removal strategies. This implies that the amount of entanglement generated in the error correction procedure for both incoherent and coherent noise is small, and therefore can be captured accurately using the MPS simulator.

The bond dimension required tends to increase as the coherent component of the leakage is increased, and decreases with increasing leakage removal. The latter trend can be understood due to the fact that as the leakage population increases, more entanglement can be generated.

Fig. 3.6 shows the bond dimensions during the process in the most computationally intensive noise setting by fixing the threshold for truncating singular values. The required bond dimensions increase linearly at initial rounds and saturate at a certain point. We also find that in the limit of many rounds of error correction, the required bond dimension is constant with respect to d, with the exception of finite-size effects for systems with $d \lesssim 50$. From this, it can be inferred that this system exhibits an area law [41, 43], that is, the entan-

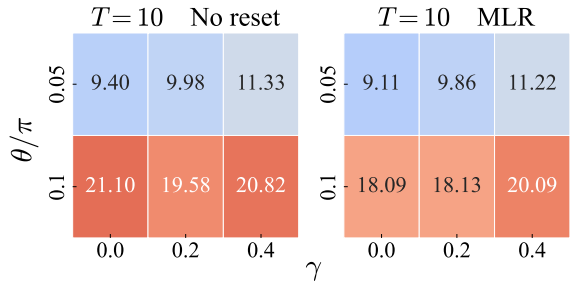


Figure 3.7: The average MPS bond dimension required to reach a truncation error of 10^{-4} when simulating the 3×7 surface code for varying leakage rotation angle θ, coupling strength to bath γ and leakage removal strategy. The average MPS bond dimension is computed by collecting all the bond dimensions at the end of each round and taking their average. The number of rounds is 7.

glement entropy is proportional to the boundary of the two regions (a constant amount of entanglement in the 1D case). This is in contrast to the general case, where exponential bond dimensions are required to represent the quantum state of a n qutrit 1D brickwork circuit with a depth n. These results all indicate that our method is scalable and accurate for the codes and noise models under consideration.

Finally, we plotted the average bond dimension for the 3×7 thin surface code in Fig. 3.7, which follows almost the same trend as the 1D repetition code. We note, however, that value is not saturated because of the smaller number of rounds, and the simulation of the $3 \times d$ surface code is more costly than that of the 1D repetition code.

3.5.2 Exact vs twirling approximation

We use our simulation method to examine the difference between coherent and incoherent leakage processes. For this purpose, we consider a well-known method to approximate coherent noise as incoherent noise, the General Twirling Approximation (GTA) [74]. Since our method allows simulations of both coherent and incoherent noise processes, we use it to compare the effect of a coherent leakage noise process to the GTA approximation of it.

In this test, for simplicity, we assume that there is no leakage spreading, that is, $\theta_{\text{spread}} = 0$. In our model for leakage, which has a single leaked state $|2\rangle$, we define the state leakage L of a 1-qubit state ρ as

$$L(\rho) = \langle 2|\rho|2\rangle \ . \tag{3.13}$$

Next, we define the leakage rate L_1 and seepage rate L_2 [122, 123] for a given

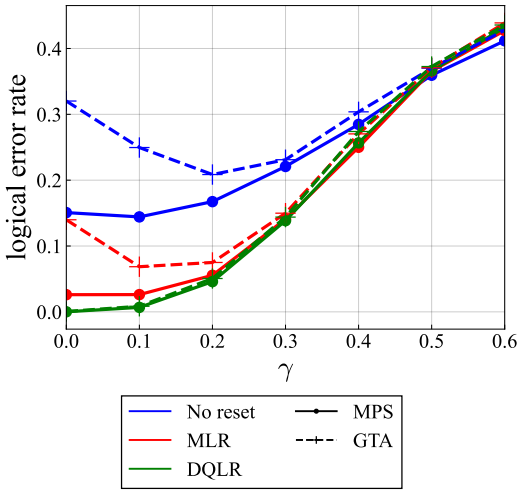


Figure 3.8: The logical error rate of the 1D repetition code for the coherent noise (solid) and approximated incoherent one (dashed) is plotted as a function of coupling strength to bath γ. The code distance, temperature, and rotation angle are fixed as $d = 19$, $T = 10$, and $\theta = 0.1\pi$. Three reset strategies, no reset, MLR, and DQLR are plotted in different colors.

noise map $\mathcal{E}$ as

$$L_1(\mathcal{E}) = L\left(\mathcal{E}\left(\frac{1}{2}\left(|0\rangle\langle 0| + |1\rangle\langle 1|\right)\right)\right), \tag{3.14}$$

$$L_2(\mathcal{E}) = 1 - L\left(\mathcal{E}\left(|2\rangle\langle 2|\right)\right). \tag{3.15}$$

These rates represent the probability of moving from the computational subspace to the leakage subspace and vice versa. We use them to define a GTA approximated quantum channel $\mathcal{E}'$ to a coherent control leakage unitary U as follows:

$$\mathcal{E}' = (1 - L_1)\mathcal{I}_1 + L_1 D_{21} + L_2 D_{12} + (1 - L_2)\mathcal{I}_2, \tag{3.16}$$

$$D_{21}(\rho) = \mathrm{Tr}[\mathbb{1}_1\rho]\,|2\rangle\langle 2|, \tag{3.17}$$

$$D_{12}(\rho) = \mathrm{Tr}[\mathbb{1}_2\rho]\left(\frac{1}{2}(|0\rangle\langle 0| + |1\rangle\langle 1|)\right), \tag{3.18}$$

$$\mathcal{I}_1(\rho) = \tilde{U}(\mathbb{1}_1\rho\mathbb{1}_1)\tilde{U}^\dagger, \tag{3.19}$$

$$\mathcal{I}_2(\rho) = \mathbb{1}_2\rho\mathbb{1}_2, \tag{3.20}$$

where $\mathbb{1}_1$ and $\mathbb{1}_2$ are the projectors onto the computational and leakage subspace, respectively, and $\tilde{U} = \mathbb{1}_1 U \mathbb{1}_1$. Finally, $\tilde{U}$ is replaced as a stochastic incoherent noise by applying the standard Pauli Twirling Approximation.

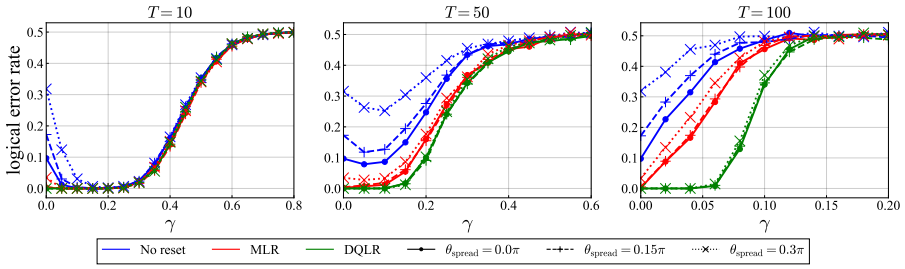


Figure 3.9: The logical error rate of the 1D repetition code is plotted as a function of coupling strength to bath γ for various temperatures and $d = 49$, $\theta = 0.05\pi$. Three reset strategies are plotted in different colors, and different θ_{spread} values are plotted by different linestyles and markers.

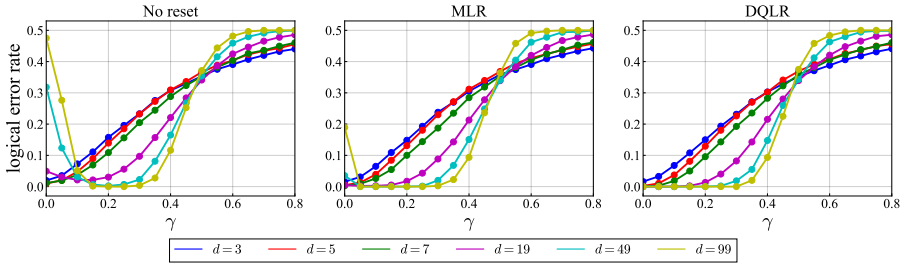


Figure 3.10: The logical error rate of the 1D repetition code for the three leakage removal strategies as a function of γ for various code distances. Parameters are set as $T = 10$, $\theta = 0.05\pi$ and $\theta_{\text{spread}} = 0.3\pi$.

Fig. 3.8 shows the difference in logical error rates between exact and GTA noise. As can be seen from the figure, introducing an approximation can overestimate the logical error rate by a large amount (by more than a factor of three in the MLR case). These phenomena, where twirling approximations over-predict the logical error rate, have also been observed in earlier studies [124–126]. This indicates that the introduction of approximations, like GTA, could potentially lead to misinterpretations of important properties such as inherent error tolerance and thresholds that the code possesses. Moreover, these results demonstrate that the simplified metrics for leakage, such as the leakage and seepage rates, do not necessarily fully capture the actual impact of leakage on the logical error rate.

3.5.3 Comparison of leakage processes and removal strategies

We conducted numerical experiments to investigate the effect of leakage by computing logical error rates as leakage removal strategies and the parameters of the noise model were varied.

There are three primary causes of leakage in the noise model: thermal excitation, control leakage, and leakage spreading. In high-temperature regions, the influence of thermal excitation appears strongly, as indicated in Eq. (3.10), which increases the leakage population. Fig. 3.9 shows the logical error rate of the 1D repetition code as a function of γ, the coupling strength to the bath, as the temperature changes. When adopting less effective leakage removal strategies such as no reset or MLR, which cannot perfectly remove leakage, we observed that the logical error rate significantly increases in the high-temperature region. Additionally, including leakage spreading further deteriorates the performance. In the case of DQLR, the presence or absence of leakage spreading hardly affects the results, indicating that it can nearly ideally eliminate leakage on qubits.

When the temperature is low, leakage is primarily introduced by over-rotation, leading to non-monotonic behavior in the logical error rate. Fig. 3.10 shows the logical error probability for the three leakage removal strategies in a low-temperature environment for various code distances. For large code distance d, in the small γ region, the logical error rate sharply increases as γ decreases, when we use either no reset or MLR strategy. This trend becomes more pronounced as the code distance d increases. This is because when coherent leakage is strong, the coupling to the thermal bath induces amplitude damping noise, which effectively reduces the leakage population. If the amplitude damping is weak, i.e., when γ is small, the leaked state persists and further spreads to its neighbors, fundamentally destroying the logical state composed of many physical qubits. It should be noted that these unusual phenomena cannot be obtained from simulations of small-scale systems.

We also plotted the logical error rate as a function of over-rotation angle θ in Fig. 3.11. When $\gamma = 0$, these plots show the effect of leakage purely due to fast control. One can see that for fast-control leakage, the effect on the logical error

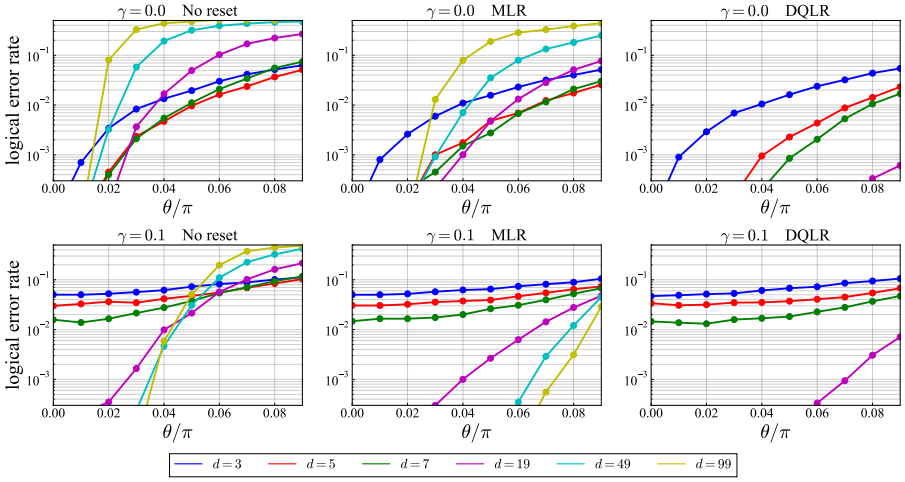


Figure 3.11: The logical error rate of the 1D repetition code for the three leakage removal strategies as a function of rotation strength θ for various code distances. Parameters are set as $T = 10$ and $\theta_{\text{spread}} = 0.3\pi$. Upper: the results with $\gamma = 0$, meaning there are no amplitude damping errors. Lower: the results with $\gamma = 0.1$. The logical error rate is calculated based on the results from several thousand samples, so only points above 3×10^{-4} are plotted for accuracy. Especially, for DQLR, with $d = 99$ and 49, exactly zero logical errors were detected during the simulations.

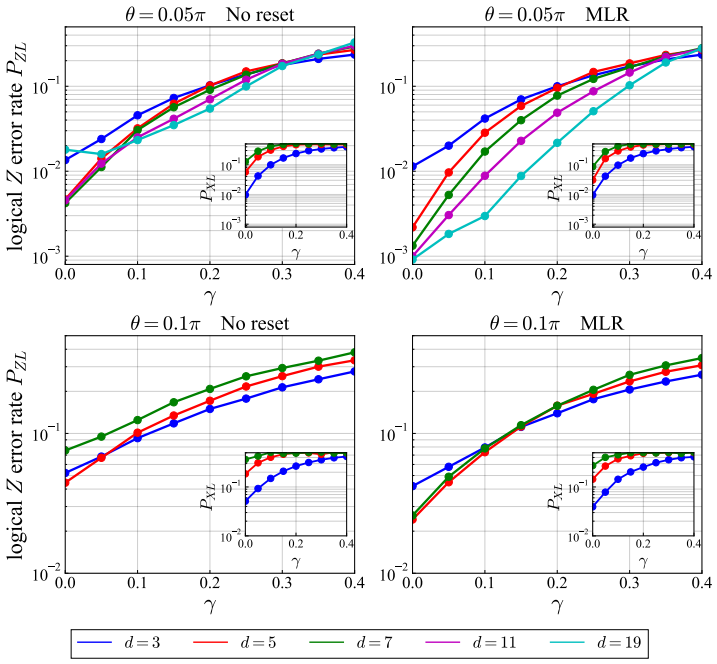


Figure 3.12: The logical Z error rate P_{ZL} of the $3 \times d$ surface code for the two leakage removal strategies and rotation angles $\theta = 0.05\pi, 0.1\pi$ as a function of coupling strength to bath γ for various Z-distance d. Parameters are set as $T = 10$ and $\theta_{\text{spread}} = 0$. The inset is the logical X error rate P_{XL} on the same setting.

rate becomes significant for larger system sizes. On the other hand, DQLR is very effective at eliminating leakage caused by control leakage and lowering the logical error rate, especially for these large sizes. For small γ one can observe competition between leakage caused by fast control and amplitude damping that tends to return leaked states to the qubit subspace. As seen above, amplitude damping also tends to reduce the effect of fast-control leakage on larger systems.

Lastly, we discuss the results of the MPS simulation for the $3 \times d$ thin surface code. Due to the simulation cost and the complexity of the procedure, we did not consider leakage spreading, and only No reset and MLR strategies were adopted for the leakage removal. Fig. 3.12 shows the logical error rate against γ for various Z-distance d when the amount of control leakage and the leakage removal strategies are changed. The logical Z error rates broadly follow the same trend as the 1D repetition code, where a larger θ results in more leakage and a higher logical error rate, but these can be suppressed by employing better strategies. Additionally, we also observed a reversal phenomenon where the logical error rate becomes larger when γ is small and d is large, which results in the disappearance of the threshold. We anticipate that such phenomena would

also be observable in standard $d \times d$ surface codes. We plotted the logical X error rates in the inset of Fig. 3.12. As we expected, the logical X error rates increase as code distance increases, but the leakage removal strategy is still effective in suppressing the noise effect.

In this chapter, we just use one CPU node to execute the simulation. For the $d \geq 5$ simulation, the cost of performing SVD becomes a crucial bottleneck of the computation time, so massive parallelization on cluster machines or fast SVD execution using GPUs is required. More realistic problem settings and large-scale simulations are left as topics for future research.

3.6 Discussion

In this chapter, we propose a method for simulating leakage processes in quantum error correcting codes of up to several hundred qutrits using the MPS method. This approach enables us to investigate the severe impact of coherent leakage errors and thermal noise on QEC codes without simplifications such as the general twirling approximation.

We demonstrate that applying such approximations can lead to substantially different performance estimates compared to the original noise model. For instance, we observe that an incoherent approximation to a coherent leakage noise model can significantly overestimate the logical error rate. The result of the numerical calculation emphasizes that the simplified indicators for leakage, like the leakage and seepage rates, fail to fully encapsulate the actual impact of leakage on the logical error rate. From a practical perspective, we also highlight the importance of various leakage removal strategies. We show that the effect of these strategies varies drastically depending on the specific type of leakage process.

Furthermore, this chapter also introduces a new application of quantum circuit simulations using tensor networks. In recent years, tensor-network methods for quantum computation [127, 128] have greatly advanced and have been employed for various tasks, such as the verification of quantum advantage [60–64, 103, 104] and for benchmarking and pre-training in variational quantum algorithms [57, 66, 129, 130]. To effectively apply tensor networks to a problem, it is essential to utilize geometric and entanglement structures of the system. In this chapter, we exploited a low-entanglement structure to discover non-trivial behavior in the processes of 1D quantum error correction codes under a leakage noise model.

Looking forward, it is expected that other tensor network algorithms will also be utilized for evaluating the memory performance beyond 1D and quasi-1D codes. For example, $d \times d$ surface code requires bond dimension that scales exponentially with d when using an MPS ansatz. To investigate codes with higher dimensional structure, more sophisticated tensor-network algorithms will likely be required, such as PEPS and isoTNS [131, 132].

Chapter 4

Tensor Networks for the Compilation of State Preparation Circuits of Multivariate Normal Distributions

The quantum state preparation of probability distributions is an important subroutine for many quantum algorithms. When embedding D-dimensional multivariate probability distributions by discretizing each dimension into 2^n points, we need a state preparation circuit comprising a total of nD qubits, which is often difficult to compile. In this chapter, we propose a scalable method to generate state preparation circuits for D-dimensional multivariate normal distributions, utilizing tree tensor networks (TTN). We establish theoretical guarantees that multivariate normal distributions with 1D correlation structures can be efficiently represented using TTN. Based on these analyses, we propose a compilation method that uses automatic structural optimization to find the most efficient network structure and compact circuit. We apply our method to state preparation circuits for various high-dimensional random multivariate normal distributions. The numerical results suggest that our method can dramatically reduce the circuit depth and CNOT count while maintaining fidelity compared to existing approaches. This chapter is based on [H. Manabe and Y. Sano, *Quantum* **9**, 1755 (2025)] with slight modifications to fit the context of this dissertation.

4.1 Introduction

Amplitude embedding of classical data is a key subroutine in various quantum algorithms such as quantum-linear-solver algorithms [133], Monte-Carlo methods [134], and quantum machine learning [135, 136]. In quantum state preparation, we encode 2^n classical data $f(x)$ into an n-qubit quantum state $|f(x)\rangle = U(x)|0\rangle$. Generally, achieving this requires $O(2^n)$ quantum operations [137, 138], which can become a critical bottleneck in the execution of algorithms. Thus, proposing efficient state preparation methods is essential for the practical implementation of quantum computer algorithms.

There are many studies on methods of quantum state preparation [6, 139–148]. For the Fault-Tolerant Quantum Computer (FTQC), it is possible to utilize Quantum Signal Processing-based [6] or oracle-based algorithms, where error bounds and the scaling of the gate counts can be rigorously proven. However, in the case of NISQ (Noisy Intermediate-Scale Quantum devices) and early-FTQC, there are strict limitations on executable circuit depth, connectivity, and the number of qubits. Therefore, it becomes essential to use classical optimizations to drastically reduce the circuit costs.

One of the efficient compilation methods for state preparation circuits is the Fourier series loader (FSL) method [147]. Suppose we want to encode a D-dimensional function $f(\boldsymbol{x})$, discretizing each dimension into 2^n points. Implementing this would require a state preparation circuit with a nD-qubit circuit and a depth of $O(2^{nD})$. The FSL method utilizes the exponential decay of the Fourier coefficients of smooth functions, allowing for state preparation in a depth of $O(2^{mD})$ with $m < n$ using Fourier series approximation. Nevertheless, as D increases, the circuit depth increases exponentially, making this method unsuitable for high-dimensional functions.

In this chapter, we propose a new method to compile the state preparation circuit for multivariate functions, particularly for multivariate normal distributions, which are widely used for many practical quantum algorithms in various settings [149–156]. We utilize a tensor network-based method [157–170], especially the Tree Tensor Network (TTN) ansatz [171], which is often used to represent quantum states with a 1D structure. We prove that a multivariate normal distribution can be efficiently represented as a TTN when there is a 1D structure in the correlations between variables. We demonstrate that by combining the FSL method and Tensor Cross Interpolation (TCI) [172, 173], it is possible to scalably prepare quantum states for multivariate normal distributions with 1D structures, independent of the dimension D and discretization parameter n.

Additionally, we propose a new approach that combines circuit compilation with automatic structural optimization [174]. Automatic structural optimization is a method to optimize network structure in TTN to represent the state more efficiently by reconnecting the legs of the local tensor throughout the process. This allows for the appropriate detection of structures in multivariate normal distributions and finding more compact circuits, even when the correlations between variables are unknown beforehand or when there is no 1D

structure.

We have demonstrated our method to compile state preparation circuits for several high-dimensional multivariate normal distributions. Our method can scalably generate state preparation circuits for multivariate normal distributions with 1D structures. We also numerically calculate the relationship between the fidelity of the circuit generated and the bond dimension, confirming the theoretical results. Furthermore, we investigated the impact of the automatic structural optimization algorithms on the performance of circuit compilation. Our numerical results show that the algorithm appropriately detects the 1D structure of a multivariate normal distribution and reconfigures the network with high success probability, even for large D. Finally, we show that we can reduce the circuit depth and CNOT count by factors of 10 to 1000 compared to existing methods for random multivariate normal distributions.

This chapter is organized as follows. We describe tensor network techniques used in this chapter in Sec. 4.2. We explain the proposed state preparation method in Sec. 4.3. In Sec. 4.4, we present the numerical results of our methods and discuss their performance. We conclude the chapter by summarizing our contribution in Sec. 4.5.

4.2 Tensor network methods

In this section, we briefly explain the tensor network methods used in this chapter. For a more detailed discussion on tensor networks, please refer to the reference [40].

4.2.1 Matrix product states and tree tensor networks

Tensor networks are a framework for representing a large tensor in the form of a network of smaller tensors. Specifically in quantum information theory, they are used to represent the wave function of quantum many-body systems. Matrix Product States (MPS), one type of tensor network ansatz, can be described as follows:

$$|\psi_{\mathrm{mps}}\rangle = \sum_{\{s_i\}} \sum_{\{\alpha_i\}} \left[A^{[1]s_1}_{\alpha_1} A^{[2]s_2}_{\alpha_1\alpha_2} \cdots A^{[n]s_n}_{\alpha_{n-1}} \right] |s_1 s_2 \cdots s_n\rangle \,, \tag{4.1}$$

where $\{s_i\}$ and $\{\alpha_i\}$ are referred to as physical and virtual indices, respectively. The dimension of the virtual indices is referred to as *bond dimension*.

The ability to represent a multi-dimensional array as an MPS can be evaluated using its singular values. Suppose that $\psi(s_1, \ldots, s_i; s_{i+1}, \ldots, s_n)$ is the unfolding matrix between site $(1, \ldots, i)$ and $(i+1, \ldots, n)$. Consider the singular value decomposition (SVD) of the matrix in the form

$$\psi(s_1, \ldots, s_i; s_{i+1}, \ldots, s_n) = U_i \Sigma_i V_i \tag{4.2}$$

and its singular values $\{\sigma^{(i)}_j\}$ sorted in descending order. Note that the summation of the square of the singular values is 1 for a wave function $|\psi\rangle$. Then,

there exists an MPS approximation $\tilde{\psi}$ with bond dimension r whose truncation error, in the sense of the Frobenius norm, is upper bounded by its singular values [58, 172]:

$$\left\|\psi(\boldsymbol{s}) - \tilde{\psi}(\boldsymbol{s})\right\|_F \leq \sqrt{\sum_{i=1}^{n-1} \epsilon_i^2}, \tag{4.3}$$

$$\epsilon_i^2 = 1 - \sum_{j=0}^{r-1} {\sigma_j^{(i)}}^2. \tag{4.4}$$

In other words, a multi-dimensional array is well-represented by an MPS if it has a small number of dominant singular values for each unfolding matrix.

The discussion above can be generalized to Tree Tensor Networks (TTN) [171]. A TTN is a type of tensor network ansatz that does not have closed loops in the diagrammatic representation. The inner edge divides the sites into two disjoint sets, which we call the *bipartition* of the system. Similarly to the case with MPS, the error upper bound in the best TTN approximation, in terms of the Frobenius norm, depends on the distribution of the singular values of the unfolding matrix across each bipartition induced by the edges in the network.

4.2.2 Functional matrix product states and tree tensor network

For the discussion later, we introduce the functional version of MPS, known as functional MPS (functional Tensor-Train decomposition) [175–178]. For $\psi \in L^2(\mathbb{R}^n)$, we consider the following decomposition:

$$\psi(x_1, \ldots, x_n) = \sum_{\{\alpha_i\}} \gamma_1(\alpha_0, x_1, \alpha_1)\gamma_2(\alpha_1, x_2, \alpha_2) \cdots \gamma_n(\alpha_{n-1}, x_n, \alpha_n). \tag{4.5}$$

To consider the error introduced by this approximation, we use an analogue of the singular value decomposition: the Schmidt decomposition. The Schmidt decomposition of the function ψ has the form

$$\psi(x_1, \ldots, x_i, x_{i+1}, \ldots, x_n) = \sum_k \sqrt{\lambda_k^{(i)}} \psi_k^{(1)}(x_1, \ldots, x_i)\psi_k^{(2)}(x_{i+1}, \ldots, x_n), \tag{4.6}$$

where $\psi_k^{(1)}$ and $\psi_k^{(2)}$ form an orthogonal basis for each subsystem, and $\{\sqrt{\lambda_k^{(i)}}\}$ are called *Schmidt coefficients* and sorted in descending order. Then, as in the discrete version, there exists a functional MPS approximation $\tilde{\psi}$ with bond dimension r whose truncation error, in the sense of the L^2 norm, is upper bounded by its Schmidt coefficients:

$$\|\psi - \tilde{\psi}\|_{L^2(\mathbb{R}^n)} \leq \sqrt{\sum_{i=1}^{n-1} \left(\sum_{k=r}^{\infty} \lambda_k^{(i)} \right)}. \tag{4.7}$$

These formulations for the functional MPS can similarly be extended to the functional TTN.

Consider tensorizing a function evaluated at points on a grid. When discretized on a sufficiently dense and large grid, the Schmidt decomposition (4.5) can be interpreted as the singular value decomposition (4.2), and the truncation error of the functional MPS (4.6) as that of discrete MPS (4.3). Therefore, if a function can be accurately approximated by a functional MPS, we can expect that its discretized multi-index tensor can similarly be approximated by a discrete MPS with the same bond dimension. In the following discussion, we consider functional MPS and functional TTN when evaluating approximation error.

4.2.3 Tensor cross interpolation

We can construct MPS or TTN from the original multi-index array by sequentially applying SVD. However, this method requires loading the unfolding matrix of the original tensor into memory, which becomes impractical when the tensor to be approximated itself is very large.

Instead, we can use the tensor cross interpolation (TCI) algorithm [172], a method for directly constructing MPS and TTN from the elements of the multi-dimensional tensor. Let $s_i \in \mathbb{K}_i = \{1, \ldots, d\}$. We define the following multi-indices:

$$s_{\leq i} = s_1 s_2 \cdots s_i \in \mathbb{K}_1 \times \mathbb{K}_2 \times \cdots \times \mathbb{K}_i, \quad s_{>i} = s_{i+1} \cdots s_n \in \mathbb{K}_{i+1} \times \cdots \times \mathbb{K}_n. \tag{4.8}$$

We interpolate the multi-index tensor as:

$$\begin{aligned} \psi(s_1, \ldots, s_n) \simeq & \psi(s_1, \mathcal{J}_{>1})[\psi(\mathcal{J}_{\leq 1}, \mathcal{J}_{>1})]^{-1} \psi(\mathcal{J}_{\leq 1}, s_2, \mathcal{J}_{>2})[\psi(\mathcal{J}_{\leq 2}, \mathcal{J}_{>2})]^{-1} \\ & \cdots \psi(\mathcal{J}_{\leq n-1}, s_n), \end{aligned} \tag{4.9}$$

where $\mathcal{J}_{\leq i}$ and $\mathcal{J}_{>i}$ are sets of multi-indices satisfying the following condition:

$$\mathcal{J}_{\leq i+1} \subset \mathcal{J}_{\leq i} \times \mathbb{K}_{i+1}, \quad \mathcal{J}_{>i} \subset \mathbb{K}_{i+1} \times \mathcal{J}_{>i+1}. \tag{4.10}$$

If the size of each $\mathcal{J}_{\leq i}, \mathcal{J}_{>i}$ is limited by χ, the original tensor is interpolated as an MPS with bond dimension χ. The choice of multi-indices defines the accuracy of the approximation. In this chapter, we use the ALS maxvol algorithm to find indices to use. For the details of the algorithm, please refer to the reference [172].

4.2.4 Canonical form of tree tensor network

For tree tensor networks, an efficient truncation method using the *canonical form* is well-known [110]. The diagram notation of the canonical form is illustrated in Fig. 4.1(a). We impose the isometric constraint on each tensor, except for the order-2 tensor known as the *top tensor*. We can efficiently transform any TTN into canonical form by repeatedly applying SVD to each tensor. When sweeping the position of the top tensor in canonical form, we can truncate

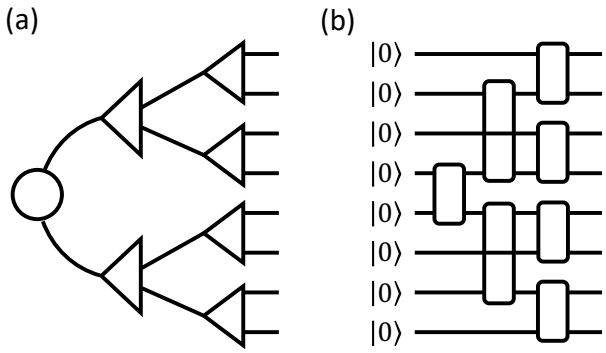


Figure 4.1: (a) The canonical form of the TTN. The top tensor is illustrated as the circle node and isometries as the triangle nodes. (b) The corresponding quantum circuit.

small singular values and reduce bond dimension, which achieves a near-optimal approximation in terms of the Frobenius norm.

After obtaining the canonical form for the 1D tensor network states, we can easily construct the state preparation circuit as illustrated in Fig. 4.1(b). The quantum circuit representation of the top tensor is achieved through the brute-force state preparation method. The other tensors are isometries and can be converted into circuits using unitary synthesis [179]. Even if the bond dimension is not a power of two, it can be represented as a quantum circuit by suitably padding tensors to become a power of two.

4.3 Methods

Our goal is to compile a quantum circuit to generate a quantized D-dimensional multivariate normal distribution by discretizing each dimension into 2^n points. In this chapter, we consider the following probability density function with mean zero for simplicity:

$$p(\boldsymbol{x}) = \frac{1}{\sqrt{(2\pi)^D \det(\Sigma)}} \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \Sigma^{-1} \boldsymbol{x}\right), \tag{4.11}$$

where $\boldsymbol{x}$ is a D-dimensional vector and Σ is the covariance matrix. Then we specify the parameter a such that $[-a/2, a/2]^D$ contains effectively all of the probability density. We use the binary encoding to discretize each dimension, such that

$$x_i = -\frac{a}{2} + \frac{a}{2^n} b_i, \quad 0 \le b_i \le 2^n - 1, \tag{4.12}$$

then our target state is expressed as follows:

$$\sum_{\boldsymbol{b}} f_{\boldsymbol{b}} \left|\boldsymbol{b}\right\rangle \equiv \sum_{\boldsymbol{b}} \left(\frac{a}{2^n}\right)^{D/2} \sqrt{p(\boldsymbol{x})} \left|\boldsymbol{b}\right\rangle. \tag{4.13}$$

4.3.1 Fourier series approximation

To scale for n, we adopt the technique of using discrete Fourier transform as proposed in [147]. The discrete Fourier transform over $\mathbb{Z}_{2^n}^D$ is expressed as follows:

$$\hat{f}_{\boldsymbol{k}} = \frac{1}{\sqrt{2^{nD}}} \sum_{b_1=0}^{2^n-1} \cdots \sum_{b_D=0}^{2^n-1} f_{\boldsymbol{b}} \exp\left(-2\pi i \frac{b_1 k_1 + \cdots + b_D k_D}{2^n}\right). \tag{4.14}$$

Then we approximate $f_{\boldsymbol{b}}$ by truncating Fourier coefficients as follows:

$$\tilde{f}_{\boldsymbol{b}} red = \frac{1}{\sqrt{2^{nD}}} \sum_{k_1=-M/2}^{M/2-1} \cdots \sum_{k_D=-M/2}^{M/2-1} \hat{f}_{\boldsymbol{k}} \exp\left(2\pi i \frac{b_1 k_1 + \cdots + b_D k_D}{2^n}\right). \tag{4.15}$$

Here, $M := 2^m$ represents the number of Fourier coefficients to be kept for each dimension. It is shown that, if $f(\cdot)$ is a sufficiently smooth function, the required number of m is smaller than n and, moreover, does not depend on n [147]. In practice, $\hat{f}_{\boldsymbol{k}}$ is calculated as follows:

$$\begin{aligned}\hat{f}_{\boldsymbol{k}} &\propto \sum_{\boldsymbol{b}} f_{\boldsymbol{b}} \exp\left(-2\pi i \frac{x_1 k_1 + \cdots + x_D k_D}{a}\right) \exp\left(-2\pi i \frac{k_1 + \cdots + k_D}{2}\right) \\ &\propto (-1)^{|k_1+\cdots+k_D|} \int_{-\frac{a}{2}}^{\frac{a}{2}} \cdots \int_{-\frac{a}{2}}^{\frac{a}{2}} \sqrt{p(\boldsymbol{x})} \exp\left(-2\pi i \frac{\boldsymbol{x}\boldsymbol{k}}{a}\right) d\boldsymbol{x}\end{aligned} \tag{4.16}$$

for a sufficiently fine discretization. Moreover, if a is taken to be sufficiently large, the following approximation also holds:

$$\hat{f}_{\boldsymbol{k}} \propto (-1)^{|k_1+\cdots+k_D|} \exp\left(-(2\pi/a)^2 \boldsymbol{k}^T \Sigma \boldsymbol{k}\right) \tag{4.17}$$

with an appropriate normalization term. Equation (4.17) shows that the value of $\hat{f}_{\boldsymbol{k}}$ decreases exponentially as the wavenumber $\boldsymbol{k}$ increases, indicating that retaining only a very small number of Fourier coefficients is sufficient to represent $f_{\boldsymbol{b}}$.

We first prepare the state that encodes the Fourier coefficients:

$$\sum_{k_1=-M/2}^{M/2-1} \cdots \sum_{k_D=-M/2}^{M/2-1} \hat{f}_{\boldsymbol{k}} |\boldsymbol{k}\rangle, \tag{4.18}$$

then we apply Inverse Quantum Fourier Transform (QFT) to obtain the target

state $\sum_{\boldsymbol{b}} \tilde{f}_{\boldsymbol{b}} |b\rangle$:

$$\begin{aligned}
&\mathrm{QFT}^\dagger \sum_{k_1=-M/2}^{M/2-1} \cdots \sum_{k_D=-M/2}^{M/2-1} \hat{f}_{\boldsymbol{k}} |\boldsymbol{k}\rangle \\
&= \sum_{\boldsymbol{k}} \hat{f}_{\boldsymbol{k}} \left(\frac{1}{\sqrt{2^{nD}}} \sum_{\boldsymbol{b}} \exp\left(2\pi i \frac{\boldsymbol{b}\cdot\boldsymbol{k}}{2^n}\right) |\boldsymbol{b}\rangle \right) \\
&= \sum_{\boldsymbol{b}} \left(\frac{1}{\sqrt{2^{nD}}} \sum_{\boldsymbol{k}} \hat{f}_{\boldsymbol{k}} \exp\left(2\pi i \frac{\boldsymbol{b}\cdot\boldsymbol{k}}{2^n}\right) \right) |\boldsymbol{b}\rangle \\
&= \sum_{\boldsymbol{b}} \tilde{f}_{\boldsymbol{b}} |\boldsymbol{b}\rangle . \qquad (4.19)
\end{aligned}$$

The circuit cost for preparing (4.18) is much smaller than that for $\sum_{\boldsymbol{b}} f_{\boldsymbol{b}} |b\rangle$, and Inverse QFT can be implemented efficiently in quantum computers with circuit depth n. Therefore, we can generate the desired state with a small cost that does not scale exponentially with the discretization size n.

4.3.2 Representing Fourier coefficients using tree tensor network

We now want to encode the truncated Fourier coefficients $\hat{f}_{\boldsymbol{k}}$ into the quantum circuit. In the case of D-dimensional probability distributions, however, the number of truncated Fourier coefficients is 2^{mD}, which can be a very large number.

Therefore, we consider representing the Fourier coefficients with an MPS:

$$\sum_{\boldsymbol{k}} \hat{f}_{\boldsymbol{k}} |\boldsymbol{k}\rangle = \sum_{\{k_i\}} \sum_{\{\alpha_i\}} \left[A^{[1]k_1}_{\alpha_1} A^{[2]k_2}_{\alpha_1\alpha_2} \cdots A^{[D]k_D}_{\alpha_{D-1}} \right] |k_1 k_2 \cdots k_D\rangle , \qquad (4.20)$$

or a TTN. The memory footprint required grows only polynomially with D.

Schmidt coefficients of multivariate normal distribution

As discussed above, the accuracy achievable in the MPS or TTN representation depends on its singular values or Schmidt coefficients. The Schmidt coefficients for a multivariate normal distribution are analytically known [180].

First, we discuss the Schmidt decomposition of the two-dimensional normal distribution. The Schmidt decomposition of the wave function $\psi(x_1, x_2)$ has the form

$$\psi(x_1, x_2) = \sum_{k=0}^{\infty} \sqrt{\lambda_k} \psi_k^{(1)}(x_1) \psi_k^{(2)}(x_2). \qquad (4.21)$$

Suppose the quantum state $\psi(x_1, x_2)$ is given by the square root of the density

function of the normal distribution with correlation coefficient ρ:

$$\psi(x_1, x_2) = \sqrt{p(x_1, x_2)}, \tag{4.22}$$

$$p(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left(-\frac{1}{2(1-\rho^2)}\left(\frac{(x_1-m_1)^2}{\sigma_1^2} - 2\rho\frac{(x_1-m_1)(x_2-m_2)}{\sigma_1\sigma_2} + \frac{(x_2-m_2)^2}{\sigma_2^2}\right)\right), \tag{4.23}$$

where m_1, m_2 are the mean values and σ_1^2, σ_2^2 are the variance of each variable. The Schmidt decomposition of this wave function [181] is given by:

$$\psi_k^{(1)}(x_1) = C_k^{(1)} H_k\left(\frac{(x_1-m_1)}{\sigma_1}\sqrt{\frac{K}{2}}\right)\exp\left(-K\frac{(x_1-m_1)^2}{4\sigma_1^2}\right), \tag{4.24}$$

$$\psi_k^{(2)}(x_2) = C_k^{(2)} H_k\left(\frac{(x_2-m_2)}{\sigma_2}\sqrt{\frac{K}{2}}\right)\exp\left(-K\frac{(x_2-m_2)^2}{4\sigma_2^2}\right), \tag{4.25}$$

where H_k denotes the k-th Hermite polynomial, $C_k^{(1)}, C_k^{(2)}$ are the normalization factors and K is calculated by

$$K = \frac{1}{\sqrt{1-\rho^2}}. \tag{4.26}$$

The squares of Schmidt coefficients $\{\lambda_k\}$ form a geometric series with an initial value

$$\lambda_0 = \frac{2}{K+1} \tag{4.27}$$

and common ratio:

$$q = \frac{K-1}{K+1}. \tag{4.28}$$

An approximation error of an MPS of bond dimension r is calculated as

$$\begin{aligned}\|\psi - \psi_{\mathrm{mps}}\|^2_{L^2(\mathbb{R}^2)} &\leq \sum_{k=r}^{\infty}\lambda_k \\ &= \lambda_0\sum_{k=r}^{\infty} q^k \\ &= \lambda_0\frac{q^r}{1-q} = q^r.\end{aligned} \tag{4.29}$$

Then, if we want to suppress the truncation error to ϵ, we need the bond dimension r that satisfies:

$$q^r \leq \epsilon^2 \iff r \geq \frac{2}{\log\frac{1}{q}}\log\frac{1}{\epsilon}. \tag{4.30}$$

In other words, the required bond dimension scales as $O(\log(1/\epsilon))$, which is an efficient approximation.

The distribution of the Schmidt coefficients of the multivariate normal distribution is also known and calculated only from the covariance matrix [180]. For simplicity, we consider the multivariate normal distribution with zero mean. The Schmidt decomposition of the multivariate function for subsystems $x_1, \ldots, x_p$ and $x_{p+1}, \ldots, x_D$, assuming that $2p \leq D$, is defined as follows:

$$\psi(x_1, \ldots, x_p, x_{p+1}, \ldots, x_D) = \sum_k \sqrt{\lambda_k} \psi_k^{(1)}(x_1, \ldots, x_p) \psi_k^{(2)}(x_{p+1}, \ldots, x_D). \tag{4.31}$$

According to the canonical-correlation analysis [182], linear transformations can be found for variables $\xi_1, \ldots \xi_p, \xi_{p+1}, \ldots, \xi_D$:

$$\xi_i = \sum_{j=1}^{p} l_{ij} x_j, \qquad 1 \leq i \leq p \tag{4.32}$$

$$\xi_i = \sum_{j=p+1}^{D} m_{ij} x_j, \quad p+1 \leq i \leq D \tag{4.33}$$

that satisfy the following properties: (1) All ξ have unit variance and zero mean and (2) the correlation between any ξ is zero except for the pairs (ξ_1, ξ_{p+1}), $(\xi_2, \xi_{p+2}), \ldots, (\xi_p, \xi_{2p})$. The coefficients $\boldsymbol{l}, \boldsymbol{m}$ and the canonical correlation coefficients $\{\rho_i\}$ between these pairs are the solutions to the following generalized eigenvalue problem:

$$\begin{pmatrix} O & \Sigma_{12} \\ \Sigma_{21} & O \end{pmatrix} \begin{pmatrix} \boldsymbol{l} \\ \boldsymbol{m} \end{pmatrix} = \rho \begin{pmatrix} \Sigma_{11} & O \\ O & \Sigma_{22} \end{pmatrix} \begin{pmatrix} \boldsymbol{l} \\ \boldsymbol{m} \end{pmatrix}, \tag{4.34}$$

where $\Sigma_{11}, \Sigma_{12}, \Sigma_{21}, \Sigma_{22}$ are the submatrices of the covariance matrix:

$$\Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}. \tag{4.35}$$

$D - 2p$ solutions are zero and the others arise in pairs as $\pm\rho_1, \pm\rho_2, \ldots, \pm\rho_p$. We choose the positive ones. Then, the target function is decomposed as the tensor product of the probability density function of the two-dimensional normal distribution $(\xi_1, \xi_{p+1}), \ldots, (\xi_p, \xi_{2p})$ with correlation $\rho_1, \ldots, \rho_p$, and one-dimensional normal distribution $\xi_{2p+1}, \ldots, \xi_D$:

$$\begin{aligned} &\psi(x_1, \ldots, x_p, x_{p+1}, \ldots, x_D) = \\ &\prod_{i=1}^{p} \left(\sum_{k_i} \sqrt{\lambda_{k_i}^{(i)}} \psi_{k_i}^{(1)}(\xi_i) \psi_{k_i}^{(2)}(\xi_{p+i}) \right) \cdot \sqrt{p(\xi_{2p+1})} \cdots \sqrt{p(\xi_D)}, \end{aligned} \tag{4.36}$$

where $\{\lambda^{(i)}\}$ is the Schmidt coefficients for the pair (ξ_i, ξ_{p+i}). The Schmidt decomposition of the multivariate normal distribution is given by defining mode

$\boldsymbol{k} = (k_1, \ldots, k_p)$ and the corresponding orthogonal basis as

$$\psi^{(1)}_{\boldsymbol{k}}(x_1, \ldots, x_p) = \prod_{i=1}^{p} \psi^{(1)}_{k_i}(\xi_i), \tag{4.37}$$

$$\psi^{(2)}_{\boldsymbol{k}}(x_{p+1}, \ldots, x_D) = \left(\prod_{i=1}^{p} \psi^{(2)}_{k_i}(\xi_{p+i}) \right) \cdot \sqrt{p(\xi_{2p+1})} \cdots \sqrt{p(\xi_D)}, \tag{4.38}$$

and the Schmidt coefficients are the product of each Schmidt coefficient of the pairs:

$$\sqrt{\lambda_{\boldsymbol{k}}} = \prod_{i=1}^{p} \sqrt{\lambda^{(i)}_{k_i}}. \tag{4.39}$$

Each $\{\lambda^{(i)}\}$ can be calculated from ρ_i and equation (4.26)-(4.28). Therefore, by solving the generalized eigenvalue problem (4.34), we can enumerate all the Schmidt coefficients of the multivariate normal distribution between any bipartition of the system and calculate the required bond dimensions for the tensor network approximation. The computational cost of calculating the above values only depends polynomially on the dimension D, not on the discretization size n.

Schmidt coefficients of a multivariate normal distribution with 1D covariance structure

The discussion above can be applied to multivariate normal distributions with any covariance matrix and bipartition. However, if we restrict to the 1D covariance structure, we can guarantee the existence of efficient MPS or TTN representations of the target state. Note that a priori rank bounds for approximations in the MPS representation for the multivariate normal distribution with weak correlations are studied in [178], This chapter differs in that we directly use the information of the Schmidt coefficients for the proof.

Theorem 1. *Let $p(\boldsymbol{x})$ be a probability density function of a D-dimensional multivariate normal distribution with zero mean and covariance matrix Σ. Assume that, for any bipartition of variables induced by an inner edge of the tree tensor network,* $\operatorname{rank} \Sigma_{12} \leq l$ *where l is constant and the canonical correlations are less than 1. For every ϵ there exists a TTN approximation $\tilde{\psi}$ of $\psi(\boldsymbol{x}) = \sqrt{p(\boldsymbol{x})}$ with*

$$\|\psi(\boldsymbol{x}) - \tilde{\psi}(\boldsymbol{x})\|_{L^2(\mathbb{R}^D)} \leq \epsilon \tag{4.40}$$

whose bond dimension χ is upper bounded by

$$\chi \leq \left(C \cdot \log \frac{l\sqrt{D}}{\epsilon} \right)^l \tag{4.41}$$

for some constant C.

This theorem shows that the required bond dimension scales only polylogarithmically in $1/\epsilon$, which indicates this kind of multivariate normal distribution can be efficiently represented as a TTN.

Proof. From (4.7), if the sum of the squares of the Schmidt coefficients for all bipartitions is less than ϵ^2/D, then an MPS approximation that satisfies (4.40) exists.

Let $q = \max_i q_i$, where q_i is the common ratio of the squared Schmidt coefficients for the pair (ξ_i, ξ_{p+i}). Consider keeping the r^l-th largest Schmidt coefficients and truncating all other small values. The truncation error is upper-bounded as follows:

$$\begin{aligned}\sum_{k=r^l}^{\infty} \lambda_k &\leq 1 - (1-q^r)^l \\ &\leq 1 - (1 - l \cdot q^r) = l \cdot q^r, \end{aligned} \tag{4.42}$$

where we use the Weierstrass product inequality and $q^r < 1$. If we want to suppress the error by ϵ^2/D, we have to satisfy the following inequality:

$$l \cdot q^r \leq \epsilon^2/D. \tag{4.43}$$

By solving this we get

$$r \geq \frac{2}{\log \frac{1}{q}} \log \frac{l\sqrt{D}}{\epsilon}. \tag{4.44}$$

□

Example 1. *The multivariate normal distribution with uniform correlation coefficients. Consider the following covariance matrix:*

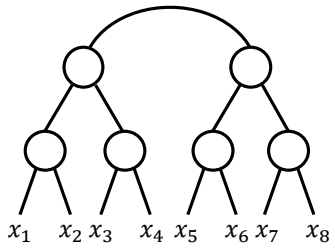
$$\Sigma = \begin{pmatrix} 1 & \rho & \rho & \cdots & \rho \\ \rho & 1 & \rho & \cdots & \rho \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \rho & \cdots & 1 \end{pmatrix}, \tag{4.45}$$

with $-1 < \rho < 1$. Then any bipartition of the system gives rank-one submatrix Σ_{12}. Therefore, this probability distribution can be efficiently represented as an MPS with any index ordering.

Example 2. *The multivariate normal distribution with 1D correlations. Consider the following covariance matrix:*

$$\Sigma = \begin{pmatrix} 1 & \rho & \rho^2 & \cdots & \rho^D \\ \rho & 1 & \rho & \cdots & \rho^{D-1} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^D & \rho^{D-1} & \rho^{D-2} & \cdots & 1 \end{pmatrix}, \tag{4.46}$$

with $-1 < \rho < 1$. Then, a bipartition of subsystems $x_1, \ldots, x_i$ and $x_{i+1}, \ldots, x_D$ gives a rank-one submatrix Σ_{12}. Therefore, this probability distribution can be efficiently represented as an MPS with the same ordering.



$$\Sigma = \begin{pmatrix} 1 & \rho & \rho^3 & \rho^3 & \rho^5 & \rho^5 & \rho^5 & \rho^5 \\ \rho & 1 & \rho^3 & \rho^3 & \rho^5 & \rho^5 & \rho^5 & \rho^5 \\ \rho^3 & \rho^3 & 1 & \rho & \rho^5 & \rho^5 & \rho^5 & \rho^5 \\ \rho^3 & \rho^3 & \rho & 1 & \rho^5 & \rho^5 & \rho^5 & \rho^5 \\ \rho^5 & \rho^5 & \rho^5 & \rho^5 & 1 & \rho & \rho^3 & \rho^3 \\ \rho^5 & \rho^5 & \rho^5 & \rho^5 & \rho & 1 & \rho^3 & \rho^3 \\ \rho^5 & \rho^5 & \rho^5 & \rho^5 & \rho^3 & \rho^3 & 1 & \rho \\ \rho^5 & \rho^5 & \rho^5 & \rho^5 & \rho^3 & \rho^3 & \rho & 1 \end{pmatrix}$$

Figure 4.2: An example of a multivariate normal distribution with a tree-like structure. The covariance decays exponentially with the distance on the tree between variables.

Example 3. *The multivariate normal distribution with tree-like correlations. Consider the elements of the covariance matrix are given by its distance:*

$$\Sigma_{ij} = \exp\left(-\frac{\text{distance}(i,j)}{\sigma}\right), \tag{4.47}$$

where the distance between site i and j is given as the length of the path on the tree as shown in Fig 4.2. Then, the bipartition of the system induced by each edge of the tree gives rank-one submatrix Σ_{12} and this probability distribution can be efficiently represented as a TTN. This generalizes the above two examples.

In addition to the case where the rank of the submatrix is upper-bounded, we can consider the following situation where correlation coefficients decay exponentially. In this case, however, the required bond dimension for approximation scales polynomially in $1/\epsilon$ instead of $\log(1/\epsilon)$.

Theorem 2. *Let $p(\boldsymbol{x})$ be a probability density function of a D-dimensional multivariate normal distribution with zero mean and covariance matrix Σ. Suppose the canonical correlations are less than 1 and decay exponentially:*

$$\rho_j \leq \alpha e^{-\theta \cdot j} \quad \alpha, \theta > 0 \tag{4.48}$$

for any bipartition of variables induced by an inner edge of the tree tensor network. For every ϵ there exists a TTN approximation $\tilde{\psi}$ of $\psi(\boldsymbol{x}) = \sqrt{p(\boldsymbol{x})}$ with

$$\|\psi(\boldsymbol{x}) - \tilde{\psi}(\boldsymbol{x})\|_{L^2(\mathbb{R}^D)} \leq \epsilon \tag{4.49}$$

whose bond dimension is upper-bounded by

$$\chi \leq \left(\frac{C_2\sqrt{D}}{\epsilon}\right)^{C_1 \log\left(C_3 \log\left(\frac{2C_1 \log\left(\frac{C_2\sqrt{D}}{\epsilon}\right)\sqrt{D}}{\epsilon}\right)\right)} \tag{4.50}$$

for some constants C_1, C_2, C_3.

This theorem indicates that the necessary bond dimension scales as $\chi = O(\mathrm{poly}(\frac{C_2\sqrt{D}}{\epsilon}))$ if we ignore the log term in the exponent. The efficiency of the approximation is worse compared to the case where the rank of the covariance submatrix is strictly upper-bounded.

Proof. By (4.28),

$$q_j = \frac{K_j - 1}{K_j + 1} = \frac{\left(1 - \sqrt{1 - \rho_j^2}\right)^2}{\rho_j^2} < \rho_j \quad (0 < \rho_j < 1). \tag{4.51}$$

Therefore, when ρ_j decays exponentially, the common ratio q_j also decays exponentially to j.

Consider the two-step approximation: First, we retain only the largest singular value for pairs (ξ_i, ξ_{p+i}) with $i > l$. Then the approximation error by introducing this is calculated as

$$\begin{aligned} \|\psi(\boldsymbol{x}) - \overline{\psi}(\boldsymbol{x})\|^2_{L^2(\mathbb{R}^D)} &= 1 - (1 - q_{l+1})(1 - q_{l+2}) \cdots (1 - q_p) \\ &\leq 1 - (1 - \alpha e^{-\theta(l+1)})(1 - \alpha e^{-\theta(l+2)}) \cdots (1 - \alpha e^{-\theta p}) \\ &\leq \sum_{i=l+1}^{p} \alpha e^{-\theta i} \leq \alpha \frac{e^{-\theta l}}{1 - e^{-\theta}}. \end{aligned} \tag{4.52}$$

By solving the following inequality:

$$\alpha \frac{e^{-\theta l}}{1 - e^{-\theta}} \leq \frac{\epsilon^2}{4D}, \tag{4.53}$$

we get $l = C_1 \log\left(C_2\sqrt{D}/\epsilon\right)$ for constant numbers C_1, C_2.

Next, we consider approximating $\overline{\psi}(\boldsymbol{x})$ with finite Schmidt coefficients. Now the covariance matrix for the multivariate normal distribution $\overline{\psi}(\boldsymbol{x})$ only have l non-zero correlation coefficients. According to theorem 1, we can approximate it with error $\epsilon/(2\sqrt{D})$ by using r^l Schmidt coefficients and $r = C_3 \log\left(2l\sqrt{D}/\epsilon\right)$ for constant number C_3. Then we get

$$\begin{aligned} \|\psi(\boldsymbol{x}) - \tilde{\psi}(\boldsymbol{x})\|_{L^2(\mathbb{R}^D)} &\leq \|\psi(\boldsymbol{x}) - \overline{\psi}(\boldsymbol{x})\|_{L^2(\mathbb{R}^D)} + \|\overline{\psi}(\boldsymbol{x}) - \tilde{\psi}(\boldsymbol{x})\|_{L^2(\mathbb{R}^D)} \\ &\leq \epsilon/\sqrt{D}. \end{aligned} \tag{4.54}$$

Therefore, the necessary bond dimension to approximate the entire quantum

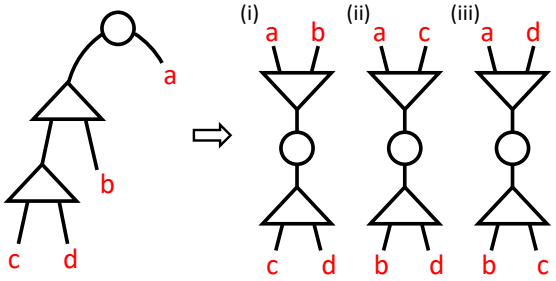


Figure 4.3: The process of automatic structural optimization of the TTN. When sweeping the top tensor, there are three patterns for choosing indices. By selecting the pattern that results in the lowest entanglement entropy, calculated from the singular values contained in the top tensor, the structure of TTN can be automatically optimized to the most suitable configuration.

state as TTN within precision ϵ scales as

$$\begin{aligned} r^l &= C_3 \log \left(\frac{2l\sqrt{D}}{\epsilon} \right)^l \\ &= C_3 \log \left(\frac{2C_1 \log\left(\frac{C_2\sqrt{D}}{\epsilon}\right)\sqrt{D}}{\epsilon} \right)^{C_1 \log\left(\frac{C_2\sqrt{D}}{\epsilon}\right)} \\ &= \left(\frac{C_2\sqrt{D}}{\epsilon} \right)^{C_1 \log\left(C_3 \log\left(\frac{2C_1 \log\left(\frac{C_2\sqrt{D}}{\epsilon}\right)\sqrt{D}}{\epsilon} \right)\right)} . \end{aligned} \tag{4.55}$$

□

4.3.3 Automatic structural optimization of the TTN

As discussed above, the required bond dimension for achieving enough accuracy can be estimated by the distribution of Schmidt coefficients among variables that are bipartitioned by the network. Therefore, whether a TTN can efficiently represent a multivariate normal distribution depends strongly on the tensor network structure. Even if a multivariate normal distribution has a one-dimensional correlation structure as in Th. 1 and 2, an efficient approximation is no longer guaranteed if an appropriate TTN corresponding to that structure is not used. In the case of a normal distribution, the optimal network structure can be determined by enumerating all network configurations and estimating their fidelity from the correlation coefficients. However, for large D, it becomes increasingly impractical, as there are $O(D!)$ ways to arrange variables, even when restricted to MPS.

Instead, in this chapter, we employ automatic structural optimization [174] to optimize network configuration. Fig. 4.3 shows the outline of the algorithm. Initially, we execute TCI using an initial network structure (e.g., MPS) to represent the multivariate normal distribution. Then, we sweep the top tensor in the network using the canonical form. During this process, we contract the three tensors in Fig. 4.3 and obtain T_{abcd}. We consider three ways of singular value decomposition:

$$T_{abcd} = U_{abe} s_e V_{ecd}, \tag{4.56}$$

$$T_{abcd} = U_{ace} s'_e V_{ebd}, \tag{4.57}$$

$$T_{abcd} = U_{ade} s''_e V_{ebc} \tag{4.58}$$

and compute the entanglement entropy as follows:

$$EE^{(ab|cd)} = -\sum_e s_e^2 \log s_e^2, \tag{4.59}$$

$$EE^{(ac|bd)} = -\sum_e s_e'^2 \log s_e'^2, \tag{4.60}$$

$$EE^{(ad|bc)} = -\sum_e s_e''^2 \log s_e''^2, \tag{4.61}$$

where we assumed that TTN is normalized. Among them, we choose the structure with the smallest entanglement entropy. By repeating this process, the entire TTN network is automatically optimized to reflect the entanglement structure of the system.

Entanglement entropy takes small values when the singular values decay rapidly. As discussed previously, the faster the correlation coefficients decay, the more rapidly the Schmidt coefficients decrease. Thus, in this context, entanglement entropy indicates the strength of correlation when dividing variables into two groups. The process of automatic structural optimization can be interpreted as assuming a tree-like correlation structure in the probability distribution and searching the most suitable network configuration where the correlation between variables is minimized.

We first execute TCI using an initial tree tensor network with a sufficiently large bond dimension enough to accurately represent the multivariate normal distribution. Then, we employ structural optimization to refine the network structure and reduce the bond dimension through truncation using the canonical form. If the accuracy of the state obtained is insufficient, we can rerun TCI using the optimized network structure to achieve higher accuracy.

4.3.4 Optimizing the quantum circuit by tree tensor network

Up to this point, we have discussed how to represent Fourier coefficients using tensor networks efficiently. After encoding the Fourier coefficients as a quantum

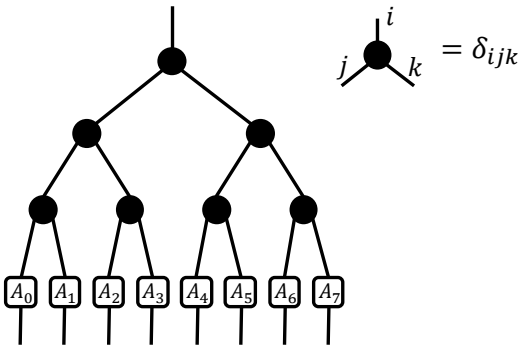


Figure 4.4: The tree tensor network representing inverse QFT. The black node represents a copy tensor. The definition of tensors $\{A_j\}_{j=0}^{n}$ is given in the main text. The number of qubits for each dimension is set as $n = 8$ in this figure.

circuit, we can get the discretized probability distribution by applying inverse QFT with a depth of n.

We can further compile the entire state preparation circuit. When expressing Fourier coefficients with an MPS or TTN, a bond dimension 2^m is inevitably required for a physical index. This bond dimension can be reduced by utilizing the fact that the inverse QFT circuit for each dimension can be represented as a TTN, as illustrated in Fig 4.4. Here, the black node in TTN represents a copy tensor δ_{ijk} with bond dimension 2^m and A_j is a $2^k \times 2$ tensor whose elements are given as

$$(A_j)_l^k = e^{2\pi ikl/2^{j+1}}. \tag{4.62}$$

We then apply bond truncation algorithms using the canonical form. While this approach loses the advantage of executing inverse QFT with a circuit depth of n, it allows for the potential creation of a more compact circuit. Additionally, as the size of each tensor decreases, the classical resources required for the circuit synthesis of each tensor are reduced.

Fig. 4.5 summarizes the comprehensive workflow for synthesizing the state preparation circuit developed in this chapter. For a target multivariate normal distribution, we first employ TCI to represent its Fourier coefficients $\hat{f}_{\boldsymbol{k}}$ within a TTN framework. This is followed by automatic structural optimization to identify the most suitable network topology that reflects the underlying correlation structure. If the initial approximation is insufficient, the TCI process may be iteratively repeated using the optimized network structure. Subsequently, the inverse QFT, represented as a TTN, is applied to obtain the target distribution $\tilde{f}_{\boldsymbol{b}}$. The entire tensor network is further compressed through bond truncation utilizing the canonical form. Finally, the resulting TTN is compiled into a quantum circuit. Bond truncation drastically reduces the memory size of each local

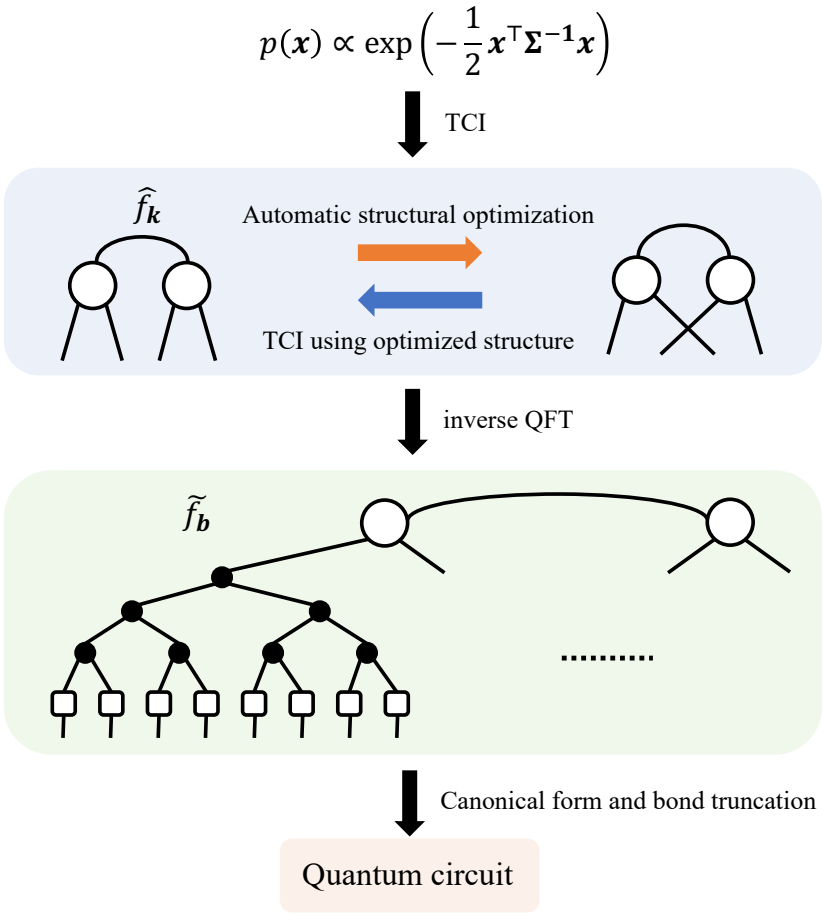


Figure 4.5: Overview of the proposed method for generating a state preparation circuit for a multivariate normal distribution. The procedure involves representing the Fourier coefficients as a TTN via TCI, performing automatic structural optimization, applying the inverse QFT, compressing the bond dimension, and finally converting the TTN into a quantum circuit.

tensor compared to the original high-dimensional representation. This reduction renders isometry synthesis of the local tensor computationally feasible. By synthesizing unitary circuits for each individual tensor and connecting them according to the network topology, we can efficiently map the entire TTN into a quantum circuit, yielding the desired state preparation circuit.

4.4 Results

Here we present our simulation results for the compilation method of the state preparation circuit of multivariate normal distribution described in Sec. 4.3. We set the parameters as $a = 20$ ($x \in [-a/2, a/2]^D$), $n = 8$ (discretizing each dimension into 2^n points) and $m = 5$ (keep 2^m Fourier coefficients for each dimension) to discretize the probability density function with sufficient accuracy. We first use TCI to express the Fourier coefficients (4.20) as an MPS with bond dimension χ', where χ' is set sufficiently large to represent the target Fourier coefficients. In this chapter, we set $\chi' = 64$.

The fidelity between the original and approximated states of the i-th step of truncation can be calculated using the singular values as follows:

$$f^{(i)} = \sum_{j=0}^{\chi-1} {\sigma_j^{(i)}}^2. \tag{4.63}$$

Note that we normalize the quantum state after each truncation step so that the summation of the squares of singular values is always one. After completing the truncation process, the fidelity of the final state is predicted as the product of the fidelity at each step:

$$f = \prod_i f^{(i)}. \tag{4.64}$$

This approach serves as a good estimate. Subsequently, the Frobenius norm is restored using the following equation:

$$\| \, |\psi\rangle - |\tilde{\psi}\rangle \, \|_F = \sqrt{2(1 - \sqrt{f})}. \tag{4.65}$$

The tensor-network algorithms are implemented using the Python library Quimb [183]. Each data point on the plots is obtained by averaging 100 random instances.

4.4.1 Multivariate normal distribution with low-rank structure

To numerically test the theorem 1, we represented a multivariate normal distribution with a low-rank covariance matrix using MPS. We utilized a simple covariance matrix with a low-rank structure similar to the Example 2, with σ uniformly sampled from $[0, \sigma_{\max}]$ and $\sigma_{\max} = 0.5$. We consider a set of l multivariate normal distributions. We partition the variables into non-overlapping subgroups $\{x_i^{(j)}\}$ as follows: for $1 \leq j \leq l$, let

$$x_1^{(j)} \in \{x_1, x_2, \ldots, x_l\}, \quad x_2^{(j)} \in \{x_{l+1}, \ldots, x_{2l}\}, \quad \cdots. \tag{4.66}$$

Each subgroup follows a multivariate normal distribution with a 1D structure. For the probability density function constructed in this manner, the condition on the rank of the covariance matrix $\mathrm{rank}(\Sigma_{12}) \leq l$ in Theorem 1 is satisfied.

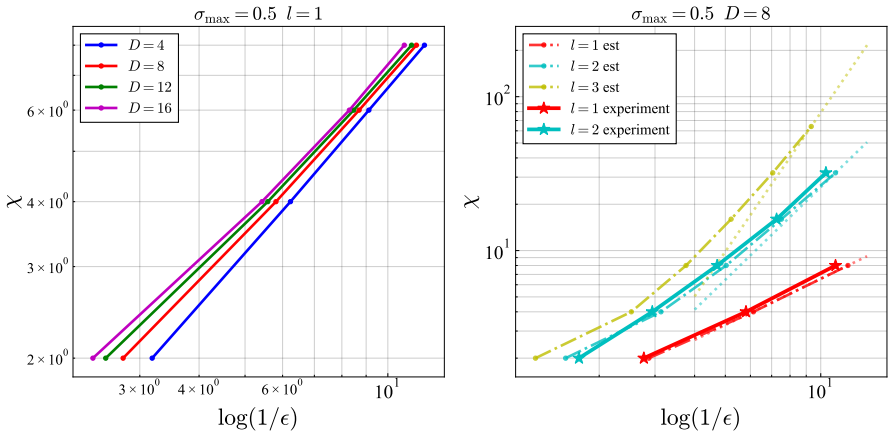


Figure 4.6: The bond dimension of the MPS is plotted as a function of the accuracy $\log(1/\epsilon)$ (Left) for various dimensions of the multivariate normal distribution and (right) for various ranks of the submatrix of the covariance matrix. "est" represents the estimation value from the distribution of the Schmidt coefficients and "experiment" represents the actual value from the numerical experiment. The dotted lines represent the predicted polynomial growth rate in $\log(1/\epsilon)$.

In Fig. 4.6, we illustrate the relationship between the accuracy obtained from MPS approximation and the corresponding bond dimension. In the left figure, the rank l is fixed at 1 and the dimension D is varied. As can be seen from the figure, the required bond dimension increases linearly with $\log(1/\epsilon)$. Moreover, the required bond dimension grows only gradually as D increases, suggesting that this approach scales effectively with dimension.

In the right plot, the dimension D is fixed, and the rank l is varied. The dash-dot line uses the distribution of Schmidt coefficients to directly estimate the achievable accuracy from the bond dimension, while the solid line calculates accuracy by obtaining the actual MPS. The results confirm the Theorem 1 that the required bond dimension increases with $\log(1/\epsilon)^l$. Furthermore, the accuracy obtained from the actual MPS is found to be very close to that predicted from the Schmidt coefficients, confirming the effectiveness of this bound in estimating the necessary bond dimension given the target accuracy.

4.4.2 Multivariate normal distribution with exponentially decaying correlation coefficients

Next, we represent a multivariate normal distribution in which canonical correlations decay exponentially. To construct such a distribution, we consider the following setup: the sites are aligned in 1D and the covariance between site i

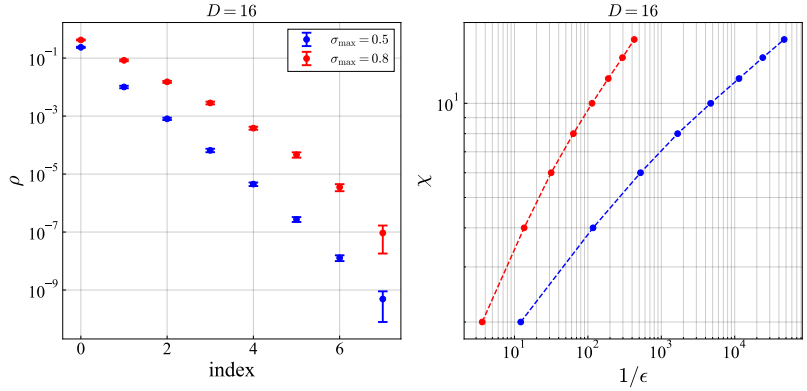


Figure 4.7: (left) The distribution of the canonical correlation coefficients when the system with $D = 16$ is divided in half. (right) The bond dimension of the MPS is plotted as a function of the accuracy $1/\epsilon$. Different maximum values of covariance $\sigma_{\max}$ are plotted in different colors.

and j is defined as

$$\mathrm{cov}_{ij} = (\sqrt{\sigma_i \sigma_j})^{|i-j|}, \tag{4.67}$$

where σ_i is uniformly sampled from $[0, \sigma_{\max}]$. We numerically verified that the correlations decay exponentially with the distance between sites. Fig. 4.7 (left) shows the distribution of canonical correlations between sites 1 to 8 and 9 to 16 for the multivariate normal distribution with dimension $D = 16$. As can be seen from the figure, the canonical correlations decay exponentially with the index number, satisfying the condition $\rho_j \leq \alpha e^{-\theta j}$ in Theorem 2.

Fig. 4.7 (right) shows the relationship between accuracy and the required bond dimension for approximating the above multivariate normal distribution with MPS. Note that the x-axis represents $1/\epsilon$, not $\log(1/\epsilon)$. As can be seen, polynomial growth in $1/\epsilon$ is confirmed, which is consistent with Theorem 2. These numerical results demonstrate that when correlations decay exponentially in 1D, it is possible to efficiently represent them using MPS regardless of the dimension D, although the efficiency significantly decreases compared to cases when the rank is strictly bounded.

4.4.3 Structural optimization with 1D tree structure

We also evaluated the performance of the structural optimization introduced in Sec. 4.3.3. We consider a multivariate normal distribution with a tree-like structure as presented in the Example 3. The initial structure of a tree is generated randomly. We approximate the distribution using a standard MPS using TCI without considering the covariance structure. Then, we employed automatic structural optimization to modify the tensor network, as depicted in Fig. 4.8 (a), and verify whether it could restore the original tree structure.

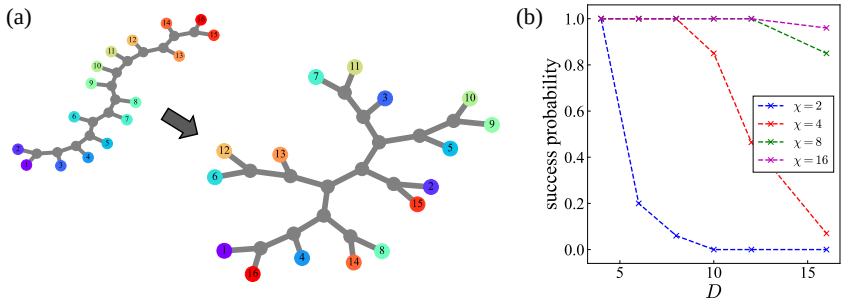


Figure 4.8: (a) Initially, the distribution is represented as an MPS, and after automatic structural optimization, the tree-like structure is recovered. (b) The success probability of the structural optimization is plotted as a function of the dimension D. Different bond dimensions in the tensor network are represented in different colors.

Fig. 4.8 (b) displays the relationship between the dimension D of the multivariate normal distribution and the success rate of structural optimization, i.e. the probability of completely restoring the original structure. Here, χ represents the maximum bond dimension of the tensor network during the process. We see that even for high-dimensional distributions like $D = 16$, automatic structural optimization can replicate the original structure with high probability. Additionally, increasing the bond dimension provides more accurate information about the bipartite entanglement entropy of the system, allowing a higher success rate. These numerical results indicate that despite being a heuristic algorithm, structural optimization is extremely powerful in capturing the 1D structure of distributions.

4.4.4 State preparation of random normal distribution

Now we move on to the compilation of the state preparation circuits for the multidimensional normal distribution without any 1D or tree-like structure.

Before discussing the results, we outline how the generated circuits are evaluated. We assume all-to-all connectivity. The main metrics adopted here are the number of CNOT gates and the depth of the circuit, both of which are very important, especially for NISQ computers. Regarding the number of CNOT gates, it is known that an isometry with a p-qubit input and a q-qubit output in the tensor network can be implemented using $O(2^{p+q})$ CNOT gates [179]. For simplicity and comparison purposes, we treat this number as 2^{p+q}, essentially equating it to the memory size required by the tensor network. Circuit depth is the necessary depth for implementing all two-qubit gates and is calculated by identifying the longest path from the root to the leaf within the TTN. If there is an isometry of size 2^{p+q} along a path, then 2^{p+q} is added to the circuit depth. Using these metrics, we estimate the cost of the circuits. Actual circuit

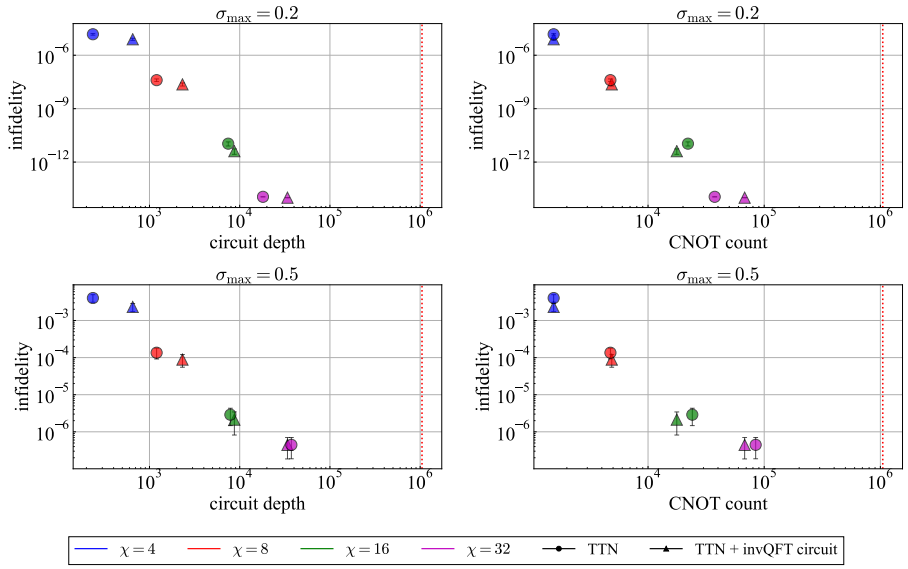


Figure 4.9: The circuit depth and CNOT count of the encoding circuit, and the infidelity of the quantum state generated by the encoding circuit. The parameters are set as $\sigma_{\max} = 0.2, 0.5$. The circle points show the result by representing inverse QFT as a TTN, discussed in Sec. 4.3.4, and triangle points show the result by using standard inverse QFT circuit. The red line implies the result without MPS approximation.

synthesis on quantum hardware is beyond the scope of this chapter and is left for future investigation.

We consider preparing the quantum state of a 4-dimensional multivariate normal distribution with a randomly generated covariance matrix. Each element of the covariance matrix is generated from the uniform random distribution $[-\sigma_{\max}, \sigma_{\max}]$, except for the diagonal elements, which are all 1.0 for regularization.

In Fig. 4.9, we present the circuit depth, CNOT count, and infidelity of the encoding circuit generated by our method. For comparison, we also plot the costs using the FSL method without tensor network approximations, which serves as a baseline [147]. As shown in the figure, our approach compiles circuits that achieve high accuracy while using 10 to 1000 times fewer resources than the baseline. Additionally, by using a lower bond dimension, it is possible to perform state preparation at a lower cost, albeit at the expense of accuracy. This is particularly useful for NISQ algorithms, where the required precision is lower, allowing the use of circuits that require very few resources. Notably, at $\sigma_{\max} = 0.2$, we can further reduce the circuit depth and CNOT count that optimize the entire circuit through a TTN representation, as discussed in Sec. 4.3.4.

On the other hand, increasing $\sigma_{\max}$ leads to a significant reduction in ac-

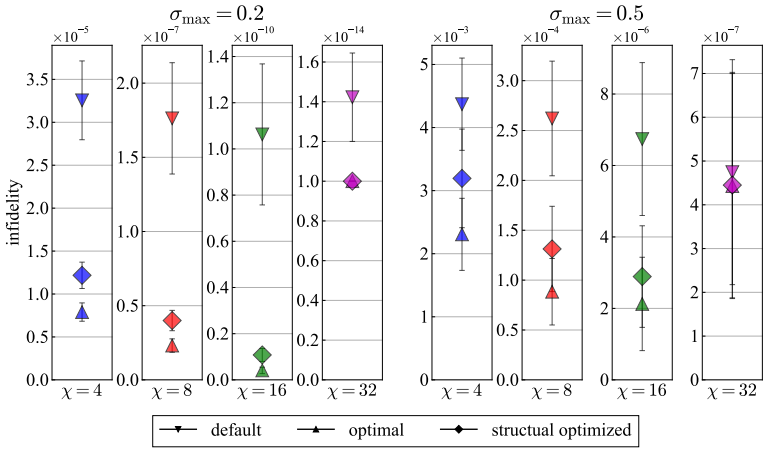


Figure 4.10: The infidelity of the encoding circuit with different strategies for determining network structure. The parameters are set as $\sigma_{\max} = 0.2, 0.5$. The lower triangle points show the result without any optimization of the ordering. The upper triangle points show the result with optimal network structure. The diamond points show the result with the automatic structural optimization method discussed in Sec. 4.3.3.

curacy. $\sigma_{\max}$ represents the strength of the correlations between variables. Stronger correlations result in larger Schmidt coefficients, making tensor network approximations more challenging.

Finally, we discuss the effect of network structure and automatic structural optimization. Fig. 4.10 shows the infidelity of the state preparation circuit by varying the strategy for determining network structure. We get the optimal network structure by enumerating all possible candidates and estimating the fidelity. It is found that using tensor network structural optimization achieves accuracy close to that of optimal network structure, compared to when optimization is not employed. This highlights the significant influence of network structure on the performance of tensor network-based quantum circuit compilation and underscores the potential of structural optimization techniques in finding the near-optimal ordering, even when there are no 1D or tree-like structures.

4.5 Discussion

In this chapter, we proposed a scalable circuit compilation method for multivariate normal distributions using TTN. We first considered multivariate normal distributions with 1D correlation structures. We examined two cases: the rank of the covariance matrix is bounded, and the correlation coefficients decay exponentially with the distance between variables. We theoretically guaranteed

that these probability distributions can be efficiently represented using TTN.

Additionally, we introduced automatic structural optimization into our circuit compilation method. This optimization rearranges the structure of tensor networks based on the bipartite entanglement entropy of the system. By applying it to multivariate probability distributions, we found that it searches for structures that appropriately reflect the 1D correlation structure of the distribution.

We combined these techniques with the FSL method, TCI, and truncation method using the canonical form. We numerically demonstrated that we can scalably compile the state preparation circuit for normal multivariate distributions with 1D structures. Moreover, even for random normal distributions without a specific structure, we confirmed numerically that our approach could significantly reduce the circuit depth and the number of CNOT gates compared to existing methods.

Multivariate normal distributions play an important role in various practical quantum algorithms. An important example is quantum algorithms for quantum field theory [149, 151]. In lattice ϕ^4 theory, the vacuum state $|\mathrm{vac}\rangle$ is represented as a multivariate normal distribution in the field variables $\phi(\boldsymbol{x})$, with the covariance matrix defined as

$$G(\boldsymbol{x}-\boldsymbol{y}) = \langle \mathrm{vac}|\phi(\boldsymbol{x})\phi(\boldsymbol{y})|\mathrm{vac}\rangle\,. \tag{4.68}$$

In the free theory, the covariance matrix $G(\boldsymbol{x}-\boldsymbol{y})$ decays exponentially with the distance $|\boldsymbol{x}-\boldsymbol{y}|$. Therefore, when the lattice is defined in one spatial dimension, the vacuum state can be well approximated using MPS, allowing efficient compilation via our proposed methods.

Another important application lies in quantum algorithms for finance. In finance, the relationship between the risks and returns of multiple assets is modeled with a multivariate normal distribution; therefore, preparing a multivariate normal quantum state is essential for quantum Monte Carlo integration applied to derivative pricing [184, 185] and risk analysis [153, 186, 187]. These financial datasets are not guaranteed to exhibit exact tree-like correlation structures. However, as demonstrated in our numerical experiments, our heuristic optimization algorithm may help reduce the cost of preparing such quantum states.

One of the advantages of our method is the ability to calculate achievable fidelity in advance. There are a lot of methods for circuit compiling for NISQ algorithms [144, 145, 157–170], which also produce low-cost circuits at the expense of fidelity. However, many algorithms are based on variational optimization and there are often no theoretical guarantees or estimates regarding the accuracy of their approximations. Our method does not depend on variational optimization and can estimate the achievable fidelity from the covariance matrix and bond dimension in the network. Although this chapter was limited to multivariate normal distributions with theoretical guarantees, it can be extended to other multivariate probability distributions and multivariable functions [188].

Another advantage of our method is its hardware compatibility. While our numerical simulations assume all-to-all connectivity, the hierarchical structure

of the TTN allows each local tensor to be encoded into a local quantum circuit and connected using SWAP gates in a layered manner. If the bond dimension is sufficiently small, the cost of isometry synthesis can also be kept low. Therefore, the proposed method is expected to be practical even on quantum hardware with limited qubit connectivity. We left fully synthesizing circuits and testing them on real hardware for future work.

From the perspective of tensor network algorithms, this research can also be considered a new application of automatic structural optimization. Automatic structural optimization, developed in the field of many-body physics, has been used to detect the entanglement structure of systems. In [189], the techniques were applied to the field of machine learning, utilizing mutual information to efficiently represent datasets. This chapter demonstrates that these techniques can also be applied to the field of quantum circuit optimization, underscoring the potential applicability across various fields.

Future directions could involve the use of other tensor network algorithms, especially those designed for two-dimensional systems, for the compilation of state preparation circuits. Moreover, since automatic structural optimization remains a heuristic algorithm, there is a potential to develop theoretically guaranteed or more sophisticated structural optimization algorithms.

An open-source implementation of the method for detecting the correlation structure of multivariate normal distributions is available and has been described in a subsequent paper [190].

Chapter 5

Tensor Network Formulation of Dequantized Algorithms for Ground State Energy Estimation

Verifying quantum advantage for practical problems, particularly the ground state energy estimation (GSEE) problem, is one of the central challenges in quantum computing theory. For that purpose, dequantization algorithms play a central role in providing a clear theoretical framework to separate the complexity of quantum and classical algorithms. However, existing dequantized algorithms typically rely on sampling procedures, leading to prohibitively large computational overheads and hindering their practical implementation on classical computers. In this chapter, we propose a tensor network-based dequantization framework for GSEE that eliminates the sampling process while preserving the asymptotic complexity of prior dequantized algorithms. In our formulation, the overhead arising from sampling is replaced by the growth of the bond dimension required to represent Chebyshev vectors as tensor network states. Consequently, physical structure, such as entanglement and locality, is naturally reflected in the computational cost. By combining this approach with tensor network approximations, such as Matrix Product States (MPS), we construct a practical dequantization algorithm that is executable within realistic computational resources. Numerical simulations demonstrate that our method can efficiently construct high-degree polynomials up to $d = 10^4$ for Hamiltonians with up to 100 qubits, explicitly revealing the crossover between classically tractable and quantum advantaged regimes. These results indicate that tensor network-based dequantization provides a crucial tool toward the rigorous, quantitative verification of quantum advantage in realistic many-body systems. This chapter is based on [H. Manabe *et al.*, arXiv: 2512.13548 (2025)] with slight modifications

to fit the context of this dissertation.

5.1 Introduction

Quantum computers have attracted significant attention for their potential to perform information processing tasks that are intractable for classical computers [12, 191]. Among these tasks, ground state energy estimation (GSEE) [24, 192–195] is a central problem in condensed matter physics and quantum chemistry, where exponential quantum speedup is anticipated. Formally, given an n-qubit Hamiltonian of the form $H = \sum_{i=1}^{\mathrm{poly}(n)} H_i$, the goal of GSEE is to estimate the ground state energy of H to within the target precision ϵ. When the Hamiltonian can be accessed on a quantum computer and a guiding state with sufficient overlap with the true ground state is available, this task can be efficiently solved using algorithms based on quantum phase estimation (QPE) or quantum singular value transformation (QSVT) with eigenvalue filtering [23, 196]. Reflecting this broad applicability and importance, there has been extensive research on implementing these algorithms on an early fault-tolerant quantum computer with reduced resource requirements or shallower circuit depth [24, 192, 193, 197–200].

However, the exponential speedup associated with the GSEE problem has a more nuanced structure than previously understood. Gharibian and Le Gall [201] showed that, given a guiding state sufficiently close to the ground state, GSEE for a k-local Hamiltonian is BQP-complete when the required energy precision ϵ scales as $1/\mathrm{poly}(n)$, whereas it becomes efficiently solvable on a classical computer when the precision is constant. In their work, they employed an approach known as *dequantization*: the process of constructing a classical analogue of a quantum algorithm by relaxing certain assumptions on the problem [201–207]. Through dequantization, they revealed that the computational hardness hinges critically on the scaling of a single parameter, the precision ϵ: different scalings separate classically tractable regimes from those requiring full quantum computational power. Hence, the approach of dequantization serves as a powerful theoretical tool for probing the boundary between classical and quantum computation and for rigorously verifying quantum advantage.

Despite its theoretical importance, the existing dequantized algorithms are computationally impractical. Many dequantization algorithms rely on Monte Carlo sampling to estimate inner products between high-dimensional vectors [206], leading to enormous sampling overheads that render practical implementation on classical hardware infeasible. For example, in the dequantized algorithm of Gharibian and Le Gall [201] for the GSEE problem of a k-local Hamiltonian, the runtime scales as $O(2^{(8+k)/\epsilon})$. Even for $\epsilon = 0.1$, which is extremely coarse and far from practically meaningful, the resulting classical cost becomes prohibitively high. As a result, prior studies on dequantization have largely remained at the level of computational complexity theory, without concrete implementations or numerical demonstrations. This motivates the development of practically executable dequantization algorithms that preserve the theoretical

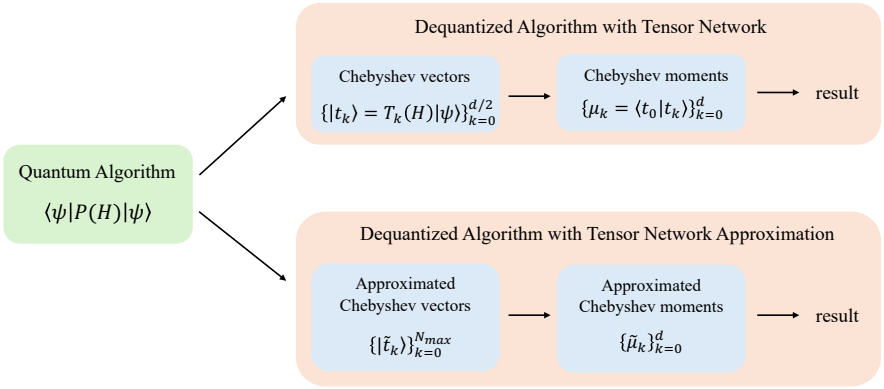


Figure 5.1: An overview of the dequantized algorithm with tensor networks for ground state energy estimation. P denotes a polynomial of degree d, and T_k denotes the kth Chebyshev polynomial. $|\widetilde{t_k}\rangle$ and $\widetilde{\mu}_k$ are approximations of the Chebyshev vector $|t_k\rangle$ and the moment μ_k respectively, and we assume $N_{\max} < d$. The formal definitions are provided later in the main text.

insights while being computationally feasible on classical devices.

In this chapter, we introduce tensor network techniques [38, 41, 42, 87, 127] to formulate a dequantized algorithm for solving the GSEE problem. We classically simulate the QSVT algorithm with tensor networks, which share the same formulation as the Chebyshev Matrix Product States (ChebMPS) method [48, 208], which was developed in condensed matter physics. We reframe this technique within the rigorous framework of dequantization to quantitatively probe the boundary of quantum advantage. An overview of our approach is shown in Fig. 5.1. We first expand the polynomial filter P used in the quantum algorithm in the Chebyshev basis. We then represent the Chebyshev vectors as tensor network states and compute the Chebyshev moments from these representations. In this formulation, all quantities are evaluated deterministically through tensor network contraction. As a result, we eliminate the sampling overhead arising from statistical error while preserving the same asymptotic complexity scaling. Moreover, this formulation allows us to utilize the locality of the Hamiltonian to reduce the effective computational cost.

Furthermore, we introduce a practical variant of the dequantized algorithm that leverages tensor network approximation methods. In this approach, we approximate the Chebyshev vectors using tensor network states and estimate the corresponding moments via linear prediction. As long as these approximation errors remain within a tolerable range, the computational cost scales linearly—rather than exponentially—in $1/\epsilon$, yielding a dequantized algorithm that is executable in practice. This formulation effectively recasts the source of

computational hardness: the bottleneck shifts from the statistical variance of sampling to the entanglement growth of the Chebyshev vectors.

We implement the proposed method and benchmark it on one- and two-dimensional transverse-field Ising models (TFIM). In particular, we use Matrix Product States (MPS) [88] to represent Chebyshev vectors. We demonstrate that our approach can efficiently construct high-degree polynomial filters up to $d = 10^4$, which is fundamentally inaccessible to prior sampling-based approaches, and execute ground state energy estimation for 100-qubit systems. As a result, while the 1D TFIM can be dequantized with tensor networks to high precision, the 2D TFIM cannot be fully dequantized with the bond dimensions used in our experiments. These results, however, explicitly visualize the crossover between classically tractable and quantum-advantaged regimes through a single precision parameter ϵ, which captures the essence of the dequantization approach. In this way, our framework can serve as a key methodology for the quantitative verification of quantum advantage in realistic many-body systems.

This chapter is organized as follows. In Sec. 5.2, we review the tensor network methods and the GSEE algorithm used in this section. In Sec. 5.3, we formulate the dequantized algorithm based on exact tensor network contractions, and in Sec. 5.4, we extend the formulation to include tensor network approximation methods. Section 5.5 presents numerical results for one- and two-dimensional transverse-field Ising models, demonstrating the practical utility of our approach. Finally, Sec. 5.6 summarizes our contributions.

5.2 Preliminaries

In this section, we provide the tools required for developing a tensor network formulation of the dequantized algorithm for the GSEE problem. We first introduce the basic concept of tensor networks, and then review the eigenvalue filtering techniques used in quantum and dequantized algorithms for GSEE.

5.2.1 Tensor network

A tensor network [38, 41, 87] is a graphical formalism that represents a high-rank tensor as a contraction of many small tensors. Tensor networks are widely used as a computational tool in fields such as condensed matter physics [39, 42, 46, 49, 106, 209], data science [57, 58, 129, 210] and quantum computing [61, 66, 101, 127] to avoid the so-called curse of dimensionality, where the memory cost increases exponentially with the number of dimensions. We define tensor networks on an undirected graph.

Definition 1 (Tensor network)**.** *A tensor network is a triple* $(G, \{T_v\}_{v\in V}, \{d_e\}_{e\in E})$ *consisting of:*

- *An undirected graph* $G = (V, E)$*.*

- *A positive integer $d_e \in \mathbb{N}$ associated with each edge $e \in E$, called the* bond dimension.

- *For each vertex $v \in V$, a tensor*

$$T_v \in \bigotimes_{e \in E_v} \mathbb{C}^{d_e}, \tag{5.1}$$

where $E_v := \{e \in E : e \ni v\}$ denotes the set of edges incident to v.

We distinguish between *internal edges*, which connect two vertices (including loops), and *dangling edges*, which are incident on only one vertex.

Definition 2 (Tensor network states)**.** *A* tensor network state *$TNS(G, \{T_v\}_{v\in V}, \{d_e\}_{e\in E})$ is a quantum state obtained by contracting all internal edges of the tensor network, i.e., summing over all indices associated with internal edges:*

$$|\mathrm{TNS}(G, \{T_v\}, \{d_e\})\rangle = \sum_{\{i_e\}_{e\in E_{\mathrm{int}}}} \prod_{v\in V} T_v\big(\{i_e\}_{e\in E_v}\big), \tag{5.2}$$

where E_{int} denotes the set of internal edges.

Throughout this chapter, we consider qubit systems and assume that $d_e = 2$ whenever e is a dangling edge.

Definition 3 (Inner product of tensor network states)**.** *Let $(G, \{T_v\}_{v\in V}, \{d_e\}_{e\in E})$ and $(G, \{T'_v\}_{v\in V}, \{d'_e\}_{e\in E})$ be two tensor network states defined on the same graph $G = (V, E)$. Suppose $d_e = d'_e$ if e is a dangling edge. Then, their inner product is defined by connecting the two tensor networks along the dangling edges and contracting:*

$$\begin{aligned}\langle \mathrm{TNS}'|\mathrm{TNS}\rangle = \sum_{\{i_e\}_{e\in E}} \sum_{\{j_e\}_{e\in E}} & \\ \times \prod_{v\in V} \overline{T'_v(\{j_e\}_{e\in E_v})}\, T_v(\{i_e\}_{e\in E_v}) & \\ \times \prod_{e\in E_{\mathrm{dang}}} \delta_{i_e,j_e}. & \end{aligned} \tag{5.3}$$

Here, the overline denotes complex conjugation, E_{dang} denotes the set of dangling edges, and δ is a Kronecker delta.

Definition 4 (Efficiently contractible tensor network states)**.** *We say that a family of tensor network states $\mathrm{TNS}(G, \{T_v\}, \{d_e\})$ that represent an n-qubit state is* efficiently contractible *if the inner product between two tensor network states $\mathrm{TNS}(G, \{T_v\}, \{d_e\})$ and $\mathrm{TNS}(G, \{T'_v\}, \{d'_e\})$ can be computed in time polynomial in the number of qubits and the maximum bond dimension, i.e., in $\mathrm{poly}(n, \max_e d_e, \max_e d'_e)$ time.*

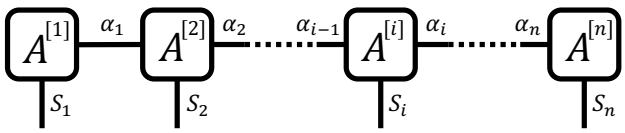


Figure 5.2: The diagram notation of MPS, a tensor network with a simple 1D structure. $\{s_i\}$ is a set of indices that corresponds to the dangling edges, and $\{\alpha_i\}$ is a set of internal indices. This figure is reproduced from Fig. 2.2 for convenience.

A typical example of an efficiently contractible tensor network state is the *matrix product states* (MPS). An MPS is a tensor network defined on a 1D chain graph and can be written as:

$$|\psi\rangle = \sum_{\{s_i\}} \sum_{\{\alpha_i\}} \left[A^{[1]s_1}_{\alpha_1} A^{[2]s_2}_{\alpha_1\alpha_2} \cdots A^{[n]s_n}_{\alpha_{n-1}} \right] |s_1 s_2 \cdots s_n\rangle \,, \tag{5.4}$$

where s_i denotes the physical index of the ith qubit and α_i denotes the internal index. We illustrate the diagram notation of MPS in Fig. 5.2.

Let $D = \max_e d_e$. Then, the cost of computing the inner product scales as $\mathcal{O}(nD^3)$, indicating that MPS is efficiently contractible. In Fig. 5.3, we express the process of calculating the inner product of MPS in diagrammatic notation.

5.2.2 Eigenvalue filtering

In this section, we briefly review the eigenvalue filtering algorithm based on Quantum Singular Value Transformation (QSVT) [23, 196], which forms the foundation of the dequantized algorithm for GSEE.

QSVT

The QSVT algorithm applies a polynomial transformation to the singular values of a general matrix A. In the context of GSEE, however, the matrix of interest is a non-negative Hermitian operator $H \geq 0$. In this case, QSVT effectively performs quantum eigenvalue transformation (QEVT), in which a polynomial is applied directly to the eigenvalues of H. Consequently, we adopt the QEVT framework throughout this chapter. We also assume that the Hamiltonian is normalized: $\|H\| \leq 1$.

We assume access to H through a block-encoding in a unitary operator U:

$$U = \begin{pmatrix} H & * \\ * & * \end{pmatrix}. \tag{5.5}$$

Let $P\colon [-1,1] \to \mathbb{R}$ be a real polynomial with degree d satisfying:

- P has parity d mod 2

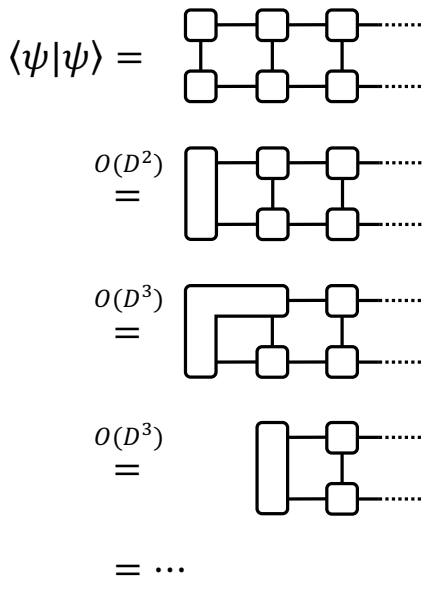


Figure 5.3: The process of calculating the inner product of two MPS. The double-layered MPS is contracted from the left-hand side. Each contraction costs $O(D^3)$, and this process is repeated n times.

- $|P(x)| \leq 1, \quad x \in [-1, 1]$.

The QEVT framework allows us to efficiently implement a unitary operator U_P such that

$$U_P = \begin{pmatrix} P(H) & * \\ * & * \end{pmatrix}. \tag{5.6}$$

Here, $P(H)$ is the matrix polynomial defined as

$$P(H) = \sum_i P(\lambda_i) \left|\lambda_i\right\rangle \left\langle\lambda_i\right|, \tag{5.7}$$

where λ_i is an eigenvalue of H and $|\lambda_i\rangle$ is the corresponding eigenvector.

Filtering function

The core of the QSVT-based GSEE algorithm lies in the design of the polynomial P. By approximating the *shifted sign function* with a polynomial and applying it as $P(H)$, one can filter out all eigenvalues of H above a given threshold, enabling efficient ground state preparation and energy estimation. Efficient polynomial approximations of the shifted sign function are well studied.

Lemma 1 (Polynomial approximation of the shifted sign function [196, 212])**.** *Let $\eta, \Delta \in (0, \frac{1}{2})$ and $c \in [-1, 1]$. There exists a polynomial $P_{\eta,\Delta,c}(x)$ of degree*

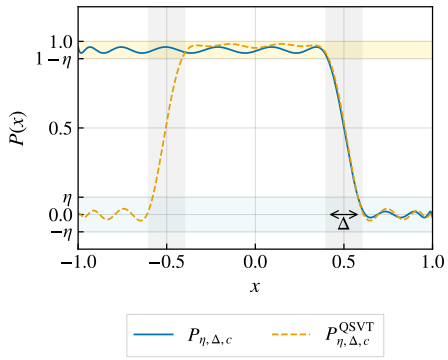


Figure 5.4: An example of polynomial approximation of the shifted sign function $P_{\eta,\Delta,c}$ and the rectangle function $P^{\mathrm{QSVT}}_{\eta,\Delta,c}$. Parameters are set as $\eta = 0.1, \Delta = 0.2, c = 0.5$ and the degree is 20. The polynomial approximation is obtained by CVXPY [211].

$O\left(\frac{1}{\Delta}\log(1/\eta)\right)$ *such that*

$$|P_{\eta,\Delta,c}(x)| \leq 1 \tag{5.8}$$

for all $x \in [-1,1]$ *and*

$$\begin{cases} P_{\eta,\Delta,c}(x) \in [1-\eta, 1] & \text{for } x \in [-1, c-\Delta/2]\,, \\ P_{\eta,\Delta,c}(x) \in [-\eta, \eta] & \text{for } x \in [c+\Delta/2, 1]\,. \end{cases} \tag{5.9}$$

When implementing this transformation via QSVT, the polynomial P must satisfy the parity constraint. Therefore, instead of directly using a general shifted sign function, we consider the following rectangle function tailored for QSVT.

Lemma 2 (Polynomial approximation of the rectangle function [196, 212])**.** *Let* $\eta, \Delta \in (0, \frac{1}{2})$ *and* $c \in [-1, 1]$*. There exists an even polynomial* $P^{\mathrm{QSVT}}_{\eta,\Delta,c}(x)$ *of degree* $O\left(\frac{1}{\Delta}\log(1/\eta)\right)$ *such that*

$$|P^{\mathrm{QSVT}}_{\eta,\Delta,c}(x)| \leq 1 \tag{5.10}$$

for all $x \in [-1,1]$ *and*

$$\begin{cases} P^{\mathrm{QSVT}}_{\eta,\Delta,c}(x) \in [1-\eta, 1] & \text{for } x \in [-c+\Delta/2,\ c-\Delta/2], \\ P^{\mathrm{QSVT}}_{\eta,\Delta,c}(x) \in [-\eta, \eta] & \text{for } x \in [-1,\ -c-\Delta/2] \\ & \qquad \cup\, [\,c+\Delta/2,\ 1\,]. \end{cases} \tag{5.11}$$

For visual intuition, we plot examples of polynomial approximations $P_{\eta,\Delta,c}(x)$ and $P^{\mathrm{QSVT}}_{\eta,\Delta,c}(x)$ in Fig. 5.4.

GSEE

Now we define the central problem studied in this chapter: ground state energy estimation [24, 192–195].

Problem 1 (GSEE)**.** *Let H be a Hamiltonian on n qubits satisfying $H \geq 0$ and $\|H\| \leq 1$, i.e., H is non-negative and normalized. Let $\mathcal{E}(H)$ denote the ground state energy of H, namely, its smallest eigenvalue. Given a precision parameter $\epsilon > 0$, the goal is to compute an estimate $\hat{E}$ such that*

$$\left|\hat{E} - \mathcal{E}(H)\right| \leq \epsilon. \tag{5.12}$$

This problem can be efficiently solved on a quantum computer using QSVT with an appropriate filter function, provided that one has access to a block encoding of H and a guiding state with sufficient overlap with the ground state.

Lemma 3 (GSEE via QSVT [23])**.** *Suppose we are given a unitary block encoding of H. Assume that a quantum state $|\psi\rangle$ can be prepared such that it has overlap $\chi := |\langle\psi|\lambda_0\rangle|$ with the ground state $|\lambda_0\rangle$ of H. Then, for any $\delta > 0$, there exists a quantum algorithm that uses*

$$\mathcal{O}\left(\frac{1}{\epsilon\chi^2}\log\left(\frac{1}{\chi}\right)\log\left(\frac{1}{\delta}\right)\right) \tag{5.13}$$

queries to the block encoding of H and outputs an ϵ-approximation of $\mathcal{E}(H)$ with probability at least $1-\delta$.

Proof. Given the initial state $|0\rangle|\psi\rangle$, applying the QSVT circuit for $P^{\mathrm{QSVT}}_{\eta,\Delta,c}(H)$ yields

$$U_P\big(|0\rangle|\psi\rangle\big) = |0\rangle\ P^{\mathrm{QSVT}}_{\eta,\Delta,c}(H)|\psi\rangle\ +\ |1\rangle|\mathrm{garbage}\rangle\,. \tag{5.14}$$

Hence, the probability of measuring the outcome 0 in the ancilla qubit is

$$p_0 = \left\|P^{\mathrm{QSVT}}_{\eta,\Delta,c}(H)|\psi\rangle\right\|^2 = \sum_i |c_i|^2\left|P^{\mathrm{QSVT}}_{\eta,\Delta,c}(\lambda_i)\right|^2, \tag{5.15}$$

where $|\psi\rangle = \sum_i c_i|\lambda_i\rangle$.

We consider the following two cases.

Case (1): $\lambda_i \geq c + \Delta/2$ for all i.

In this regime, we have $\left|P^{\mathrm{QSVT}}_{\eta,\Delta,c}(\lambda_i)\right| \leq \eta$ for all eigenvalues, and hence

$$p_0 \leq \sum_i |c_i|^2\,\eta^2 = \eta^2. \tag{5.16}$$

Case (2): $\lambda_0 \leq c - \Delta/2$.

Let $\chi = |\langle\lambda_0|\psi\rangle|$ be the ground-state overlap. Since $\left|P^{\mathrm{QSVT}}_{\eta,\Delta,c}(\lambda_0)\right| \geq 1-\eta$,

$$p_0 \geq |c_0|^2(1-\eta)^2 = \chi^2(1-\eta)^2. \tag{5.17}$$

Therefore, by choosing the polynomial approximation error as $\eta = \chi/4$ for example, equations (5.16) and (5.17) imply that the two probabilities satisfy

$$p_0 \leq \chi^2/16 \qquad \text{vs.} \qquad p_0 \geq \frac{9}{16}\chi^2. \tag{5.18}$$

By sampling the ancilla outcome, one can distinguish between the two cases. Lemma 5 in Ref. [23] implies that

$$\mathcal{O}\left(\frac{1}{\chi^2}\log\frac{1}{\delta}\right) \tag{5.19}$$

samples suffice to determine the correct case with probability at least $1-\delta$.

We now apply this classifier to perform a binary search. Assume an initial interval $[l, r]$ satisfies $l < \lambda_0 < r$. At each iteration, define

$$c = \frac{l+r}{2}, \qquad \Delta = \frac{r-l}{3}, \tag{5.20}$$

and perform the above decision procedure for the threshold c.

- If the ancilla measurement returns 0 with high probability, then we are not in Case (1), and thus

$$\lambda_0 < c + \frac{\Delta}{2} = \frac{l+2r}{3}, \qquad r \leftarrow \frac{l+2r}{3}. \tag{5.21}$$

- Otherwise we are not in Case (2), implying

$$\lambda_0 > c - \frac{\Delta}{2} = \frac{2l+r}{3}, \qquad l \leftarrow \frac{2l+r}{3}. \tag{5.22}$$

In each iteration the interval shrinks by a constant factor 2/3, and therefore after

$$\mathcal{O}\left(\log\frac{1}{\epsilon}\right) \tag{5.23}$$

steps we obtain an interval $[l, r]$ of width at most 2ϵ. We output $\hat{E} = (l+r)/2$ as the final estimate.

Each iteration uses the QSVT circuit for $P^{\mathrm{QSVT}}_{\eta,\Delta,c}(H)$, whose degree is

$$O\left(\frac{1}{\Delta}\log\frac{1}{\eta}\right), \tag{5.24}$$

and $\mathcal{O}\left(\frac{1}{\chi^2}\log\frac{1}{\delta}\right)$ measurements of the ancilla qubit. Since Δ scales with the current interval size and decreases geometrically, the total number of Hamiltonian queries is

$$\mathcal{O}\left(\frac{1}{\epsilon\,\chi^2}\log\frac{1}{\chi}\log\frac{1}{\delta}\right). \tag{5.25}$$

□

We can confirm that this quantum algorithm remains efficient when $\epsilon = 1/\mathrm{poly}(n)$.

Dequantization

We now consider *dequantizing* this quantum algorithm. In this chapter, we define dequantization as a process of deriving a classical analogue of a quantum algorithm by relaxing several assumptions of the problem [201–207, 213–218]. To dequantize, we impose some additional assumptions. We consider a k-local Hamiltonian on n qubits:

$$H = \sum_{i=1}^{m} H_i, \tag{5.26}$$

where each term H_i acts non-trivially only on k qubits. We also assume that we have sample-and-query access to the guiding state $|\psi\rangle$:

1. Query access: efficiently compute the amplitude $\langle \boldsymbol{x}|\psi\rangle$ for any bitstring $\boldsymbol{x} \in \{0,1\}^n$.
2. Sample access: efficiently sample a bitstring $\boldsymbol{x} \in \{0,1\}^n$ with probability $|\langle \boldsymbol{x}|\psi\rangle|^2$.

Under these additional assumptions, the GSEE problem can be solved with the following computational complexity:

Lemma 4 (GSEE via dequantization [201, 205])**.** *Let H be a k-local Hamiltonian on n qubits and assume that $\sum_{i=1}^{m} \|H_i\| = 1$. Suppose we are given sample-and-query access to a guiding state $|\psi\rangle$ with overlap χ. Then, there exists a classical algorithm that runs in time*

$$O^*\left(\frac{1}{\epsilon} \cdot 2^{(k+8)/\epsilon} \cdot \chi^{-8}\right) \tag{5.27}$$

, and outputs an ϵ-approximation of $\mathcal{E}(H)$ with probability at least $1 - 1/\exp(n)$.

Proof sketch. We provide only a brief proof sketch and refer to prior work for the full details. The basic idea is the same as in the QSVT-based approach: we aim to calculate

$$\langle \psi|P_{\eta,\Delta,c}(H)|\psi\rangle \tag{5.28}$$

and use this quantity to run a binary search to estimate the ground state energy. In contrast to the quantum setting, we do not need to enforce positivity of H, nor impose any parity constraints on the polynomial. Thus, we use $P_{\eta,\Delta,c}$ instead of $P^{\mathrm{QSVT}}_{\eta,\Delta,c}$.

As in the previous section, the polynomial satisfies

$$\lambda_i \geq c + \Delta/2 \quad \Rightarrow \quad |\langle \psi|P_{\eta,\Delta,c}(H)|\psi\rangle| \leq \eta \tag{5.29}$$

and

$$\lambda_0 \leq c - \Delta/2 \quad \Rightarrow \quad |\langle \psi|P_{\eta,\Delta,c}(H)|\psi\rangle| \geq \chi^2(1-\eta). \tag{5.30}$$

Therefore, choosing $\eta = \chi^2/4$ allows us to distinguish the two cases by estimating the expectation value to additive accuracy $O(\chi^2)$.

We expand the polynomial in monomials:

$$P_{\eta,\Delta,c}(x) = \sum_{r=0}^{d} a_r x^r, \tag{5.31}$$

$$\langle\psi|P_{\eta,\Delta,c}(H)|\psi\rangle = \sum_{r=0}^{d} a_r \langle\psi|H^r|\psi\rangle . \tag{5.32}$$

To estimate each term $\langle\psi|H^r|\psi\rangle$, we write

$$\langle\psi|H^r|\psi\rangle = \sum_{\boldsymbol{x}\in[m]^r} \langle\psi|H_{x_1}\cdots H_{x_r}|\psi\rangle , \tag{5.33}$$

where $[m] = \{1, 2, \ldots, m\}$. For each multi-index $\boldsymbol{x}$, define

$$q(\boldsymbol{x}) = \|H_{x_1}\|\cdots\|H_{x_r}\|. \tag{5.34}$$

We introduce the random variable

$$X = \frac{\langle\psi|H_{x_1}\cdots H_{x_r}|\psi\rangle}{q(\boldsymbol{x})}, \tag{5.35}$$

and average over t independent samples, where $\boldsymbol{x}$ is sampled according to the probability $q(\boldsymbol{x})$. By taking t sufficiently large, we can get an accurate estimator of $\mathbb{E}[X] = \langle\psi|H^r|\psi\rangle$.

The main challenge is to compute $\langle\psi|H_{x_1}\cdots H_{x_r}|\psi\rangle$ in poly(n) time rather than exp(n) time. In fact, this quantity can be efficiently estimated, given that each H_i is k-local (therefore s-sparse matrix with $s = 2^k$) and we have sample-and-query access to $|\psi\rangle$. The result of iterated matrix multiplication shows that one can estimate $\langle\psi|H_{x_1}\cdots H_{x_r}|\psi\rangle$ in time

$$O^*\Big(s^r\,\epsilon'^{-2}\,\log(1/\delta)\Big), \tag{5.36}$$

to additive error $\epsilon'\,\|H_{x_1}\|\cdots\|H_{x_r}\|$ with probability at least $1-\delta$. Here the factor ϵ'^{-2} accounts for the sampling overhead of the estimation of the inner product, and s^r accounts for the number of elements that are required for obtaining one element of $H_{x_1}\cdots H_{x_r}\,|\psi\rangle$.

Setting

$$\epsilon' = O(\chi^2/4^r) \quad t = O(4^{2r}/\chi^4) \quad \delta = O(1/8t), \tag{5.37}$$

we obtain an estimator of $\langle\psi|H^r|\psi\rangle$ with additive error $O(\chi^2/4^r)$. The resulting time cost is

$$O^*\big(t\cdot s^r\cdot 2^{4r}\cdot\chi^{-4}\log t\big) = O^*\big(s^r 2^{8r}\chi^{-8} r\big) . \tag{5.38}$$

The coefficients of the polynomial bounded on $[-1, 1]$ satisfy [219]

$$|a_r| \le 4^r. \tag{5.39}$$

Moreover, the degree of the polynomial scales as $O^*(1/\epsilon)$. Putting everything together, the overall time complexity to estimate the quantity $\langle\psi|P_{\eta,\Delta,c}(H)|\psi\rangle$ in additive error $O(\chi^2)$ scales as

$$O^*\left(\frac{1}{\epsilon}\cdot 2^{(k+8)/\epsilon}\cdot\chi^{-8}\right). \tag{5.40}$$

□

From the complexity, we see that the $1/\epsilon$ term appears in the exponent whose base is the constant $(2^{(k+8)})$. Thus, when the target precision ϵ is a constant, the GSEE problem can, in principle, be solved efficiently on a classical computer.

However, in practice, the constant overhead $2^{(k+8)/\epsilon}$ is prohibitively large, making it impossible to execute this algorithm in a realistic regime. For example, if we set $k = 2$, the term $2^{10/\epsilon}$ already implies that only rather coarse precision $\epsilon \gtrsim 0.1$ is feasible even when we have massive computing resources. Since the Hamiltonian spectrum is normalized to the interval $[-1, 1]$, such a requirement on the precision makes the algorithm entirely impractical. To the best of our knowledge, no work has implemented such sampling-based dequantized algorithms on a classical computer or evaluated their practical performance.

5.3 Dequantized algorithm with tensor networks

As discussed in the previous section, existing dequantized algorithms suffer from a large sampling overhead, making them practically infeasible to implement. In this chapter, we propose an alternative formulation based on tensor networks that eliminates the need for sampling. To do so, we need some additional assumptions that the guiding state and Hamiltonian are efficiently contractible.

5.3.1 Additional assumptions on the problem

In all tensor network-based dequantized algorithms considered in this chapter, we assume that the guiding state is efficiently contractible. In practice, this means that the guiding state is given in a tensor network representation such as an MPS. Importantly, an efficiently contractible guiding state automatically satisfies the sample-and-query access assumption used in prior dequantization results. To see this, recall that query access requires the ability to compute $|\langle\psi|\boldsymbol{x}\rangle|$ for any bitstring $\boldsymbol{x}$. This quantity is a special case of contracting a tensor network with a product state and can therefore be computed efficiently whenever the guiding state is efficiently contractible. Similarly, sampling from the distribution $p(\boldsymbol{x}) = |\langle\psi|\boldsymbol{x}\rangle|^2/\|\psi\|^2$ requires the ability to evaluate marginals like $p(x_1)$, which are also efficiently computable under this assumption. Thus, the efficiently contractible assumption implies the sample-and-query access model of Lemma 4 and can be viewed as a special case of the assumptions used in previous dequantization work.

We also define the class of Hamiltonians that can be handled by our algorithm. While this class does not include all Hamiltonians, it does include physically important families such as k-local Hamiltonians.

Definition 5 (Efficiently contractible Hamiltonian)**.** *Let* $|\mathrm{TNS}(G, \{T_v\}, \{d_e\})\rangle$ *be an efficiently contractible tensor network state. Consider a Hamiltonian* H *such that the action of* H *on* $|\mathrm{TNS}\rangle$ *yields another tensor network state on the same graph:*

$$H\,|\mathrm{TNS}(G, \{T_v\}, \{d_e\})\rangle = |\mathrm{TNS}(G, \{T'_v\}, \{d'_e\})\rangle \tag{5.41}$$

and satisfies

$$\max_{e\in E} d'_e = \mathrm{poly}\left(\max_{e\in E} d_e, |V|\right). \tag{5.42}$$

We call such an operator H *an* efficiently contractible Hamiltonian*.*

A typical example is a k-local Hamiltonian:

$$H = \sum_{i=1}^{m} H_i, \tag{5.43}$$

where $k = O(1)$ and $m = \mathrm{poly}(n)$. Each local term H_i can be expressed as a sum of Pauli products:

$$H_i = \sum_{j=1}^{4^k} \alpha_{ij} P_{ij}. \tag{5.44}$$

For a single Pauli term P_{ij}, the action $P_{ij}\,|\mathrm{TNS}\rangle$ does not increase any bond dimension d_e. A sum of tensor network states with the same graph can be represented by a TNS:

$$|\mathrm{TNS}(G, \{T_v\}, \{d_e\})\rangle = \tag{5.45}$$

$$|\mathrm{TNS}(G, \{T_v^{(1)}\}, \{d_e^{(1)}\})\rangle + |\mathrm{TNS}(G, \{T_v^{(2)}\}, \{d_e^{(2)}\})\rangle \tag{5.46}$$

with increased bond dimension:

$$d_e = d_e^{(1)} + d_e^{(2)}. \tag{5.47}$$

Therefore, after applying H to the initial tensor network state,

$$\begin{aligned} &|\mathrm{TNS}(G, \{T'_v\}, \{d'_e\})\rangle \\ &= H\,|\mathrm{TNS}(G, \{T_v\}, \{d_e\})\rangle \\ &= \sum_{i=1}^{m}\sum_{j=1}^{4^k} \alpha_{ij} P_{ij}\,|\mathrm{TNS}(G, \{T_v\}, \{d_e\})\rangle\,, \end{aligned} \tag{5.48}$$

the resulting bond dimensions d'_e satisfy

$$\max_e d'_e = m \cdot 4^k \cdot \max_e d_e. \tag{5.49}$$

Even if the Hamiltonian is not k-local, a similar argument holds for a Pauli Hamiltonian:

$$H = \sum_{i=1}^{m} P_i, \tag{5.50}$$

where P_i is an n-qubit Pauli operator and $m = \text{poly}(n)$. Because the action $P_i |\text{TNS}\rangle$ does not increase the bond dimension, the resulting bond dimension d'_e satisfies

$$\max_e d'_e = m \cdot \max_e d_e. \tag{5.51}$$

5.3.2 Chebyshev polynomial expansion

The core of the GSEE algorithm based on eigenvalue filtering lies in evaluating the quantity $\langle\psi|P(H)|\psi\rangle$. In this chapter, we expand the polynomial filter $P(H)$ in terms of Chebyshev polynomials and compute the above expression using recurrence relations [48, 208, 220].

Let $T_k(x)$ denote the Chebyshev polynomial of the first kind, defined by the recurrence relations:

$$T_0(x) = 1, \tag{5.52}$$
$$T_1(x) = x, \tag{5.53}$$
$$T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x) \quad \text{for } k \geq 2. \tag{5.54}$$

We expand the polynomial $P(x)$ in a Chebyshev series of degree d:

$$P(x) = \frac{a_0}{2} + \sum_{k=1}^{d} a_k T_k(x), \tag{5.55}$$

where a_k are the expansion coefficients. Then, the quantity of interest becomes

$$\begin{aligned}\langle\psi|P(H)|\psi\rangle &= \frac{a_0}{2} + \sum_{k=1}^{d} a_k \langle\psi|T_k(H)|\psi\rangle \\ &= \frac{a_0}{2} + \sum_{k=1}^{d} a_k \mu_k,\end{aligned} \tag{5.56}$$

where we define the *Chebyshev moments* as $\mu_k = \langle\psi|T_k(H)|\psi\rangle$.

Let $|t_k\rangle := T_k(H)|\psi\rangle$ be the kth *Chebyshev vector*, which can be recursively computed as:

$$|t_0\rangle = |\psi\rangle, \tag{5.57}$$
$$|t_1\rangle = H|\psi\rangle, \tag{5.58}$$
$$|t_k\rangle = 2H|t_{k-1}\rangle - |t_{k-2}\rangle. \tag{5.59}$$

Then, Chebyshev moments are calculated from the Chebyshev vectors as follows [48, 208, 220]:

$$\mu_{2k} = 2\,\langle t_k|t_k\rangle - \mu_0, \tag{5.60}$$

$$\mu_{2k+1} = 2\,\langle t_{k+1}|t_k\rangle - \mu_1. \tag{5.61}$$

Therefore, to calculate (5.56) of order $2d$, Chebyshev vectors up to order d are required.

The use of Chebyshev polynomials, rather than monomials, is motivated by their superior numerical stability. For any integer k, the Chebyshev vector $T_k(H)\,|\psi\rangle$ admits the eigenbasis expansion

$$T_k(H)\,|\psi\rangle = \sum_i c_i\,T_k(\lambda_i)\,|\lambda_i\rangle\,, \tag{5.62}$$

which oscillates as $T_k(\lambda_i) = \cos(k\theta_i)$ when we write $\lambda_i = \cos\theta_i$. Hence the Chebyshev vectors remain $O(1)$ in norm for all k: they do not exhibit exponential growth or decay.

Moreover, if a function $f(x)$ satisfying $|f(x)| \le 1$ on $[-1, 1]$ is expanded into Chebyshev polynomials,

$$f(x) = \frac{a_0}{2} + \sum_{k\ge 1} a_k\,T_k(x), \tag{5.63}$$

then the coefficients satisfy the uniform bounds

$$|a_0| \le 1, \qquad |a_k| \le \frac{4}{\pi} \quad (k \ge 1). \tag{5.64}$$

This is in sharp contrast to the monomial expansion (5.39), whose coefficients grow exponentially in k. In the latter case, the estimation error of $\langle\psi|H^k|\psi\rangle$ is amplified by a factor of $O(4^k)$, making reliable estimation rapidly infeasible.

These stability properties, the $O(1)$ norm of Chebyshev vectors and Chebyshev coefficients, are essential for implementing dequantized algorithms in practical numerical settings, where issues such as loss of accuracy and floating-point roundoff can easily arise. They are particularly crucial when approximation techniques are applied, as discussed later.

5.3.3 Complexity analysis via tensor networks

We analyze the computational cost of evaluating $\langle\psi|P(H)|\psi\rangle$ using tensor network contraction.

Lemma 5 (Computing moments with tensor network states)**.** *Assume that the Hamiltonian H and the guiding state $|\psi\rangle = |\mathrm{TNS}(G, \{T_v\}, \{d_e\})\rangle$ are efficiently contractible. Then the moment $\mu_d = \langle\psi|T_d(H)|\psi\rangle$ can be computed in time*

$$\mathrm{poly}((D_H)^{d/2}, D, n), \tag{5.65}$$

where D_H is the maximum bond dimension growth induced by applying H, d is the degree of the polynomial, and D is the maximum bond dimension of the initial guiding state.

Proof. We represent the Chebyshev vectors as tensor network states:

$$|t_k\rangle = T_k(H)\,|\mathrm{TNS}(G, \{T_v\}, \{d_e\})\rangle\,. \tag{5.66}$$

Let $D^{(k)}$ be the maximum bond dimension of the kth Chebyshev vector $|t_k\rangle$. D_H is the maximum growth in bond dimension when H acts on a tensor network state:

$$\max_e d'_e \leq D_H \max_e d_e, \tag{5.67}$$

where d'_e is the bond dimension of $H\,|\psi\rangle$. Then from the recurrence relation (5.57)–(5.59),

$$D^{(0)} = D, \tag{5.68}$$

$$D^{(1)} = D_H D, \tag{5.69}$$

$$D^{(k)} = D_H D^{(k-1)} + D^{(k-2)}. \tag{5.70}$$

To leading order, this implies

$$D^{(k)} = \mathcal{O}((D_H)^k D). \tag{5.71}$$

Because each Chebyshev vector can be represented by a tensor network state whose bond dimension is polynomial in D and $(D_H)^k$, their inner products can also be calculated in time polynomial in $D, (D_H)^k$ and n. By using (5.60) and (5.61), we can efficiently compute $\langle\psi|T_d(H)|\psi\rangle$ using Chebyshev vectors up to $O(d/2)$ degree, which proves the claim. □

Theorem 3. *Assume that the Hamiltonian H and the guiding state $|\psi\rangle = |\mathrm{TNS}(G, \{T_v\}, \{d_e\})\rangle$ are efficiently contractible, and the guiding state has an overlap $\chi = |\,\langle\psi|\lambda_0\rangle\,|$ with the ground state of H. Let $D = \max_e d_e$ and D_H be the maximum bond dimension growth induced by applying H. Then, there exists a classical algorithm that computes an ϵ-approximation of the ground state energy $\mathcal{E}(H)$ in time*

$$\mathrm{poly}((D_H)^{1/(2\epsilon)}, D, n). \tag{5.72}$$

Proof. We consider an approximate shifted sign function $P_{\eta,\epsilon,c}(x)$ with degree $d = \mathcal{O}(1/\epsilon \cdot \log(1/\eta))$, and expand it in the Chebyshev basis:

$$P_{\eta,\epsilon,c}(x) = \frac{a_0}{2} + \sum_{k=1}^{d} a_k T_k(x). \tag{5.73}$$

From equation (5.64), the coefficients $\{a_k\}$ are uniformly bounded by a constant.

We can compute the quantity $\langle\psi|P_{\eta,\epsilon,c}(H)|\psi\rangle$ using the Chebyshev moments:

$$\langle\psi|P_{\eta,\epsilon,c}(H)|\psi\rangle = \frac{a_0}{2} + \sum_{k=1}^{d} a_k \mu_k. \tag{5.74}$$

By Lemma 5, these moments can be computed in time $\mathrm{poly}(D_H^{d/2}, D, n)$. The procedure for estimating the ground state energy from $\langle\psi|P_{\eta,\epsilon,c}(H)|\psi\rangle$ is the same as in Lemma 3 and 4. □

As an example, we estimate the computational cost when the guiding state is given as an MPS and the Hamiltonian is a Pauli Hamiltonian. In this case, the increase in bond dimension from applying the Hamiltonian is $D_H = m$, where m is the number of Pauli terms. Taking into account the cost of inner product computation for MPS, the total cost becomes

$$O^*\left(D^3 m^{1.5/\epsilon}\right). \tag{5.75}$$

Compared to the sampling-based approach whose cost scales as $O^*(1/\epsilon \cdot 2^{(k+8)/\epsilon} \cdot \chi^{-8})$, this formulation removes the polynomial dependence on the overlap χ and eliminates heavy factors such as $2^{(k+8)/\epsilon}$ that arise from sampling overhead. On the other hand, the dependence on m, the number of terms in the local Hamiltonian, becomes explicit. Since $m = O(n)$ in many physically relevant cases, the tensor network-based approach becomes more expensive in regimes where n is large.

However, it is important to note that this is a worst-case estimate. For instance, in the case of a nearest-neighbor 2-local Hamiltonian, we have $D_H = 3$, leading to an overall complexity of

$$O^*\left(D^3 3^{1.5/\epsilon}\right), \tag{5.76}$$

which is significantly smaller. Furthermore, as discussed in the next section, the use of tensor network approximation techniques allows one to heuristically suppress the bond dimension, thereby reducing the computational cost in practice.

5.4 Practical dequantized algorithm with tensor network approximation

In the previous sections, we analyzed the computational complexity of the GSEE problem using tensor networks. Under similar assumptions to those in prior works based on sample-and-query access, the GSEE problem can be solved with polynomial time complexity in n when the target precision is constant. Moreover, by exploiting the locality of the Hamiltonian, our method achieves significantly lower computational cost than sampling-based approaches. However, in both cases, the required bond dimension grows exponentially with the polynomial degree d, which limits the practicality of the algorithm [221, 222]. As such, the current dequantized algorithm remains within the scope of computational complexity theory.

To address this limitation, we propose to use tensor network approximation in this chapter. There exists a wide range of approximation techniques for tensor networks, and in many physically relevant systems, the bond dimension can be

significantly reduced. In our case, since the Chebyshev vectors are represented as tensor networks, the core challenge becomes how efficiently these vectors can be approximated using tensor network ansatz. Furthermore, by extrapolating the Chebyshev moments with a linear prediction, we can effectively access much higher polynomial degrees.

5.4.1 Approximated Chebyshev vectors

We consider representing Chebyshev vectors $|t_k\rangle$ as an approximate tensor network state $|\widetilde{t_k}\rangle$, such as an MPS. We define the accuracy of the approximated state $\Delta^{(k)}$ as the difference from the actual state:

$$\Delta^{(k)} = \| \, |t_k\rangle - |\widetilde{t_k}\rangle \, \|. \tag{5.77}$$

Then, the following property holds.

Theorem 4. *Assume the Hamiltonian H and the guiding state $|\psi\rangle$ satisfy the conditions of Theorem 3. Let $d = O^*(1/\epsilon)$ be the degree required to construct $\langle t_0|P_{\eta,\epsilon,c}(H)|t_0\rangle$. Consider the approximated tensor network states $|\widetilde{t_k}\rangle$ with maximum bond dimension D_1. If the accuracy of these approximated states satisfies*

$$\sum_{k=1}^{d} \Delta^{(k)} \leq \frac{\pi}{32}\chi^2, \tag{5.78}$$

then we can compute an ϵ-approximation of the ground state energy $\mathcal{E}(H)$ in time

$$\mathrm{poly}(1/\epsilon, D_1, n). \tag{5.79}$$

This theorem claims that, if we can obtain the approximated Chebyshev vectors with high quality, we can solve the GSEE problem in polynomial time in the degree d. In this case, the exponential runtime overhead associated with d has been removed, resulting in a practically feasible dequantized algorithm.

Proof. We estimate $\langle t_0|P_{\eta,\epsilon,c}(H)|t_0\rangle$ using approximate tensor network states:

$$\langle\psi|P_{\eta,\epsilon,c}(H)|\psi\rangle \simeq \frac{a_0}{2} + \sum_{k=1}^{d} a_k \, \langle t_0|\widetilde{t_k}\rangle \, . \tag{5.80}$$

Its error can be upper bounded by:

$$\begin{aligned} &| \, \langle\psi|P_{\eta,\epsilon,c}(H)|\psi\rangle - \frac{a_0}{2} - \sum_{k=1}^{d} a_k \, \langle t_0|\widetilde{t_k}\rangle \, | \\ &\leq \sum_{k=1}^{d} |a_k| | \, \langle t_0|t_k\rangle - \langle t_0|\widetilde{t_k}\rangle \, | \\ &\leq \frac{4}{\pi} \cdot \sum_{k=1}^{d} \Delta^{(k)}. \end{aligned} \tag{5.81}$$

If the error of the tensor network approximation is upper bounded by (5.78), then the estimation error of $\langle\psi|P_{\eta,\epsilon,c}(H)|\psi\rangle$ is upper bounded by:

$$|\langle\psi|P_{\eta,\epsilon,c}(H)|\psi\rangle - \frac{a_0}{2} - \sum_{k=1}^{d} a_k \langle t_0|\widetilde{t_k}\rangle| \leq \frac{\chi^2}{8}. \tag{5.82}$$

From Eqs (5.29) and (5.30), we can successfully distinguish two cases and execute a binary search to estimate the ground state energy. □

For later discussions, let $\widetilde{\mu}_i$ be the approximated moments:

$$\widetilde{\mu}_{2k} = 2\langle\widetilde{t}_k|\widetilde{t}_k\rangle - \widetilde{\mu}_0, \tag{5.83}$$

$$\widetilde{\mu}_{2k+1} = 2\langle\widetilde{t}_{k+1}|\widetilde{t}_k\rangle - \widetilde{\mu}_1. \tag{5.84}$$

We also define the moment approximation error $\Delta_m^{(k)}$ as:

$$\Delta_m^{(k)} = |\mu_k - \widetilde{\mu}_k|. \tag{5.85}$$

Lemma 6. *Assume that the Hamiltonian H and the guiding state $|\psi\rangle$ satisfy the conditions of Theorem 3. Let $d = O^*(1/\epsilon)$ be the polynomial degree required to construct $\langle t_0|P_{\eta,\epsilon,c}(H)|t_0\rangle$. If the approximated moments $\{\widetilde{\mu}_k\}_{k=0}^d$ satisfy*

$$\sum_{k=1}^{d} \Delta_m^{(k)} \leq \frac{\pi}{32}\chi^2, \tag{5.86}$$

then we can compute an ϵ-approximation of the ground state energy $\mathcal{E}(H)$ in time

$$\text{poly}(d). \tag{5.87}$$

Proof. The estimation error of $\langle t_0|P_{\eta,\epsilon,c}(H)|t_0\rangle$ can be bounded as follows:

$$\begin{aligned}
&|\langle\psi|P_{\eta,\epsilon,c}(H)|\psi\rangle - \frac{a_0}{2} - \sum_{k=1}^{d} a_k\widetilde{\mu}_k| \\
&\leq \sum_{k=1}^{d} |a_k||\mu_k - \widetilde{\mu}_k| \\
&\leq \frac{4}{\pi}\sum_{k=1}^{d} \Delta_m^{(k)}.
\end{aligned} \tag{5.88}$$

□

5.4.2 Approximation method: MPS and linear prediction

To apply the above theorem, it is necessary to perform an efficient tensor network approximation of the Chebyshev vectors. In this chapter, we focus in particular on Matrix Product States (MPS), which are widely used to represent one-dimensional quantum many-body systems.

Matrix Product States (MPS)

MPS is efficiently contractible tensor network states typically used for 1D quantum systems. We assume that the guiding state $|t_0\rangle$ is represented as an MPS. The approximate Chebyshev vectors are generated via the recurrence relations:

$$|\widetilde{t}'_1\rangle = H\,|t_0\rangle\,, \tag{5.89}$$

$$|\widetilde{t}'_k\rangle = 2H\,|\widetilde{t}_{k-1}\rangle - |\widetilde{t}_{k-2}\rangle\,. \tag{5.90}$$

Applications of H increase the bond dimension of the MPS. We impose a maximum bond dimension χ_{mps} and truncate if the bond dimension of the network exceeds this threshold:

$$|\widetilde{t}_k\rangle = \text{truncate}(|\widetilde{t}'_k\rangle). \tag{5.91}$$

In this chapter, we use a standard MPS compression algorithm based on the canonical form and bond truncation with singular value decomposition (SVD) [42, 88]. Higher-quality approximations can also be obtained using iterative methods such as two-site DMRG [39], which we did not employ due to their runtime overhead and relatively small improvements.

In practice, the truncation error at step k is usually quantified by the *cosine error*:

$$\Delta_c^{(k)} = \left| 1 - \frac{\langle \widetilde{t}_k | \widetilde{t}'_k \rangle}{\| \,|\widetilde{t}_k\rangle\,\| \,\| \,|\widetilde{t}'_k\rangle\,\|} \right|. \tag{5.92}$$

Note that the cosine error is a local fitting error for each step and differs from the global error $\Delta^{(k)}$ used in Theorem 4. Nevertheless, we will see in the section on numerical experiments that this cosine error serves as a good indicator of the actual approximation error.

The truncation error at step k is also quantified by *truncation error* $\Delta_t^{(k)}$, defined as:

$$\Delta_t^{(k)} = \| \,|\widetilde{t}'_k\rangle - |\widetilde{t}_k\rangle\,\|. \tag{5.93}$$

This quantity is also efficiently calculated within the MPS framework. The local error does not grow exponentially due to the property of the Chebyshev recursion, and we can obtain a uniform upper bound for local truncation errors to guarantee that the dequantization algorithm works.

Theorem 5. *Assume the conditions of Theorem 4 hold. Let δ be a uniform upper bound on the local truncation error defined in Eq.* (5.93)*, i.e., $\Delta_t^{(k)} \leq \delta$ for all k. Then, the global accuracy condition (Eq.* (5.78)*) is satisfied if:*

$$\delta \leq \frac{3\pi\chi^2}{16d^3} = O(\chi^2 d^{-3}). \tag{5.94}$$

Proof. Let $|e_k\rangle = |t_k\rangle - |\widetilde{t}_k\rangle$ be the global error vector at step k. The local error introduced at step k is $|\eta_k\rangle = |\widetilde{t}'_k\rangle - |\widetilde{t}_k\rangle$, with norm $\| \,|\eta_k\rangle\,\| = \Delta_t^{(k)} \leq \delta$. The error propagates according to the linear recurrence:

$$|e_k\rangle = 2H\,|e_{k-1}\rangle - |e_{k-2}\rangle + |\eta_k\rangle\,. \tag{5.95}$$

The local truncation error $|\eta_j\rangle$ introduced at step j accumulates at a later step k ($k \geq j$). This contribution is given by:

$$U_{k-j}(H)\,|\eta_j\rangle\,, \tag{5.96}$$

where $U_k(x)$ denotes the Chebyshev polynomial of the second kind of degree k. A key property of these polynomials is that their maximum absolute value on the interval $[-1, 1]$ grows linearly with their degree:

$$\max_{x\in[-1,1]} |U_n(x)| = n + 1. \tag{5.97}$$

Since the eigenvalues of H lie within $[-1, 1]$, the operator norm is bounded by $\|U_{k-j}(H)\| \leq k - j + 1$. Consequently, the norm of the local error $|\eta_j\rangle$ is amplified by a factor of at most $(k - j + 1)$ when it propagates to step k.

Assuming the worst-case accumulation, the norm of the global error at step k is bounded by:

$$\| \,|e_k\rangle\, \| \leq \sum_{j=1}^{k} (k - j + 1) \| \,|\eta_j\rangle\, \| \leq \delta \sum_{m=1}^{k} m \approx \frac{k^2}{2}\delta. \tag{5.98}$$

Summing these global errors up to degree d as required by Theorem 4:

$$\sum_{k=1}^{d} \| \,|e_k\rangle\, \| \lesssim \sum_{k=1}^{d} \frac{k^2}{2}\delta \approx \frac{d^3}{6}\delta. \tag{5.99}$$

Requiring this total error to be bounded by $\frac{\pi}{32}\chi^2$, we obtain the condition $\frac{d^3}{6}\delta \leq \frac{\pi}{32}\chi^2$, which yields the stated bound. □

This theorem provides a sufficient condition for the approximated dequantized algorithm to execute with theoretical guarantees. We emphasize, however, that this bound is likely conservative, as it is derived from a worst-case error analysis. Specifically, the $O(d^{-3})$ dependence already imposes a requirement of $\Delta_t \sim 10^{-6}$ even for moderate degrees such as $d \sim 100$. Given that the cosine error Δ_c, used as a local fitting error in our numerical simulations, scales quadratically with the truncation error ($\Delta_c = O(\Delta_t^2)$), strictly satisfying this theoretical bound would correspond to a high-accuracy regime (e.g., $\Delta_c \sim 10^{-12}$). In practice, our numerical results indicate that the algorithm can perform accurately even when the errors exceed this threshold, as we will discuss later.

Linear prediction

In our approximation strategy, we represent the Chebyshev vectors as tensor network states. Due to entanglement growth, accurately representing these vectors requires sufficiently large bond dimensions. However, computing tens of thousands of approximate tensor network states with such a large bond dimension is computationally demanding.

As shown in Lemma 6, only the approximate moments $\widetilde{\mu}_k$ are required for the ground state energy estimation, rather than the full tensor network states. To this end, we adopt a linear prediction (LP) scheme [223, 224] to extrapolate the moment sequence $\{\mu_k\}$ beyond the range directly accessible to MPS.

Recall that the initial state is written in the eigenbasis of H:

$$|\psi\rangle = \sum_i c_i \, |\lambda_i\rangle \tag{5.100}$$

and define $\theta_i = \arccos(\lambda_i)$. Each moment can be written as

$$\begin{aligned} \mu_k &= \langle\psi|T_k(H)|\psi\rangle \\ &= \sum_i |c_i|^2 \cos(k\theta_i), \end{aligned} \tag{5.101}$$

showing that the sequence is a finite linear combination of oscillatory modes. If the initial state $|\psi\rangle$ has non-zero overlap with only a few eigenstates (i.e., only a small number of coefficients $|c_i|^2$ are dominant), then the moment sequence (5.101) is governed by only a few frequencies $\{\theta_i\}$. In this case, the effective dimensionality of the signal is low, and the sequence is well approximated by a low-order autoregressive model.

Suppose we calculate the Chebyshev vectors up to degree $N_{\max}$ and have a moment sequence $\{\mu_k\}_{k=0}^{2N_{\max}}$. We model the tail of the moment sequence using the relation

$$\mu_n \simeq -\sum_{j=1}^{n_{\text{fit}}} a_j \mu_{n-j}, \tag{5.102}$$

where $\{a_j\}$ are real coefficients. These coefficients are obtained by a least-squares fit of Eq. (5.102) over the final $n \in \{2N_{\max} - n_{\text{fit}}, \ldots, 2N_{\max} - 1\}$ moments. Once the coefficients are determined, the moments for $n > 2N_{\max}$ are generated recursively using Eq. (5.102).

In this way, the linear prediction procedure allows us to obtain $\{\mu_n\}$ for $n \gg 2N_{\max}$ without further MPS calculations. Using these extrapolated moments, we can then approximately evaluate the higher-degree polynomial for $\langle\psi|P(H)|\psi\rangle$. Note that this is a heuristic approach, and its accuracy depends on the initial state and the system we consider. A theoretical characterization of when linear prediction yields reliable extrapolations in this context remains an important topic for future work.

5.5 Results

In this section, we implement the proposed method and evaluate its performance through numerical experiments on benchmark models. The tensor network simulations are implemented using the quimb [183] library, and all computations are executed on a machine with an AMD EPYC 7532 32-core processor and an NVIDIA A100 40-GB GPU.

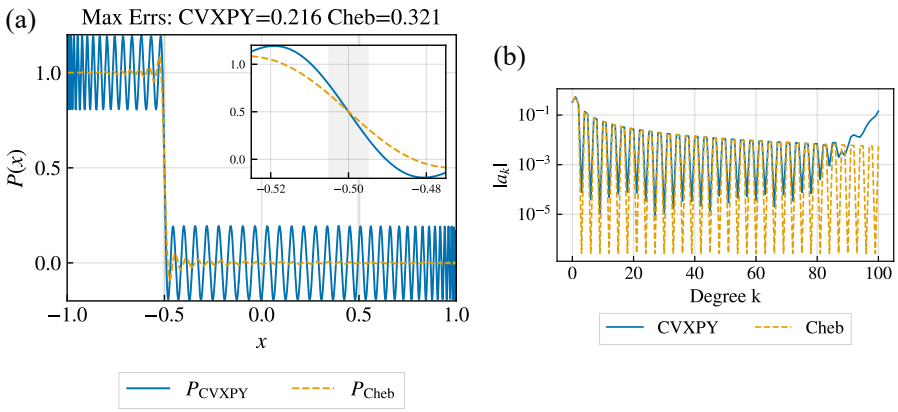


Figure 5.5: The comparison between two polynomial approximations. (a) The polynomials that approximate the shifted sign function are plotted. P_{CVXPY} is the one obtained by CVXPY and P_{Cheb} is obtained by Chebyshev truncation. The discontinuity point is set to $c = -0.5$. The inset is the zoom-in version around $x = c$. (b) The Chebyshev coefficients $|a_k|$ are plotted as a function of k.

5.5.1 Problem settings

Target model

In this chapter, we consider the one-dimensional and two-dimensional transverse-field Ising models (1D TFIM and 2D TFIM) as benchmark models. The 1D TFIM is defined on an open chain:

$$H_{\mathrm{1D}} = -J \sum_{i=1}^{L-1} X_i X_{i+1} - h \sum_{i=1}^{L} Z_i, \tag{5.103}$$

and the 2D TFIM is defined on an open lattice with $L \times L$ sites:

$$H_{\mathrm{2D}} = -J \sum_{\langle i,j \rangle} X_i X_j - h \sum_{i} Z_i. \tag{5.104}$$

Depending on the ratio h/J, the system exhibits a quantum phase transition ($h/J = 1.0$ for the 1D TFIM and $h/J \simeq 3.044$ for the 2D TFIM). Because of its simplicity and rich physics, the TFIM is widely used as a benchmark model for testing numerical methods based on tensor networks. In this chapter, we normalize $\|H\| \leq 1$ so that the eigenvalue filter can be applied.

Chebyshev polynomial approximation

Several methods have been proposed for constructing Chebyshev polynomial approximations to the shifted sign function, including those based on numerical

optimization [198, 225]. However, optimization-based approaches become prohibitively slow and unstable, especially when the degree exceeds a few thousand.

In this chapter, we instead generate the required polynomials from a Chebyshev expansion of the corresponding error function. Fig. 5.5(a) compares two polynomial approximations of the shifted sign function: one obtained using CVXPY and the other obtained by truncating the Chebyshev expansion of the error function. The gap parameter is set to $\Delta = 0.01$ and the polynomial degree is set to $d = 100$, and no parity constraint is imposed. The maximum approximation error of the CVXPY polynomial is lower than that of the Chebyshev truncation: 0.216 versus 0.321. This can be confirmed by inspecting the inset of Fig. 5.5(a): the CVXPY polynomial can reproduce the discontinuity better. However, the CVXPY-generated polynomial exhibits strong oscillations outside the gap region, whereas the Chebyshev truncation remains stable except for the Gibbs oscillations near the discontinuity.

Fig. 5.5(b) plots the magnitude of the coefficients $|a_k|$ for both polynomials. For the Chebyshev truncation, the coefficients decay exponentially with k, while the CVXPY polynomial exhibits larger coefficients at higher orders. This indicates that the CVXPY solution reduces the approximation error by actively employing higher-degree terms. Nevertheless, as seen from Eq. (5.88), the error in the quantity of interest involves the product of $|a_k|$ and $\Delta_m^{(k)}$. Since $\Delta_m^{(k)}$ grows with k, a polynomial with exponentially decaying coefficients, such as that obtained by Chebyshev truncation, is far more desirable for classical simulation.

Although Chebyshev truncation is not optimal in the max-norm, its error differs from the optimal one by at most a constant factor. Furthermore, the Chebyshev truncation approach remains numerically stable and computationally efficient even for degrees up to tens of thousands. For these reasons, we adopt Chebyshev truncation as our method for constructing the polynomial approximations.

In principle, more sophisticated optimization methods may yield polynomials with higher accuracy and improved stability. Such optimized constructions could be beneficial not only for classical simulations but also for practical implementations of QSVT, where the degree directly determines the circuit depth. Developing such optimized and robust polynomial constructions is an interesting direction for future work.

Numerical experiment settings

We adopt an MPS-based implementation of the dequantization algorithm for our numerical simulations. To run the algorithm, we require an initial MPS with a sufficiently large overlap with the true ground state of the model. We prepare the initial states by running DMRG using TeNPy [226], with the bond dimension restricted to χ_{init}.

We then compute the Chebyshev vectors according to Eqs. (5.89), (5.90) and (5.91), using an MPS approximation whose bond dimension is truncated to χ_{mps} at every step. After obtaining $N_{\max}$ Chebyshev vectors, we compute the

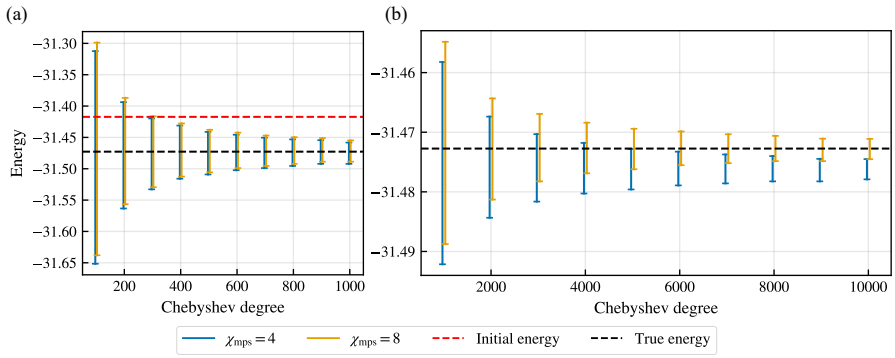


Figure 5.6: The estimated energy is plotted as a function of Chebyshev degree (a) from 100 to 1000 and (b) from 1000 to 10000 for the 1D TFIM with $L = 25, J = 1.0, h = 1.0$. Each color represents a different maximum bond dimension χ_{mps} used for the Chebyshev vector. The red dashed line represents the energy of the initial state, and the black dashed line represents the exact ground state energy. The initial energy is prepared with $\chi_{\mathrm{init}} = 2$.

$2N_{\max}$ Chebyshev moments using Eqs. (5.83) and (5.84). We then apply linear prediction to extrapolate the moment sequence if needed.

Given access to the moments, we can evaluate expectation values $\langle\psi|P(H)|\psi\rangle$ for polynomial P. This enables a binary-search procedure for ground state energy estimation: by repeatedly shrinking the gap of a shifted sign function, we can exponentially refine the estimate of the ground energy.

For simplicity, in this chapter we fix the gap parameter as $\Delta = 1/d$, where d is the maximal moment order used. Empirically, we have found that this choice suffices to construct a high-accuracy polynomial approximation via Chebyshev truncation. For a fixed gap, we first construct a family of polynomials $\{P_{\eta,\Delta,x}\}_x$, where x ranges from -1 to 1. We then evaluate the cumulative function

$$C(x) := \frac{a_0^{(x)}}{2} + \sum_{k=1}^{d} a_k^{(x)}\,\mu_k, \tag{5.105}$$

where $a_k^{(x)}$ denotes the Chebyshev coefficients of $P_{\eta,\Delta,x}$. We take the value of x for which $C(x)$ is closest to the threshold $\chi^2/2$, and return the interval $[x-\Delta/2, x+\Delta/2]$ as the resulting estimate with error Δ. The solution obtained in this way is reachable by the original binary-search procedure.

5.5.2 Numerical results

To obtain a deeper understanding of the proposed algorithm, we first perform simulations in a regime where full state-vector simulation is still feasible. Fig. 5.6 shows the results of the ground-state energy estimation for the 1D TFIM. We

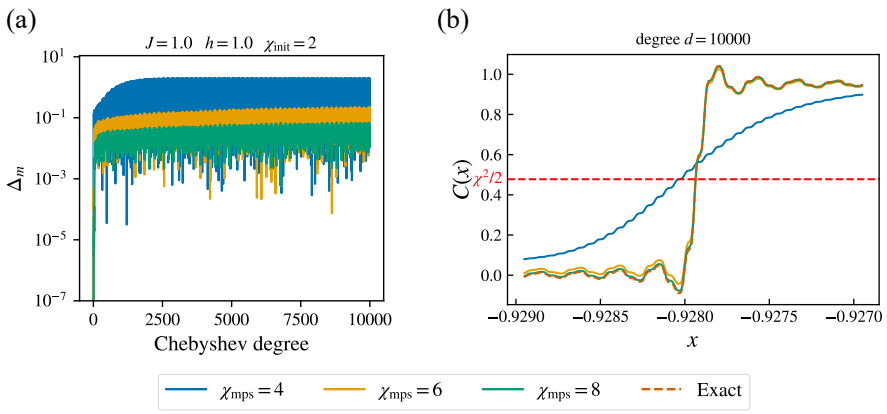


Figure 5.7: (a) The moment error $\Delta_m^{(k)}$ is plotted every 10 steps as a function of Chebyshev degree k for the 1D TFIM model. Each color represents a different maximum bond dimension of MPS. Parameters are set to $J = 1.0, h = 1.0, \chi_{\text{init}} = 2$. (b) The cumulative function $C(x)$ using d moments is plotted every 10 steps as a function of x. The dashed line shows the result with the exact moments obtained from the state-vector simulation. The red horizontal line is the threshold $\chi^2/2$ to estimate the ground state energy.

consider a system of size $L = 25$ at the critical point $J = 1.0, h = 1.0$. The initial state is prepared using DMRG with bond dimension $\chi_{\text{init}} = 2$, and each Chebyshev vector is approximated as an MPS with maximum bond dimension $\chi_{\text{mps}} = 4$ or 8. As shown in the figure, the dequantized algorithm yields a ground-state energy estimate that is more accurate than the energy of the initial state, indicating that our algorithm is working correctly. Moreover, increasing the Chebyshev degree reduces the gap of the approximated threshold function, enabling higher-precision energy estimation. The degrees we reach, $d = 10^4$, are far beyond what is achievable in the exact dequantized algorithm, including those that use Monte Carlo sampling. This demonstrates that the approximated tensor network approach brings the algorithm into a practical regime while preserving the essence of the dequantized framework.

We also plot the moment error $\Delta_m^{(k)}$ for various maximum bond dimensions χ_{mps} in Fig. 5.7(a). As expected, the moment error decreases as we increase the bond dimension and represent the Chebyshev vector more faithfully. Fig. 5.7(b) shows the cumulative function $C(x)$ constructed from the sequence of moments up to order 10^4. We observe that a larger bond dimension yields a more accurate reconstruction of $C(x)$, which in turn leads to a more reliable energy estimate. These results indicate that increasing the bond dimension systematically improves the quality of the results, just as in conventional tensor network methods in condensed matter physics and data science.

Fig. 5.8 presents the moment error and cosine error for the 1D TFIM under

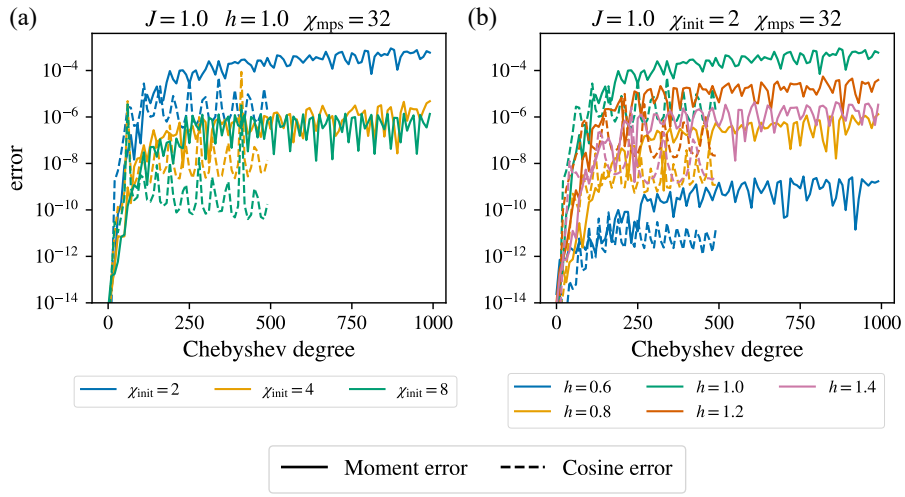


Figure 5.8: The cosine error at each step and the moment error for the 1D TFIM are plotted as functions of Chebyshev degree. (a) Different colors represent the bond dimension χ_{init} for the initial state preparation. (b) Different colors represent the different values of transverse field h.

various parameter choices. The dashed curves represent the cosine error obtained at each MPS truncation step, while the solid curves show the deviation from the exact Chebyshev moments computed using state-vector simulation. Fig. 5.8(a) shows the effect of changing the initial-state bond dimension χ_{init}. Interestingly, a larger χ_{init} leads to a more accurate representation of the Chebyshev vectors, even when we use the same bond dimension χ_{mps}. A good initial state is close to the ground state, which exhibits low entanglement entropy. Therefore, it is reasonable to expect that the application of the Hamiltonian to generate the Chebyshev vectors will keep the resulting states near the low-entanglement manifold. In the QSVT-based GSEE algorithm, a better initial state overlap χ leads to a more efficient runtime. Our results suggest, however, that the improvement of the initial state also lowers the cost of the corresponding dequantized algorithm, thereby potentially reducing the quantum advantage.

Fig. 5.8(b) shows how the error varies with the transverse field h. The hardest instance occurs at $h = 1.0$, while the error decreases as h deviates from 1.0. These observations are consistent with the intuition that, in the 1D TFIM, $h = 1.0$ corresponds to the critical point, where the initial state contains a broader set of eigenstates, making the corresponding Chebyshev vectors inherently more difficult to approximate.

Moreover, Fig. 5.8 demonstrates that the behavior of the cosine error exhibits the same trend as the actual error in the Chebyshev moments. This implies that the local fitting error strongly reflects the global error. Despite the moment error being accessible only when an exact state-vector simulation is tractable,

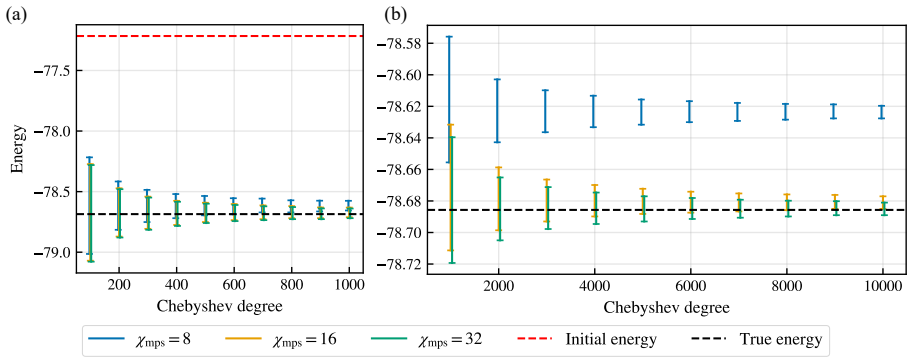


Figure 5.9: The estimated energy is plotted as a function of Chebyshev degree for the 2D TFIM with $L = 5, J = 1.0, h = 3.0$. Each color represents a different maximum bond dimension χ_{mps} for the Chebyshev vectors. The red dashed line represents the energy of the initial state, and the black dashed line represents the exact ground state energy. The initial energy is prepared with $\chi_{\text{init}} = 2$.

the cosine error can always be computed efficiently from the MPS. This confirms that the precision of the Chebyshev vectors can be reliably estimated from the cosine error.

We also perform simulations for the 2D TFIM and compare the results with the exact state-vector simulation. Fig. 5.9 shows the results of the ground state energy estimation and Fig. 5.10 shows the error in the Chebyshev moments for 2D TFIM. Both figures exhibit trends similar to the 1D case: the dequantized algorithm yields energy estimates that are more accurate than that of the initial state, and increasing either the bond dimension or the quality of the initial state leads to more accurate approximations of the Chebyshev vectors. The difficulty of the approximation also depends on the model parameters. Note that the hardest point of h deviates from the actual critical point because of finite size effects. However, the overall accuracy is significantly worse than in the 1D case. This degradation arises from the intrinsic difficulty of approximating a two-dimensional model using a one-dimensional tensor network ansatz. In the 2D TFIM, the entanglement entropy of the ground state obeys an area law $S \sim \alpha L$. Therefore, one can expect that the Chebyshev vectors also obey the same scaling, meaning that they can no longer be efficiently represented by MPS. These results indicate that substantially larger bond dimensions are required to achieve high-precision approximations in two dimensions as the system size increases.

Finally, we present results for a 100-qubit system, which is a typical target size where state-vector simulation is infeasible and a quantum computer would be required. The MPS simulation for this size becomes computationally costly if we set the bond dimension χ_{mps} large. Therefore, we combine the Chebyshev MPS simulation with the linear prediction technique described in Sec. 5.4.2.

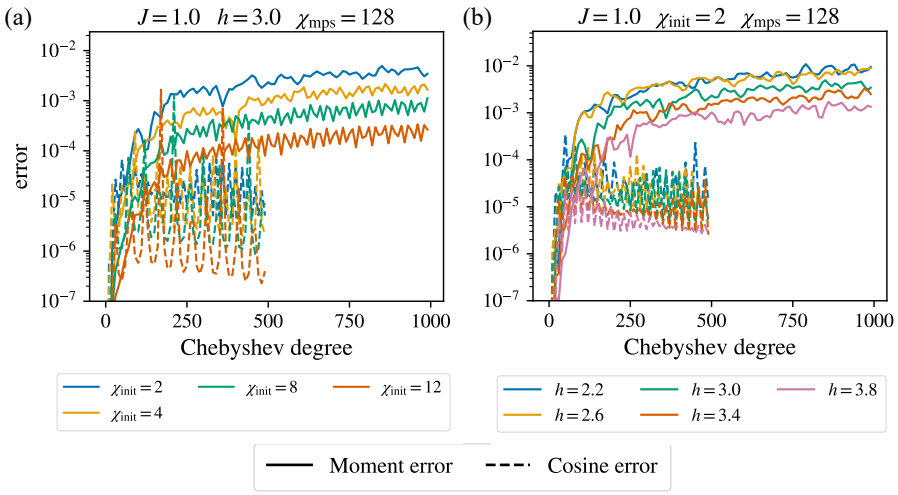


Figure 5.10: The cosine error for each step and the exact moment error for the 2D TFIM are plotted every 10 steps as functions of Chebyshev degree. (a) Different colors represent the bond dimension for the initial state preparation. (b) Different colors represent the different transverse field h.

Fig. 5.11 shows the results of ground state energy estimation for the 1D TFIM with $L = 100, J = 1.0$ and $h = 1.0$, for several choices of the maximum bond dimension. As shown in Fig. 5.11(a), $\chi_{\text{mps}} = 16$ is already sufficient to accurately approximate the Chebyshev vectors, resulting in a reliable energy estimate. Fig. 5.11(b) further shows that the cumulative function begins to display the expected jump discontinuity as the bond dimension increases, approaching the ideal shape. These observations quantitatively demonstrate that ground state energy estimation for the 1D TFIM can be easily dequantized and does not exhibit quantum advantage.

We also plot the results obtained via linear prediction as dashed lines. In both panels (a) and (b), the extrapolated results closely match those from the direct MPS calculations. In particular, for $\chi_{\text{mps}} = 16$, the linear prediction method reproduces the energy estimation and the cumulative function with near-perfect accuracy, owing to the high-quality first 1000 moments. These results indicate that the quantum algorithm may, in practice, be dequantized using much lower-order moments than one would expect.

Fig. 5.12 shows the results of ground state energy estimation for the 2D TFIM on an $L = 10$ lattice, corresponding to a 100-qubit system. Because classical computation of the Chebyshev MPS becomes increasingly expensive (each step takes several minutes for the bond dimensions considered here), we compute the moments up to order 10^3 using MPS, and use these to extrapolate the sequence up to order 10^4. In contrast to the 1D case, the bond dimensions we use in the experiments are insufficient to obtain an accurate energy estimate

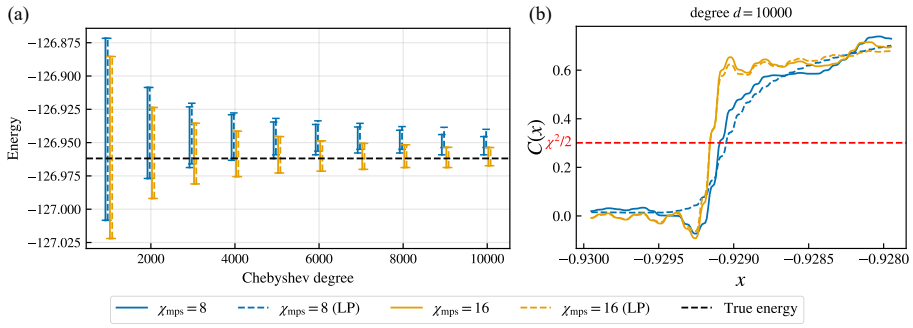


Figure 5.11: (a) The estimated energy is plotted as a function of Chebyshev degree for the 1D TFIM with $L = 100, J = 1.0, h = 1.0$. Each color represents a different maximum bond dimension χ_{mps} for the Chebyshev vectors. Dashed lines represent the results obtained from linear prediction (LP). The black dashed line represents the ground state energy obtained by DMRG with a sufficiently large bond dimension. The initial energy is prepared with $\chi_{\mathrm{init}} = 2$. (b) The cumulative function $C(x)$ obtained from moments up to 10^4 degree is plotted for different settings. The red horizontal line is the threshold $\chi^2/2$ to estimate the ground state energy.

for the 2D system due to its entanglement growth. As a result, the quantum algorithm for the high precision regime cannot be dequantized.

Nevertheless, an important observation is that the accuracy improves systematically as the bond dimension increases. This behavior suggests that, if the bond dimensions are increased further, the results will approach those of the ideal QSVT algorithm. This provides a key conceptual advantage of the tensor network-based dequantization framework. It enables an intuitive comparison between classical and quantum computational regimes through the single precision parameter ϵ, which is the spirit of the dequantization. The required precision determines the polynomial degree d, which directly determines the quantum computational cost. On the classical side, the task is to accurately compute Chebyshev moments up to order d. Because the approximation error decreases systematically with increasing bond dimension, this framework allows us to estimate the necessary classical computational resources to achieve a desired accuracy.

If increasing the bond dimension appears sufficient for the dequantized algorithm to succeed, then no quantum advantage exists in that regime. Conversely, if the Chebyshev vectors and the moment sequence cannot be approximated by any classical methods while the required polynomial degree d remains within the feasible range for a quantum computer, then the corresponding task may lie in a regime where quantum computation provides genuine utility.

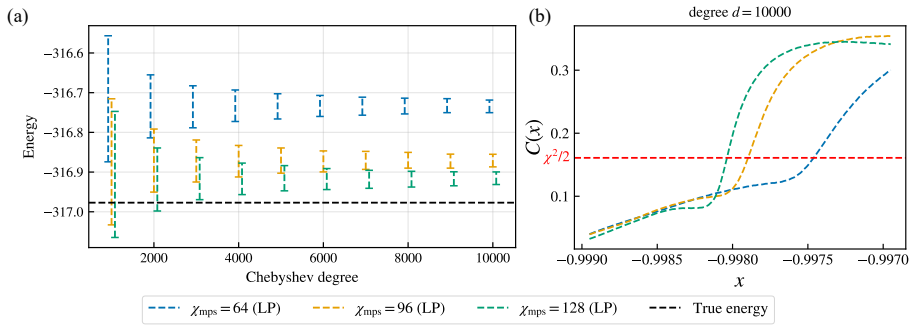


Figure 5.12: (a) The estimated energy is plotted as a function of Chebyshev degree for the 2D TFIM with $L = 10, J = 1.0, h = 3.0$. Each color represents a different maximum bond dimension χ_{mps} for the Chebyshev vectors. Moments beyond order 10^3 are estimated using linear prediction (LP). The black dashed line represents the ground state energy obtained by DMRG with sufficient bond dimension. The initial energy is prepared with $\chi_{\mathrm{init}} = 2$. (b) The cumulative function $C(x)$ obtained by moments up to 10^4 degree is plotted for different settings. Red horizontal line is the threshold $\chi^2/2$ to estimate the ground state energy.

5.6 Discussion

In this chapter, we have proposed a dequantization framework for ground state energy estimation based on tensor networks. We first showed that the tensor network-based dequantized algorithm reproduces the computational complexity of prior dequantization results while eliminating the need for Monte Carlo sampling. We then introduced practical variants that incorporate tensor network approximations, which run in time linear in $1/\epsilon$ when the approximation error is within a tolerable range. Finally, we demonstrated through numerical experiments that the dequantized algorithm works for the 1D and 2D TFIM up to 100 qubits and Chebyshev degrees of order 10^4. Although the 2D TFIM could not be fully dequantized with the bond dimensions used in our experiments, this limitation is itself informative: it suggests the boundary between the classically tractable and quantum-advantageous regions.

The ultimate goal of our research is to identify the boundary between classical and quantum computational power, and to determine the regimes in which quantum computation provides a genuine advantage. There are several possible approaches to answering this question. One approach is to compare against the best-known classical heuristics. For the GSEE problem, a variety of efficient and accurate numerical algorithms have long been developed, such as DMRG. However, because classical and quantum algorithms have fundamentally different structures, performing such comparisons in a systematic and model-independent manner is challenging. Another approach is the classical simulation of quantum

circuits [61, 64]. This provides a unified comparison framework because all quantum algorithms are expressed as quantum circuits and can be evaluated under a single metric of classical simulation cost. While this method has been successfully applied to NISQ algorithms, it becomes extremely challenging for FTQC algorithms involving many qubits and deep circuits. Furthermore, such simulations generally cannot exploit problem-specific structure or incorporate approximation methods effectively.

The tensor network-based dequantization framework addresses several of these issues. First, the classical and quantum algorithms share the same structure, and their complexity can be compared using a single accuracy parameter ϵ, or equivalently, the polynomial degree of the filter function. Classical simulations are limited by the degree for which the approximation remains successful. In this sense, the hardness of the classical simulation is fully characterized by the difficulty of approximating the Chebyshev vectors. On the other hand, the cost of a quantum algorithm, at least in theory, scales only polynomially with the degree. This perspective allows us to intuitively separate the regime in which classical methods remain effective from the regime in which quantum advantage is possible, using a single parameter ϵ or the Chebyshev degree d.

The role of approximation is also clear in this framework. Classical executability reduces to how accurately tensor network states can represent the Chebyshev vectors and their moments. Different tensor network ansatz, like tree tensor network [171], multi-scale renormalization ansatz [106], and projected entangled pair states [46], may yield substantially better approximations depending on the structure of the underlying physical system.

An interesting direction for future work is to apply this dequantization methodology to other QSVT-based algorithms. The core requirement of our method is the ability to approximate the Chebyshev vectors $|t_k\rangle$ and the moments μ_k. Thus, any QSVT-based algorithm whose output can be reconstructed solely from these quantities becomes immediately dequantizable. To establish quantum advantage in such settings, it is therefore necessary to show that the required polynomial degree is sufficiently large and that the Chebyshev vectors are classically hard to approximate.

Chapter 6

Conclusion

In this thesis, we investigated how tensor network methods can be used to analyze, design, and benchmark quantum algorithms across different layers of the quantum computing stack. We focused on three concrete tasks: the simulation of quantum error correction under realistic noise, the compilation of quantum circuits for state preparation, and the analysis of quantum advantage through dequantization.

In Chapter 3, we developed a scalable tensor network-based simulation framework for QEC codes, which involve large numbers of qubits and repeated multi-qubit gate operations. By employing MPS, we simulated large-scale QEC circuits without relying on approximations such as Pauli twirling, enabling accurate investigation of the effect of realistic leakage noise. Furthermore, we successfully compared the effectiveness of different leakage removal strategies.

In Chapter 4, we demonstrated that the tensor network framework can be effectively applied to quantum circuit compilation, enabling the construction of state preparation circuits that exploit the structure of multivariate normal distributions. We provided theoretical guarantees for efficient circuit representations in the case of distributions with one-dimensional correlation structures. In addition, we proposed a heuristic compilation method that combines tensor network-based numerical techniques, including automatic structural optimization, to handle more general distributions. For several benchmark models, we showed that our method achieves substantial cost reductions compared to existing approaches.

In Chapter 5, we introduced a tensor network-based dequantization framework for ground state energy estimation. By reformulating dequantized algorithms in terms of tensor network representations of Chebyshev vectors, we eliminated stochastic sampling for estimating high-dimensional inner products and replaced statistical overhead with entanglement-controlled computational cost. Numerical experiments revealed the crossover between classically tractable and quantum-advantaged regimes, providing a practical and quantitative method for assessing quantum advantage.

A central insight of this thesis is that the theoretical properties of tensor

networks are directly connected to both algorithmic guarantees and scalability. On the one hand, tensor network theory provides rigorous criteria for when quantum states, operators, or algorithmic objects can be represented efficiently, enabling performance analysis and provable bounds. On the other hand, tensor network–specific numerical techniques allow these theoretical insights to be translated into practical simulations and optimizations beyond naive state-vector simulations. This dual role forms the core contribution of this thesis.

From a theoretical perspective, tensor networks have long been studied as efficient representations of low-entanglement quantum many-body states. This property underlies the first and third contributions of this thesis. In the simulation of QEC codes, repeated syndrome measurements project the system onto states with limited entanglement, which can be efficiently captured by MPS. Similarly, in the dequantization algorithms, we represent Chebyshev vectors as low-entanglement tensor network states. In both cases, the essential question is whether the relevant quantum objects admit an efficient tensor network ansatz.

At the same time, as emphasized in the introduction, tensor networks should not be viewed merely as tools for representing quantum states. Instead, they provide a language for describing quantum algorithms. This viewpoint is central to the second and third contributions of this thesis. In the circuit compilation problem, theoretical guarantees for tensor network representations of classical probability distributions and multivariate functions are directly translated into guarantees for circuit depth and gate complexity. These ideas originate from data science rather than condensed matter physics, highlighting the interdisciplinary nature of the tensor network framework. In the dequantization framework, we further combined tensor network representations of quantum states with tensor network descriptions of algorithms, which are developed in the field of complexity theory, enabling the construction of dequantized algorithms with rigorous theoretical guarantees.

These theoretical insights were integrated with state-of-the-art numerical tensor network techniques. The one-dimensional ansatz employed in this thesis, including MPS and TTN, admits numerically stable and scalable optimization. By combining these methods with modern high-performance computing resources such as GPUs and supercomputers, we demonstrated that classical analysis of quantum systems at the scale of tens to hundreds of qubits is feasible.

Taken together, tensor networks serve as a unifying framework between physics-driven structure (such as entanglement and locality) and computation-driven structure (including algorithms, circuits, and classical information) across the quantum computing stack. This perspective helps unify lines of tensor network research that have developed in seemingly disparate communities and clarifies why tensor networks are particularly well-suited for software-level approaches to practical quantum computing.

More broadly, the possibility of such unification is rooted in the structural nature of quantum computation itself. Quantum algorithms are composed of local linear operations, a structure that is naturally and efficiently captured by tensor network representations. At the same time, quantum computing is

intrinsically an interdisciplinary subject, lying at the intersection of quantum many-body physics and computer science. As a computer built upon the principles of quantum mechanics, a quantum computer is inherently governed by the physics of quantum many-body systems, and concepts such as quantum error correction share deep connections with holographic perspectives in high-energy physics. On the other hand, a quantum computer is, fundamentally, a computer, and therefore necessarily requires a computer science perspective, including coding theory, circuit compilation, and algorithm design. Ideally, a practical quantum computer requires the synthesis of both: a physical understanding of the underlying hardware and a computational understanding of the information processes.

Tensor networks are among the very few theoretical frameworks that simultaneously accommodate these viewpoints. By integrating these perspectives within a single tensor network framework, this thesis shows that physical nature and computational structure can be analyzed in a unified and coherent manner. As a result, tensor network methods emerge not merely as numerical tools for individual tasks, but as an essential and foundational framework for the design, analysis, and validation of quantum computers.

Several directions for future research are possible. First, while this thesis focused on one-dimensional tensor network ansatz, extending the framework to higher-dimensional and alternative ansatz remains an important direction. Second, the methods developed here address only a subset of the challenges faced in quantum computing, and applying this framework to other key tasks is an open problem. Finally, deeper theoretical aspects of tensor networks—such as infinite systems, symmetric tensor networks, alternative tensor decompositions other than SVD, and diagrammatic frameworks including quantum tensor networks and ZX-calculus—offer promising avenues for further unifying theoretical analysis with scalable numerical methods.

List of Publications

Publications included in the thesis

1. **H. Manabe**, Y. Suzuki, and A. S. Darmawan, "Efficient Simulation of Leakage Errors in Quantum Error Correcting Codes Using Tensor Network Methods", *New Journal of Physics* **27**, 114512 (2025).

2. **H. Manabe** and Y. Sano, "The State Preparation of Multivariate Normal Distributions Using Tree Tensor Network", *Quantum* **9**, 1755 (2025).

3. **H. Manabe**, T. Sugimoto, and K. Fujii, "Tensor Network Formulation of Dequantized Algorithms for Ground State Energy Estimation", arXiv:2512.13548 (2025).

Other publications

1. R. Watanabe, **H. Manabe**, T. Hikihara, and H. Ueda, "TTNOpt: Tree Tensor Network Package for High-Rank Tensor Compression", *Computer Physics Communications* **319**, 109918 (2026).

2. Y. Morohoshi, A. Nakayama, **H. Manabe**, and K. Mitarai, "Learning Functions of Hamiltonians with Hamiltonian Fourier Features", arXiv:2504.16370 (2025).

3. F. Kobayashi, **H. Manabe**, G. A. L. White, T. Farrelly, K. Modi, and T. M. Stace, "Tensor-Network Decoders for Process Tensor Descriptions of Non-Markovian Noise", arXiv:2412.13739 (2024).

Acknowledgements

First, I would like to greatly thank my supervisor, Prof. Keisuke Fujii, for his indispensable guidance and support throughout my study. His deep understanding of quantum computers, together with his forward-looking perspective on the field, played a central role in shaping my development as a researcher. I would also like to thank Prof. Kenji Harada, who supervised me during my bachelor's and master's courses at Kyoto University. His deep understanding of tensor networks and his creative use of them greatly influenced the formation of my identity as a researcher.

I am grateful to all my collaborators, Prof. Masayoshi Kubo, Ryotaro Banno, Masataka Minoji, Dr. Takato Mori, Prof. Hiroaki Matsueda, Dr. Hiroyuki Ootomo, Prof. Rio Yokota, Dr. Yasunari Suzuki, Prof. Andrew S. Darmawan, Prof. Nobuyuki Yoshioka, Dr. Yuichi Sano, Dr. Fumiyoshi Kobayashi, Dr. Gregory A. L. White, Dr. Terry Farrelly, Prof. Kavan Modi, Prof. Thomas M. Stace, Yuto Morohoshi, Akimoto Nakayama, Prof. Kosuke Mitarai, Ryo Watanabe, Prof. Toshiya Hikihara, Prof. Hiroshi Ueda and Prof. Takanori Sugimoto for fruitful collaborations and stimulating discussions. Working with them has been an invaluable experience, and their deep insights have greatly contributed to the progress of my research.

I would like to thank all members of the Fujii Group for their constructive and insightful discussions. In particular, I would like to acknowledge Prof. Hideaki Hakoshima, Dr. Yuichiro Yoshida and Dr. Junichi Haruna for many insightful discussions and conversations.

I would like to express my gratitide to Ayano Yoshida, Shotaro Yamaguchi, Shotaro Miyashita, Hayato Hirai, Tatsuki Oba and Rihito Kanazawa, with whom I worked as a team member in the Scientific Game Jam Tokyo 2024. I am also thankful to Prof. Kenji Ono for his support during this project.

Finally, I would like to express my deepest gratitide to my family for their support and encouragement throughout my academic career.

Bibliography

[1] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978. doi:10.1145/359340.359342.

[2] Peter W. Shor. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. doi:10.1137/S0097539795293172.

[3] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6):467–488, 1982. doi:10.1007/BF02650179.

[4] Seth Lloyd. Universal Quantum Simulators. *Science*, 273(5278):1073–1078, 1996. doi:10.1126/science.273.5278.1073.

[5] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. Simulated Quantum Computation of Molecular Energies. *Science*, 309(5741):1704–1707, 2005. doi:10.1126/science.1113479.

[6] Sam McArdle, András Gilyén, and Mario Berta. Quantum state preparation without coherent arithmetic, 2022. arXiv:2210.14892.

[7] Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, et al. Quantum Chemistry in the Age of Quantum Computing. *Chemical Reviews*, 119(19):10856–10915, 2019. doi:10.1021/acs.chemrev.8b00803.

[8] Y. Nakamura, Yu A. Pashkin, and J. S. Tsai. Coherent control of macroscopic quantum states in a single-Cooper-pair box. *Nature*, 398(6730):786–788, 1999. doi:10.1038/19718.

[9] Jens Koch, Terri M. Yu, Jay Gambetta, A. A. Houck, D. I. Schuster, J. Majer, Alexandre Blais, M. H. Devoret, S. M. Girvin, and R. J. Schoelkopf. Charge-insensitive qubit design derived from the Cooper pair box. *Physical Review A*, 76(4):042319, 2007. doi:10.1103/PhysRevA.76.042319.

[10] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, et al. Superconducting quantum circuits at the surface code threshold for fault tolerance. *Nature*, 508(7497):500–503, 2014. doi:10.1038/nature13171.

[11] Morten Kjaergaard, Mollie E. Schwartz, Jochen Braumüller, Philip Krantz, Joel I.-Jan Wang, Simon Gustavsson, and William D. Oliver. Superconducting Qubits: Current State of Play. *Annual Review of Condensed Matter Physics*, 11(1):369–395, 2020. doi:10.1146/annurev-conmatphys-031119-050605.

[12] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. doi:10.1038/s41586-019-1666-5.

[13] Rajeev Acharya, Dmitry A. Abanin, Laleh Aghababaie-Beni, Igor Aleiner, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Nikita Astrakhantsev, et al. Quantum error correction below the surface code threshold. *Nature*, 638(8052):920–926, 2025. doi:10.1038/s41586-024-08449-y.

[14] D. Jaksch, J. I. Cirac, P. Zoller, S. L. Rolston, R. Côté, and M. D. Lukin. Fast Quantum Gates for Neutral Atoms. *Physical Review Letters*, 85(10):2208–2211, 2000. doi:10.1103/PhysRevLett.85.2208.

[15] M. Saffman, T. G. Walker, and K. Mølmer. Quantum information with Rydberg atoms. *Reviews of Modern Physics*, 82(3):2313–2363, 2010. doi:10.1103/RevModPhys.82.2313.

[16] Dolev Bluvstein, Simon J. Evered, Alexandra A. Geim, Sophie H. Li, Hengyun Zhou, Tom Manovitz, Sepehr Ebadi, Madelyn Cain, Marcin Kalinowski, Dominik Hangleiter, et al. Logical quantum processor based on reconfigurable atom arrays. *Nature*, 626(7997):58–65, 2024. doi:10.1038/s41586-023-06927-3.

[17] J. I. Cirac and P. Zoller. Quantum Computations with Cold Trapped Ions. *Physical Review Letters*, 74(20):4091–4094, 1995. doi:10.1103/PhysRevLett.74.4091.

[18] J. M. Pino, J. M. Dreiling, C. Figgatt, J. P. Gaebler, S. A. Moses, M. S. Allman, C. H. Baldwin, M. Foss-Feig, D. Hayes, K. Mayer, et al. Demonstration of the trapped-ion quantum-CCD computer architecture. *Nature*, 592(7853):209–213, 2021. doi:10.1038/s41586-021-03318-4.

[19] Anthony Ransford, M. S. Allman, Jake Arkinstall, J. P. Campora, Samuel F. Cooper, Robert D. Delaney, Joan M. Dreiling, Brian Estey, Caroline Figgatt, Alex Hall, et al. Helios: A 98-qubit trapped-ion quantum computer, 2025. arXiv:2511.05465.

[20] N. Cody Jones, Rodney Van Meter, Austin G. Fowler, Peter L. McMahon, Jungsang Kim, Thaddeus D. Ladd, and Yoshihisa Yamamoto. Layered Architecture for Quantum Computing. *Physical Review X*, 2(3):031007, 2012. doi:10.1103/PhysRevX.2.031007.

[21] Craig Gidney and Martin Ekerå. How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits. *Quantum*, 5:433, 2021. doi:10.22331/q-2021-04-15-433.

[22] Craig Gidney. How to factor 2048 bit RSA integers with less than a million noisy qubits, 2025. arXiv:2505.15917.

[23] John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Grand Unification of Quantum Algorithms. *PRX Quantum*, 2(4):040203, 2021. doi:10.1103/PRXQuantum.2.040203.

[24] Lin Lin and Yu Tong. Heisenberg-Limited Ground-State Energy Estimation for Early Fault-Tolerant Quantum Computers. *PRX Quantum*, 3(1):010318, 2022. doi:10.1103/PRXQuantum.3.010318.

[25] Quantum AI team and collaborators. Cirq, 2020. doi:10.5281/zenodo.4062499.

[26] Héctor Abraham et al. Qiskit: An open-source framework for quantum computing, 2019. doi:10.5281/zenodo.2562110.

[27] Aleks Kissinger and John van de Wetering. PyZX: Large Scale Automated Diagrammatic Reasoning, 2019. doi:10.4204/EPTCS.318.14.

[28] Seyon Sivarajah, Silas Dilkes, Alexander Cowtan, Will Simmons, Alec Edgington, and Ross Duncan. T|ket⟩: A retargetable compiler for NISQ devices. *Quantum Science and Technology*, 6(1):014003, 2020. doi:10.1088/2058-9565/ab8e92.

[29] Yunseong Nam, Neil J. Ross, Yuan Su, Andrew M. Childs, and Dmitri Maslov. Automated optimization of large quantum circuits with continuous parameters. *npj Quantum Information*, 4(1):23, 2018. doi:10.1038/s41534-018-0072-4.

[30] Francisco J. R. Ruiz, Tuomas Laakkonen, Johannes Bausch, Matej Balog, Mohammadamin Barekatain, Francisco J. H. Heras, Alexander Novikov, Nathan Fitzpatrick, Bernardino Romera-Paredes, John van de Wetering, et al. Quantum circuit optimization with AlphaTensor. *Nature Machine Intelligence*, 7(3):374–385, 2025. doi:10.1038/s42256-025-01001-1.

[31] Ross Duncan, Aleks Kissinger, Simon Perdrix, and John van de Wetering. Graph-theoretic Simplification of Quantum Circuits with the ZX-calculus. *Quantum*, 4:279, 2020. doi:10.22331/q-2020-06-04-279.

[32] Matthew Amy, Dmitri Maslov, and Michele Mosca. Polynomial-time T-depth Optimization of Clifford+T circuits via Matroid Partitioning. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 33(10):1476–1489, 2014. doi:10.1109/TCAD.2014.2341953.

[33] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, 86(3):032324, 2012. doi:10.1103/PhysRevA.86.032324.

[34] Pavel Panteleev and Gleb Kalachev. Asymptotically good Quantum and locally testable classical LDPC codes. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 375–388, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519935.3520017.

[35] Nikolas P. Breuckmann and Jens Niklas Eberhardt. Quantum Low-Density Parity-Check Codes. *PRX Quantum*, 2(4):040101, 2021. doi:10.1103/PRXQuantum.2.040101.

[36] Nicolas Delfosse and Naomi H. Nickerson. Almost-linear time decoding algorithm for topological codes. *Quantum*, 5:595, 2021. doi:10.22331/q-2021-12-02-595.

[37] Thomas Häner and Damian S. Steiger. 0.5 Petabyte Simulation of a 45-Qubit Quantum Circuit. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–10, 2017. doi:10.1145/3126908.3126947.

[38] Roman Orus. A Practical Introduction to Tensor Networks: Matrix Product States and Projected Entangled Pair States. *Annals of Physics*, 349:117–158, 2014. doi:10.1016/j.aop.2014.06.013.

[39] Steven R. White. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69(19):2863–2866, 1992. doi:10.1103/PhysRevLett.69.2863.

[40] Jacob C. Bridgeman and Christopher T. Chubb. Hand-waving and Interpretive Dance: An Introductory Course on Tensor Networks. *Journal of Physics A: Mathematical and Theoretical*, 50(22):223001, 2017. doi:10.1088/1751-8121/aa6dc3.

[41] Ignacio Cirac, David Perez-Garcia, Norbert Schuch, and Frank Verstraete. Matrix Product States and Projected Entangled Pair States: Concepts, Symmetries, and Theorems. *Reviews of Modern Physics*, 93(4):045003, 2021. doi:10.1103/RevModPhys.93.045003.

[42] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011. doi:10.1016/j.aop.2010.09.012.

[43] J. Eisert, M. Cramer, and M. B. Plenio. Area laws for the entanglement entropy - a review. *Reviews of Modern Physics*, 82(1):277–306, 2010. doi:10.1103/RevModPhys.82.277.

[44] M. B. Hastings. An Area Law for One Dimensional Quantum Systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(08):P08024–P08024, 2007. doi:10.1088/1742-5468/2007/08/P08024.

[45] G. Vidal. Classical Simulation of Infinite-Size Quantum Lattice Systems in One Spatial Dimension. *Physical Review Letters*, 98(7):070201, 2007. doi:10.1103/PhysRevLett.98.070201.

[46] F. Verstraete and J. I. Cirac. Renormalization algorithms for Quantum-Many Body Systems in two and higher dimensions, 2004. arXiv:cond-mat/0407066.

[47] E. M. Stoudenmire and Steven R. White. Studying Two-Dimensional Systems with the Density Matrix Renormalization Group. *Annual Review of Condensed Matter Physics*, 3(Volume 3, 2012):111–128, 2012. doi:10.1146/annurev-conmatphys-020911-125018.

[48] Andreas Holzner, Andreas Weichselbaum, Ian P. McCulloch, Ulrich Schollwöck, and Jan Von Delft. Chebyshev matrix product state approach for spectral functions. *Physical Review B*, 83(19):195115, 2011. doi:10.1103/PhysRevB.83.195115.

[49] Guifré Vidal. Efficient Simulation of One-Dimensional Quantum Many-Body Systems. *Physical Review Letters*, 93(4):040502, 2004. doi:10.1103/PhysRevLett.93.040502.

[50] Steven R. White and Adrian E. Feiguin. Real-Time Evolution Using the Density Matrix Renormalization Group. *Physical Review Letters*, 93(7):076401, 2004. doi:10.1103/PhysRevLett.93.076401.

[51] D. Pérez-García, M. M. Wolf, M. Sanz, F. Verstraete, and J. I. Cirac. String Order and Symmetries in Quantum Spin Lattices. *Physical Review Letters*, 100(16):167202, 2008. doi:10.1103/PhysRevLett.100.167202.

[52] Norbert Schuch, David Perez-Garcia, and Ignacio Cirac. Classifying quantum phases using Matrix Product States and PEPS. *Physical Review B*, 84(16):165139, 2011. doi:10.1103/PhysRevB.84.165139.

[53] Xie Chen, Zheng-Cheng Gu, and Xiao-Gang Wen. Classification of gapped symmetric phases in one-dimensional spin systems. *Physical Review B*, 83(3):035107, 2011. doi:10.1103/PhysRevB.83.035107.

[54] Fernando Pastawski, Beni Yoshida, Daniel Harlow, and John Preskill. Holographic quantum error-correcting codes: Toy models for the bulk/boundary correspondence. *Journal of High Energy Physics*, 2015(6):149, 2015. doi:10.1007/JHEP06(2015)149.

[55] Patrick Hayden, Sepehr Nezami, Xiao-Liang Qi, Nathaniel Thomas, Michael Walter, and Zhao Yang. Holographic duality from random tensor networks. *Journal of High Energy Physics*, 2016(11):9, 2016. doi:10.1007/JHEP11(2016)009.

[56] Ahmed Almheiri, Xi Dong, and Daniel Harlow. Bulk Locality and Quantum Error Correction in AdS/CFT. *Journal of High Energy Physics*, 2015(4):163, 2015. doi:10.1007/JHEP04(2015)163.

[57] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

[58] I. V. Oseledets. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing*, 33(5):2295–2317, 2011. doi:10.1137/090752286.

[59] Igor L. Markov and Yaoyun Shi. Simulating quantum computation by contracting tensor networks. *SIAM Journal on Computing*, 38(3):963–981, 2008. doi:10.1137/050644756.

[60] Yong, Liu, Xin, Liu, Fang, Li, Haohuan Fu, Yuling Yang, Jiawei Song, Pengpeng Zhao, et al. Closing the "Quantum Supremacy" Gap: Achieving Real-Time Simulation of a Random Quantum Circuit Using a New Sunway Supercomputer. *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–12, 2021. doi:10.1145/3458817.3487399.

[61] Cupjin Huang, Fang Zhang, Michael Newman, Xiaotong Ni, Dawei Ding, Junjie Cai, Xun Gao, Tenghui Wang, Feng Wu, Gengyan Zhang, et al. Efficient parallelization of tensor network contraction for simulating quantum computation. *Nature Computational Science*, 1(9):578–587, 2021. doi:10.1038/s43588-021-00119-7.

[62] Feng Pan, Keyang Chen, and Pan Zhang. Solving the Sampling Problem of the Sycamore Quantum Circuits. *Physical Review Letters*, 129(9):090502, 2022. doi:10.1103/PhysRevLett.129.090502.

[63] Johnnie Gray and Stefanos Kourtis. Hyper-optimized tensor network contraction. *Quantum*, 5:410, 2021. doi:10.22331/q-2021-03-15-410.

[64] Joseph Tindall, Matthew Fishman, E. Miles Stoudenmire, and Dries Sels. Efficient Tensor Network Simulation of IBM's Eagle Kicked Ising Experiment. *PRX Quantum*, 5(1):010308, 2024. doi:10.1103/PRXQuantum.5.010308.

[65] Siddhartha Patra, Saeed S. Jahromi, Sukhbinder Singh, and Roman Orus. Efficient tensor network simulation of IBM's largest quantum processors. *Physical Review Research*, 6(1):013326, 2024. doi:10.1103/PhysRevResearch.6.013326.

[66] William Huggins, Piyush Patel, K. Birgitta Whaley, and E. Miles Stoudenmire. Towards Quantum Machine Learning with Tensor Networks. *Quantum Science and Technology*, 4(2):024001, 2019. doi:10.1088/2058-9565/aaea94.

[67] Trevor Vincent, Lee J. O'Riordan, Mikhail Andrenkov, Jack Brown, Nathan Killoran, Haoyu Qi, and Ish Dhand. Jet: Fast quantum circuit simulations with parallel task-based tensor-network contraction. *Quantum*, 6:709, 2022. doi:10.22331/q-2022-05-09-709.

[68] Sergey Bravyi, Martin Suchara, and Alexander Vargo. Efficient algorithms for maximum likelihood decoding in the surface code. *Physical Review A*, 90(3):032326, 2014. doi:10.1103/PhysRevA.90.032326.

[69] Andrew J. Ferris and David Poulin. Tensor Networks and Quantum Error Correction. *Physical Review Letters*, 113(3):030501, 2014. doi:10.1103/PhysRevLett.113.030501.

[70] Peter W. Shor. Scheme for reducing decoherence in quantum computer memory. *Physical Review A*, 52(4):R2493–R2496, 1995. doi:10.1103/PhysRevA.52.R2493.

[71] Daniel Gottesman. Stabilizer Codes and Quantum Error Correction, 1997. arXiv:quant-ph/9705052.

[72] Sebastian Krinner, Nathan Lacroix, Ants Remm, Agustin Di Paolo, Elie Genois, Catherine Leroux, Christoph Hellings, Stefania Lazar, Francois Swiadek, Johannes Herrmann, et al. Realizing Repeated Quantum Error Correction in a Distance-Three Surface Code. *Nature*, 605(7911):669–674, 2022. doi:10.1038/s41586-022-04566-8.

[73] Youwei Zhao, Yangsen Ye, He-Liang Huang, Yiming Zhang, Dachao Wu, Huijie Guan, Qingling Zhu, Zuolin Wei, Tan He, Sirui Cao, et al. Realization of an Error-Correcting Surface Code with Superconducting Qubits. *Physical Review Letters*, 129(3):030501, 2022. doi:10.1103/PhysRevLett.129.030501.

[74] Rajeev Acharya, Igor Aleiner, Richard Allen, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Juan Atalaya, Ryan Babbush, et al. Suppressing quantum errors by scaling a surface code logical qubit. *Nature*, 614(7949):676–681, 2023. doi:10.1038/s41586-022-05434-1.

[75] Kevin C. Miao, Matt McEwen, Juan Atalaya, Dvir Kafri, Leonid P. Pryadko, Andreas Bengtsson, Alex Opremcak, Kevin J. Satzinger, Zijun Chen, Paul V. Klimov, et al. Overcoming leakage in quantum error correction. *Nature Physics*, 19(12):1780–1786, 2023. doi:10.1038/s41567-023-02226-w.

[76] Panos Aliferis and Barbara M. Terhal. Fault-tolerant quantum computation for local leakage faults. *Quantum Info. Comput.*, 7(1):139–156, 2007. doi:10.26421/QIC7.1-2-9.

[77] F. Motzoi, J. M. Gambetta, P. Rebentrost, and F. K. Wilhelm. Simple Pulses for Elimination of Leakage in Weakly Nonlinear Qubits. *Physical Review Letters*, 103(11):110501, 2009. doi:10.1103/PhysRevLett.103.110501.

[78] Austin G. Fowler. Coping with qubit leakage in topological codes. *Physical Review A*, 88(4):042308, 2013. doi:10.1103/PhysRevA.88.042308.

[79] Natalie C. Brown and Kenneth R. Brown. Leakage mitigation for quantum error correction using a mixed qubit scheme. *Physical Review A*, 100(3):032325, 2019. doi:10.1103/PhysRevA.100.032325.

[80] Boris Mihailov Varbanov, Francesco Battistel, Brian Michael Tarasinski, Viacheslav Petrovych Ostroukh, Thomas Eugene O'Brien, Leonardo DiCarlo, and Barbara Maria Terhal. Leakage detection for a transmon-based surface code. *npj Quantum Information*, 6(1):1–13, 2020. doi:10.1038/s41534-020-00330-w.

[81] C. C. Bultink, T. E. O'Brien, R. Vollmer, N. Muthusubramanian, M. W. Beekman, M. A. Rol, X. Fu, B. Tarasinski, V. Ostroukh, B. Varbanov, et al. Protecting quantum entanglement from leakage and qubit errors via repetitive parity measurements. *Science Advances*, 6(12):eaay3050, 2020. doi:10.1126/sciadv.aay3050.

[82] M. McEwen, D. Kafri, Z. Chen, J. Atalaya, K. J. Satzinger, C. Quintana, P. V. Klimov, D. Sank, C. Gidney, A. G. Fowler, et al. Removing leakage-induced correlated errors in superconducting quantum error correction. *Nature Communications*, 12(1):1761, 2021. doi:10.1038/s41467-021-21982-y.

[83] F. Battistel, B.M. Varbanov, and B.M. Terhal. Hardware-Efficient Leakage-Reduction Scheme for Quantum Error Correction with Superconducting Transmon Qubits. *PRX Quantum*, 2(3):030314, 2021. doi:10.1103/PRXQuantum.2.030314.

[84] Scott Aaronson and Daniel Gottesman. Improved Simulation of Stabilizer Circuits. *Physical Review A*, 70(5):052328, 2004. doi:10.1103/PhysRevA.70.052328.

[85] Andrew S. Darmawan and David Poulin. Tensor-Network Simulations of the Surface Code under Realistic Noise. *Physical Review Letters*, 119(4):040502, 2017. doi:10.1103/PhysRevLett.119.040502.

[86] Sergey Bravyi, Matthias Englbrecht, Robert Koenig, and Nolan Peard. Correcting coherent errors with surface codes. *npj Quantum Information*, 4(1):55, 2018. doi:10.1038/s41534-018-0106-y.

[87] Kouichi Okunishi, Tomotoshi Nishino, and Hiroshi Ueda. Developments in the Tensor Network – from Statistical Mechanics to Quantum Entanglement. *Journal of the Physical Society of Japan*, 91(6):062001, 2022. doi:10.7566/JPSJ.91.062001.

[88] D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix product state representations. *Quantum Info. Comput.*, 7(5):401–430, 2007. doi:10.26421/QIC7.5-6-1.

[89] D. Ristè, S. Poletto, M.-Z. Huang, A. Bruno, V. Vesterinen, O.-P. Saira, and L. DiCarlo. Detecting bit-flip errors in a logical qubit using stabilizer measurements. *Nature Communications*, 6(1):6983, 2015. doi:10.1038/ncomms7983.

[90] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Yu Chen, et al. State preservation by repetitive error detection in a superconducting quantum circuit. *Nature*, 519(7541):66–69, 2015. doi:10.1038/nature14270.

[91] Yasunari Suzuki, Keisuke Fujii, and Masato Koashi. Efficient Simulation of Quantum Error Correction Under Coherent Error Based on the Nonunitary Free-Fermionic Formalism. *Physical Review Letters*, 119(19):190503, 2017. doi:10.1103/PhysRevLett.119.190503.

[92] Zijun Chen, Kevin J. Satzinger, Juan Atalaya, Alexander N. Korotkov, Andrew Dunsworth, Daniel Sank, Chris Quintana, Matt McEwen, Rami Barends, Paul V. Klimov, et al. Exponential suppression of bit or phase flip errors with repetitive error correction. *Nature*, 595(7867):383–387, 2021. doi:10.1038/s41586-021-03588-y.

[93] Yasunari Suzuki, Yoshiaki Kawase, Yuya Masumura, Yuria Hiraga, Masahiro Nakadai, Jiabao Chen, Ken M. Nakanishi, Kosuke Mitarai, Ryosuke Imai, Shiro Tamiya, et al. Qulacs: A fast and versatile quantum circuit simulator for research purpose. *Quantum*, 5:559, 2021. doi:10.22331/q-2021-10-06-559.

[94] Gian Giacomo Guerreschi, Justin Hogaboam, Fabio Baruffa, and Nicolas P. D. Sawaya. Intel Quantum Simulator: A cloud-ready high-performance simulator of quantum circuits. *Quantum Science and Technology*, 5(3):034007, 2020. doi:10.1088/2058-9565/ab8505.

[95] Iris Cong, Harry Levine, Alexander Keesling, Dolev Bluvstein, Sheng-Tao Wang, and Mikhail D. Lukin. Hardware-Efficient, Fault-Tolerant Quantum Computation with Rydberg Atoms. *Physical Review X*, 12(2):021049, 2022. doi:10.1103/PhysRevX.12.021049.

[96] D. Hayes, D. Stack, B. Bjork, A. C. Potter, C. H. Baldwin, and R. P. Stutz. Eliminating Leakage Errors in Hyperfine Qubits. *Physical Review Letters*, 124(17):170501, 2020. doi:10.1103/PhysRevLett.124.170501.

[97] Daniel Sank, Zijun Chen, Mostafa Khezri, J. Kelly, R. Barends, B. Campbell, Y. Chen, B. Chiaro, A. Dunsworth, A. Fowler, et al. Measurement-Induced State Transitions in a Superconducting Qubit: Beyond the Rotating Wave Approximation. *Physical Review Letters*, 117(19):190503, 2016. doi:10.1103/PhysRevLett.117.190503.

[98] Ross Shillito, Alexandru Petrescu, Joachim Cohen, Jackson Beall, Markus Hauru, Martin Ganahl, Adam G.M. Lewis, Guifre Vidal, and Alexandre Blais. Dynamics of Transmon Ionization. *Physical Review Applied*, 18(3):034031, 2022. doi:10.1103/PhysRevApplied.18.034031.

[99] Mostafa Khezri, Alex Opremcak, Zijun Chen, Kevin C. Miao, Matt McEwen, Andreas Bengtsson, Theodore White, Ofer Naaman, Daniel Sank, Alexander N. Korotkov, et al. Measurement-induced state transitions in a superconducting qubit: Within the rotating-wave approximation. *Physical Review Applied*, 20(5):054008, 2023. doi:10.1103/PhysRevApplied.20.054008.

[100] K. De Raedt, K. Michielsen, H. De Raedt, B. Trieu, G. Arnold, M. Richter, Th. Lippert, H. Watanabe, and N. Ito. Massively parallel quantum computer simulator. *Computer Physics Communications*, 176(2):121–136, 2007. doi:10.1016/j.cpc.2006.08.007.

[101] Guifré Vidal. Efficient Classical Simulation of Slightly Entangled Quantum Computations. *Physical Review Letters*, 91(14):147902, 2003. doi:10.1103/PhysRevLett.91.147902.

[102] Chu Guo, Yong Liu, Min Xiong, Shichuan Xue, Xiang Fu, Anqi Huang, Xiaogang Qiang, Ping Xu, Junhua Liu, Shenggen Zheng, et al. General-Purpose Quantum Circuit Simulator with Projected Entangled-Pair States and the Quantum Supremacy Frontier. *Physical Review Letters*, 123(19):190501, 2019. doi:10.1103/PhysRevLett.123.190501.

[103] Yiqing Zhou, E. Miles Stoudenmire, and Xavier Waintal. What Limits the Simulation of Quantum Computers? *Physical Review X*, 10(4):041038, 2020. doi:10.1103/PhysRevX.10.041038.

[104] Changhun Oh, Minzhao Liu, Yuri Alexeev, Bill Fefferman, and Liang Jiang. Classical algorithm for simulating experimental Gaussian boson sampling. *Nature Physics*, 20(9):1461–1468, 2024. doi:10.1038/s41567-024-02535-8.

[105] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout Van Den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, et al. Evidence for the utility of quantum computing before fault tolerance supplement. *Nature*, 618(7965):500–505, 2023. doi:10.1038/s41586-023-06096-3.

[106] G. Vidal. Entanglement Renormalization. *Physical Review Letters*, 99(22):220405, 2007. doi:10.1103/PhysRevLett.99.220405.

[107] Andrew S. Darmawan and David Poulin. Linear-time general decoding algorithm for the surface code. *Physical Review E*, 97(5):051302, 2018. doi:10.1103/PhysRevE.97.051302.

[108] Kyungjoo Noh, Liang Jiang, and Bill Fefferman. Efficient classical simulation of noisy random quantum circuits in one dimension. *Quantum*, 4:318, 2020. doi:10.22331/q-2020-09-11-318.

[109] Juan Osorio Iregui, Matthias Troyer, and Philippe Corboz. Infinite Matrix Product States vs Infinite Projected Entangled-Pair States on the Cylinder: A comparative study. *Physical Review B*, 96(11):115113, 2017. doi:10.1103/PhysRevB.96.115113.

[110] Ulrich Schollwoeck. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics*, 326(1):96–192, 2011. doi:10.1016/j.aop.2010.09.012.

[111] David K. Tuckett, Andrew S. Darmawan, Christopher T. Chubb, Sergey Bravyi, Stephen D. Bartlett, and Steven T. Flammia. Tailoring surface codes for highly biased noise. *Physical Review X*, 9(4):041031, 2019. doi:10.1103/PhysRevX.9.041031.

[112] Jérémie Guillaud and Mazyar Mirrahimi. Repetition Cat Qubits for Fault-Tolerant Quantum Computation. *Physical Review X*, 9(4):041053, 2019. doi:10.1103/PhysRevX.9.041053.

[113] Andrew S. Darmawan, Benjamin J. Brown, Arne L. Grimsmo, David K. Tuckett, and Shruti Puri. Practical quantum error correction with the XZZX code and Kerr-cat qubits. *PRX Quantum*, 2(3):030345, 2021. doi:10.1103/PRXQuantum.2.030345.

[114] J. Pablo Bonilla Ataides, David K. Tuckett, Stephen D. Bartlett, Steven T. Flammia, and Benjamin J. Brown. The XZZX Surface Code. *Nature Communications*, 12(1):2172, 2021. doi:10.1038/s41467-021-22274-1.

[115] Oscar Higgott, Thomas C. Bohdanowicz, Aleksander Kubica, Steven T. Flammia, and Earl T. Campbell. Improved Decoding of Circuit Noise and Fragile Boundaries of Tailored Surface Codes. *Physical Review X*, 13(3):031007, 2023. doi:10.1103/PhysRevX.13.031007.

[116] Michael A. Nielsen and Isaac L. Chuang. Quantum Computation and Quantum Information: 10th Anniversary Edition, 2010. doi:10.1017/CBO9780511976667.

[117] Keisuke Fujii. *Quantum Computation with Topological Codes: From Qubit to Topological Fault-Tolerance*, volume 8 of *SpringerBriefs in Mathematical Physics*. Springer Singapore, Singapore, 2015. doi:10.1007/978-981-287-996-7.

[118] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. Topological quantum memory. *Journal of Mathematical Physics*, 43(9):4452–4505, 2002. doi:10.1063/1.1499754.

[119] Brian Skinner, Jonathan Ruhman, and Adam Nahum. Measurement-Induced Phase Transitions in the Dynamics of Entanglement. *Physical Review X*, 9(3):031009, 2019. doi:10.1103/PhysRevX.9.031009.

[120] Yimu Bao, Soonwon Choi, and Ehud Altman. Theory of the phase transition in random unitary circuits with measurements. *Physical Review B*, 101(10):104301, 2020. doi:10.1103/PhysRevB.101.104301.

[121] Chase Roberts, Ashley Milsted, Martin Ganahl, Adam Zalcman, Bruce Fontaine, Yijian Zou, Jack Hidary, Guifre Vidal, and Stefan Leichenauer. TensorNetwork: A Library for Physics and Machine Learning, 2019. arXiv:1905.01330.

[122] Christopher J. Wood and Jay M. Gambetta. Quantification and Characterization of Leakage Errors. *Physical Review A*, 97(3):032306, 2018. doi:10.1103/PhysRevA.97.032306.

[123] Bujiao Wu, Xiaoyang Wang, Xiao Yuan, Cupjin Huang, and Jianxin Chen. Leakage Benchmarking for Universal Gate Sets. *Entropy*, 26(1):71, 2024. doi:10.3390/e26010071.

[124] Yu Tomita and Krysta M. Svore. Low-distance surface codes under realistic quantum noise. *Physical Review A*, 90(6):062320, 2014. doi:10.1103/PhysRevA.90.062320.

[125] Amara Katabarwa and Michael R. Geller. Logical error rate in the Pauli twirling approximation. *Scientific Reports*, 5(1):14670, 2015. doi:10.1038/srep14670.

[126] Amara Katabarwa. A dynamical interpretation of the Pauli Twirling Approximation and Quantum Error Correction, 2017. arXiv:1701.03708.

[127] Mario Collura, Guglielmo Lami, Nishan Ranabhat, and Alessandro Santini. *Tensor Network Techniques for Quantum Computation*. SISSA Medialab S.r.l., 2024. doi:10.22323/9788898587049.

[128] Aleksandr Berezutskii, Minzhao Liu, Atithi Acharya, Roman Ellerbrock, Johnnie Gray, Reza Haghshenas, Zichang He, Abid Khan, Viacheslav Kuzmin, Dmitry Lyakh, et al. Tensor networks for quantum computing. *Nature Reviews Physics*, pages 1–13, 2025. doi:10.1038/s42254-025-00853-1.

[129] Zhao-Yu Han, Jun Wang, Heng Fan, Lei Wang, and Pan Zhang. Unsupervised Generative Modeling Using Matrix Product States. *Physical Review X*, 8(3):031012, 2018. doi:10.1103/PhysRevX.8.031012.

[130] James Dborin, Fergus Barratt, Vinul Wimalaweera, Lewis Wright, and Andrew G Green. Matrix product state pre-training for quantum machine learning. *Quantum Science and Technology*, 7(3):035014, 2022. doi:10.1088/2058-9565/ac7073.

[131] Glen Evenbly. Gauge fixing, canonical forms, and optimal truncations in tensor networks with closed loops. *Physical Review B*, 98(8):085155, 2018. doi:10.1103/PhysRevB.98.085155.

[132] Michael P. Zaletel and Frank Pollmann. Isometric Tensor Network States in Two Dimensions. *Physical Review Letters*, 124(3):037201, 2020. doi:10.1103/PhysRevLett.124.037201.

[133] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum Algorithm for Linear Systems of Equations. *Physical Review Letters*, 103(15):150502, 2009. doi:10.1103/PhysRevLett.103.150502.

[134] Ashley Montanaro. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, 2015. doi:10.1098/rspa.2015.0301.

[135] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017. doi:10.1038/nature23474.

[136] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum Circuit Learning. *Physical Review A*, 98(3):032309, 2018. doi:10.1103/PhysRevA.98.032309.

[137] Mikko Möttönen, Juha J. Vartiainen, Ville Bergholm, and Martti M. Salomaa. Transformation of quantum states using uniformly controlled rotations. *Quantum Info. Comput.*, 5(6):467–473, 2005.

[138] Martin Plesch and Časlav Brukner. Quantum-state preparation with universal gate decompositions. *Physical Review A*, 83(3):032302, 2011. doi:10.1103/PhysRevA.83.032302.

[139] Alexei Kitaev and William A. Webb. Wavefunction preparation and resampling using a quantum computer, 2009. arXiv:0801.0342.

[140] Almudena Carrera Vazquez and Stefan Woerner. Efficient State Preparation for Quantum Amplitude Estimation. *Physical Review Applied*, 15(3):034027, 2021. doi:10.1103/PhysRevApplied.15.034027.

[141] Christian W. Bauer, Plato Deliyannis, Marat Freytsis, and Benjamin Nachman. Practical considerations for the preparation of multivariate Gaussian states on quantum computers, 2021. arXiv:2109.10918.

[142] Arthur G. Rattew, Yue Sun, Pierre Minssen, and Marco Pistoia. The Efficient Preparation of Normal Distributions in Quantum Registers. *Quantum*, 5:609, 2021. doi:10.22331/q-2021-12-23-609.

[143] Arthur G. Rattew and Bálint Koczor. Preparing Arbitrary Continuous Functions in Quantum Registers With Logarithmic Complexity, 2022. arXiv:2205.00519.

[144] Kouhei Nakaji, Shumpei Uno, Yohichi Suzuki, Rudy Raymond, Tamiya Onodera, Tomoki Tanaka, Hiroyuki Tezuka, Naoki Mitsuda, and Naoki Yamamoto. Approximate amplitude encoding in shallow parameterized quantum circuits and its application to financial market indicators. *Physical Review Research*, 4(2):023136, 2022. doi:10.1103/PhysRevResearch.4.023136.

[145] Elton Yechao Zhu, Sonika Johri, Dave Bacon, Mert Esencan, Jungsang Kim, Mark Muir, Nikhil Murgai, Jason Nguyen, Neal Pisenti, Adam Schouela, et al. Generative quantum learning of joint probability distribution functions. *Physical Review Research*, 4(4):043092, 2022. doi:10.1103/PhysRevResearch.4.043092.

[146] Gabriel Marin-Sanchez, Javier Gonzalez-Conde, and Mikel Sanz. Quantum algorithms for approximate function loading. *Physical Review Research*, 5(3):033114, 2023. doi:10.1103/PhysRevResearch.5.033114.

[147] Mudassir Moosa, Thomas W. Watts, Yiyou Chen, Abhijat Sarma, and Peter L. McMahon. Linear-depth quantum circuits for loading Fourier approximations of arbitrary functions. *Quantum Science and Technology*, 9(1):015002, 2023. doi:10.1088/2058-9565/acfc62.

[148] Julien Zylberman and Fabrice Debbasch. Efficient Quantum State Preparation with Walsh Series, 2023. arXiv:2307.08384.

[149] Stephen P. Jordan, Keith S. M. Lee, and John Preskill. Quantum Algorithms for Quantum Field Theories. *Science*, 336(6085):1130–1133, 2012. doi:10.1126/science.1217069.

[150] Anirban Narayan Chowdhury and Rolando D. Somma. Quantum algorithms for Gibbs sampling and hitting-time estimation. *Quantum Info. Comput.*, 17(1-2):41–64, 2017. doi:10.26421/QIC17.1-2-3.

[151] Stephen P. Jordan, Keith S. M. Lee, and John Preskill. Quantum Computation of Scattering in Scalar Quantum Field Theories, 2019. arXiv:1112.4833.

[152] Daniel J. Egger, Ricardo García Gutiérrez, Jordi Cahué Mestre, and Stefan Woerner. Credit Risk Analysis Using Quantum Computers. *IEEE Transactions on Computers*, 70(12):2136–2145, 2021. doi:10.1109/TC.2020.3038063.

[153] Sascha Wilkens and Joe Moorhouse. Quantum computing for financial risk measurement. *Quantum Information Processing*, 22(1):51, 2023. doi:10.1007/s11128-022-03777-2.

[154] Xiangyu Li, Xiaolong Yin, Nathan Wiebe, Jaehun Chun, Gregory K. Schenter, Margaret S. Cheung, and Johannes Mülmenstädt. Potential quantum advantage for simulation of fluid dynamics. *Physical Review Research*, 7(1):013036, 2025. doi:10.1103/PhysRevResearch.7.013036.

[155] Vladimir Skavysh, Sofia Priazhkina, Diego Guala, and Thomas R. Bromley. Quantum monte carlo for economics: Stress testing and macroeconomic deep learning. *Journal of Economic Dynamics and Control*, 153:104680, 2023. doi:10.1016/j.jedc.2023.104680.

[156] Kohei Morimoto, Yusuke Takase, Kosuke Mitarai, and Keisuke Fujii. Continuous optimization by quantum adaptive distribution search. *Physical Review Research*, 6(2):023191, 2024. doi:10.1103/PhysRevResearch.6.023191.

[157] Shi-Ju Ran. Encoding of Matrix Product States into Quantum Circuits of One- and Two-Qubit Gates. *Physical Review A*, 101(3):032310, 2020. doi:10.1103/PhysRevA.101.032310.

[158] Adam Holmes and A. Y. Matsuura. Efficient Quantum Circuits for Accurate State Preparation of Smooth, Differentiable Functions. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 169–179. IEEE Computer Society, 2020. doi:10.1109/QCE49297.2020.00030.

[159] Paula García-Molina, Javier Rodríguez-Mediavilla, and Juan José García-Ripoll. Quantum Fourier analysis for multivariate functions and applications to a class of Schr\"odinger-type partial differential equations. *Physical Review A*, 105(1):012433, 2022. doi:10.1103/PhysRevA.105.012433.

[160] Peng-Fei Zhou, Rui Hong, and Shi-Ju Ran. Automatically differentiable quantum circuit for many-qubit state preparation. *Physical Review A*, 104(4):042601, 2021. doi:10.1103/PhysRevA.104.042601.

[161] Prithvi Gundlapalli and Junyi Lee. Deterministic and Entanglement-Efficient Preparation of Amplitude-Encoded Quantum Registers. *Physical Review Applied*, 18(2):024013, 2022. doi:10.1103/PhysRevApplied.18.024013.

[162] Matan Ben-Dov, David Shnaiderov, Adi Makmal, and Emanuele G. Dalla Torre. Approximate encoding of quantum states using shallow circuits. *npj Quantum Information*, 10(1):1–8, 2024. doi:10.1038/s41534-024-00858-1.

[163] Jason Iaconis, Sonika Johri, and Elton Yechao Zhu. Quantum state preparation of normal distributions using matrix product states. *npj Quantum Information*, 10(1):1–11, 2024. doi:10.1038/s41534-024-00805-0.

[164] Ar A. Melnikov, A. A. Termanova, S. V. Dolgov, F. Neukart, and M. R. Perelshtein. Quantum state preparation using tensor networks. *Quantum Science and Technology*, 8(3):035027, 2023. doi:10.1088/2058-9565/acd9e7.

[165] Javier Gonzalez-Conde, Thomas W. Watts, Pablo Rodriguez-Grasa, and Mikel Sanz. Efficient quantum amplitude encoding of polynomial functions. *Quantum*, 8:1297, 2024. doi:10.22331/q-2024-03-21-1297.

[166] Manuel S. Rudolph, Jacob Miller, Danial Motlagh, Jing Chen, Atithi Acharya, and Alejandro Perdomo-Ortiz. Synergy Between Quantum Circuits and Tensor Networks: Short-cutting the Race to Practical Quantum Advantage, 2023. arXiv:2208.13673.

[167] Jason Iaconis and Sonika Johri. Tensor Network Based Efficient Quantum Data Loading of Images, 2023. arXiv:2310.05897.

[168] Bernhard Jobst, Kevin Shen, Carlos A. Riofrío, Elvira Shishenina, and Frank Pollmann. Efficient MPS representations and quantum circuits from the Fourier modes of classical image data. *Quantum*, 8:1544, 2024. doi:10.22331/q-2024-12-03-1544.

[169] Raghav Jumade and Nicolas PD Sawaya. Data is often loadable in short depth: Quantum circuits from tensor networks for finance, images, fluids, and proteins, 2023. arXiv:2309.13108.

[170] Yuichi Sano and Ikko Hamamura. Quantum State Preparation for Probability Distributions with Mirror Symmetry Using Matrix Product States, 2024. arXiv:2403.16729.

[171] Y.-Y. Shi, L.-M. Duan, and G. Vidal. Classical simulation of quantum many-body systems with a tree tensor network. *Physical Review A*, 74(2):022320, 2006. doi:10.1103/PhysRevA.74.022320.

[172] Ivan Oseledets and Eugene Tyrtyshnikov. TT-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010. doi:10.1016/j.laa.2009.07.024.

[173] Dmitry V. Savostyanov. Quasioptimality of maximum-volume cross interpolation of tensors. *Linear Algebra and its Applications*, 458:217–244, 2014. doi:10.1016/j.laa.2014.06.006.

[174] Toshiya Hikihara, Hiroshi Ueda, Kouichi Okunishi, Kenji Harada, and Tomotoshi Nishino. Automatic structural optimization of tree tensor networks. *Physical Review Research*, 5(1):013031, 2023. doi:10.1103/PhysRevResearch.5.013031.

[175] I. V. Oseledets. Constructive Representation of Functions in Low-Rank Tensor Formats. *Constructive Approximation*, 37(1):1–18, 2013. doi:10.1007/s00365-012-9175-x.

[176] Daniele Bigoni, Allan P. Engsig-Karup, and Youssef M. Marzouk. Spectral tensor-train decomposition. *SIAM Journal on Scientific Computing*, 38(4):A2405–A2439, 2016. doi:10.1137/15M1036919.

[177] Alex A. Gorodetsky, Sertac Karaman, and Youssef M. Marzouk. A continuous analogue of the tensor-train decomposition, 2018. arXiv:1510.09088.

[178] Paul B. Rohrbach, Sergey Dolgov, Lars Grasedyck, and Robert Scheichl. Rank Bounds for Approximating Gaussian Densities in the Tensor-Train Format, 2020. arXiv:2001.08187.

[179] Raban Iten, Roger Colbeck, Ivan Kukuljan, Jonathan Home, and Matthias Christandl. Quantum Circuits for Isometries. *Physical Review A*, 93(3):032318, 2016. doi:10.1103/PhysRevA.93.032318.

[180] Yu I. Bogdanov, N. A. Bogdanova, D. V. Fastovets, and V. F. Luckichev. Schmidt decomposition and multivariate statistical analysis. In *International Conference on Micro- and Nano-Electronics 2016*, volume 10224, pages 706–715. SPIE, 2016. doi:10.1117/12.2266891.

[181] A. Yu. Bogdanov, Yu. I. Bogdanov, and K. A. Valiev. Schmidt information and entanglement of quantum systems. *Moscow University Computational Mathematics and Cybernetics*, 31(1):33–42, 2007. doi:10.3103/S0278641907010074.

[182] W. D. Ray. The Advanced Theory of Statistics, Vol. 3: Design and Analysis and Time Series. *Royal Statistical Society. Journal. Series A: General*, 147(3):523, 1984. doi:10.2307/2981604.

[183] Johnnie Gray. Quimb: A python package for quantum information and many-body calculations. *Journal of Open Source Software*, 3(29):819, 2018. doi:10.21105/joss.00819.

[184] Queenie Sun, Nicholas Grablevsky, Huaizhang Deng, and Pooya Azadi. Quantum Computing for Multi Period Asset Allocation, 2024. arXiv:2410.11997.

[185] Nikitas Stamatopoulos and William J. Zeng. Derivative Pricing using Quantum Signal Processing. *Quantum*, 8:1322, 2024. doi:10.22331/q-2024-04-30-1322.

[186] Stefan Woerner and Daniel J. Egger. Quantum risk analysis. *npj Quantum Information*, 5(1):1–8, 2019. doi:10.1038/s41534-019-0130-6.

[187] Koichi Miyamoto. Quantum algorithm for calculating risk contributions in a credit portfolio. *EPJ Quantum Technology*, 9(1):1–16, 2022. doi:10.1140/epjqt/s40507-022-00153-y.

[188] Joseph Tindall, Miles Stoudenmire, and Ryan Levy. Compressing multivariate functions with tree tensor networks, 2024. arXiv:2410.03572.

[189] Kenji Harada, Tsuyoshi Okubo, and Naoki Kawashima. Tensor tree learns hidden relational structures in data to construct generative models. *Machine Learning: Science and Technology*, 6(2):025002, 2025. doi:10.1088/2632-2153/adc2c7.

[190] Ryo Watanabe, Hidetaka Manabe, Toshiya Hikihara, and Hiroshi Ueda. TTNOpt: Tree tensor network package for high-rank tensor compression. *Computer Physics Communications*, 319:109918, 2026. doi:10.1016/j.cpc.2025.109918.

[191] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout van den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, et al. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965):500–505, 2023. doi:10.1038/s41586-023-06096-3.

[192] Zhiyan Ding and Lin Lin. Even Shorter Quantum Circuit for Phase Estimation on Early Fault-Tolerant Quantum Computers with Applications to Ground-State Energy Estimation. *PRX Quantum*, 4(2):020331, 2023. doi:10.1103/PRXQuantum.4.020331.

[193] Guoming Wang, Daniel Stilck França, Ruizhe Zhang, Shuchen Zhu, and Peter D. Johnson. Quantum algorithm for ground state energy estimation using circuit depth with exponentially improved dependence on precision. *Quantum*, 7:1167, 2023. doi:10.22331/q-2023-11-06-1167.

[194] Nobuyuki Yoshioka, Tsuyoshi Okubo, Yasunari Suzuki, Yuki Koizumi, and Wataru Mizukami. Hunting for quantum-classical crossover in condensed matter problems. *npj Quantum Information*, 10(1):45, 2024. doi:10.1038/s41534-024-00839-4.

[195] Seunghoon Lee, Joonho Lee, Huanchen Zhai, Yu Tong, Alexander M. Dalzell, Ashutosh Kumar, Phillip Helms, Johnnie Gray, Zhi-Hao Cui, Wenyuan Liu, et al. Evaluating the evidence for exponential quantum advantage in ground-state quantum chemistry. *Nature Communications*, 14(1):1952, 2023. doi:10.1038/s41467-023-37587-6.

[196] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019. doi:10.1145/3313276.3316366.

[197] Lin Lin and Yu Tong. Near-optimal ground state preparation. *Quantum*, 4:372, 2020. doi:10.22331/q-2020-12-14-372.

[198] Yulong Dong, Lin Lin, and Yu Tong. Ground-State Preparation and Energy Estimation on Early Fault-Tolerant Quantum Computers via Quantum Eigenvalue Transformation of Unitary Matrices. *PRX Quantum*, 3(4):040305, 2022. doi:10.1103/PRXQuantum.3.040305.

[199] Hongkang Ni, Haoya Li, and Lexing Ying. On low-depth algorithms for quantum phase estimation. *Quantum*, 7:1165, 2023. doi:10.22331/q-2023-11-06-1165.

[200] Oriel Kiss, Utkarsh Azad, Borja Requena, Alessandro Roggero, David Wakeham, and Juan Miguel Arrazola. Early Fault-Tolerant Quantum Algorithms in Practice: Application to Ground-State Energy Estimation. *Quantum*, 9:1682, 2025. doi:10.22331/q-2025-04-01-1682.

[201] Sevag Gharibian and François Le Gall. Dequantizing the Quantum Singular Value Transformation: Hardness and Applications to Quantum Chemistry and the Quantum PCP Conjecture. *SIAM Journal on Computing*, 52(4):1009–1038, 2023. doi:10.1137/22M1513721.

[202] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 217–228, 2019. doi:10.1145/3313276.3316310.

[203] Ewin Tang. Quantum principal component analysis only achieves an exponential speedup because of its state preparation assumptions. *Physical Review Letters*, 127(6):060503, 2021. doi:10.1103/PhysRevLett.127.060503.

[204] Ewin Tang. Dequantizing algorithms to understand quantum advantage in machine learning. *Nature Reviews Physics*, 4(11):692–693, 2022. doi:10.1038/s42254-022-00511-w.

[205] François Le Gall. Classical Algorithms for Constant Approximation of the Ground State Energy of Local Hamiltonians, 2024. arXiv:2410.21833.

[206] François Le Gall. Robust Dequantization of the Quantum Singular Value Transformation and Quantum Machine Learning Algorithms. *computational complexity*, 34(1):2, 2025. doi:10.1007/s00037-024-00262-3.

[207] Nai-Hui Chia, András Gilyén, Tongyang Li, Han-Hsuan Lin, Ewin Tang, and Chunhao Wang. Sampling-based sublinear low-rank matrix arithmetic framework for dequantizing quantum machine learning. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 387–400, 2020. doi:10.1145/3357713.3384314.

[208] Jad C. Halimeh, Fabian Kolley, and Ian P. McCulloch. Chebyshev matrix product state approach for time evolution. *Physical Review B*, 92(11):115130, 2015. doi:10.1103/PhysRevB.92.115130.

[209] T. Nishino and K. Okunishi. Corner Transfer Matrix Renormalization Group Method. *Journal of the Physical Society of Japan*, 65(4):891–894, 1996. doi:10.1143/JPSJ.65.891.

[210] Andrzej Cichocki, Namgil Lee, Ivan Oseledets, Anh-Huy Phan, Qibin Zhao, and Danilo P. Mandic. Tensor Networks for Dimensionality Reduction and Large-scale Optimization: Part 1 Low-Rank Tensor Decompositions. *Foundations and Trends® in Machine Learning*, 9(4-5):249–429, 2016. doi:10.1561/2200000059.

[211] Steven Diamond and Stephen Boyd. CVXPY: A python-embedded modeling language for convex optimization. *J. Mach. Learn. Res.*, 17(1):2909–2913, 2016.

[212] Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Uniform Spectral Amplification, 2017. arXiv:1707.05391.

[213] Kazuki Sakamoto and Keisuke Fujii. On the quantum computational complexity of classical linear dynamics with geometrically local interactions: Dequantization and universality, 2025. arXiv:2505.10445.

[214] Yukun Zhang, Yusen Wu, and Xiao Yuan. A Dequantized Algorithm for the Guided Local Hamiltonian Problem, 2024. arXiv:2411.16163.

[215] Yusen Wu, Yukun Zhang, and Xiao Yuan. An Efficient Classical Algorithm for Simulating Short Time 2D Quantum Dynamics, 2024. arXiv:2409.04161.

[216] Dominik S. Wild and Álvaro M. Alhambra. Classical Simulation of Short-Time Quantum Dynamics. *PRX Quantum*, 4(2):020340, 2023. doi:10.1103/PRXQuantum.4.020340.

[217] Seongwook Shin, Yong Siah Teo, and Hyunseok Jeong. Dequantizing quantum machine learning models using tensor networks. *Physical Review Research*, 6(2):023218, 2024. doi:10.1103/PhysRevResearch.6.023218.

[218] Ainesh Bakshi and Ewin Tang. An Improved Classical Singular Value Transformation for Quantum Machine Learning. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, Proceedings, pages 2398–2453. Society for Industrial and Applied Mathematics, 2024. doi:10.1137/1.9781611977912.86.

[219] Alexander A. Sherstov. Making polynomials robust to noise. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, pages 747–758, New York New York USA, 2012. ACM. doi:10.1145/2213977.2214044.

[220] Alexander Weisse, Gerhard Wellein, Andreas Alvermann, and Holger Fehske. The Kernel Polynomial Method. *Reviews of Modern Physics*, 78(1):275–306, 2006. doi:10.1103/RevModPhys.78.275.

[221] Santiago Cifuentes, Samson Wang, Thais L. Silva, Mario Berta, and Leandro Aolita. Quantum computational complexity of matrix functions, 2024. arXiv:2410.13937.

[222] Ashley Montanaro and Changpeng Shao. Quantum and Classical Query Complexities of Functions of Matrices. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 573–584, Vancouver BC Canada, 2024. ACM. doi:10.1145/3618260.3649665.

[223] F. Alexander Wolf, Ian P. McCulloch, Olivier Parcollet, and Ulrich Schollwöck. Chebyshev matrix product state impurity solver for dynamical mean-field theory. *Physical Review B*, 90(11):115124, 2014. doi:10.1103/PhysRevB.90.115124.

[224] Martin Ganahl, Patrik Thunström, Frank Verstraete, Karsten Held, and Hans Gerd Evertz. Chebyshev expansion for Impurity Models using Matrix Product States. *Physical Review B*, 90(4):045144, 2014. doi:10.1103/PhysRevB.90.045144.

[225] Christopher F. Kane, Niladri Gomes, and Michael Kreshchuk. Nearly optimal state preparation for quantum simulations of lattice gauge theories. *Physical Review A*, 110(1):012455, 2024. doi:10.1103/PhysRevA.110.012455.

[226] Johannes Hauschild and Frank Pollmann. Efficient numerical simulations with Tensor Networks: Tensor Network Python (TeNPy). *SciPost Physics Lecture Notes*, page 5, 2018. doi:10.21468/SciPostPhysLectNotes.5.