



UNIVERSITÀ
DEGLI STUDI
FIRENZE

FLORE

Repository istituzionale dell'Università degli Studi di Firenze

Simulation, benchmarking and control of quantum many-qubit systems

Questa è la Versione finale referata (Post print/Accepted manuscript) della seguente pubblicazione:

Original Citation:

Simulation, benchmarking and control of quantum many-qubit systems / Mehran Khosrojerdi. - (2026).

Availability:

The webpage <https://hdl.handle.net/2158/1446116> of the repository was last updated on 2026-01-09T11:40:17Z

Terms of use:

Open Access

La pubblicazione è resa disponibile sotto le norme e i termini della licenza di deposito, secondo quanto stabilito dalla Policy per l'accesso aperto dell'Università degli Studi di Firenze (<https://www.sba.unifi.it/upload/policy-oa-2016-1.pdf>)

Publisher copyright claim:

La data sopra indicata si riferisce all'ultimo aggiornamento della scheda del Repository FloRe - The above-mentioned date refers to the last update of the record in the Institutional Repository FloRe

(Article begins on next page)



Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



UNIVERSITÀ
DEGLI STUDI
FIRENZE

DOTTORATO DI RICERCA IN FISICA E ASTRONOMIA

CICLO XXXVIII – PNRR NextGeneration EU

Simulation, benchmarking and control of quantum many-qubit systems

Settore Scientifico Disciplinare

PHYS-04/A

Dottorando

Dott. Mehran Khosrojerdi

Supervisore

Prof. Leonardo Banchi

Coordinatore

Prof. Giovanni Modugno

*Dedicated to
My lovely family*

Abstract

Understanding and controlling complex quantum systems is a central challenge in modern physics and quantum technologies. This thesis presents a unified framework that combines quantum machine learning, tensor network methods, and many-body quantum theory to simulate, benchmark and control one-dimensional quantum systems at zero temperature. By integrating these approaches, it is possible to efficiently represent quantum states, capture entanglement structures, and identify quantum phases from limited measurement data.

This thesis reviews existing literature on quantum machine learning, tensor networks, and many-body quantum systems, highlighting foundational methods, computational techniques, and recent advances in phase characterization. Building on this theoretical and methodological background, the thesis presents original projects that apply these tools to practical problems. These projects include supervised and unsupervised classification of quantum phases, recognition of phases from reduced subsystems, and applications to supervised quantum image processing, demonstrating that learning algorithms informed by tensor network representations can accurately map phase diagrams and detect subtle correlations within quantum systems.

Overall, this work establishes a practical and theoretical toolkit that bridges machine learning, tensor networks, and many-body quantum physics, providing insights into quantum phase characterization and offering scalable strategies for analyzing complex quantum systems.

Contents

1	Introduction	1
2	Quantum Machine Learning	5
2.1	Introduction	5
2.2	Preliminaries	7
2.3	Quantum Models Applied to Classical Datasets	9
2.3.1	Concept and Framework	9
2.3.2	Quantum Data Encoding Strategies	10
2.3.3	Quantum Models	12
2.3.4	Review of Related Works	17
2.4	Classical Models Trained on Quantum-Generated Data	20
2.4.1	Concept and Framework	20
2.4.2	Feature Map	21
2.4.3	Kernel-Based Classical Model	23
2.4.4	Review of Related Works	31
2.5	Fully Quantum Learning Frameworks	33
2.5.1	Concept and Framework	33
2.5.2	Review of Related Works	35
2.6	Optimization and Challenges	37
2.6.1	Hybrid Quantum-Classical Optimization in Variational Quantum Models	37
2.6.2	Structural Optimization Guided by Information-Theoretic Principles	38
2.6.3	Optimization in Quantum Kernel Methods	38
2.6.4	Challenges in Quantum Machine Learning Optimization	39
2.6.5	Optimization in Our Project	39
2.7	Summary and Outlook	39
3	Tensor Networks	41
3.1	Introduction	41
3.2	Preliminaries and General Principles	42
3.3	Tensor Network Algorithms	54
3.3.1	Density Matrix Renormalization Group	55
3.3.2	Time Evolving Block Decimation	57
3.3.3	Tensor Train Recursive Sketching	59
3.3.4	Tensor Cross Interpolation	62

3.4	Applications of Tensor Networks	66
3.4.1	Continuum Function	66
3.4.2	Quantum Fourier transform	69
3.4.3	Interpolation and Extrapolation	71
3.4.4	Supervised Learning	71
3.4.5	Gradient-Free Optimization in the Tensor Train Format	72
3.5	Summary and Outlook	74
4	Phase Diagram of Many-Body Quantum Models	75
4.1	Introduction	75
4.2	Simulation Methods Overview	77
4.2.1	Order Parameters and Phase Indicators	77
4.2.2	Exact Diagonalization	79
4.2.3	Quantum Monte Carlo	81
4.2.4	Tensor Networks	83
4.3	Axial Next-Nearest-Neighbor Ising model	83
4.4	Cluster-Ising model	88
4.5	Haldane Anisotropic Spin Integer model	94
4.6	Summary and Outlook	98
5	Learning to Classify Quantum Phases of Matter with a Few Measurements	99
5.1	Introduction	99
5.2	Problem description	100
5.3	Handling symmetries	102
5.4	Results	103
5.4.1	Axial Next-Nearest-Neighbor Ising model	103
5.4.2	Cluster-Ising Model	106
5.5	Conclusions	107
6	Unsupervised Learning to Recognize Quantum Phases of Matter	109
6.1	Introduction	109
6.2	From classical data to quantum features	112
6.3	Clustering	114
6.4	Results	116
6.4.1	Axial Next-Nearest-Neighbor Ising Model	116
6.4.2	Cluster-Ising Model with a symmetry-protected topological phase	119
6.5	Conclusion and Discussion	120
7	Quantum Phase Recognition from Reduced Subsystems	123
7.1	Introduction	123
7.2	Methodology	125
7.3	Results	129
7.3.1	Cluster-Ising Model	129
7.3.2	Haldane Anisotropic Spin 1	133
7.4	Conclusion and Discussion	137

8 Analysis of Quantum Image Representations for Supervised Classification	139
8.1 Introduction	139
8.2 Quantum Image Processing	141
8.2.1 Tensor Network Representation	141
8.2.2 Flexible Representation of Quantum Images	143
8.2.3 Novel Enhanced Quantum Representation	144
8.2.4 Quantum Probability Image Encoding	145
8.3 Classical and Quantum Kernel Methods	146
8.3.1 Compression-accuracy trade-offs	147
8.4 Results	148
8.5 Discussion	152
Bibliography	155

Chapter 1

Introduction

Quantum technologies are rapidly transforming our ability to study and control complex systems. Among these, Quantum Machine Learning and tensor networks are key tools for understanding quantum matter, simulating many-body systems, and extracting useful information. This thesis combines these frameworks to study the learning and characterization of quantum matter, providing both theoretical insights and practical methods.

Understanding quantum phase transitions and accurately characterizing the phase diagrams of quantum matter are central challenges in both fundamental physics and emerging quantum technologies. Quantum phases can exhibit exotic properties, such as topological order and long-range entanglement, which are of fundamental importance for quantum computation, quantum materials, and condensed matter physics. A major obstacle in studying such systems is the exponential growth of the Hilbert space with system size: even relatively small many-body systems rapidly exceed classical computational capabilities, rendering exact simulations infeasible. These difficulties are particularly pronounced for topological phases, which cannot be characterized by local order parameters and often lack simple symmetry-breaking signatures. As a result, conventional analytical and numerical tools, including exact diagonalization, become impractical or insufficient beyond a few tens of degrees of freedom. This motivates the development of scalable, structure-aware methods capable of efficiently representing quantum states and reliably identifying both conventional and topological phases, which form the foundation of the approaches explored in this thesis.

From a learning perspective, identifying quantum phases, especially topological ones, poses fundamental challenges. The absence of local order parameters, the non-local nature of topological invariants, and the strong role of entanglement make it difficult for standard machine learning models to extract physically meaningful features from raw data. Classical machine learning approaches often rely on hand-crafted observables or large labeled datasets, while quantum machine learning models face limitations arising from noise, trainability, and scalability on near-term hardware. At the same time, tensor networks provide compact and physically grounded representations of quantum states, but their integration with learning frameworks remains incomplete. This thesis addresses this gap by systematically combining tensor network representations with quantum and classical learning methods to study how

entanglement structure influences learnability and phase recognition in many-body quantum systems. These challenges naturally motivate learning approaches that can exploit quantum structure directly, leading to the development of quantum machine learning models.

Quantum Machine Learning aims to leverage quantum mechanics to improve learning performance [1, 2]. Recent progress in Noisy Intermediate-Scale Quantum (NISQ) devices, together with the success of classical machine learning, has inspired studies on various quantum data encoding strategies and their potential advantages for quantum machine learning [3–13]. Quantum machine learning offers new ways to represent data, enhance expressivity, and improve generalization through quantum properties like superposition and entanglement. Research focuses on three main paradigms: quantum models trained on classical data [14–41], classical models applied to quantum-generated data [42–58], and fully quantum learning frameworks [59–70]. These approaches explore encoding strategies, model expressivity, and optimization challenges, and have been applied to problems like function approximation and quantum phase recognition, which is discussed in detail in Chapter 2.

Tensor networks provide an efficient framework to represent and manipulate high-dimensional quantum states and operators. Originally developed for quantum many-body physics [71–88], tensor networks encode complex systems as networks of local tensors, exploiting entanglement to avoid the exponential growth of Hilbert space. Key structures include matrix product states, projected entangled pair states, and the multiscale entanglement renormalization ansatz [89–97]. Tensor networks have applications in physics, quantum chemistry, and machine learning [98–112]. In this thesis, tensor networks in chapter 3 introduces the foundations of traditional and modern algorithms and their use in simulations and learning tasks [113–126].

Studying the phase diagrams of many-body quantum systems helps reveal how systems transition between different phases under varying parameters. Such knowledge is essential for understanding quantum materials, designing devices, and benchmarking quantum computers. Simulations are central in this context, as exact solutions are usually infeasible. Methods like Order Parameters, Exact Diagonalization, Quantum Monte Carlo, and Tensor Networks allow exploration of ground states, excitations, and phase boundaries [127–156]. In this thesis, we study one-dimensional systems at zero temperature, enabling accurate mapping of quantum phases and entanglement properties, with tensor networks serving as the primary tool for analyzing many-body systems.

We study three representative models: the axial next-nearest-neighbor Ising model [157–170], the Cluster–Ising model [36, 171–189], and the Haldane spin-1 anisotropic model [190–202]. These models illustrate frustration, topological order [203], and entanglement signatures in quantum phases, which is discussed in detail in chapter 4.

The structure of this thesis reflects a progression from foundational frameworks to specific applications. Chapter 2 presents quantum machine learning, reviewing key models, encoding strategies, and learning paradigms. Chapter 3 introduces tensor networks, detailing their algorithms and applications. Chapter 4 explores many-body quantum phase diagrams through analytical and numerical studies. Building on these foundations in the following chapters, I present four research projects that apply quantum machine learning techniques to the study of many-body systems. The

first three projects, in chapters 5, 6, and 7, focus on quantum phases of matter, where the goal is to achieve reliable quantum phase recognition using different machine learning approaches, including supervised and unsupervised strategies as well as reduced subsystem analysis. The final project, in chapter 8, extends these ideas to the domain of supervised quantum image processing. Each project is discussed in detail in its respective chapter, where the corresponding article can also be found.

Overall, this thesis demonstrates how physically informed representations and learning strategies can overcome fundamental obstacles in quantum phase recognition, particularly for topological and strongly entangled systems.

Chapter 2

Quantum Machine Learning

This chapter presents a comprehensive review of research in Quantum Machine Learning, examining how quantum mechanical principles can be integrated into machine learning frameworks. The discussion is organized around three main paradigms: quantum models applied to classical datasets, classical models trained on quantum-generated data, and fully quantum learning frameworks. For each paradigm, we outline the key concepts and theoretical foundations, describe representative modeling and encoding strategies, and review influential articles that have shaped the field. Additionally, we discuss optimization techniques and the challenges they pose in quantum machine learning. Altogether, this chapter provides a structured overview of existing literature that connects theoretical insights with methodological developments in quantum-enhanced learning.

2.1 Introduction

Quantum Machine Learning (QML) stands at the intersection of quantum computing and artificial intelligence, aiming to harness quantum mechanical principles to enhance learning performance. The rapid progress in NISQ technologies, combined with the success of classical machine learning, has motivated a growing interest in understanding when and how quantum models can outperform classical ones. Beyond potential computational speedups, QML offers new mechanisms for data representation, expressivity, and generalization arising from fundamental quantum properties such as superposition and entanglement.

A prominent line of research focuses on quantum models applied to classical datasets, where Parameterized Quantum Circuits (PQCs), quantum kernels, and Quantum Neural Networks (QNNs) are trained on classical data. These models provide a testbed for exploring quantum expressivity, encoding strategies, and noise resilience. Casas et al. [14] showed that PQCs can realize multidimensional Fourier expansions, demonstrating their ability to approximate highly complex classical functions. Schuld et al. [15] and Caro et al. [16] revealed that encoding choices determine the accessible frequency spectrum and generalization capacity of quantum models. Studies by Chen et al. [17] and Kübler et al. [18] analyzed overfitting and inductive bias, while Thanasilp et al. [19] and Canatar et al. [20] addressed kernel concentration effects and bandwidth tuning for improved generalization. Noise

effects on trainability were further investigated by Heyraud et al. [21] and Wang et al. [22]. In terms of architectures, Abbas et al. [23] introduced the effective dimension as a generalization measure for QNNs, with follow-up studies by Haug and Kim [24] and Caro et al. [25] highlighting the efficiency of QNNs trained on limited data. Practical demonstrations, such as quantum image processing discussed in Chapter 8, confirm that quantum kernels can achieve comparable accuracy to classical methods while reducing resource demands. More generally, Jerbi et al. [26] and Liu et al. [27] provided unifying theoretical frameworks and examples of provable quantum speedups on classical datasets.

A complementary direction studies classical models trained on quantum-generated datasets, where classical learners are applied to data originating from quantum states or processes. This hybrid framework is experimentally accessible while preserving key quantum informational features. Huang, Kueng, and Preskill [42] showed that classical learners can match quantum performance in average cases with sufficient measurement data, but face exponential limitations for complex entangled systems. In related work, Huang et al. [43] emphasized that the potential for quantum advantage depends primarily on the informational content of the data rather than the algorithm itself. Jerbi et al. [44] demonstrated that the richness of measurements dictates classical learnability of quantum dynamics, while Caro et al. [45] formalized generalization bounds for classical and quantum learners. Applications presented in Chapters 5–7 show how classical supervised and unsupervised algorithms can identify quantum phases and structures using limited measurement data, revealing that classical models can efficiently extract meaningful information from quantum datasets when correlations are adequately captured.

A third major paradigm involves fully quantum learning frameworks, where both the data and the models are quantum. In this regime, quantum measurements, parameterized unitaries, and quantum processes are trained directly on quantum states. Foundational work by Cheng et al. [59] and Heidari and Szpankowski [60] established the learnability of quantum measurements and introduced the Quantum Probably Approximately Correct framework. Magner and Padakandla [61] linked quantum learnability to geometric properties of measurements via Denoised Empirical Risk Minimization. Further developments include learning dynamical processes, as shown by Gibbs et al. [62] through resource-efficient QNNs, and theoretical analyses by Caro et al. [63] and Haug and Kim [24], who explored out-of-distribution generalization and the role of the Data Quantum Fisher Information Metric. Hadiashar et al. [64] completed this picture by establishing information-theoretic lower bounds for quantum learning. Collectively, these works reveal how fully quantum frameworks extend classical learning theory into the quantum domain, enabling efficient learning of measurements, dynamics, and entangled states.

This chapter is structured as follows. We begin with preliminary 2.2 concepts, followed by a discussion of quantum models applied to classical datasets, which introduces the main concepts and frameworks, and discussion over quantum data encoding strategies, common quantum models, and a detailed review of related works 2.3. Next, in Classical Models Trained on Quantum-Generated Data, we present the conceptual framework, explore feature map strategies, discuss kernel-based classical models, and provide a comprehensive literature review highlighting practical applications 2.4. The Fully Quantum Learning Framework section then

outlines its conceptual foundations and surveys key theoretical and experimental works in this area 2.5. Finally, the chapter concludes with an Optimization and Challenges section, addressing crucial aspects of quantum–classical optimization in variational quantum models, structure optimization guided by information-theoretic principles, optimization in quantum kernel models, and a discussion of the main challenges in QML optimization 2.6. Together, these sections provide an integrated overview of current QML paradigms and the optimization principles guiding their development.

2.2 Preliminaries

QML is an emerging interdisciplinary field that investigates how principles of quantum mechanics can be leveraged to enhance machine learning algorithms and computational models [1, 2]. Before introducing quantum machine learning models, we briefly review the fundamental principles of classical machine learning, including model parametrization, linear and nonlinear function approximation, neural networks, and gradient-based optimization. In classical machine learning, tasks typically involve optimizing a model $f(x, \theta)$ with respect to parameters θ based on input data x , with the goal of achieving high accuracy and generalization in predictions. The central question in QML is whether introducing quantum components into this framework, such as quantum data representations, quantum feature spaces, or quantum parameterized models, can provide sensible computational advantages. These may include quantum speed-ups, enhanced expressivity, or the exploitation of uniquely quantum phenomena such as superposition and entanglement for improved learning performance. Conceptually, this can be viewed as extending the classical function into a quantum domain (i.e. $|f(x, \theta)\rangle$).

From a broader perspective, QML research can be categorized into four fundamental paradigms, depending on whether the data and the processing system are classical or quantum. These paradigms form a canonical framework, as illustrated in Figure 2.2.1, which serves as a roadmap for understanding how classical and quantum resources can be combined or contrasted in learning systems.

In classical machine learning, the goal is to learn a mathematical relationship between input data x and their corresponding outputs y . For example, in an image classification task, x might represent pictures of cats and dogs, and y represents their labels “cat” or “dog.” Although the true mapping $g(x)$ that connects inputs to outputs is unknown, the task of machine learning is to approximate it using a model $f(x, \theta)$ with adjustable parameters θ :

$$\hat{y} = f(x, \theta). \quad (2.2.1)$$

A simple example is a linear model, written as

$$\hat{y} = \theta x + b. \quad (2.2.2)$$

Here, θ represents the weight, which determines how strongly each input feature influences the output, while b is the bias, which allows the model to make correct predictions even when all input features are zero. In geometric terms, the bias shifts the model’s prediction line (or surface) up or down, improving its flexibility.

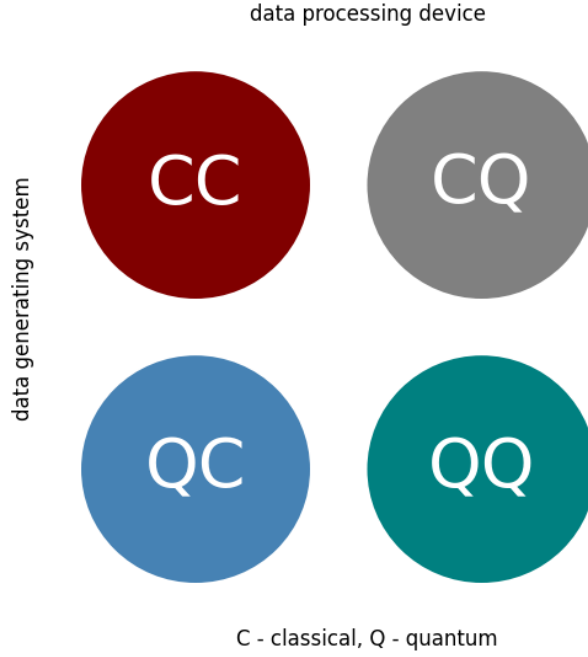


Figure 2.2.1. Classification of the four fundamental paradigms in machine learning, based on the type of dataset and processing model. The vertical axis denotes the dataset type: (C) for classical and (Q) for quantum data. The horizontal axis denotes the processing model: (C) for classical ML models and (Q) for quantum ML models. Each circle is labeled by two letters, where the first letter indicates the dataset type and the second letter indicates the model type (e.g., (CQ) represents a classical dataset processed by a quantum ML model).

However, linear models can only capture straight-line relationships. To describe more complex patterns, machine learning models introduce nonlinear activation functions $\sigma(\cdot)$, leading to

$$\hat{y} = \sigma(\theta x + b). \quad (2.2.3)$$

By stacking several such layers of transformations, each with its own parameters and activation functions, we obtain a neural network, capable of modeling highly nonlinear relationships between inputs and outputs. According to the Universal Approximation Theorem, a sufficiently large neural network can approximate almost any continuous function, which explains their success in many classical applications such as image recognition and speech processing.

Training such models involves finding the best parameters θ and b that minimize a cost function measuring the difference between predicted and true outputs. This is typically achieved through an iterative optimization process called gradient descent, where the parameters are gradually adjusted to improve prediction accuracy.

In essence, classical machine learning learns patterns from data by tuning weights and biases so that the model's predictions closely match reality, which is a principle that also inspires the development of quantum machine learning models.

In summary, this section has reviewed the fundamental principles of classical machine learning to establish a conceptual foundation for understanding QML. While classical algorithms rely entirely on classical data and computations, QML integrates quantum mechanical principles into one or more stages of the learning process. Specifically, QML frameworks may involve quantum-generated data, quantum models, or hybrid configurations that combine both. The following sections will explore these paradigms in detail and discuss key research contributions that have investigated their theoretical underpinnings and practical implications.

2.3 Quantum Models Applied to Classical Datasets

Quantum models applied to classical datasets form a central research direction in QML, where classical data are processed through quantum circuits to exploit potential quantum advantages such as enhanced feature representation or expressivity. The general idea is to transform conventional data into quantum states, process them via parameterized quantum operations, and use measurement outcomes for learning tasks through a hybrid quantum–classical optimization loop.

In this section, we first introduce the concept and framework of quantum models applied to classical datasets, providing a comprehensive overview of this learning paradigm. We then discuss the quantum data encoding strategies, which represent one of the most critical steps in this setting, as the choice of encoding significantly influences the model’s performance and expressivity. Next, we present the common quantum models typically employed for processing classical data. Finally, we review related works that have explored and evaluated this approach in various applications.

2.3.1 Concept and Framework

The pipeline typically begins with classical data, such as feature vectors, tabular datasets, or images, which are mapped into the quantum domain through data encoding or feature maps. These encodings embed classical inputs into quantum states using unitary transformations, taking forms such as angle, amplitude, or basis encoding depending on how the classical information is represented in the Hilbert space. Once encoded, the data are processed by a quantum model, most commonly a PQC or Variational Quantum Circuit (VQC), where a sequence of parameterized unitary operations acts on the quantum states to capture nonlinear dependencies within the data.

After the quantum computation, a measurement step extracts classical information from the quantum state, often in the form of expectation values corresponding to observables. These measurement outcomes are then evaluated using a classical optimization routine, which updates the circuit parameters to minimize a cost function, such as classification error or mean squared loss—defined in the classical domain. This process forms a hybrid quantum–classical learning loop, where the quantum processor handles state transformations and feature mapping, while the classical processor performs parameter updates. The training continues iteratively until convergence, yielding an optimized quantum model capable of performing supervised or unsupervised learning tasks on classical data.

The quantum model at the core of this framework can take several forms. The most common are VQCs and PQC-based QNNs, which act as flexible function approximators parameterized by tunable quantum gates. Other models include Quantum Approximate Optimization Algorithms (QAOA), which adapt combinatorial optimization frameworks for learning, and Quantum Kernel Methods, which implicitly map data to high-dimensional Hilbert spaces to evaluate similarity functions via quantum feature maps. Additionally, Quantum Boltzmann Machines and Quantum Generative Adversarial Networks extend probabilistic and generative paradigms to quantum data representations. The diversity of these architectures highlights the versatility of quantum models in capturing complex classical relationships while exploiting quantum superposition and entanglement to enhance representational efficiency.

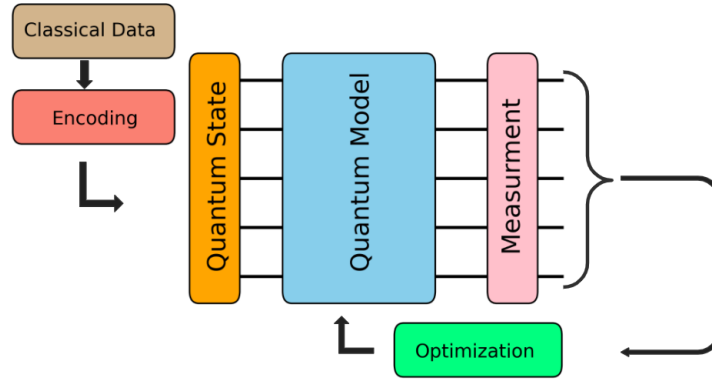


Figure 2.3.1. Schematic illustration of the pipeline for training quantum models on classical data.

The schematic in figure 2.3.1 summarizes this hybrid workflow: classical inputs are first encoded into quantum states through a feature map, processed by a quantum model, measured to produce classical outputs, and optimized via a feedback loop. The interaction between classical and quantum components enables learning processes that exploit quantum resources without requiring fully quantum data.

2.3.2 Quantum Data Encoding Strategies

Most data encountered in real-world applications, such as measurement data, images, or phase-space coordinates, is inherently classical. One of the fundamental challenges in QML is determining how to represent classical data within a quantum computer. In other words, how can we efficiently encode classical information into a quantum state suitable for quantum processing?

This process, often referred to as quantum data encoding or state preparation, involves mapping classical inputs to quantum states through well-defined encoding schemes [6–8]. The choice of encoding method depends on the structure and type of the classical data. Several encoding strategies have been proposed, including basis encoding, amplitude encoding, angle encoding, high-order embedding, and variational or trained embeddings. Each method provides distinct advantages and trade-offs in terms of resource requirements, circuit depth, and expressivity.

Below, we briefly review the most common encoding approaches used in QML.

1. **Basis encoding** maps discrete data, which is represented by binary or integer values, directly to the basis states of a quantum system [9]. Binary dataset D where each $x^m \in D$ is a binary string of the form $x^m = (b_1^m, \dots, b_N^m)$ with $b_i^m \in \{0, 1\}$ for $i = 1, \dots, N$. Then the corresponding quantum state can be written as a uniform superposition over all data samples:

$$|D\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^M |x^m\rangle, \quad (2.3.1)$$

Although conceptually simple, basis encoding often results in highly sparse amplitude vectors in high-dimensional spaces, and the required quantum circuits can become inefficient to implement for large datasets.

2. **Amplitude encoding** encodes continuous data, which represented as complex values, x into the probability amplitude vectors of quantum states. Basically amplitude encoding method demonstrates how information of classical data encoded into the probabilities associated with basis states [10] and represented by

$$|x\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=1}^{2^n} x_i |i\rangle. \quad (2.3.2)$$

Amplitude encoding takes advantage of the exponential growth in the hilbert space and just by adding qubits to the system we can encode exponential amounts of data. Therefore, encodes data into the amplitudes of a quantum state, allowing for efficient representation of large datasets.

3. **Angle encoding** represents continuous data, which represented as floating-point values, x as angles in quantum circuits. Assume, n qubits in zero states that rotate by the value of classical data [10]. This encoding can be illustrated by

$$|x\rangle = \bigotimes_j^n R_k(x_j) |0^n\rangle = \bigotimes_j^n e^{-ix_j \sigma_k / 2} |0\rangle, \quad (2.3.3)$$

where R_k denotes a rotation around axis $k \in \{x, y, z\}$ of the Bloch sphere. The number of qubits typically equals the input dimension, and each feature value controls one qubit rotation. Angle encoding is simple, hardware-efficient, and scales linearly with the data dimension, making it one of the most practical encoding strategies for near-term quantum devices.

4. The **High order embedding** often realized through Instantaneous Quantum Polynomial (IQP) circuits, enables encoding of more complex data dependencies [11, 12]. This encoding constructs quantum states of the form:

$$|x\rangle = (U_z(x) H^{\otimes n})^r |0^n\rangle, \quad (2.3.4)$$

where r is the depth of the circuit, indicating the repeating times of $U_z(x)H^{\otimes n}$. Also, $H^{\otimes n}$ is a layer of Hadamard gates acting on all qubits. $U_z(x)$ is the key step in IQP encoding scheme:

$$U_z(x) = \prod_{[i,j] \in S} R_{Z_i Z_j}(x_i x_j) \bigotimes_{k=1}^n R_z(x_k), \quad (2.3.5)$$

where S is the set containing all pairs of qubits to be entangled using R_{ZZ} gates. For instance, we consider a simple two-qubit gate rotation $R_{Z_1 Z_2}(\theta)$. Its mathematical form $e^{-i\frac{\theta}{2}Z_1 \otimes Z_2}$ can be seen as a two-qubit rotation gate around ZZ , which makes these two qubits entangled. In U_z , an R_{ZZ} gate needs to be added between every pair of qubits in set S . In this built-in IQP encoding method, users are allowed to define their customized set S .

5. **Variational or trained Embedding** refers to learning the encoding of classical information into quantum states using a parameterized (variational) quantum circuit [13]. Here, the encoding parameters are optimized during training alongside the model's parameters, allowing the quantum system to adaptively learn an optimal data representation for a given task. This approach provides flexibility and task-specific expressivity, though at the cost of higher optimization complexity.

2.3.3 Quantum Models

Quantum models can be broadly classified into two main categories: deterministic quantum models and variational quantum models.

Deterministic quantum models refer to systems in which, given a specific quantum state and a defined sequence of quantum operations, the outcome is certain and reproducible. In other words, the result is determined with complete predictability rather than probabilistically. A classical example of a deterministic quantum model is the Deutsch algorithm [28], which demonstrates how quantum computation can solve specific problems more efficiently than its classical counterpart.

In contrast, variational quantum models [29] represent one of the most prevalent frameworks for implementing quantum algorithms on near-term quantum devices. These models address the challenge of finding suitable quantum circuits for a given computational task by employing optimization techniques in conjunction with quantum processing. Instead of designing a fixed quantum circuit a priori, variational quantum models learn optimal circuit parameters through an iterative hybrid quantum-classical loop that minimizes a problem-specific objective function.

The central goal of a variational quantum model is to determine the parameter configuration that best performs the desired computational task. This optimization process is conducted on a classical computer, while the quantum circuit executes the parameterized operations. Thus, variational quantum models operate as hybrid quantum-classical algorithms, where the quantum component encodes and manipulates data, and the classical component updates parameters based on the optimization feedback.

To guide the optimization, a cost function is defined to quantify how closely the current quantum circuit output matches the target outcome. Minimizing this

cost function effectively corresponds to improving the model’s performance. When the cost function reaches its minimum, the model has ideally achieved the intended computational goal.

In general, the operation of a variational quantum model can be described in three main stages:

1. Quantum circuit construction: A parameterized quantum circuit $U(\theta)$, often referred to as an ansatz, is initialized with a set of free parameters θ .
2. Measurement and cost evaluation: An observable M is measured, and a cost function is computed based on the measurement outcomes.
3. Classical optimization: The free parameters θ are iteratively updated by a classical optimizer, which interacts with the quantum device to minimize the cost function.

Several widely recognized examples of variational quantum models include the Variational Quantum Eigensolver (VQE) [30,31], the QAOA [32,33], the VQC [34,35], and QNNs [23]. Each of these models demonstrates distinct applications of the variational framework, which will be reviewed in the following parts.

Variational Quantum Eigensolver

The VQE [30,31] is a prominent example of a variational quantum model specifically designed to determine the minimum eigenvalue of a given observable, most commonly a Hamiltonian. The Hamiltonian typically encodes the total energy of a physical system, although the VQE framework can be applied to any Hermitian operator. Due to its extensive use in quantum chemistry, where it is employed to estimate the ground-state energy of molecular systems, the terminology of energy minimization is widely adopted in VQE-related literature.

To compute the minimum eigenvalue, the method begins with the selection of a trial wave function $|\Psi\rangle = |\psi(\theta)\rangle$, which serves as a parameterized representation of the system’s state. The parameters θ are iteratively optimized to minimize the system’s expected energy. The assumption is that, for an optimal parameter set θ , the corresponding wave function approximates the ground state of the system.

The foundation of the VQE lies in the Rayleigh–Ritz variational principle, which provides a connection between the expectation value of a Hamiltonian and its minimum eigenstate. The principle states that the ground-state energy E_0 of a Hamiltonian H is always less than or equal to the expectation value of H with respect to any normalized trial state $|\Psi\rangle$:

$$E_0 \leq \frac{\langle \Psi | H | \Psi \rangle}{\langle \Psi | \Psi \rangle}. \quad (2.3.6)$$

This relationship allows one to approximate the ground-state energy by minimizing the expectation value $\langle \psi(\theta) | H | \psi(\theta) \rangle$ over the parameter space of the ansatz.

To implement this procedure in a variational quantum framework, a parameterized quantum circuit (ansatz) is designed to generate a family of trial states $|\psi(\theta)\rangle$ that ideally includes the true ground state $|\Psi_0\rangle$. The optimization process follows

the typical hybrid quantum–classical loop used in variational algorithms, which can be summarized in four main steps:

1. Initialization: Define the system’s Hamiltonian H and select an appropriate parameterized quantum circuit $U(\theta)$ (the ansatz).
2. State preparation: Apply the ansatz on the quantum device to prepare the trial quantum state $|\psi(\theta)\rangle$.
3. Measurement: Measure the expectation value $E(\theta) = \langle\psi(\theta)|H|\psi(\theta)\rangle$.
4. Classical optimization: Use a classical optimizer (e.g., gradient descent or a heuristic algorithm) to iteratively update the parameters θ in order to minimize $E(\theta)$.

The VQE is particularly well-suited for NISQ devices, as it employs relatively shallow circuits and can tolerate moderate levels of hardware noise. However, the effectiveness of VQE critically depends on the choice of ansatz, which must balance two competing requirements: it should be expressive enough to capture the true ground state, yet simple enough to allow efficient training and execution on current quantum hardware.

Quantum Approximate Optimization Algorithm

The QAOA [32,33] is one of the most notable algorithms within the class of variational quantum models. It is primarily designed to tackle combinatorial optimization problems by encoding the problem’s objective function into a cost Hamiltonian whose ground state corresponds to the optimal solution.

In QAOA, each quantum state within the parameterized circuit represents a potential solution, and the circuit parameters are optimized to minimize the expectation value of the cost Hamiltonian. The approach draws inspiration from the adiabatic theorem, which states that if a quantum system starts in the ground state of an initial Hamiltonian and evolves slowly enough, it will remain in the ground state of the system throughout its evolution.

To model this, QAOA defines a time-dependent Hamiltonian as

$$H(t) = (1 - t)H_M + tH_C, \quad t \in [0, 1], \quad (2.3.7)$$

where H_M is the mixer Hamiltonian, whose ground state is known and easy to prepare, and H_C is the cost Hamiltonian, which encodes the optimization problem.

Instead of continuously evolving this Hamiltonian as in adiabatic quantum computing, QAOA discretizes the evolution into a finite number of layers. Each layer alternates between applications of the cost and mixer Hamiltonians, parameterized by two sets of variational parameters, γ and β :

$$U_C(\gamma_k) = e^{-i\gamma_k H_C}, \quad U_M(\beta_k) = e^{-i\beta_k H_M}. \quad (2.3.8)$$

which is illustrated by figure 2.3.2

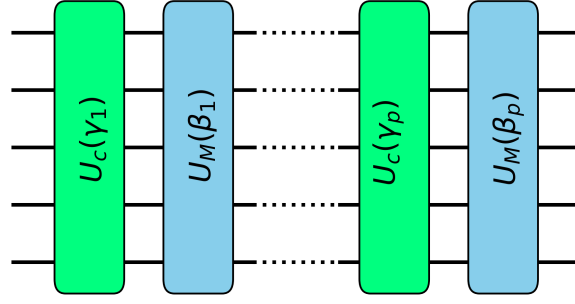


Figure 2.3.2. QAOA circuit for a p number of layers. Each layer consists of a cost unitary $U_C(\gamma_k)$, which encodes the problem Hamiltonian H_C , followed by a mixer unitary $U_M(\beta_k)$, which introduces transitions between states. The parameters γ_k, β_k are optimized to approximate the ground state of H_C .

After applying p alternating layers, the final quantum state is

$$|\psi(\beta, \gamma)\rangle = \left(\prod_{k=1}^p U_M(\beta_k) U_C(\gamma_k) \right) |\psi_0\rangle, \quad (2.3.9)$$

where $|\psi_0\rangle$ is the ground state of the mixer Hamiltonian, typically the equal superposition state

$$|\psi_0\rangle = \frac{1}{\sqrt{2^n}} \sum_{x=0}^{2^n-1} |x\rangle. \quad (2.3.10)$$

The algorithm proceeds by measuring the expectation value of the cost Hamiltonian

$$\langle H_C \rangle = \langle \psi(\beta, \gamma) | H_C | \psi(\beta, \gamma) \rangle, \quad (2.3.11)$$

and optimizing the parameters β and γ using a classical optimization routine to minimize this expectation value. The resulting parameters define a quantum state that approximates the ground state of H_C , and hence provides a near-optimal solution to the original optimization problem.

In summary, the QAOA workflow consists of four main stages:

1. Initialization: Define the cost Hamiltonian H_C representing the optimization problem and the mixer Hamiltonian H_M , often chosen as $H_M = \sum_{j=1}^n X_j$, where X_j are Pauli- X operators.
2. Circuit preparation: Initialize the system in the ground state of H_M and apply p alternating layers of $U_C(\gamma_k)$ and $U_M(\beta_k)$.
3. Measurement: Compute the expectation value of the cost Hamiltonian.
4. Classical optimization: Use a classical optimizer to update β and γ to minimize $\langle H_C \rangle$.

QAOA thus embodies a hybrid quantum–classical optimization loop, where the quantum component explores the solution space through parameterized unitary transformations, and the classical component iteratively refines the parameters. Its structure makes it particularly suited to noisy intermediate-scale quantum (NISQ) devices, balancing expressivity and feasibility by using shallow, efficiently implementable circuits.

Variational Quantum Classifier

The VQC is a class of variational models designed to tackle a wide range of classification tasks. VQCs function by encoding input data into a quantum state through a feature map, implemented as a quantum circuit $U(x)$ that transforms classical data into its quantum representation. The encoded quantum state is then processed by a VQC $V(\theta)$, parameterized by a set of trainable parameters θ .

The next step involves measurement, where the quantum state is projected onto a computational basis. Repeated measurements yield a probability distribution over the basis states, which in a supervised learning context corresponds to a probability distribution over possible class labels. For instance, Figure 2.3.3 illustrates the VQC process in the case of binary classification.

The parameters of the variational circuit are optimized using a classical optimization algorithm, which minimizes a loss function that quantifies the model's classification performance. The overall optimization loop is depicted in Figure 2.3.3. Once the circuit parameters have been optimized, the trained VQC can be employed to classify new data samples. Each new data point is first encoded into a quantum state via the feature map, then passed through the variational circuit to generate a probability distribution over possible output classes. The class associated with the highest probability is selected as the predicted label for the input sample.

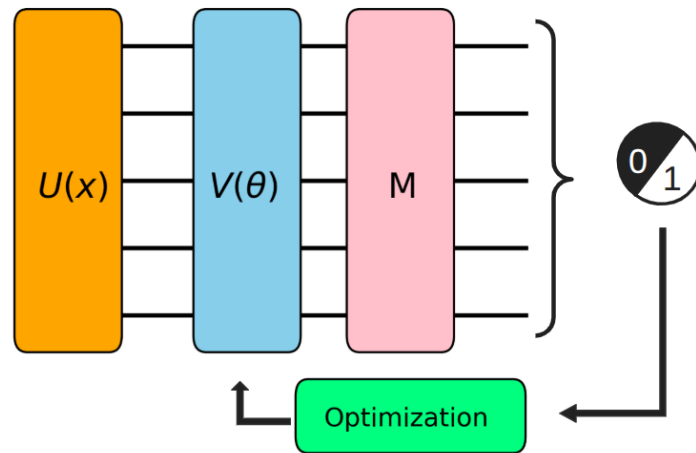


Figure 2.3.3. VQC for a binary classification task. The input data x is first encoded into a quantum state via a feature map $U(x)$, then processed by a parameterized variational circuit $V(\theta)$. Measurement block M produces a probability distribution over the computational basis, which corresponds to predicted binary class probabilities. The parameters θ are optimized using a classical algorithm to minimize a classification loss.

Quantum Neural Networks

QNNs represent the quantum counterpart of classical neural networks, designed to exploit quantum mechanical principles such as superposition, entanglement, and interference to process and learn from data. A QNN typically consists of a series of parameterized quantum gates acting on qubits, where the data is first encoded

into a quantum state via a unitary operation $U(x)$. These gates, parameterized by a set of learnable blocks ($V(\theta_1), V(\theta_2), \dots$), transform the quantum state through successive layers, analogous to the weight matrices in classical neural networks. Figure 2.3.4 illustrates a simplified representation of a QNN algorithm.

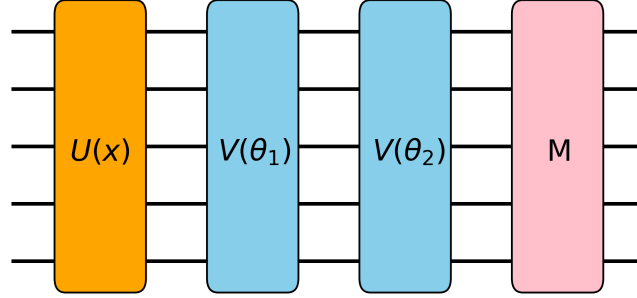


Figure 2.3.4. Schematic of a QNN. Input data is encoded into a quantum state via $U(x)$ and processed through successive parameterized layers ($V(\theta_1), V(\theta_2), \dots$), which transform the state using quantum effects. A final measurement produces the output for learning and prediction.

One of the main conceptual differences between QNNs and classical neural networks lies in the treatment of non-linearity. Classical neural networks introduce non-linearity through activation functions, which enable the model to learn complex, non-linear relationships. In contrast, quantum operations must remain unitary and are therefore inherently linear. As a result, introducing non-linearity in QNNs is challenging and is typically achieved indirectly through the probabilistic nature of quantum measurements. This probabilistic measurement process produces a non-linear mapping between the input data and output probabilities, providing QNNs with limited but meaningful expressive power.

QNNs offer several potential advantages over classical versions. Due to quantum parallelism, they can, in principle, process and represent high-dimensional data more efficiently. Furthermore, QNNs are naturally suited for learning patterns in quantum data, which classical models cannot directly handle. However, significant challenges remain: training QNNs is computationally demanding, as gradient evaluation often suffers from issues such as barren plateaus, where gradients vanish exponentially with system size. Additionally, hardware noise, limited qubit coherence times, and the difficulty of scaling quantum circuits currently restrict the practical applicability of QNNs [36–39].

2.3.4 Review of Related Works

In the following, we review recent studies that investigate this “quantum model–classical data” setting. These works explore diverse aspects such as encoding design, expressivity, generalization, and robustness to noise, collectively providing a theoretical and empirical foundation for understanding how quantum models can achieve learning advantages on classical datasets.

A fundamental concern in QML is the expressive capacity of PQCs. Casas et al. [14] investigate this by showing that PQCs can implement multidimensional

Fourier series expansions. This result is significant because it demonstrates that quantum circuits, depending on their architecture and qudit dimensions, can approximate highly complex functions of classical data. By analyzing how circuit depth, connectivity, and qudit levels influence the number of independent Fourier components that can be represented, this work provides a theoretical basis for understanding how quantum models can emulate complex classical functions. The insights are directly applicable to classical datasets, where accurately modeling subtle functional dependencies is crucial.

Closely related is the role of data encoding, which determines how classical inputs are embedded into quantum states. Schuld et al. [15] systematically explore the connection between encoding strategies and the expressive power of quantum models. They show that the encoding Hamiltonians used to map classical data into quantum states effectively dictate the range of Fourier frequencies that the model can represent. For instance, repeated encoding layers expand this frequency spectrum, enabling the model to capture more intricate functional dependencies. This work emphasizes that encoding is not merely a preprocessing step and indeed it is a central design choice that constrains or enables the model’s expressive potential. Caro et al. [16] extend this insight by deriving generalization bounds that explicitly depend on the encoding map. Their results reveal that even highly expressive circuits can fail to generalize if the encoding is poorly designed, highlighting a delicate balance between expressivity and generalization in QML.

Beyond expressivity, understanding the trainability and overfitting tendencies of quantum circuits is essential. Chen et al. [17] investigate how hardware-efficient ansatzes can overfit classical data despite high expressivity. They show that certain unitary constraints embedded in the circuit act as implicit regularizers, limiting overfitting and promoting more stable learning. This work underscores that high expressivity alone is not sufficient; the structure of the quantum circuit also influences generalization. Similarly, in quantum kernel methods, which embed classical data into high-dimensional Hilbert spaces, the relationship between expressivity and generalization is subtle. Kübler et al. [18] demonstrate that quantum kernels can represent complex functions that classical methods struggle with. However, excessively expressive kernels can harm generalization, as the function space may become too large to learn effectively from finite data. Their work emphasizes the importance of aligning quantum inductive biases with the target function to achieve learning advantages on classical datasets.

A critical limitation identified in quantum kernel methods is the phenomenon of exponential concentration, investigated by Thanasilp et al. [19]. They show that, as the number of qubits increases, quantum kernel values can converge to a nearly fixed constant, making the model insensitive to variations in the input data. This occurs due to factors such as the entanglement structure, measurement strategy, and the nature of the data embedding. Exponential concentration effectively reduces a model’s capacity to generalize, despite apparently successful training. Canatar et al. [20] address this issue by introducing a bandwidth parameter in quantum kernels, showing that careful tuning can restore generalization ability by controlling the spectrum of the kernel integral operator. For datasets with underlying structure, Glick et al. [40] propose covariant quantum kernels, which exploit group symmetries in the data to improve learning efficiency and reduce generalization error. Together,

these studies highlight the intricate interplay between kernel design, expressivity, and inductive bias in quantum learning models.

Practical deployment on noisy quantum devices introduces additional challenges. Heyraud et al. [21] examine the effect of decoherence and dissipation on quantum kernel machines, introducing the concept of effective kernel rank which is the number of independent features that remain extractable under noise. They find that while noise can degrade predictive accuracy, it may also act as an implicit regularizer, mitigating overfitting. Complementing this, Wang et al. [22] theoretically analyze global depolarizing noise, showing that noisy quantum kernels can converge to constant predictions, limiting generalization despite achieving low training error. These findings underscore that robust quantum learning requires careful consideration of both circuit design and noise mitigation, particularly on NISQ devices.

QNNs provide an alternative framework with promising expressivity and trainability. Abbas et al. [23] introduce the effective dimension, derived from Fisher information, as a metric for both generalization potential and susceptibility to barren plateaus. Their results indicate that certain QNNs architectures outperform classical feedforward networks, offering superior expressibility and more favorable training dynamics. Haug and Kim [24] extend this by employing the Data Quantum Fisher Information Metric (DQFIM), showing that symmetry-breaking in training data can paradoxically enhance generalization, and that even a small number of training states may suffice for effective learning. Caro et al. [25] and Yang et al. [41] provide complementary insights, demonstrating that QNNs can generalize effectively from limited classical data, often requiring only a subset of parameters to change significantly during training. These studies collectively highlight QNNs' capacity to efficiently capture complex patterns from classical datasets, even in resource-constrained scenarios.

In image processing tasks, as discussed in Chapter 8, our focus was on enhancing the efficiency of Quantum Image Processing (QIP) to meet the increasing demands posed by large-scale data and complex computational tasks. We conducted a comparative analysis of four quantum image representations: Tensor Network Representation (TNR), Flexible Representation of Quantum Images (FRQI), Novel Enhanced Quantum Representation (NEQR), and Quantum Probability Image Encoding (QPIE). The results demonstrated that quantum kernel methods can achieve classification accuracies comparable to classical approaches while requiring exponentially fewer resources for image storage.

Expanding beyond conventional PQCs and kernels, Jerbi et al. [26] propose a unifying framework for QML models, encompassing kernels, explicit linear models, and data re-uploading circuits. Their analysis shows that alternating data encoding with variational unitaries can map efficiently to linear models in Hilbert space, achieving learning tasks with fewer qubits and reduced data requirements. This insight is particularly relevant for near-term quantum devices, where resource constraints are a critical consideration. Finally, Liu et al. [27] present a landmark demonstration of provable quantum speed-up in supervised learning on classical datasets. Using datasets derived from the discrete logarithm problem, they show that no classical algorithm can achieve better than random guessing, whereas a quantum kernel method attains high accuracy in polynomial time. Importantly, this advantage persists in the presence of realistic sampling noise, providing a concrete

example of quantum-enhanced learning on classical data.

In summary, these studies collectively illuminate the landscape of quantum models applied to classical datasets. They reveal that achieving practical advantages requires a careful combination circuit design, encoding strategy, kernel engineering, noise mitigation, symmetry considerations, and other algorithmic and resource-related factors. While quantum models offer exceptional expressive power and novel mechanisms for generalization, realizing these advantages in practice demands a detailed understanding of both theoretical limits and hardware constraints. Together, these works provide a roadmap for harnessing quantum computation to tackle classical learning problems, bridging fundamental research with the practical challenges of near-term quantum devices.

2.4 Classical Models Trained on Quantum-Generated Data

To understand how classical learning models interact with quantum-generated datasets, it is first necessary to outline how such data are created and transformed for classical analysis. This section first introduces the overall concept and framework of this hybrid setting, then focuses on feature map, which form its core. Next, it examines kernel-based classical models and the role of the quantum kernel as a bridge between quantum data and classical algorithms. Finally, it concludes with a review of related studies exploring this approach in practice.

2.4.1 Concept and Framework

A quantum-generated dataset typically originates from the outcomes of quantum experiments or simulations. These datasets may consist of quantum states (pure or mixed), measurement results, or expectation values derived from quantum circuits. In many cases, each data point represents the result of measuring a quantum state prepared by a parameterized circuit like $U(\theta) |0\rangle$, where the parameters θ encode specific physical conditions, Hamiltonian parameters, or control inputs 2.3.2. Depending on the task, the dataset can represent various phenomena, ranging from ground-state configurations in condensed matter systems to the time evolution of quantum observables or the outputs of quantum communication protocols.

Since quantum data exist in Hilbert space, direct access to their full information is not feasible for classical computers. Therefore, the first key step involves feature extraction or encoding, which transforming quantum states into classical feature vectors through measurement or simulation. Measurements in different bases or expectation values of observables are used to create a classical surrogate representation of the underlying quantum state. These features retain key correlations, coherence patterns, and phase information that can be leveraged by classical learning algorithms.

However, in many cases, such raw classical data are insufficient to preserve the high-dimensional correlations that make quantum systems unique. To address this, researchers employ quantum-inspired feature mappings or kernels. A quantum kernel is a function that measures the similarity between two quantum states, often defined

as

$$K(x_i, x_j) = |\langle \psi(x_i) | \psi(x_j) \rangle|^2, \quad (2.4.1)$$

where $\psi(x)$ is the quantum feature map. These kernels effectively embed classical representations of quantum data into high-dimensional Hilbert spaces, allowing classical models, such as Support Vector Machine (SVM) or Gaussian Processes, to operate on them efficiently. This approach builds a conceptual “bridge” between quantum-generated data and classical learners, preserving quantum correlations while allowing optimization to proceed on classical hardware.

On the classical side, a variety of models can be trained on quantum-derived features, including kernel machines, neural networks, and ensemble methods. Kernel-based models are particularly well-suited for such tasks because they naturally accommodate non-linear, high-dimensional relationships that arise from quantum measurements. Neural networks, on the other hand, can adaptively learn to approximate these relationships, and their architectures can be tailored to the structure of quantum data for instance, convolutional layers for spatial correlations or recurrent layers for temporal quantum dynamics.

During training, the model parameters are optimized by minimizing a cost function that quantifies prediction error on the classical representation of quantum data. In hybrid or feedback-based schemes, this optimization can involve iterative communication between classical and quantum components: the classical optimizer updates circuit parameters, measurements generate new data, and the process continues until convergence. This interplay is illustrated conceptually in figure 2.4.1, where quantum-generated datasets are transformed through a quantum kernel or feature map into a form digestible by classical models, while parameter optimization ensures that the learned representation remains faithful to the quantum source.

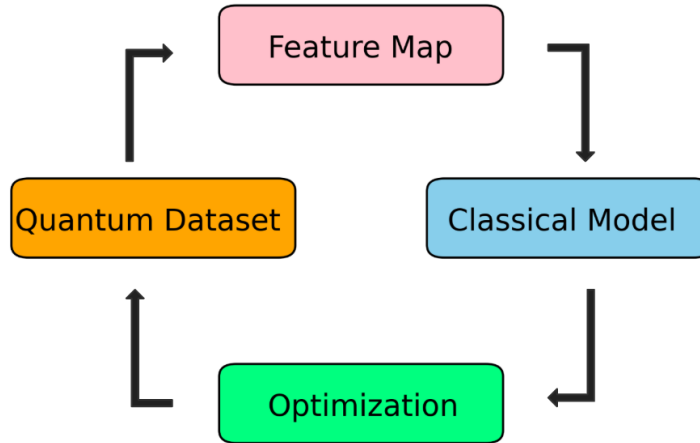


Figure 2.4.1. Schematic illustration of the pipeline for training classical models on quantum-generated data.

2.4.2 Feature Map

Feature map is the following step that maps data to the feature space. A feature map can transform data into a space where it is easier to process. More formally, let X be a set of input data. A feature map ϕ is a function that acts as $\phi : X \rightarrow F$

where F is the feature space. The outputs of the map on the individual data points, $\phi(\mathbf{x})$ for all $\mathbf{x} \in X$, are called feature vectors. Therefore, we map the dataset from data space to higher dimensional space [11]. Let's have a look at these two examples for more clarification:

1. In this example, we observe that the data is not linearly separable in the original feature space, which exists in one dimension. After applying a mapping to a higher-dimensional feature space, specifically two dimensions (i.e., mapping x to (x, x^2)), the data becomes linearly separable. As a result, a simple linear classifier, like a plane, can effectively separate the two classes 2.4.2.

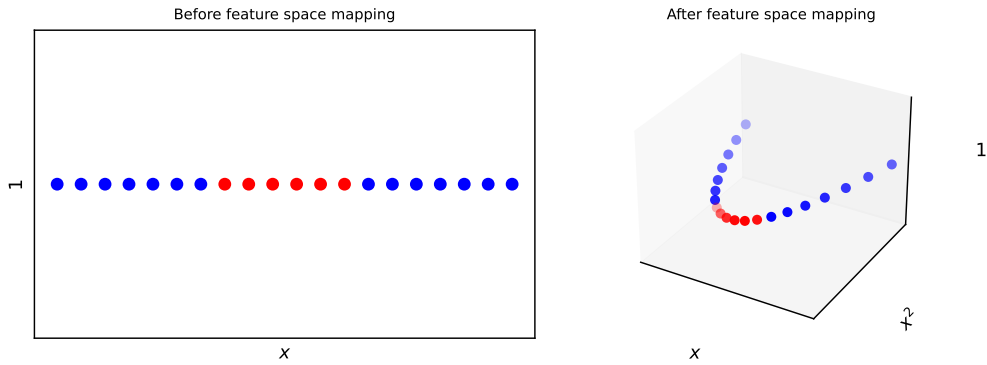


Figure 2.4.2. Plots illustrating how nonlinear data becomes linearly separable after mapping to a higher-dimensional feature space. The left plot shows the original 2D space $(1, x)$, where the classes are not separable. The right plot shows the transformed 3D space $(1, x, x^2)$, where the data becomes linearly separable, enabling a simple linear classifier to distinguish the two classes.

2. In this example, we observe that the data initially resides in a two-dimensional space defined by x and y , and is not linearly separable in this space. By applying a feature mapping that introduces a new dimension, specifically $z = x^2 + y^2$, the data is mapped into a three-dimensional space. In this new feature space, the data becomes linearly separable, enabling us to separate the classes using a linear decision boundary, such as a plane 2.4.3.

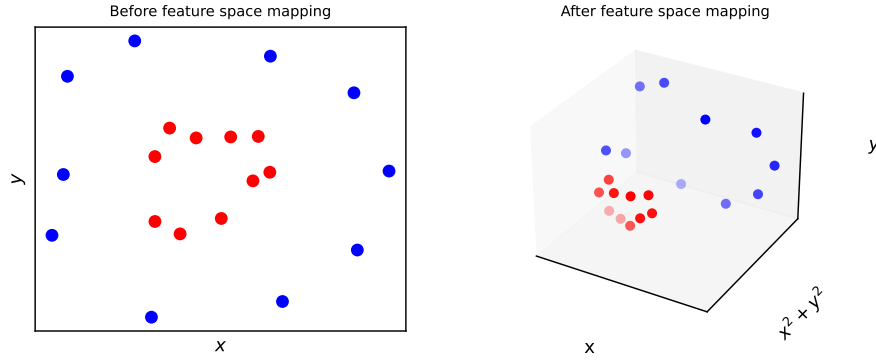


Figure 2.4.3. Illustration of how nonlinear data in the original 2D feature space (x, y) becomes linearly separable after applying a feature map to a 3D space $(x, y, x^2 + y^2)$. In the higher-dimensional space, the transformed data can be separated by a linear decision boundary, such as a plane.

For both of these examples, we have to know that we could map the dataset to higher dimension feature space (i.e. (x, y) to $(1, x, y, xy, x^2, y^2$ and *etc*)), which in this way we have a complete separation of data.

Let's move to the quantum enhanced feature space, which can drastically reduce storage requirements by utilizing quantum superposition. This fundamental quantum property enables qubits to represent multiple states simultaneously. For instance, a n qubit state $|\phi_1, \phi_2, \dots, \phi_n\rangle$ can be written as:

$$|\phi_1, \phi_2, \dots, \phi_n\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle. \quad (2.4.2)$$

In quantum computing, all binary numbers in the set $\{0, 1, \dots, 2^n - 1\}$ can exist simultaneously within an n -qubit quantum register, whereas a classical computer would require a 2^n -bit register. This distinction highlights the advantage of quantum computing in reducing the size of the feature space. This reduction is achieved through quantum superposition, which enables an n -qubit register to represent an exponentially larger set of binary numbers without increasing physical storage requirements. Consequently, superposition plays a key role in promoting an enhanced feature space in QML classifiers.

The fact that, in many practical applications we don't really need to know the entire explicit feature vectors, we only need to have some sense of where it lies within in the feature space relative to other feature vectors that we are trying to classify.

2.4.3 Kernel-Based Classical Model

In QML, kernel functions play a central role in connecting quantum representations of data with classical learning models. The kernel captures the inner product between quantum feature states, defining the geometry of the associated Hilbert space and shaping how effectively data can be separated within it. The choice of kernel therefore has a direct impact on the expressivity and performance of QML models. More

broadly, the kernel lies at the core of all kernel-based learning methods, serving as a bridge between data representation and geometric structure in a higher-dimensional feature space. It is inherently linked to the feature map (see Section 2.4.2), which embeds classical data into this space and determines the nature of the induced kernel function.

Formally, a kernel function is defined as a real-valued mapping:

$$K : \Omega \times \Omega \rightarrow \mathbb{R}, \quad (2.4.3)$$

where Ω denotes the input domain. The function takes two data points from the dataset and measures the similarity between them by evaluating the inner product of their feature-space representations. Mathematically, this relationship is expressed as:

$$K(x, y) = \langle F(x), F(y) \rangle, \quad (2.4.4)$$

where $F(\cdot)$ denotes the feature map that transforms the input data into the (possibly infinite-dimensional) feature space.

The interpretation of the kernel function is naturally tied to the distance between feature vectors in this space. Specifically, the squared Euclidean distance between two mapped points can be expressed as:

$$\|F(x) - F(y)\|^2 = K(x, x) + K(y, y) - 2K(x, y). \quad (2.4.5)$$

This expression reveals that the kernel implicitly encodes geometric relationships among data points without requiring explicit computation of the mapping. This insight forms the basis of the kernel trick, which allows inner products in high-dimensional feature spaces to be computed directly from input-space data using the kernel function itself. As a result, one can leverage the representational power of high-dimensional spaces while maintaining computational tractability.

The kernel function thus determines not only the similarity measure but also the structure of the induced feature space and the nature of the feature map. To serve as a valid kernel, it must satisfy Mercer's condition [46], ensuring that the corresponding Gram matrix is positive semi-definite and that $K(x, y)$ represents an inner product in a reproducing kernel Hilbert space.

Several canonical kernels are widely used in classical machine learning. Two prominent examples include:

1. Polynomial kernel:

$$K_{\text{poly}}(x, y) = (\langle x, y \rangle + 1)^d .$$

2. Radial Basis Function (RBF) kernel:

$$K_{\text{RBF}}(x, y) = \exp\left(-\lambda\|x - y\|^2\right) .$$

Both kernels satisfy Mercer's condition and effectively capture nonlinear relationships in the data without explicitly computing feature vectors. The following examples demonstrate how the choice of kernel influences model performance and classification capability.

1. This example demonstrates how the RBF kernel effectively addresses a nonlinear classification problem 2.4.4. The kernel represents the inner product of feature vectors $F(x)$ and $F(y)$, and given the data distribution, the RBF form provides an appropriate similarity measure:

$$K(x, y) = \langle F(x), F(y) \rangle = \exp(-\lambda \|x - y\|^2). \quad (2.4.6)$$

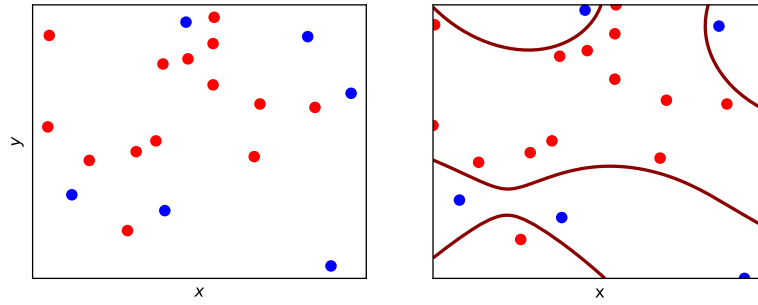


Figure 2.4.4. Visualization of how the RBF kernel captures nonlinear similarity between a typical dataset.

2. This example illustrates how different kernel choices influence classification boundaries for a particular data distribution. Each kernel induces a unique geometry in the feature space, resulting in distinct decision boundaries 2.4.5.

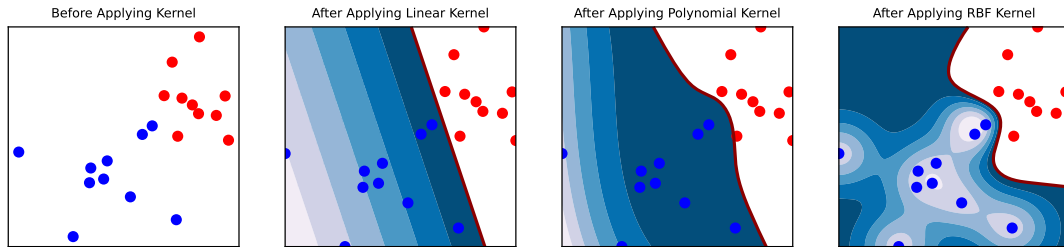


Figure 2.4.5. Comparison of decision boundaries produced by different kernels for the same dataset. Each kernel induces a distinct feature-space geometry, resulting in different classification boundaries. For this particular dataset, even the linear kernel achieves a clean separation, while polynomial and RBF kernels also provide effective and smooth decision boundaries.

3. Here, various kernels are compared on a more complex, nonlinearly separable dataset. Linear and polynomial kernels fail to capture the intricate structure of the data, whereas the RBF kernel successfully models the underlying nonlinear relationships, achieving superior classification performance 2.4.6.

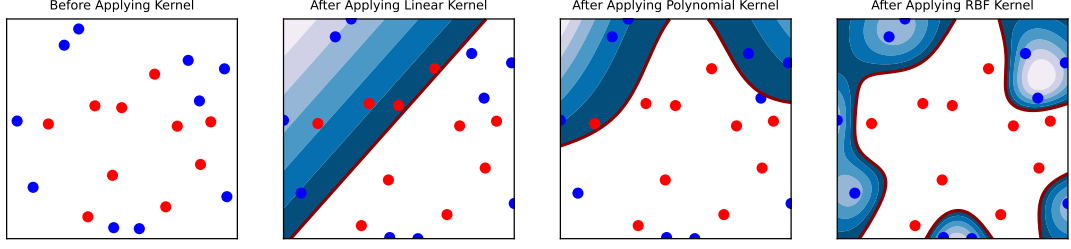


Figure 2.4.6. Comparison of kernel performance on a highly nonlinear dataset. While linear and polynomial kernels are unable to capture the complex structure of the data, the RBF kernel effectively models the nonlinear patterns, resulting in a significantly more accurate classification boundary.

In the quantum setting, data points from the input space Ω are mapped to quantum states using a rank-one projector via a unitary circuit U :

$$\Phi : x \in \Omega \mapsto \Phi(x) = |\phi(x)\rangle\langle\phi(x)| = U(x)|0\rangle\langle 0|U^\dagger(x). \quad (2.4.7)$$

The resulting kernel matrix is obtained using the Hilbert-Schmidt (trace) inner product between these density matrices, which can also be expressed in terms of the fidelity between the quantum states:

$$K(x, y) = \langle\Phi(x), \Phi(y)\rangle_{HS} = |\langle 0|U^\dagger(x)U(y)|0\rangle|^2. \quad (2.4.8)$$

This quantity is always real, independent of global phases, and positive, ensuring a valid kernel.

In quantum circuits, the quantum kernel function is defined as the inner product of two quantum states obtained from the circuit measurements. At a high level, the distinction between the data encoding block $U(x)$ and the quantum model block $V(\theta)$ can be blurred. In this setting, the encoding block not only maps classical data into the quantum Hilbert space but also becomes trainable, effectively teaching the model how to embed the data. During each iteration of training, the encoding block is optimized alongside the measurement process, resulting in a progressively refined kernel that improves performance [13].

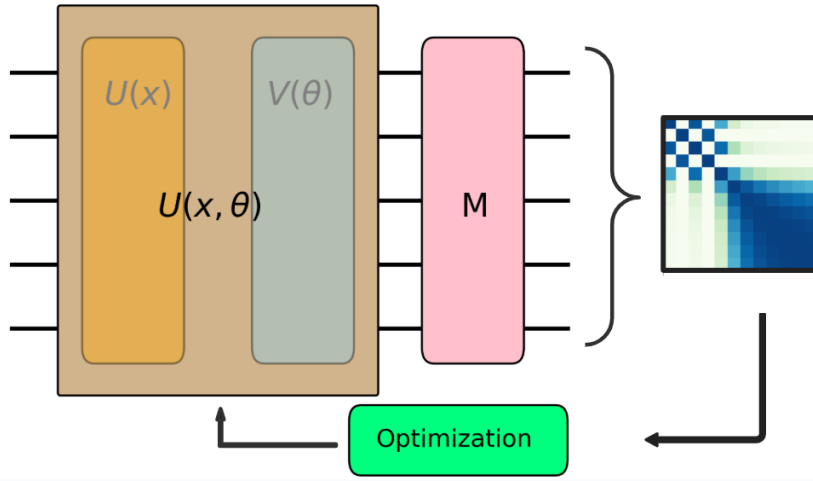


Figure 2.4.7. Quantum circuit is used to generate and iteratively train a quantum kernel by optimizing the joint encoding–model unitary $U(x, \theta)$. Measurements produce quantum states whose inner products form the kernel matrix, which is updated through a classical optimization loop to improve learning performance.

Figure 2.4.7 illustrates the training process for a quantum kernel. The encoding and model blocks are combined into a trainable unit $U(x, \theta)$. After the measurement block produces the quantum state $|\psi\rangle$, the kernel is constructed from these states. The kernel is then used to train a classical model by evaluating a cost function that measures prediction error. Optimization is performed over the kernel, which updates the parameters of $U(x, \theta)$. This process of measurement, kernel evaluation, and parameter optimization is repeated iteratively until convergence, yielding a kernel well-suited for the learning task.

In some cases, the kernel can be analyzed analytically, without full numerical evaluation. For example, consider a dataset of product states:

$$x = (x_1, \dots, x_n) \in \mathbb{R}^n, \quad (2.4.9)$$

with a quantum circuit composed of single-qubit gates for each datum:

$$U(x) = \bigotimes_{i=1}^n U_i(x_i). \quad (2.4.10)$$

where each $U_i(x_i)$ could be a simple rotation, e.g., $\exp(ix_i/2)$. In this case, the kernel factorizes:

$$K(x, y) = |\langle 0 | U^\dagger(x) U(y) | 0 \rangle|^2 = \prod_{i=1}^n |\langle 0 | U_i^\dagger(x_i) U_i(y_i) | 0 \rangle|^2 = \prod_{i=1}^n \cos((x_i - y_i)/2)^2. \quad (2.4.11)$$

Since each term can be computed individually, the kernel is fully classically tractable and effectively corresponds to a classical feature map.

There are several machine learning models that are fed by the kernel Gram matrix, which captures the pairwise similarities between data points. Examples include SVM and Support Vector Regression (SVR), which use the kernel matrix to

find optimal decision boundaries in a high-dimensional feature space; Kernel Ridge Regression (KRR) and Gaussian Process Regression (GPR), which rely on it for modeling nonlinear relationships; Kernel Principal Component Analysis (Kernel PCA) and Kernel Fisher Discriminant Analysis (KFDA), which use it for nonlinear dimensionality reduction and feature extraction; and Spectral Clustering, which employs the kernel (similarity) matrix to identify clusters through graph Laplacian eigen-decomposition.

In our studies, we have focused on the SVM as a supervised learning model and Spectral Clustering as an unsupervised learning approach.

Support Vector Machine

SVM [47, 48] are a family of supervised learning models used for classification and regression. They operate by constructing decision hyperplanes in a Hilbert space, which is known as the feature space, to separate data points belonging to different classes. The training process aims to determine the optimal hyperplane that maximizes the margin between the two classes, while classification is performed by evaluating on which side of the hyperplane a new input lies.

Given a training dataset

$$(\mathbf{x}_i, y_i)_{i=1}^M, \quad \mathbf{x}_i \in \mathbb{R}^d, ; y_i \in -1, +1, \quad (2.4.12)$$

the SVM seeks a decision function

$$f(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b, \quad (2.4.13)$$

where $\phi(\mathbf{x})$ represents a (possibly nonlinear) mapping of the input into a higher-dimensional feature space, $\mathbf{w}$ is the weight vector, and b is a bias term. The optimal hyperplane is obtained by solving the primal optimization problem

$$\min_{\mathbf{w}, b, \xi}; \frac{1}{2} |\mathbf{w}|^2 + C \sum_{i=1}^M \xi_i, \quad (2.4.14)$$

subject to the constraints

$$y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad (2.4.15)$$

where $C > 0$ is a regularization parameter balancing the trade-off between maximizing the margin and minimizing misclassification.

Equations (2.4.14) and (2.4.15) define a convex quadratic optimization problem. While it can be solved directly in the primal form, it is often more convenient to consider the dual formulation, which expresses the problem in terms of Lagrange multipliers α_i . The dual problem is written as

$$\arg \max_{0 \leq \alpha_i < C} \left[\sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right], \quad (2.4.16)$$

subject to $\sum_{i=1}^M \alpha_i y_i = 0$, where $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j)$ denotes the kernel function. The kernel defines the Gram matrix, which measures pairwise similarities in feature

space and allows nonlinear decision boundaries to be learned implicitly through the kernel trick. Common choices include the linear, polynomial, and radial basis function (RBF) kernels.

Once the optimal Lagrange multipliers α_i are determined, the classifier parameters can be reconstructed as

$$\mathbf{w} = \sum_{i=1}^M \alpha_i y_i \phi(\mathbf{x}_i), \quad (2.4.17)$$

and the bias term as

$$b = \sum_{i=1}^M \left(\frac{y_i - \sum_{j=1}^M \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)}{M} \right). \quad (2.4.18)$$

The resulting decision function for a new data point $\mathbf{x}$ is

$$y = \text{sign} \left[b + \sum_{i=1}^M \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \right]. \quad (2.4.19)$$

In the quantum version of SVM, quantum kernel estimation is employed to compute the kernel matrix K using a quantum computer, enhancing the model's ability to capture complex data distributions and nonlinear relationships [49]. Here, the classical input $\mathbf{x}$ is embedded into a quantum state through a feature map

$$\mathbf{x} \mapsto \rho(\mathbf{x}) = |\psi(\mathbf{x})\rangle\langle\psi(\mathbf{x})|, \quad (2.4.20)$$

where $\rho(\mathbf{x})$ is a density operator representing the data in the quantum Hilbert space. The decision boundary in this space can be expressed as

$$f(\mathbf{x}) = \text{Tr}[W\rho(\mathbf{x})] + b = 0, \quad (2.4.21)$$

where W is an observable encoding the model parameters. The optimization process follows the same dual formulation as in Eq. (2.4.16), but with a quantum kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = \text{Tr}[\rho(\mathbf{x}_i)\rho(\mathbf{x}_j)], \quad (2.4.22)$$

estimated on a quantum processor. Once the optimal α_i are found, W and b can be analytically reconstructed as

$$W = \sum_{i=1}^M y_i \alpha_i \rho(\mathbf{x}_i), \quad b = \sum_{i=1}^M \left(\frac{y_i - \sum_{j=1}^M \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)}{M} \right). \quad (2.4.23)$$

The classification of a new sample is then given by

$$y = \text{sign} \left[b + \sum_{i=1}^M \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) \right]. \quad (2.4.24)$$

Quantum kernel estimation effectively transforms data into a high-dimensional quantum feature space, where linear separability is achieved. This capability enables quantum-enhanced SVMs to potentially outperform classical models on complex datasets, though designing optimal quantum feature maps remains an open challenge [50, 51].

Spectral Clustering

Clustering methods aim to divide data samples into groups (clusters) such that points within the same cluster share greater similarity with one another than with points in different clusters. These similarities are quantified through a kernel or affinity function, which measures how closely two samples are related in a chosen feature space. Collecting all pairwise affinities yields the kernel matrix, which forms the basis of spectral clustering.

Given a dataset

$$\mathcal{X} = \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N, \quad \mathbf{x}_i \in \mathbb{R}^d, \quad (2.4.25)$$

the pairwise similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is quantified using a kernel function $k(\mathbf{x}_i, \mathbf{x}_j)$ [52]. In our study, the kernel is defined as the correlation function, computed as the inner product between two data matrices:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle, \quad (2.4.26)$$

which produces a symmetric, positive semi-definite kernel matrix $K \in \mathbb{R}^{N \times N}$. This matrix serves as the basis for constructing the similarity graph used in spectral clustering.

From the kernel matrix K , a similarity graph $G = (V, E)$ is constructed, where each data point $\mathbf{x}_i$ corresponds to a node $v_i \in V$, and weighted edges E represent the similarities between points. An edge between nodes i and j is drawn if the corresponding kernel value exceeds a predefined similarity threshold $\tau \in [0, 1)$. One common approach to build a weighted adjacency matrix $A = [A_{ij}]$ from a kernel (similarity) matrix is to apply a threshold

$$A_{ij} = \begin{cases} K_{ij}, & K_{ij} > \tau, \\ 0, & \text{otherwise.} \end{cases} \quad (2.4.27)$$

This yields a sparsified graph capturing only sufficiently strong affinities. The degree matrix $D = \text{diag}(d_1, d_2, \dots, d_N)$ is a diagonal matrix where each element represents the total connection weight of a node,

$$d_i = \sum_{j=1}^N A_{ij}. \quad (2.4.28)$$

Using A and D , the unnormalized graph Laplacian is defined as

$$L = D - A. \quad (2.4.29)$$

The Laplacian encodes the structure of the graph and serves as the foundation for spectral embedding. Spectral Clustering identifies groups within the data by analyzing the eigen-structure of the graph Laplacian. Let $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$ denote the ordered eigenvalues of L , with corresponding eigenvectors $u_1, u_2, \dots, u_N$. We select the first eigenvector associated with the smallest eigenvalues and form the embedding matrix

$$\Delta = [u_1, u_2, \dots, u_c] \in \mathbb{R}^{N \times c}. \quad (2.4.30)$$

The final clustering step is performed by applying the k -means algorithm to the rows of Δ , grouping them into the number of distinct clusters based on their new coordinates in the embedded space [53]. This approach effectively transforms a nonlinear clustering problem in the input space into a linear one in the spectral domain. Since spectral clustering requires the number of clusters to be specified, selecting an appropriate value is crucial. Selecting too many clusters leads to over-segmentation, resulting in small and potentially insignificant groups, while selecting too few may merge distinct structures and reduce interpretability [54, 55]. Several strategies have been proposed to estimate a suitable number of clusters for a given dataset.

A classical approach is based on examining the eigenvalues of the graph Laplacian. Suppose the Laplacian is defined by 2.4.29. Solving the eigenvalue problem

$$L\varphi = \lambda\varphi \quad (2.4.31)$$

we can infer the suitable number of clusters from the structure of the eigenvalues. Additionally, the corresponding eigenvectors provide insight into how the data points should be grouped within the cluster space.

Another widely used technique is the Elbow Method, which evaluates the variance explained as a function of the number of clusters [57, 58]. Specifically, it analyzes the Within-Cluster Sum of Squares (WCSS) across increasing values of the number of clusters. The Elbow point, where WCSS ceases to decrease substantially, marks a balance between model complexity and cluster quality. Beyond this point, adding more clusters yields only marginal improvements and can lead to overfitting.

A third approach is the Silhouette Method, which measures how well each data point fits within its assigned cluster relative to other clusters [56]. For each element (i), the silhouette width is defined as

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}, \quad (2.4.32)$$

where $a(i)$ is the average distance from (i) to points within its own cluster, and $b(i)$ is the minimum average distance to points in neighboring clusters, using the same metric employed in the clustering process. In this study, all distances are computed using the quantum kernel matrix.

2.4.4 Review of Related Works

Building upon this conceptual framework, we now turn to a review of the major works that have explored this hybrid setting, where classical models are trained on datasets originating from quantum processes. This research direction represents one of the most practically viable approaches within QML, enabling the study of quantum systems through accessible classical computation while maintaining key quantum informational properties in the data.

Huang, Kueng, and Preskill (2021) [42] made one of the most influential contributions in this direction by examining how classical and quantum learners compare when trained on quantum data. In their framework, the dataset consists of measurement results obtained from quantum systems, while quantum learners can also

access the unmeasured quantum states directly. Their analysis showed that, for many average-case learning tasks, a sufficiently large number of measurements allows classical models to achieve performance comparable to quantum models. However, in more complex or worst-case scenarios, particularly when the task requires predicting global properties of highly entangled states, the classical approach becomes exponentially inefficient. This work laid a theoretical foundation for understanding when quantum advantages in learning emerge, emphasizing that the efficiency gap is strongly dependent on the structure and complexity of the underlying quantum data.

A complementary perspective is offered by Huang et al. in “The Power of Data in QML” (2021) [43]. This study highlights that the source and quality of data are often more decisive than the algorithmic model itself. The authors demonstrate that when quantum-generated data contain sufficient classical information, traditional models can learn effectively; yet, for datasets encoding intrinsically quantum correlations, classical learners face fundamental limitations. Their results emphasize that the potential advantage of quantum learning does not arise solely from computational speed but from the information content of the data. In essence, the more “quantum” the dataset, meaning it captures superposition, entanglement, or nonlocal correlations, the greater the potential advantage of quantum-based models over classical ones.

Further expanding this line of inquiry, Jerbi et al. (2023) [44] explored the extent to which classical models can predict the evolution of quantum systems when they only have access to classical measurement data. Their results indicate that if sufficiently rich and flexible measurements are performed, classical learners can, in principle, reconstruct aspects of quantum dynamics. However, when measurements are constrained, as in most realistic experimental settings, learning complex quantum evolutions becomes infeasible, requiring exponentially more samples. This analysis provides a nuanced understanding of the limits of classical learning in dynamic quantum environments and identifies where quantum access is indispensable.

Caro et al. (2024) extended this discussion through an information-theoretic framework [45]. In their study authors derive generalization bounds that quantify how well a learning model trained on quantum data can perform on unseen samples. Their results apply to both classical and quantum learners and help formalize the trade-off between data complexity, model capacity, and generalization performance. A key insight of this work is that classical learners can generalize efficiently from quantum data when they capture essential correlations without overfitting specific measurement outcomes. This theoretical clarity helps identify the regimes in which classical models remain competitive and when quantum learners become necessary for scalable generalization.

Beyond theoretical insights, several practical studies have demonstrated the applicability of classical models to quantum-generated datasets in condensed matter and quantum physics. For instance, as discussed in Chapter 5, classical models, particularly kernel-based classifiers, can successfully identify quantum phases using limited measurement data. By constructing observables from classical data derived from quantum states, the approach enables the classification of ground states without requiring universal quantum gate operations. This has strong experimental relevance, offering a data-efficient route to phase recognition when direct quantum control is

restricted.

Similarly, the project described in Chapter 6 illustrates how classical unsupervised algorithms can reveal hidden structures in quantum data. Without requiring labeled samples, the method uses fidelity-based similarity measures between quantum states to cluster them into distinct phases. Applied to spin-half chains simulated via tensor networks, the algorithm, which enhanced by silhouette and elbow methods, accurately reconstructed phase diagrams, demonstrating that classical learning can autonomously uncover quantum phase structures.

Finally, Chapter 7 introduces a supervised learning framework based on Density Matrix Renormalization Group (DMRG) data. In this approach, classical models are trained on reduced subsystems of many-body quantum states, where key information is retained through Schmidt decomposition. Remarkably, phase recognition remains accurate even when training on small subsystems, indicating that classical learners can extract and generalize relevant quantum features from compact datasets. This work highlights the potential of data-efficient strategies that balance experimental feasibility with predictive power.

Overall, these studies collectively show that classical models trained on quantum-generated datasets occupy a critical middle ground in QML. While they cannot fully replicate the capabilities of quantum learners in highly entangled or information-rich regimes, they provide scalable and experimentally accessible methods for extracting meaningful insights from quantum data. As the field advances, hybrid strategies that combine classical efficiency with selective quantum access are likely to play a central role in realizing the practical potential of quantum-enhanced learning.

2.5 Fully Quantum Learning Frameworks

In a fully quantum learning framework, both the data and the learning model are represented as quantum objects. This setting differs from hybrid approaches, where classical algorithms process quantum data, or classical data are used to train quantum models. Instead, here both the information source and the hypothesis class are genuinely quantum, allowing the learning process to exploit features such as superposition, entanglement, and the geometry of quantum state space. This framework captures the essence of QML operating entirely within the quantum domain. In this section first, we provide an overview of the concepts and framework that define this setting and then, we review the main research works related to this fully quantum learning paradigm.

2.5.1 Concept and Framework

The first element in this framework is the quantum data, represented by quantum states that encode the examples from which the model learns. Each sample may correspond to a pure state $|\psi_i\rangle$, a mixed state ρ_i , or even an entangled multipartite state. Such datasets can originate from quantum physical systems, simulations of quantum processes, or outputs of other quantum devices. Since quantum information cannot be perfectly copied or fully measured due to the no-cloning and measurement disturbance principles, learning must be achieved indirectly, often using repeated preparations of the same state or partial measurements that extract statistical

information. In this sense, quantum datasets provide a richer but more challenging information landscape than classical ones.

The quantum model acts on these quantum data and constitutes the hypothesis that is optimized during training. Depending on the learning task, the model can take different forms. One common choice is a Positive Operator-Valued Measure (POVM), which performs a generalized quantum measurement to map input quantum states to classical outcomes. Learning in this setting corresponds to identifying the optimal measurement that best classifies or discriminates the input states. Another approach employs PQCs or unitary models, where the model represents a tunable quantum transformation acting on the data. These models are widely used for learning quantum dynamics, state transformations, or encoding complex decision boundaries in the quantum state space. Both categories, measurement-based and unitary-based quantum models, have been investigated in the literature (e.g., Cheng, et al., 2015 [65]; Magner and Padakandla, 2024 [66]; Heidari and Szpankowski, 2024 [60]; Caro, Cerezo, et al., 2023 [63]).

After the model acts on the quantum data, a measurement produces classical information that can be compared to target outcomes. The discrepancy between predicted and desired results defines a cost function, which quantifies the performance of the model. The optimization process then adjusts the parameters of the quantum model, either the measurement operators in a POVM or the gate parameters in a quantum circuit, to minimize this cost. This forms a closed learning loop: the quantum data are processed by the model, measured to produce classical feedback, and the model is iteratively updated based on this feedback signal. Through successive training rounds, the quantum model gradually learns to perform its target task with higher accuracy.

This conceptual structure is illustrated in figure 2.5.1. The first block represents the quantum dataset composed of states used for training. The second block corresponds to the quantum model, which can represent either a measurement or a parameterized unitary transformation. The subsequent measurement yields classical outcomes, which are used to evaluate the loss function and optimize the model parameters. This cycle captures the essence of fully quantum learning: quantum data processed by a quantum model, guided by quantum-to-classical feedback and iterative optimization.

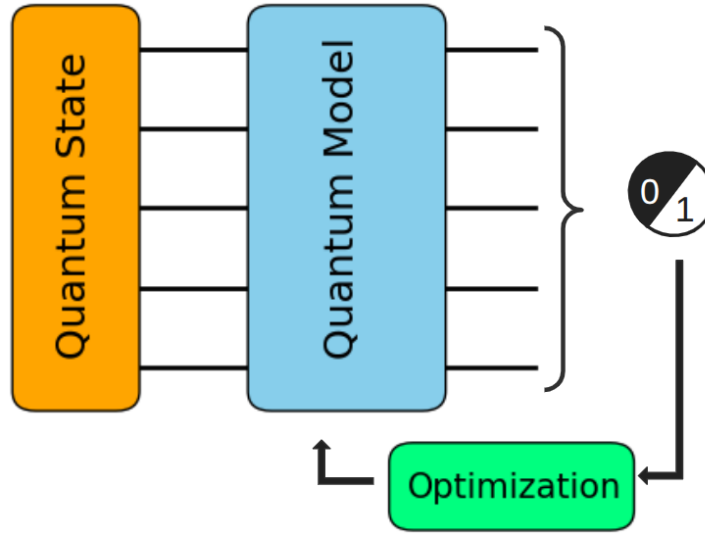


Figure 2.5.1. Fully quantum learning framework. Quantum states form the input dataset and are processed by a quantum model (POVM or parameterized circuit). A binary measurement, which is shown as a half-white, half-black circle, yields classical outcomes used to optimize the model parameters in an iterative training loop.

2.5.2 Review of Related Works

A number of studies have been conducted within this framework, investigating how quantum models, ranging from quantum measurements to parameterized unitary operations, can be trained on quantum data such as entangled states or outputs of unknown quantum processes, thereby establishing the theoretical foundations for learning entirely within the quantum domain.

A prominent line of research focuses on learning unknown quantum measurements. Cheng, et al. [59] studied the scenario where the model is an unknown quantum measurement, represented as a Positive Operator-Valued Measure (POVM), and the data consists of quantum states (density matrices) used as inputs. The learning task is to infer the internal structure of the measurement from observed input-output pairs. By extending classical concepts such as the fat-shattering dimension to quantum systems, the authors demonstrate that the amount of quantum data required scales linearly with the system dimension. This establishes that fully quantum models can be learned efficiently and highlights deep connections between classical statistical learning theory and quantum information theory.

Heidari and Szpankowski [60] further advanced this line of research by formalizing learning using the Quantum Probably Approximately Correct (QPAC) framework, in which quantum states are inputs and the hypothesis is a quantum measurement. A major challenge in this setting is that quantum states cannot be copied or fully measured without disturbance, making standard learning methods like empirical risk minimization inadequate. To address this, they developed Quantum Shadow Risk Minimization, which leverages shadow tomography to efficiently estimate properties of quantum states. Their results establish rigorous upper and lower bounds on the number of quantum samples needed to learn, demonstrating that, under certain conditions, learning efficiency in the fully quantum regime can approach that of

classical models.

Magner and Padakandla [61] explored the conditions under which quantum measurement models are learnable. They show that, similar to classical learning, a quantum analogue of the fat-shattering dimension determines learnability. Additionally, they introduce the concept of approximate joint measurability covering, linking the geometric structure of measurements to learnability. To overcome limitations of classical empirical risk minimization in the quantum setting, the authors propose Denoised Empirical Risk Minimization, which averages out quantum randomness. These works collectively establish a rigorous theoretical foundation for learning quantum models from quantum data, particularly measurements acting as classifiers of quantum states.

Other studies have focused on learning unitary dynamics and parameterized quantum models. Wang, et al. [61] analyzed how the degree of entanglement in training data affects the learning of unknown quantum dynamics. They found a dual effect: while highly entangled data can reduce prediction error when a sufficient number of measurements is available, it can worsen performance when only limited measurements are possible. This counterintuitive result highlights the importance of carefully designing QML protocols, especially for near-term devices with restricted resources. Similarly, Caro, et al., [63] addressed out-of-distribution generalization, showing that models trained on simple product states can generalize to highly entangled states when learning an unknown unitary. This finding is particularly relevant for near-term quantum hardware, where preparing complex states is challenging.

Other research has investigated learning dynamical processes and generalization in parameterized quantum models. Gibbs, et al. [62] introduced the Resource-Efficient Fast-Forwarding algorithm, a QML-based approach for predicting quantum dynamics using a fixed-depth QNNs. By training on short-time evolution data, the model can predict long-time dynamics efficiently, providing significant improvements in fidelity on Noisy Intermediate-Scale Quantum hardware. Haug and Kim [24] extended the theoretical understanding of fully quantum models by introducing the Data Quantum Fisher Information Metric framework. This framework quantifies the learnable capacity of a quantum model relative to the training data, predicting critical thresholds for overparameterization and generalization. Remarkably, breaking symmetries in the training data can improve learning efficiency, and in some cases, out-of-distribution generalization performs better than in-distribution learning.

Finally, foundational studies such as the work by Hadiashar, et al. [64] establish information-theoretic lower bounds for quantum learning. By analyzing quantum Probably Approximately Correct and agnostic learning models, they confirm that classical sample complexity bounds still hold in the quantum regime. They also investigate quantum-specific problems such as the Quantum Coupon Collector, demonstrating how quantum resources influence learning efficiency and highlighting the fundamental limits of quantum sample complexity.

Taken together, these studies illustrate a rich landscape for fully QML, where both models and data are quantum. Quantum measurements and parameterized unitaries act as the hypothesis class, while input quantum states, ranging from simple product states to highly entangled systems, form the training data. The research highlights several key advantages of this approach, including potential

reductions in sample complexity, enhanced generalization to entangled states, and efficient simulation of quantum dynamics. At the same time, it emphasizes the fundamental differences from classical learning, such as the no-cloning constraint, the impact of entanglement, and the need for quantum-specific learning strategies like Quantum Shadow Risk Minimization, Denoised Empirical Risk Minimization, and Data Quantum Fisher Information Metric-based frameworks. Overall, fully quantum QML not only extends classical learning theory into the quantum domain but also offers practical insights for leveraging quantum systems in real-world applications.

2.6 Optimization and Challenges

Optimization is a fundamental component of QML, encompassing the processes through which quantum models learn from data and achieve generalization to unseen instances. Unlike classical machine learning, QML operates in Hilbert space, where computations are inherently quantum and measurement outcomes are probabilistic. Consequently, optimization in QML is typically hybrid, relying on classical routines to update parameters while quantum circuits evaluate objective functions.

At a high level, optimization in QML occurs on two interconnected levels:

1. **Parameter optimization:** This involves tuning continuous parameters of PQCs or QNNs to minimize a cost function, often corresponding to empirical risk or energy. Classical optimizers, such as gradient-based methods (Adam, gradient descent), SPSA, Nelder–Mead, or convex solvers, are employed iteratively to adjust parameters based on quantum measurement outcomes.
2. **Structural or architectural optimization:** Beyond parameter tuning, QML models require careful design of the quantum circuit, encoding strategies, and measurement protocols to ensure expressivity, train-ability, and generalization. Structural optimization includes the selection of circuit ansatz, encoding gates, scaling of input features, entanglement structures, qubit number, and measurement protocols. This level of optimization is crucial for avoiding barren plateaus, where gradients vanish exponentially with circuit size, or exponential concentration in kernel methods, which renders learning infeasible.

Together, these levels form a multi-dimensional optimization landscape, where both algorithmic and structural choices interact to determine the model's performance. Effective QML optimization thus requires not only classical parameter updates but also principled architectural design informed by theoretical and empirical insights.

2.6.1 Hybrid Quantum-Classical Optimization in Variational Quantum Models

Variational quantum algorithms (VQAs), which include QNNs and PQCs, constitute the backbone of most QML frameworks. A central strategy for optimization in these models is the hybrid quantum-classical loop, where quantum circuits evaluate expectation values and classical algorithms iteratively update parameters.

Caro et al. [16] emphasize a multi-dimensional optimization paradigm. At the training level, PQC parameters are updated to minimize empirical risk, while at

the model-selection level, structural risk minimization (SRM) balances empirical performance and generalization. By introducing generalization bounds dependent on architectural and encoding hyperparameters, the study demonstrates that optimization must jointly consider both continuous parameters and discrete architectural choices to achieve models that generalize while remaining quantum-efficient.

Similarly, Schuld et al. [15] formalize optimization as a dual process: continuous parameter training and discrete model design refinement. Parameters are updated via classical optimizers (e.g., Adam), while structural optimization tunes circuit depth, data-encoding Hamiltonians, and observables based on a Fourier-theoretic analysis. This ensures PQCs maintain a balance between expressivity and generalization.

Casas et al. [14] extend this framework by introducing expressibility constraints, aligning the number of trainable parameters N_p with the degrees of freedom like ν required to represent target functions. Architectural choices, including circuit ansatz type, local dimension (qubits vs. qudits), and encoding strategies, are optimized to satisfy this condition. Parameter updates are performed using classical routines (e.g., Nelder–Mead), illustrating the synergy between algorithmic and structural optimization in variational circuits.

2.6.2 Structural Optimization Guided by Information-Theoretic Principles

A persistent challenge in variational optimization is the barren plateau problem, where gradients vanish exponentially, hindering training. Abbas et al. [23] leverage information geometry, utilizing the Fisher information matrix and effective dimension to guide structural choices. Architectures are selected to be both trainable and expressive, maintaining well-conditioned loss landscapes and avoiding flat regions.

Haug et al. [24] introduce the Data Quantum Fisher Information Metric (DQFIM) to quantify model capacity, guiding choices of circuit parameters, training data size, and architecture. Optimization strategies aim for overparametrization for sufficient parameters to reach the global minimum and overcompleteness to rich data for generalization. They also show that non-symmetric or out-of-distribution data can improve generalization efficiency, and that circuit Lie algebras scaling polynomially with system size enhance trainability.

Du et al. [67] emphasize structural tuning based on covering numbers, which quantify the expressivity of a circuit ansatz. By adjusting depth, gate type, and qudit dimension, the model achieves “modest expressivity”, which is enough to represent target functions without causing barren plateaus or overfitting. Structural risk minimization further ensures generalization under hardware constraints, integrating noise-aware design and error mitigation.

2.6.3 Optimization in Quantum Kernel Methods

Quantum kernel methods offer another paradigm for QML optimization, where the focus is on trainable classical weights and structural kernel design. Heyraud et al. [21] show that linear weights applied to quantum feature vectors can be optimized analytically via least-squares minimization, while structural optimization adjusts encoding strategies, system size, and measurement protocols to improve expres-

sivity and robustness. Decoherence may act as an implicit regularizer, enhancing generalization.

Thanasilp et al. [19] address exponential concentration, a major barrier in quantum kernel learning. Structural optimization, choosing embeddings, circuit depth, entanglement structure, and measurement type, is necessary to maintain trainability. Wang et al. [22] and Glick et al. [40] further demonstrate the dual optimization of analytical weights and kernel parameters, ensuring robust learning under noisy and resource-constrained environments.

Canatar et al. [20] and Kubler et al. [18] optimize kernel bandwidth and kernel-target alignment, balancing bias-variance trade-offs and controlling function space expressivity. Gyurik et al. [68] combine SRM with empirical risk minimization in linear quantum classifiers, tuning observables' rank and Frobenius norm to control model complexity. Caro et al. [25] employ hybrid classical-quantum algorithms (VAns, qFactor) to optimize both discrete and continuous parameters while leveraging theoretical generalization bounds to efficiently explore variable architectures.

2.6.4 Challenges in Quantum Machine Learning Optimization

Optimization in QML is inherently constrained by physical and structural limitations. Li et al. [69] show that deep encoding circuits induce barren plateaus, severely limiting parameter trainability. Wang et al. [70] demonstrate that amplitude encoding can create loss barriers: if quantum states representing different classes are too similar, the loss function cannot decrease, rendering parameter optimization ineffective. These results highlight the crucial interplay between classical optimization, encoding strategies, and circuit design. All in all, QML optimization integrates three key elements: parameter updates, structural design, and data considerations. Achieving reliable performance requires balancing expressivity, trainability, and generalization, while respecting hardware constraints and noise limitations.

2.6.5 Optimization in Our Project

In the present project, we adopt a dataset-level optimization approach rather than full hybrid optimization over the model. By optimizing directly over the dataset, we significantly reduce computational cost while still leveraging QML-inspired methods. This approach allows us to focus on data-driven optimization rather than circuit-level tuning, maintaining the advantages of quantum-inspired learning without incurring the overhead of full parameter and structural optimization. Chapters 5, 6, and 7 provide a detailed treatment of this approach and demonstrate its effectiveness.

2.7 Summary and Outlook

In this chapter, we begin by providing an overview of the application of quantum mechanics in machine learning across different settings, specifically, when classical or quantum data interact with either classical or quantum computational devices. We introduce the main paradigms of QML and present the conceptual foundations and general frameworks underlying these approaches.

Subsequently, we review key research papers and projects that exemplify each of these settings, highlighting their methodologies, advantages, and current challenges. The discussion then turns to quantum data encoding strategies, where we explore how classical information can be efficiently mapped into quantum states to leverage quantum computational advantages.

Furthermore, we examine common quantum models used in QML, analyzing their structures and learning mechanisms. Special attention is given to quantum feature maps and kernel-based approaches, which serve as essential tools for establishing connections between quantum representations and classical machine learning models. Together, these sections aim to provide a comprehensive understanding of the theoretical and practical aspects of QML, setting the stage for deeper exploration in subsequent chapters.

Chapter 3

Tensor Networks

Tensor Networks have emerged as a powerful framework for representing and manipulating high-dimensional tensors, with broad applications in areas such as quantum many-body physics, numerical computation, and machine learning. This chapter reviews the theoretical foundations of Tensor Networks, including graphical notation, tensor contractions, singular value decomposition, boundary conditions, and the structure of Matrix Product States and Matrix Product Operators, emphasizing their role in efficiently encoding entanglement and correlations. Key computational algorithms are discussed, such as the Density Matrix Renormalization Group, Time-Evolving Block Decimation, and Tensor Train Recursive Sketching, highlighting their effectiveness in ground state determination, time evolution, and generative modeling. We further survey diverse applications, including solving differential equations, numerical integration, quantum Fourier transforms, interpolation, extrapolation, supervised learning, and gradient-free optimization, illustrating the versatility of Tensor Networks in both physics and data-driven contexts. Overall, this chapter provides a unified perspective on the principles, algorithms, and practical uses of tensor networks, serving as a reference for researchers across disciplines.

3.1 Introduction

In recent years, Tensor Networks (TNs) have emerged as a powerful and versatile framework, with applications spanning a wide range of scientific and engineering fields. Originally developed to tackle the difficulties of quantum many-body physics [71–76] and recently have since found use in areas such as machine learning [98–106]. A key motivation for their development lies in the study of quantum many-body systems, which remain among the most challenging problems in condensed matter physics. A central difficulty in quantum many-body physics is that the Hilbert space dimension grows exponentially with the number of particles, making Exact Diagonalization (ED) feasible only for very small systems. Moreover, strong correlations give rise to complex patterns of entanglement that pose additional challenges for standard numerical approaches. Standard numerical approaches, such as Quantum Monte Carlo (QMC), often encounter severe obstacles, most notably the sign problem, while simulations of dynamics and finite-temperature behavior are limited by the rapid growth of entanglement. Together, these challenges highlight the need for new methods such

as TNs, which offer a systematic and efficient way to address the complexity of strongly correlated quantum systems.

TNs efficiently encode high-dimensional tensors representing quantum states, operators, and processes in terms of local building blocks connected along a graph. By leveraging the entanglement structure of quantum states, TNs overcome the “curse of dimensionality” when entanglement is limited. In one dimension, Matrix Product States (MPS) and Matrix Product Operators (MPO) provide compact representations of ground states, dynamics, and thermal states, while in higher dimensions, projected entangled pair states (PEPS) and the multiscale entanglement renormalization ansatz (MERA) capture area-law states and critical systems [71, 72, 77, 89–94, 126]. TNs exploit both a graphical viewpoint, where connectivity reflects entanglement flow, and a formal backbone, with canonical forms, Schmidt spectra, and tensor constraints encoding symmetries and global order [78–81, 89, 92].

The flexibility of TN methods has enabled their application across diverse domains. In one dimension, MPS-based algorithms such as Density Matrix Renormalization Group (DMRG), Time-Evolving Block Decimation (TEBD), Tensor Train Recursive Sketching (TTRS), and Tensor Cross Interpolation (TCI) allow precise simulations of ground states, time evolution, and learning properties [71–76, 90, 98–106, 113–115]. Higher-dimensional extensions, such as PEPS and MERA, capture critical and strongly correlated systems beyond 1D [90, 91, 93, 94, 204]. TNs methods accommodate long-range interactions, mixed states, and various boundary conditions, and are widely used in quantum chemistry, non-equilibrium transport, disordered systems, lattice gauge theories, and impurity problems [82, 95, 96, 107, 116–120]. Overall, TNs offer a unifying framework bridging condensed matter physics, quantum information, numerical analysis, and theoretical physics, while enabling practical computations for large, strongly correlated systems.

This chapter is organized into three parts. We begin with the foundations of TNs, introducing graphical notation, tensor contractions, and the matrix product state and operator formalisms 3.2. Next, we discuss key algorithms such as DMRG, TEBD, TTRS, and TCI 3.3. Finally, we explore applications ranging from physics and applied mathematics to machine learning, including differential equations, quantum Fourier transform, interpolation, and optimization 3.4.

3.2 Preliminaries and General Principles

Tensors provide a natural framework to describe multidimensional data and their interactions. By definition, a tensor is a multidimensional array of complex numbers, characterized by its rank, or the number of indices it carries. A rank-0 tensor is a scalar, a single number; a rank-1 tensor is a vector, with one index; a rank-2 tensor is a matrix, with two indices; and higher-rank tensors generalize this notion to more indices, capturing increasingly complex relationships.

Manipulating tensors typically involves contractions, the summation over shared indices. Even simple operations illustrate how quickly the notation can become cumbersome: the scalar product of two vectors gives a scalar,

$$S = \sum_{\alpha} A_{\alpha} B_{\alpha}, \quad (3.2.1)$$

while a matrix product contracts one index to produce another matrix,

$$M_{\alpha,\gamma} = \sum_{\beta} A_{\alpha,\beta} B_{\beta,\gamma}. \quad (3.2.2)$$

More complex contractions involve several tensors and multiple shared indices, such as

$$F_{\gamma\omega\rho\sigma} = \sum_{\alpha,\beta,\delta,\nu,\mu} A_{\alpha,\beta,\delta,\sigma} B_{\beta,\gamma,\mu} C_{\delta,\nu,\mu,\omega} E_{\nu,\rho,\alpha}, \quad (3.2.3)$$

where $\alpha, \beta, \delta, \nu$, and μ are shared indices and the result is tensor ranked-4. Considering even larger sets of tensors, the notation becomes extremely dense:

$$E_{\gamma}^{t_1} E_{\gamma_1, \gamma_2}^{t_2} E_{\gamma_2, \gamma_3}^{t_3} E_{\gamma_3}^{t_4} = \sum_{\alpha, \beta, s} M_{\alpha_1}^{s_1} M_{\alpha_1, \alpha_2}^{s_2} M_{\alpha_2, \alpha_3}^{s_3} M_{\alpha_3}^{s_4} \cdot W_{\beta_1}^{s_1, t_1} W_{\beta_1, \beta_2}^{s_2, t_2} W_{\beta_2, \beta_3}^{s_3, t_3} W_{\beta_3}^{s_4, t_4}. \quad (3.2.4)$$

Here, the layers of indices can obscure the underlying structure, making it difficult to see how tensors are connected or which indices are being contracted.

To address this, Roger Penrose introduced graphical tensor notation in 1971 [78]. In this notation, tensors are represented as shapes, and each leg corresponds to one index. Connecting two legs represents contracting the corresponding indices, while free legs correspond to uncontracted indices. This visual representation dramatically reduces the complexity of tensor algebra: intricate sums over many indices are replaced by diagrams, making the structure of operations immediately clear. As a result, Penrose notation not only reduces errors and clutter but also provides intuitive insight into the relationships between tensors, which has proven invaluable in physics, quantum information, and even machine learning applications.

In this graphical language, the basic objects are represented as follows (Fig. 3.2.1): a scalar is a node with no legs; a vector a node with one leg; a matrix a node with two legs; a rank-3 tensor a node with three legs; and a rank-4 tensor a node with four legs. By connecting these building blocks, one can form TNs, where the geometry of the diagram directly encodes the structure of the multilinear relationships, making complex operations far easier to understand and manipulate.

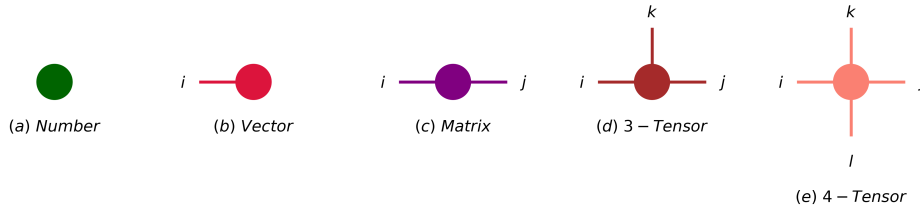


Figure 3.2.1. Penrose graphical notation for basic tensors. (a) A scalar is represented as a node without legs, (b) a vector as a node with one leg, (c) a matrix as a node with two legs, (d) a rank-3 tensor as a node with three legs, and (e) a rank-4 tensor as a node with four legs. The number of legs corresponds directly to the rank (order) of the tensor.

To build intuition, figure 3.2.2 shows several common operations in Penrose notation. Each subfigure illustrates how connecting tensor legs encodes familiar algebraic operations.

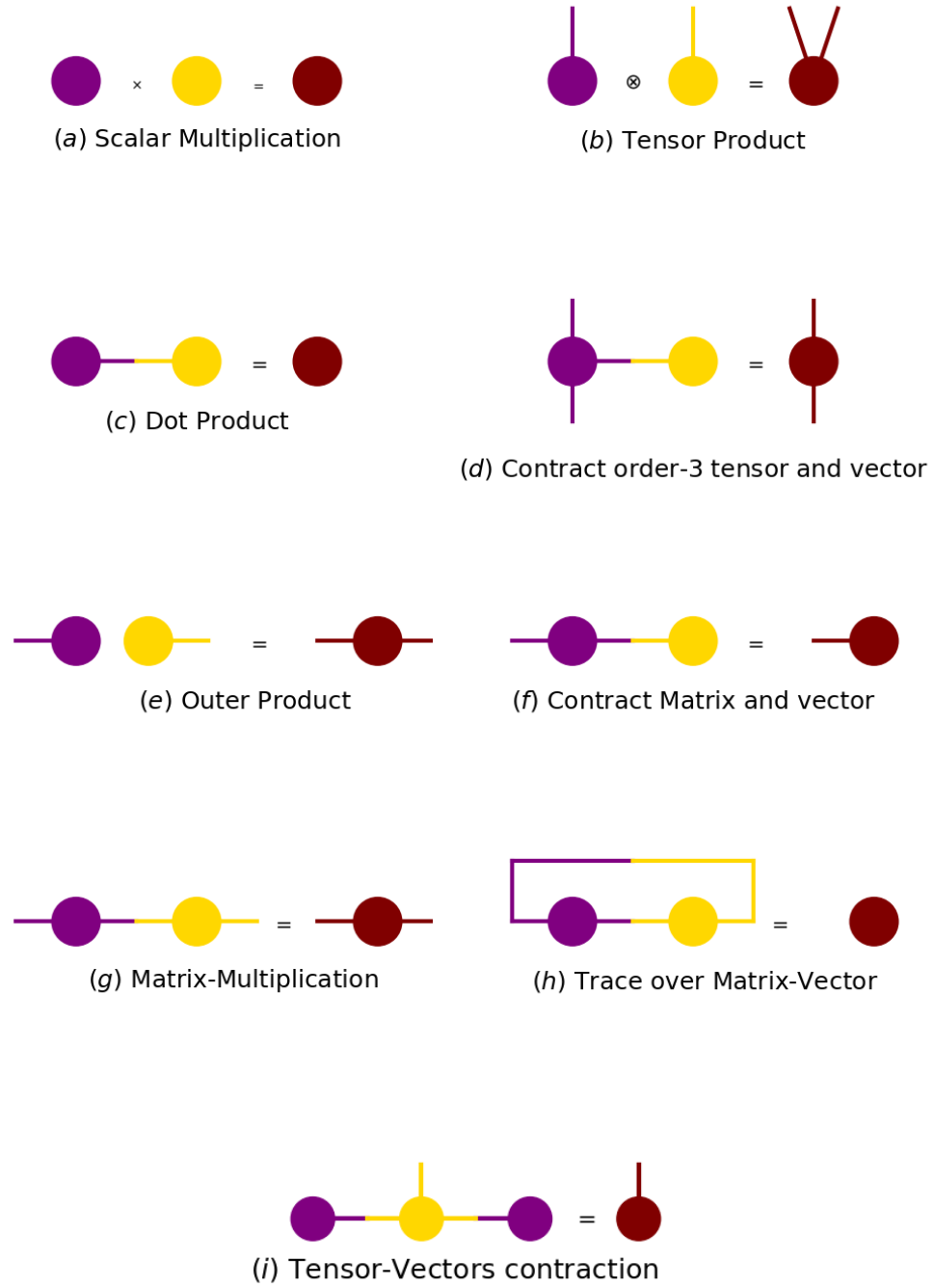


Figure 3.2.2. Illustration of typical tensor operations in Penrose notation. (a) scalar–scalar multiplication; (b) tensor product of two vectors, yielding a matrix; (c) dot product of two vectors, yielding a scalar; (d) contraction of a rank-3 tensor with a vector, producing a matrix; (e) outer product of two vectors, yielding a matrix; (f) contraction of a matrix with a vector, producing a vector; (g) contraction of two matrices along one index, producing another matrix; (h) contraction of two matrices along two indices, giving a scalar (trace); (i) contraction of two vectors with a rank-3 tensor, leaving a vector

In (a), two scalars are combined, yielding another scalar since there are no legs to connect. In (b), the tensor product of two vectors produces a matrix with two legs. In (c), contracting two vectors corresponds to their dot product, resulting in a scalar. Subfigure (d) shows the contraction of a rank-3 tensor with a vector along one shared index, leaving a matrix with two open legs. In (e), the outer product of two vectors again produces a matrix. In (f), a matrix contracts with a vector along one index, giving another vector. Subfigure (g) depicts the contraction of two matrices along a shared index, which yields another matrix, while in (h) contracting along two shared indices gives a scalar which is the familiar trace operation. Finally, in (i), two vectors contract with a rank-3 tensor along two of its indices, leaving a single free leg corresponding to a resulting vector.

Using this graphical language, we can now evaluate the complex contraction between two sets of tensors, M and W , from Eq. (3.2.4) in a far more intuitive way, as shown in Figure 3.2.3. By visualizing the connections, the structure of the computation becomes immediately clear, highlighting how Penrose notation reduces both complexity and cognitive load in tensor operations.



Figure 3.2.3. Graphical representation of the contraction between two sets of tensors, M and W , in Penrose notation. Connecting legs indicate contracted indices, while free legs represent uncontracted indices, making the structure of the computation immediately clear.

So far, we have highlighted the power of Penrose graphical notation in simplifying tensor operations and reducing algebraic complexity. Nevertheless, a fundamental challenge remains: the number of parameters required to represent a tensor grows exponentially with its rank. For instance, a vector (rank-1 tensor) with indices of dimension two has 2 components, a matrix (rank-2 tensor) has $2^2 = 4$ components, and a rank-3 tensor has $2^3 = 8$ components. Extending this reasoning for $n = 50$ in figure 3.2.4. A rank-50 tensor would contain 2^{50} components, on the order of 10^{15} , which is computationally infeasible to store or manipulate directly. This exponential scaling renders high-rank tensors impractical for representing systems with many particles, even when aided by powerful notations.

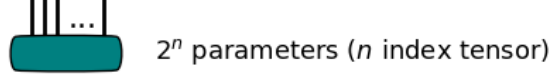


Figure 3.2.4. Graphical representation of a generic n -index tensor. The blue box denotes the tensor, while the attached legs represent its indices, each annotated with the corresponding dimension.

To address this, one may exploit the fact that many physically relevant states or operators admit efficient approximations in terms of low-rank structures. A cornerstone of this approach is the Singular Value Decomposition (SVD), which factorizes a matrix into orthogonal (or unitary) components and singular values. Crucially, truncating small singular values provides a low-rank approximation that drastically reduces storage and computational cost while preserving essential structure. This insight was formalized in the seminal work of Eckart and Young [83], which established the foundation for low-rank approximations of matrices. In the context of TNs, such decompositions serve as the key ingredient that makes the efficient simulation of many-body systems possible.

The SVD is a fundamental matrix factorization that plays a central role in both mathematics and physics [71, 72, 90, 91, 97, 115]. Given a rectangular matrix $A \in \mathbb{C}^{m \times n}$, the SVD asserts that one can always decompose it as

$$A = U \Sigma V^\dagger, \quad (3.2.5)$$

where $U \in \mathbb{C}^{m \times m}$ and $V \in \mathbb{C}^{n \times n}$ are unitary matrices, and $\Sigma \in \mathbb{R}^{m \times n}$ is a diagonal matrix with non-negative real entries known as the singular values σ .

Figure 3.2.5 provides a schematic illustration of the SVD: the matrix A is factorized into the product of two unitary transformations, U and V and a diagonal scaling Σ , which makes the decomposition both conceptually clear and practically useful in TNs algorithms.

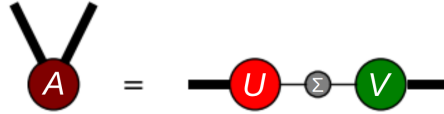


Figure 3.2.5. SVD of a matrix A . The matrix is factorized as $A = U \Sigma V^\dagger$, where U and V are orthogonal matrices and Σ is a diagonal matrix containing the singular values. Truncating small singular values yields a low-rank approximation of A .

Since $\Sigma \in \mathbb{R}^{m \times n}$ is generally rectangular, the number of non-zero singular values along its diagonal cannot exceed the smaller dimension of A . For convenience, Σ is often represented as a square matrix of size $\max(n, m)$ in diagrams, but only the first $\min(n, m)$ entries along the diagonal are non-zero. These non-zero entries correspond to the full rank of the matrix and set an upper bound on the amount of

independent information that can be transmitted across the corresponding “cut” in a TNs.

In practice, one often truncates small singular values to reduce computational cost while retaining the most significant contributions. This truncated number of singular values is precisely what defines the bond dimension χ along a network edge. Each bond in a TNs is thus assigned a bond dimension $\chi \leq \min(m, n)$, which determines the effective rank of the tensor along that bond. Choosing a larger χ allows the network to capture more correlations and provides a more accurate representation of the original tensor, whereas a smaller χ reduces memory and computational requirements at the cost of approximation.

In short, the bond dimension χ is directly linked to the singular values in Σ which counts how many singular values are retained and therefore controls the rank of the tensor along each bond in the network.

Extending the SVD from matrices to higher-order tensors allows one to efficiently represent a large tensor as a network of low-rank components. For a tensor with n indices, SVD can be applied to uncover the most significant correlations. By truncating the singular values to a selected bond dimension χ , the effective rank along each bond is controlled, reducing computational complexity while preserving the essential structure. The resulting factors are then reshaped into smaller tensors connected by bonds, thereby decomposing the original tensor into a structured TNs.

This procedure is applied recursively over different partitions of the indices until the full tensor is expressed as a TNs of low-rank tensors. Each bond in the network carries correlations between the tensors, with the bond dimension χ determining the accuracy and efficiency of the representation. Figure 3.2.6 illustrates this process: a high-order tensor is decomposed step by step into a network of smaller tensors connected by bonds, demonstrating how SVD enables the systematic construction of low-rank TNs.

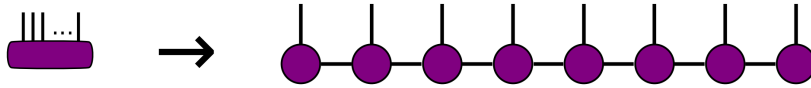


Figure 3.2.6. Visualization of an n -index tensor (left) and its decomposition into a TN (right). The original high-order object is reshaped into interconnected lower-rank tensors.

Exactly this process gives rise to the MPS representation [71, 80, 125]. An MPS expresses the coefficients of a high-order tensor as a product of local tensors connected along virtual bonds. Explicitly, for a system of L sites with Open Boundary Condition (OBC), the quantum state can be written as

$$|\psi\rangle = \sum_{\sigma_1, \dots, \sigma_L} \sum_{\alpha_1, \dots, \alpha_{L-1}} A_{1, \alpha_1}^{[1] \sigma_1} A_{\alpha_1, \alpha_2}^{[2] \sigma_2} \cdots A_{\alpha_{L-1}, 1}^{[L] \sigma_L} |\sigma_1, \dots, \sigma_L\rangle, \quad (3.2.6)$$

where each tensor $A^{[i] \sigma_i}$ acts as a matrix (or higher-rank array) that links the local physical index σ_i to the bond indices. The contractions along the internal bonds encode correlations and entanglement across different parts of the system, making MPS a compact yet powerful representation of high-dimensional states.

Before SVD, a high-order tensor representing N sites with local dimension p has p^N coefficients. After applying SVD across all sites, the number of parameters required to describe an MPS scales as $O(Np\chi^2)$, where χ is the bond dimension controlling the size of the virtual indices connecting neighboring tensors. Summing over all sites, this parameter count remains vastly smaller than the full Hilbert-space size p^N , yet it efficiently captures essential correlations and entanglement patterns in the system.

Central tensors, which connect two neighbors, contain approximately $p\chi^2$ parameters, while edge tensors in an open-boundary MPS, having only one bond index, require roughly $p\chi$ parameters. Graphically, each central tensor is represented with a vertical leg for the physical index and horizontal legs for the bond indices. Constructing an MPS involves contracting these tensors along shared bonds, visually represented by connecting lines between adjacent tensors, providing an intuitive and compact picture of the many-body wave function (see Fig. 3.2.7).

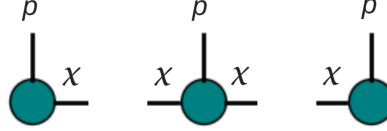


Figure 3.2.7. Graphical representation of a MPS. Each site tensor carries one physical index and one or two bond indices: the boundary tensors on the left and right have only a single bond index, while the bulk tensors possess two bond indices connecting them to their neighbors.

In practice, when building an MPS through successive SVD factorizations, the bond dimension tends to grow rapidly with each bipartition of the system, and without truncation it becomes exponential in the number of sites (e.g., $1 \rightarrow 2 \rightarrow 4 \rightarrow 8 \dots$). To keep the representation efficient, only the largest singular values are retained up to a chosen cutoff χ . This cutoff sets the effective bond dimension of the MPS, balancing computational cost with the accuracy of the approximation.

In an MPS, the growth of the bond dimension depends on the canonical form used. In the right-canonical form, decomposition proceeds from left to right, and the bond dimensions increase until they saturate at the cutoff χ . For instance, with $\chi = 7$, the sequence of a one dimension 6 sites might evolve as $1 \rightarrow 2 \rightarrow 4 \rightarrow 7 \rightarrow 7 \rightarrow 7$. In contrary to the left-canonical form, the process is reversed: the decomposition is performed from right to left, so the bond dimension grows toward the left and then decreases toward the boundary, e.g. $7 \leftarrow 7 \leftarrow 7 \leftarrow 4 \leftarrow 2 \leftarrow 1$.

In the context of TNs, boundary conditions play a crucial role in defining the system's geometry and the nature of its interactions. As depicted in Figure 3.2.8, OBC involve tensors connected at their edges, with no connections beyond the system's boundaries, effectively simulating a finite system with exposed edges. Conversely, Periodic Boundary Condition (PBC) connect the edges of the TNs, forming a continuous loop, thereby mimicking an infinite system by eliminating edge effects. These choices influence the system's entanglement properties and computational complexity; for instance, PBC often facilitate more efficient simulations of bulk properties by reducing surface effects, whereas OBC are more representative of

real-world systems with boundaries. The selection between OBC and PBC depends on the specific physical model and the desired accuracy in capturing boundary effects [84–86].

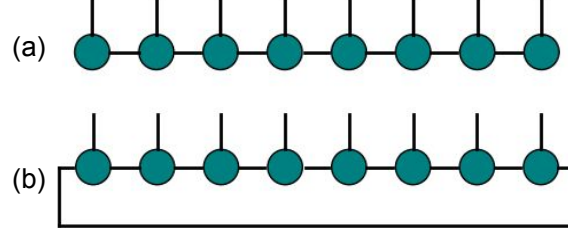


Figure 3.2.8. (a) MPS representations for different boundary conditions. For OBC, the state is written as $|\psi\rangle = \sum_{\sigma_1, \dots, \sigma_L} A_{1, \alpha_1}^{[1]\sigma_1} A_{\alpha_1, \alpha_2}^{[2]\sigma_2} \dots A_{\alpha_{L-1}, 1}^{[L]\sigma_L} |\sigma_1 \dots \sigma_L\rangle$, where the first and last tensors have a single bond index. For (b) PBC, all tensors have two bond indices and the MPS takes the form $|\psi\rangle = \sum_{\sigma_1, \dots, \sigma_L} \text{Tr} (A^{[1]\sigma_1} A^{[2]\sigma_2} \dots A^{[L]\sigma_L}) |\sigma_1 \dots \sigma_L\rangle$.

While boundary conditions determine how the edges of a TNs are connected, another important structural property is Translational Invariance (TI). In a general finite-size MPS, each tensor can be different, meaning the network lacks any repeating pattern. However, translational invariance can be enforced by introducing a unit cell of tensors that is periodically repeated along the one-dimensional chain. As shown in Figure 3.2.9, this idea allows the MPS to be extended to the thermodynamic limit, where the system effectively becomes infinite. For instance, if the unit cell consists of a single tensor, the state remains invariant under a one-site translation. Likewise, if the unit cell is made of two tensors, the system is invariant under a two-site translation, and the same logic extends to larger unit cells. This repetition not only simplifies the structure but also makes the representation more efficient, particularly when studying infinite systems where local properties repeat across the lattice [90].

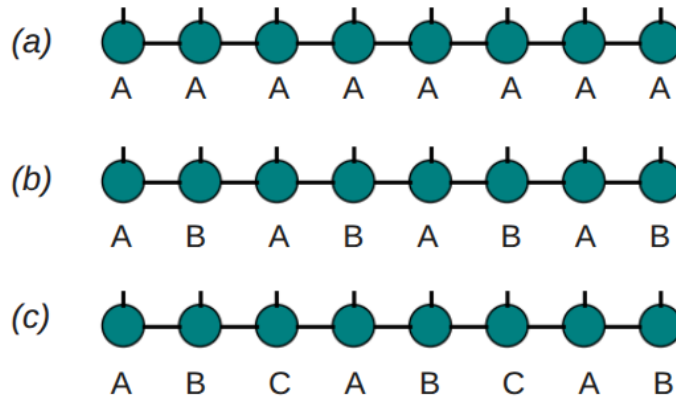


Figure 3.2.9. finite-MPS with (a) 1-site unit cell, (b) 2-site unit cell, and (c) 3-site unit cell.

Another fundamental feature of one-dimensional TNs is their relation to entanglement and the structure of the Hilbert space. In principle, MPS can represent any quantum state in the many-body Hilbert space if the bond dimension χ is allowed to grow exponentially with system size [108, 123, 124]. However, this exponential growth quickly becomes infeasible as proved earlier. Remarkably, it has been shown that the low-energy states of gapped local Hamiltonians in one dimension can be captured to high accuracy with a finite bond dimension [113]. This efficiency arises because such states obey the area law of entanglement entropy, which places strong restrictions on the accessible portion of Hilbert space.

This idea is often visualized as the onion structure of Hilbert space (Figure 3.2.10). States that respect the one-dimensional area law form a relatively thin but physically relevant layer of the full Hilbert space, which MPS with finite bond dimension can capture effectively. As the bond dimension χ increases, the set of accessible states gradually expands, allowing MPS to approximate more entangled states.

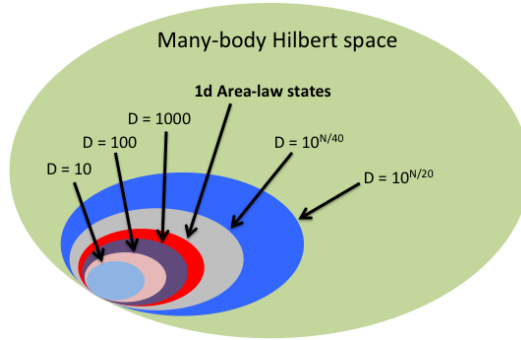


Figure 3.2.10. Onion structure of the Hilbert space (reproduced from [90] and D represent the bond dimension).

The bond dimension χ plays a central role in determining the entanglement that an TNs can encode. Larger χ allows the network to capture more correlations and represent more entangled states, while smaller χ reduces computational cost at the expense of accuracy. In one-dimensional TNs, the combination of network geometry and χ sets an upper limit on the entanglement across any bipartition: if $\chi = 1$, the state reduces to a simple product state with no entanglement. Hence, χ governs not only the size of the parameter space but also the expressive power of the network for representing quantum correlations [71, 72, 89, 90, 92].

Formally, the entanglement entropy of a block of sites scales as

$$S(L) = -\text{Tr}(\rho_L \log \rho_L) = O(\log \chi), \quad (3.2.7)$$

where ρ_L is the reduced density matrix of the block. In gapped one-dimensional systems, $S(L)$ saturates to a constant for large block sizes L , illustrating why MPS with moderate bond dimensions remain efficient.

Having established the structural features of MPS, we now turn to their contractions, which are the building blocks of practical calculations. The most fundamental operation is the computation of overlaps between two MPS, which provides the inner product of the corresponding many-body quantum states.

While the algebraic expressions for overlaps can appear cumbersome, the graphical representation in TNs notation makes the process transparent. Each contraction corresponds to summing over the shared bond indices, with the resulting diagram clearly showing how the two MPS are connected (See figure 3.2.11).

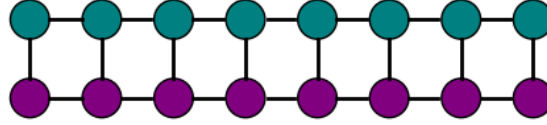


Figure 3.2.11. Graphical representation of the inner product between two MPS, $|\psi\rangle$ and $|\phi\rangle$ with identical physical indices. Each site tensor from the two states is contracted along its physical index, resulting in a scalar that represents $\langle\phi|\psi\rangle$.

In addition to full overlaps, it is often useful to consider partial contractions over specific sites within an MPS. Suppose we have two MPS and we compute their overlap. We can focus on a single site by effectively tracing out that site from one of the chains, leaving the rest of the network contracted. This operation produces a reduced network that represents the overlap of the remaining sites while isolating the contribution of the selected site, as illustrated in Figure 3.2.12. Such partial contractions are fundamental in variational algorithms, gradient computations, and evaluating the response of the network to local changes, since they allow the effect of individual tensors to be studied independently while maintaining the overall MPS structure [71, 72].

$$\frac{\partial}{\partial \text{site 3}} \left(\text{Diagram of two contracted MPS chains} \right) = \text{Diagram of two contracted MPS chains with site 3 isolated}$$

Figure 3.2.12. Overlap of two MPS with a partial derivative taken at site 3. All tensors are contracted normally except at site 3, where the derivative acts, allowing computation of the gradient with respect to the tensor at that site.

Along overlaps application, another fundamental operation in the MPS framework is the computation of expectation values of local operators. To evaluate a local observable or a two-site correlator, the operator is embedded into the full chain, acting non-trivially only on the chosen site(s) while remaining the identity on all others. This simplifies the contraction considerably, since the network is only modified at the relevant positions. As shown in Figure 3.2.13, the procedure closely resembles the overlap calculation, with the operator insertion introducing an additional summation over the corresponding physical indices [71, 72].

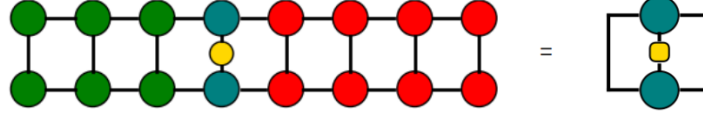


Figure 3.2.13. Graphical representation of the overlap of two MPS with a local operator inserted at a specific site. The tensors on the left and right of the operator are fully contracted, while the central tensor represents the local expectation value, illustrating how $\langle \psi | \hat{O}_{\text{local}} | \phi \rangle$ is computed.

Another important operation in the MPS framework is the summation of two states. Given two MPS, their sum can also be represented as an MPS. The procedure is conceptually straightforward: for each site, the corresponding tensors from the two original MPS are combined into a single block, arranged along the diagonal of the new tensor. This is done for all sites, effectively creating a new network where each tensor carries the combined structure of the two original tensors. The resulting MPS represents the sum of the two states while preserving the TNs structure, although the bond dimension of the new MPS is generally larger, reflecting the increased correlations introduced by the summation. This construction allows operations such as superpositions or linear combinations of states to be handled naturally within the MPS formalism [71, 103, 109].

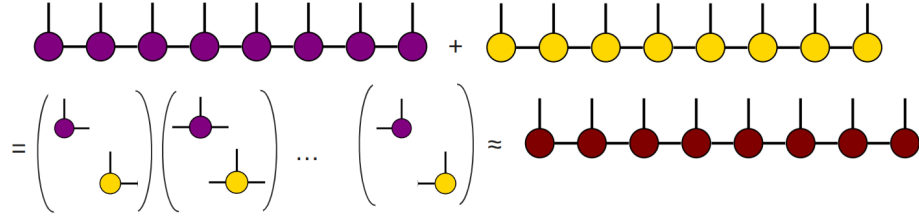


Figure 3.2.14. Sum of two MPS resulting in a new MPS. The addition increases the bond dimension, reflecting the combined entanglement structure of the original states.

Having seen how MPS provide an efficient representation of states, it is natural to ask how operators acting on these states can be expressed in the same language. This leads to the concept of MPO. An MPO is a TNs representation of a linear operator, built as a chain of local tensors connected by internal bonds. Unlike an MPS, which has a single physical index per site, an MPO carries two physical indices per site: one corresponding to the input state and the other to the output. This makes it possible to rewrite a general operator in the form

$$\hat{O} = \sum_{\sigma, \sigma'} M^{[1]\sigma_1, \sigma'_1} M^{[2]\sigma_2, \sigma'_2} \dots M^{[L]\sigma_L, \sigma'_L} |\sigma_1, \dots, \sigma_L\rangle \langle \sigma'_1, \dots, \sigma'_L|, \quad (3.2.8)$$

where each local tensor $M^{[i]}$ encodes the contribution of site i to the operator, and the contractions along the bonds ensure the global structure.

The usefulness of MPOs extends far beyond quantum states. In general, any operator, whether it represents a Hamiltonian, a time-evolution map, a transfer

matrix, or even more abstract transformations, can be decomposed into an MPO. The advantage lies in efficiency: many physically and mathematically relevant operators, such as those with local or short-range structure, admit MPO representations with small bond dimensions. This makes them tractable for both analytical manipulations and numerical algorithms. Moreover, the graphical form of MPOs highlights their close analogy to MPS: they are essentially “operator-valued” TNs, where local actions are encoded in individual tensors, and global structure emerges through the contraction of bonds. As illustrated in Fig. 3.2.15, this dual physical index structure provides a natural and flexible way to represent operators in high-dimensional spaces.

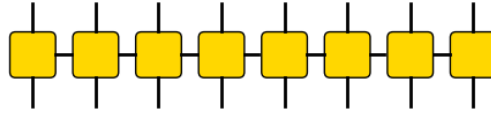


Figure 3.2.15. Graphical representation of a MPO. Each site is represented by a tensor with two physical indices (input and output) and two bond indices connecting neighboring tensors, encoding a many-body operator in a structured form.

Once the MPO formalism is established, the natural next step is to understand how an MPO acts on an MPS. Conceptually, this operation corresponds to applying an operator to a state, and in the tensor-network picture, it is realized by stacking the MPO tensors on top of the corresponding MPS tensors, as illustrated in Fig. 3.2.16. The contraction proceeds site by site. At each site, the MPO tensor (with two physical indices) combines with the MPS tensor (with one physical index), effectively mapping the input physical state to the output. This produces a new local tensor whose bond dimension is enlarged [79].

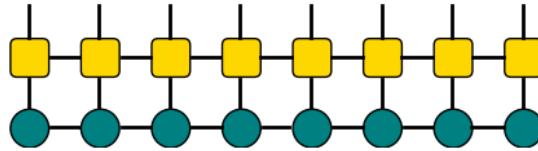


Figure 3.2.16. Contraction between a MPO and a MPS sharing the same physical indices. Each site tensor of the MPO is contracted with the corresponding MPS tensor along the physical indices, producing a new MPS that encodes the action of the operator on the state.

Building on this, we can now consider the expectation value of an operator with respect to an MPS. In the tensor-network language, this corresponds to contracting the MPS representing the quantum state with the MPO representing the operator, and then with the conjugate MPS. Graphically, this forms a layered network where the MPO sits “sandwiched” between the ket and bra MPS, as illustrated in Fig. 3.2.17. Each site contributes a local tensor contraction, combining the physical indices of the MPS and MPO while connecting the virtual bonds across the lattice. The resulting network efficiently encodes the global expectation value without ever

explicitly constructing the full Hilbert-space operator or state.

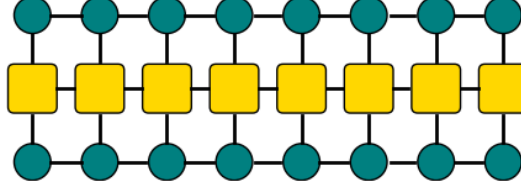


Figure 3.2.17. Computation of the expectation value of a MPO. The MPO is “sandwiched” between two MPS, with tensors contracted along their shared physical indices and bond indices, yielding $\langle \psi | \hat{O} | \psi \rangle$.

Once we have the expectation value, it is often useful to consider how this quantity changes with respect to a specific tensor in the MPS, i.e., taking the partial derivative with respect to the tensor at a particular site. Analytically, this can become quite cumbersome: each MPS tensor is entangled with its neighbors through the virtual bonds, and the MPO introduces additional layers of contraction. Writing out the derivative explicitly involves carefully tracking how every bond and physical index contributes, quickly leading to a complex web of terms. However, in the tensor-network picture, this process can be visualized and simplified dramatically. After taking the derivative with respect to a specific site, the corresponding tensor effectively “drops out” of the network, as if we are momentarily ignoring it, leaving the rest of the MPS and MPO intact (see Fig. 3.2.18). This graphical perspective not only clarifies the structure of the derivative but also provides a practical way to implement gradient-based optimization algorithms for MPS, avoiding the messy algebra of full analytical expressions.

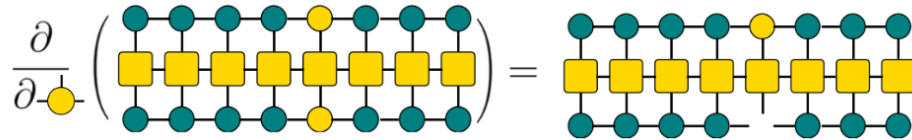


Figure 3.2.18. Partial derivative of Expectation value of a MPO with respect to an MPS, with respect to the tensor at a specific site. All other tensors are fully contracted, while the derivative at the chosen site allows computation of the gradient of $\langle \psi | \hat{O} | \psi \rangle$ with respect to that tensor.

3.3 Tensor Network Algorithms

The power of TNs lies not only in their compact representations but also in the efficient algorithms built upon them. This section reviews several key methods that demonstrate how TNs can model and compute high-dimensional systems with remarkable accuracy and efficiency.

We begin with two foundational one-dimensional algorithms: the DMRG [113–124] and the TEBD [125,126]. Both exploit the MPS structure to efficiently represent

and optimize quantum states. DMRG is celebrated for its precision in finding ground states of strongly correlated one-dimensional systems, while TEBD excels in simulating both real- and imaginary-time dynamics. Together, they highlight how TNs can capture the essence of complex many-body physics with compact and tractable representations.

More recently, TTRS [98, 99] has emerged as a powerful approach in generative modeling. Unlike DMRG and TEBD, which focus on physical systems, TTRS applies tensor network principles to learn high-dimensional probability distributions directly from data. By recursively sketching empirical distributions and extracting MPS tensors through linear algebra operations such as SVD, it provides a fast, deterministic, and scalable alternative to traditional gradient-based learning methods.

Finally, the TCI algorithm [102–105] extends tensor network techniques to the efficient approximation of high-dimensional functions. Operating in an active-learning fashion, TCI treats the tensor as a black-box function that can be queried at selected points. It adaptively identifies the most informative entries and reconstructs the rest through interpolation. By combining Interpolative Decomposition (ID) with tensor-train representations, TCI achieves highly accurate reconstructions with exponentially fewer evaluations bridging classical numerical methods and modern data-driven modeling.

3.3.1 Density Matrix Renormalization Group

A central task in many-body physics is finding the ground state of a given Hamiltonian. The conventional approach searches the ground state in the full Hilbert space. In the MPS framework, however, we restrict attention to a variational subspace spanned by MPS and look for the lowest-energy state within that subspace. This variational problem can be written using a Lagrangian to enforce normalization, e.g.

$$\langle \psi | \hat{H} | \psi \rangle - \lambda \langle \psi | \psi \rangle. \quad (3.3.1)$$

The minimizing state $|\psi\rangle$ corresponds to the best MPS approximation of the ground state and the Lagrange multiplier λ gives the associated energy estimate. The equation 3.3.1 is illustrated in figure 3.3.1.

Directly minimizing the functional is difficult because the variational parameters (the MPS tensors) appear as products, yielding a highly non-linear optimization. The DMRG strategy sidesteps this by optimizing one site tensor at a time while keeping all others fixed. If we freeze every tensor except the site l tensor M^{σ_l} , the dependence on the remaining variables becomes quadratic, and the local optimization reduces to a linear-algebra problem. Solving this local problem lowers the energy and produces an improved variational state; repeating the same procedure site-by-site and sweeping across the lattice refines the MPS until the energy converges [113–124].

We now show the overlap while keeping M^{σ_l} explicit: It is determined that:

$$\langle \psi | \psi \rangle = \sum_{\sigma_l} \sum_{a_{l-1} a_l} \sum_{a'_{l-1} a'_l} \Psi_{a_{l-1}, a'_{l-1}}^A M_{a_{l-1} a_l}^{\sigma_l *} M_{a'_{l-1} a'_l}^{\sigma_l} \Psi_{a_l, a'_l}^B, \quad (3.3.2)$$

with the left and right environment tensors defined by

$$\Psi_{a_{l-1},a'_{l-1}}^A = \sum_{\sigma_1, \dots, \sigma_{l-1}} (M^{\sigma_{l-1}\dagger} \dots M^{\sigma_1\dagger} M^{\sigma_1} \dots M^{\sigma_{l-1}})_{a_{l-1}a'_{l-1}}, \quad (3.3.3)$$

$$\Psi_{a_l, a'_l}^B = \sum_{\sigma_{l+1}, \dots, \sigma_L} (M^{\sigma_{l+1}} \dots M^{\sigma_L} M^{\sigma_L\dagger} \dots M^{\sigma_{l+1}\dagger})_{a_l a'_l}. \quad (3.3.4)$$

When the Hamiltonian is expressed as a MPO and the MPS is in a mixed-canonical form, the expectation value $\langle \psi | \hat{H} | \psi \rangle$ can be written as

$$\langle \psi | \hat{H} | \psi \rangle = \sum_{\sigma_l, \sigma'_l} \sum_{a'_{l-1}, a'_l} \sum_{a_{l-1}, a_l} \sum_{b_{l-1}, b_l} L_{b_{l-1}}^{a_{l-1}, a'_{l-1}} W_{b_{l-1}, b_l}^{\sigma_l, \sigma'_l} R_{b_l}^{a_l, a'_l} M_{a_{l-1}, a_l}^{\sigma_l*} M_{a'_{l-1}, a'_l}^{\sigma'_l}. \quad (3.3.5)$$

Taking the stationary condition of equation 3.3.1 with respect to the complex conjugate of the local tensor entries $M_{a_{l-1}, a_l}^{\sigma_l*}$ yields

$$\sum_{\sigma'_l} \sum_{a'_{l-1}, a'_l} \sum_{b_{l-1}, b_l} L_{b_{l-1}}^{a_{l-1}, a'_{l-1}} W_{b_{l-1}, b_l}^{\sigma_l, \sigma'_l} R_{b_l}^{a_l, a'_l} M_{a'_{l-1}, a'_l}^{\sigma'_l} - \lambda \sum_{a'_{l-1}, a'_l} \Psi_{a_{l-1}, a'_{l-1}}^A M_{a'_{l-1}, a'_l}^{\sigma_l} \Psi_{a_l, a'_l}^B = 0. \quad (3.3.6)$$

This equation is a generalized eigenvalue problem after reshaping indices. Defining the effective local Hamiltonian H , the overlap (or normalization) matrix N , and the vectorized local tensor ν by

$$H = \sum_{b_{l-1}, b_l} L_{b_{l-1}}^{a_{l-1}, a'_{l-1}} W_{b_{l-1}, b_l}^{\sigma_l, \sigma'_l} R_{b_l}^{a_l, a'_l}, \quad (3.3.7)$$

$$N = \Psi_{a_{l-1}, a'_{l-1}}^A \Psi_{a_l, a'_l}^B \delta_{\sigma_l, \sigma'_l}, \quad (3.3.8)$$

$$\nu = M_{a_{l-1}, a_l}^{\sigma_l}, \quad (3.3.9)$$

we arrive at the matrix equation 3.3.1

$$H\nu - \lambda N\nu = 0. \quad (3.3.10)$$

whose smallest eigenvalue λ gives the local energy estimate and whose eigenvector ν determines the updated $M_{a_{l-1}, a_l}^{\sigma_l}$ after reshaping.

In practice, N can be ill-conditioned, making the generalized eigenproblem numerically delicate. For OBC, however, this difficulty is eliminated if the MPS is left-normalized up to site $\ell - 1$ and right-normalized from site $\ell + 1$ onward. Then Ψ^A and Ψ^B simplify and N reduces to the identity, so the generalized problem becomes the standard eigen-problem.

$$\sum_{\sigma'_l} \sum_{a'_{l-1}, a'_l} \sum_{b_{l-1}, b_l} L_{b_{l-1}}^{a_{l-1}, a'_{l-1}} W_{b_{l-1}, b_l}^{\sigma_l, \sigma'_l} R_{b_l}^{a_l, a'_l} M_{a'_{l-1}, a'_l}^{\sigma'_l} - \lambda M_{a_{l-1}, a_l}^{\sigma_l} = 0, \quad (3.3.11)$$

or compactly $H\nu = \lambda\nu$ as depicted in figure 3.3.1

A practical DMRG sweep cycle follows these steps:

1. Initialization: Start with an initial MPS, usually chosen in right-normalized form.
2. Build environments Compute the right environment tensors R for all sites.
3. Right sweep: Move from the left end ($\ell = 1$) to the right ($\ell = L - 1$):
 - Solve the local eigenvalue problem for the tensor at site ℓ .
 - Left-normalize this tensor (using SVD or QR) and pass the remaining factor to the next site as its starting guess.
4. Left sweep: Move back from the right end ($\ell = L$) to the left ($\ell = 2$):
 - Solve the local eigenvalue problem for the tensor at site ℓ .
 - Right-normalize this tensor and pass the leftover factor to the previous site.
5. Iterate: Repeat right and left sweeps until the energy no longer changes significantly.

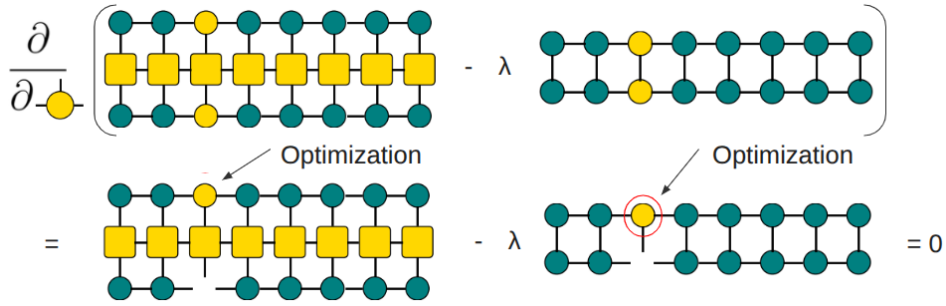


Figure 3.3.1. Single-step illustration of the DMRG Algorithm. A partial trace is performed over site 3, yielding the effective Hamiltonian $H_{eff}^{[3]}\psi = \lambda\psi$. This step shows how the local eigenvalue λ is computed during the optimization of the MPS.

During these iterations the energy typically decreases monotonically, but two practical caveats deserve mention. First, starting from a random MPS can produce poor initial guesses for the local eigensolvers, slowing convergence in early sweeps. Second, the sweep algorithm is variational but not convex — it can converge to a local rather than the global minimum depending on initialization and algorithmic details.

3.3.2 Time Evolving Block Decimation

Another important TNs algorithm in one dimensional system is TEBD [125, 126]. Time evolution refers to the process of propagating a quantum state in time according to a Hamiltonian, which governs the system's dynamics. In real-time evolution, this is achieved by applying a unitary transformation to the initial state, described mathematically as e^{-iHt} , where H is the Hamiltonian and t is time (see equation 3.3.12).

$$|\Psi(\delta t)\rangle = \exp(-iH\delta t)|\Psi(0)\rangle. \quad (3.3.12)$$

Alternatively, in imaginary-time evolution, the time parameter is replaced by an imaginary value ($t \rightarrow -i\tau$), resulting in the operator $e^{-H\tau}$.

During the time evolution process, the system gradually relaxes and evolves toward its ground state. This method is particularly effective for energy minimization tasks, as it systematically reduces the contributions from higher-energy states. In practice, time evolution is implemented by iteratively applying the time evolution operator to an initial state, often represented using variational ansatzes such as MPS. Through successive iterations, the state converges to the ground state, making this approach a powerful tool for optimization in quantum many-body systems.

Time evolution within the framework of MPS is conceptually straightforward. One starts from an initial MPS and applies the time-evolution operator, resulting in a new state. During this procedure, the bond dimension χ typically increases due to the growth of quantum correlations and entanglement. To maintain computational efficiency, the updated state is subsequently compressed back to a manageable bond dimension, with the goal of preserving the essential physical properties. By iteratively repeating this procedure, the system is propagated forward in time.

The central challenge in simulating time evolution arises from the fact that the full time-evolution operator,

$$U(\delta t) = e^{-iH\delta t}, \quad (3.3.13)$$

is generally intractable to apply directly, since the Hamiltonian H contains many local interaction terms that do not commute with one another. In such cases, a standard approach is to employ the Trotter–Suzuki decomposition. The idea is to approximate the exponential of a sum of operators by a product of exponentials of simpler terms, thereby rendering the problem computationally feasible.

Mathematically, if the Hamiltonian is written as a sum of local terms, (i.e., $H = \sum_i h_i$), then the first-order Trotter decomposition gives:

$$e^{-iH\delta t} \approx e^{-ih_1\delta t} e^{-ih_2\delta t} \dots e^{-ih_L\delta t} + O(\delta t^2), \quad (3.3.14)$$

where h_i contains the interaction between sites h_i and h_{i+1} . Here, the approximation error arises because the local terms do not commute: $[h_i, h_{i+1}] \neq 0$ and this is the reason why the order of operations matters. To handle this, the Hamiltonian is split into two parts:

- $\hat{H}_{\text{odd}}$: interactions on odd bonds (e.g., sites 1–2, 3–4, ...),
- $\hat{H}_{\text{even}}$: interactions on even bonds (e.g., sites 2–3, 4–5, ...).

Within each group, the terms commute, which makes the implementation efficient. The system can be evolved by first applying the odd-bond operator and then the even-bond operator. After each step, the state is compressed to control bond growth and truncation errors.

The algorithm for one time step can be summarized as:

1. Apply the MPO for odd bonds to $|\psi(t)\rangle$,
2. Apply the MPO for even bonds to the updated state,
3. Compress the resulting MPS to reduce its bond dimensions.

This procedure yields an updated state:

$$|\psi(t + \delta t)\rangle \approx e^{-i\hat{H}_{\text{even}}\delta t} e^{-i\hat{H}_{\text{odd}}\delta t} |\psi(t)\rangle, \quad (3.3.15)$$

which is shown in Fig. 3.3.2, is then repeated iteratively until the energy (or another chosen observable) converges.

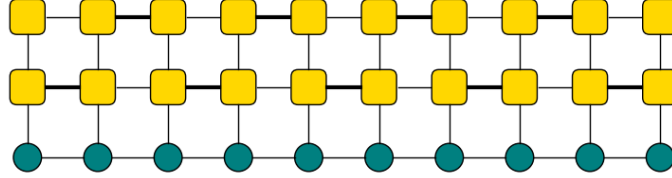


Figure 3.3.2. A complete first-order Trotter time step, encompassing both odd and even bonds, is depicted. The thick and thin lines represent MPO bonds with dimensions greater than 1 and equal to 1, respectively. In the top row, the MPO at the first and last sites correspond to trivial scalar identity operators, denoted as one.

To improve accuracy, a second-order Trotter decomposition is often used:

$$e^{-iH\delta t} \approx e^{-iH_{\text{odd}}\delta t/2} e^{-iH_{\text{even}}\delta t} e^{-iH_{\text{odd}}\delta t/2} + O(\delta t^3). \quad (3.3.16)$$

This reduces the per-step error to $O(\delta t^3)$ while keeping computational cost nearly the same. Higher-order decompositions can be applied in the same spirit for even greater accuracy. This improved scheme is illustrated in Fig. 3.3.3, where the odd-bond operators are applied in two half steps around a full even-bond update. This symmetric structure reduces the local error while leaving the overall computational cost essentially unchanged.

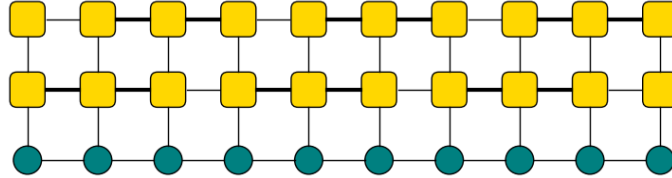


Figure 3.3.3. A complete second-order Trotter time step for a system comprising 10 sites. The time-evolution operators, expressed as MPO, feature alternating thick bonds interspersed with a single-dimensional auxiliary bond

Finally, during time evolution, bond dimensions χ typically grow rapidly. To obtain reliable results, one often extrapolates in the limits $\chi \rightarrow \infty$ or truncation error $\epsilon \rightarrow 0$, similar to procedures used in ground-state calculations.

3.3.3 Tensor Train Recursive Sketching

TTRS is a highly efficient, linear algebra-based algorithm for generative modeling, designed to approximate a high-dimensional probability distribution p using a MPS or Tensor Train. Introduced only recently [98, 99], TTRS bypasses gradient descent

entirely and provides a fast, deterministic alternative that scales linearly with the size of the training set.

Suppose the ground-truth distribution of the data is given by $p(s_1, s_2, \dots, s_N)$, which can be viewed as a large tensor:

$$\sum_d p(s_1, s_2, \dots, s_N) = 1. \quad (3.3.17)$$

This is an ordinary classical probability distribution, not a quantum probability amplitude, and all of its entries are non-negative. In practice, p is an approximation of a collection of binary d -bit strings sampled from the distribution. From these samples we can form an empirical distribution, denoted by $\hat{p}$. Intuitively, $\hat{p}$ is constructed from the available data points and reflects the observed sample frequencies. While the true distribution p is smooth (Fig. 3.3.4, green curve), the empirical distribution is often irregular and noisy, with many sharp spikes due to limited sampling.

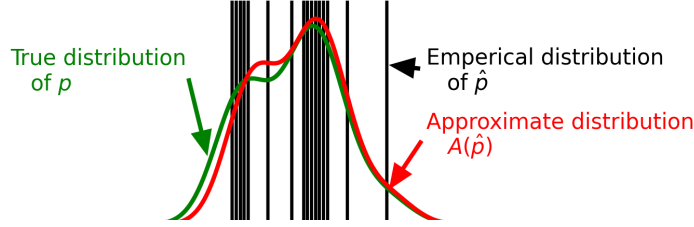


Figure 3.3.4. Empirical distribution (black vertical lines), true distribution (green line), and approximate distribution (red line). The figure illustrates how the approximation captures the main features of the true distribution compared to the sampled data.

The central idea behind TTRS is to handle the bias–variance trade-off that arises in such situations. The empirical distribution $\hat{p}$, obtained from finite data, typically suffers from high variance and undersampling gaps. This can be formally expressed as

$$p - A(\hat{p}) = (p - A(p)) + (A(p) - A(\hat{p})), \quad (3.3.18)$$

where p is the true distribution, $A(\hat{p})$ is the model approximation of the empirical distribution, and $A(p)$ is the approximation of the true distribution within the model class. The term $p - A(p)$ corresponds to the bias, while $A(p) - A(\hat{p})$ corresponds to the variance. The goal is to minimize the overall error $p - A(\hat{p})$ by striking a balance between bias and variance, thereby reconstructing a faithful approximation of p .

In practice, broadening the approximation (e.g., smoothing the empirical distribution with Gaussian kernels) reduces variance at the cost of increased bias, but this bias generally drives the model toward smoother and more generalizable functions.

TTRS incorporates this idea through sketching, which can be thought of as intentionally blurring the empirical distribution to suppress sampling noise while preserving the essential structure of the data. To better understand how the algorithm works, let us consider a concrete example. Suppose we want to reconstruct a distribution $p(s_1, s_2, \dots, s_6)$ where each s_i is a binary variable. The TTRS algorithm proceeds iteratively, building up the MPS one tensor at a time.

We begin by separating the first element s_1 from each sample and applying a sketching operation over the remaining variables $(s_2, s_3, \dots, s_6)$, as illustrated in

figure 3.3.5. The result of this procedure, aggregated over all samples, produces a tensor ϕ_1 .

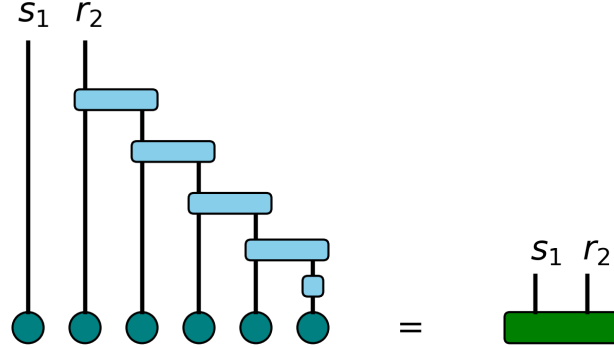


Figure 3.3.5. Illustration of the first step of TTRS. The gray boxes represent the samples $(s_1, s_2, \dots, s_6)$. The first index s_1 is isolated, while the remaining variables are sketched together to form the tensor ϕ_1 , which is then decomposed via SVD.

This tensor can be arranged as a matrix with indices (s_1, r_2) , where r_2 is a right-sketch index that compresses the contribution of the remaining variables. Applying SVD to this matrix yields two factors; the left factor is stored as the first MPS tensor U_1 , while the remainder can be discarded.

The process is then repeated for the second variable s_2 . In this case, the tensor ϕ_2 has indices (l_1, s_2, r_3) , where l_1 is the left-sketch index originating from the contraction with U_1 , and r_3 is a new right-sketch index representing the contribution of $(s_3, \dots, s_6)$ (see figure. 3.3.6). By reshaping ϕ_2 into a matrix and applying SVD, the relevant factor associated with the physical index s_2 is extracted and stored as the second MPS tensor U_2 . Again, the remaining components are discarded.

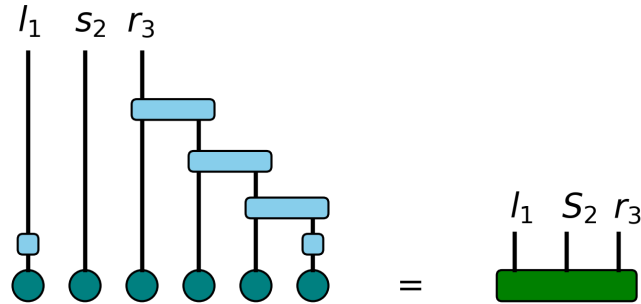


Figure 3.3.6. Illustration of the second step of TTRS. The tensor ϕ_2 is constructed with indices (l_1, s_2, r_3) , where l_1 is the left-sketch index from the first step and r_3 encodes the remaining variables.

This recursive sketching and decomposition procedure continues for all subsequent variables $(s_3, s_4, \dots, s_5)$. At each step, one physical index is isolated, the corresponding tensor ϕ_k is constructed with left and right sketch indices, and SVD is applied to extract the next MPS tensor U_k . For the final index s_6 , no further decomposition is required, since all preceding tensors $(U_1, U_2, \dots, U_5)$ have already

been determined. The last tensor is directly obtained from ϕ_6 .

By the end of this recursive, window-sliding procedure, we obtain the full MPS representation $(U_1, U_2, \dots, U_6)$ which provides an efficient approximation of the true distribution $p(s_1, s_2, \dots, s_6)$. Importantly, the use of SVD at each step not only ensures stability but also adaptively selects the bond dimension according to the singular value spectrum, thereby tuning the model complexity to the underlying data.

Through this systematic sketching and decomposition process, TTRS constructs an MPS that captures the essential structure of the probability distribution rather than overfitting to the noisy empirical samples, making it both a computationally efficient and theoretically sound approach to high-dimensional generative modeling.

3.3.4 Tensor Cross Interpolation

TCI is a modern tensor network algorithm that extends concepts from linear algebra and matrix decomposition to high-dimensional spaces. TCI enables the algorithm to adaptively explore and approximate functions that are too large or complex to represent explicitly. The key innovation of TCI lies in combining ID with TNs, allowing exponentially large tensors to be represented efficiently using tensor trains. By identifying the most representative entries of the tensor and interpolating the remaining ones, TCI achieves a balance of efficiency, scalability, and interpretability [102–105].

At the heart of TCI is the concept of ID from linear algebra. ID allows a matrix M with low-rank structure to be approximated using a small subset of its columns:

ID exploits the low-rank structure of a matrix M to construct an efficient approximation $M \approx CZ$, where C contains selected original columns of M , and Z is an interpolation matrix that reconstructs the remaining columns as linear combinations of those in C . ID is highly interpretable because each column in C corresponds to an actual column of the original matrix, unlike SVD, where the basis vectors are linear combinations that are harder to interpret.

For example, consider a movie-rating matrix where rows are movies and columns are users. Even with thousands of movies and millions of users, a small number of representative users can suffice to approximate all ratings. This is because user preferences often cluster around a few dominant patterns (genres), making the matrix effectively low-rank.

TCI extends the concept of ID from matrices to high-dimensional tensors. A tensor can be reshaped to a tensor train along a chosen index while treating all remaining indices as a combined column space. Applying ID to this matrix identifies the most informative slices of the tensor, which are preserved, while the remaining slices are reconstructed through interpolation.

This procedure can be applied recursively, gradually decomposing the high-order tensor into a sequence of lower-order components, thereby constructing a MPS representation. The process can be summarized as follows:

1. Separate one physical index from the rest of the tensor.
2. Apply ID to select the most representative columns (slices) of the resulting matrix.

3. Record the corresponding indices (pivot sets) of the selected columns for later reconstruction.
4. Incorporate the next index and repeat the process recursively, thereby compressing the tensor step by step.

After processing all indices, the tensor is represented as a chain of third-order tensors, each with two internal (bond) dimensions and one physical dimension. This MPS efficiently captures the structure of the original tensor using polynomially fewer parameters, while exactly reproducing the function values corresponding to the selected pivot entries.

To illustrate the procedure, consider a tensor with $n = 6$ indices, each of dimension $d = 2$. In the first step, we separate the last index s_6 from the rest and compute the ID. Since this is the initial step, we retain both columns of the resulting matrix $[1, 2]$.



Figure 3.3.7. First application of the ID separating index s_6 .

We record the selected columns corresponding to the configurations of s_6 . Next, we separate the fifth index, s_5 , and again apply ID. At this stage, there are four possible column combinations, $[[1, 1], [1, 2], [2, 1], [2, 2]]$, but only two are retained, assuming that these are the most informative.



Figure 3.3.8. Second application of ID, separating index s_5 and recording (s_5, s_6) .

The process continues in the same manner. When separating index s_4 , we again apply ID, resulting in four possible columns, $[[1, 2, 1], [1, 2, 2], [2, 2, 1], [2, 2, 2]]$, of which we retain three as the most significant.



Figure 3.3.9. Third application of ID, separating index s_4 and recording (s_4, s_5, s_6) .

This recursive process is repeated for all remaining indices until the full decomposition is achieved.

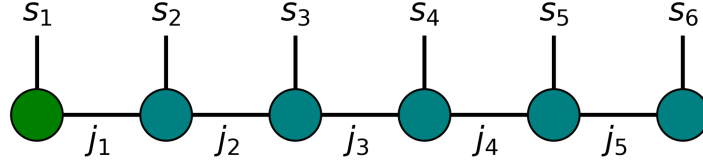


Figure 3.3.10. Final stage of recursive ID application, completing the decomposition across all indices.

At the end of this procedure, the tensor is expressed in a compressed form that can be used to reconstruct or predict values of the original high-dimensional function. Unlike traditional machine learning methods that rely on a fixed dataset, TCI treats the tensor as a function $f(s_1, s_2, \dots, s_n)$ that can be queried adaptively. During the decomposition process, the algorithm actively selects and evaluates the function at specific indices determined by the pivot sets, using these values to iteratively refine its approximation of the full tensor. Once the tensor network is constructed, it can be used to reconstruct or predict values at arbitrary configurations. For instance, based on the example above, for entries that were directly selected as pivots, such as $f(221122)$, the reconstruction is exact, whereas for configurations not included among the pivots, such as $f(111111)$, the corresponding values are estimated through interpolation within the learned tensor network. To learn and generate new tensors between these configurations, we follow this procedure.

1. Initialize the MPS with guessed pivot sets.
2. Contract adjacent tensors and evaluate the original function at selected indices.
3. Perform ID on the newly evaluated matrix, updating column/row pivots.
4. Repeat by sweeping left-to-right and right-to-left across the tensor, updating the MPS at each step.

This iterative procedure is similar to machine learning, as the algorithm adaptively improves its approximation based on new information from the function. The result is a highly efficient and interpretable approximation of the original function.

Let us now illustrate this process with a continuation of the previous ID example. Suppose we begin by contracting the first two tensors:

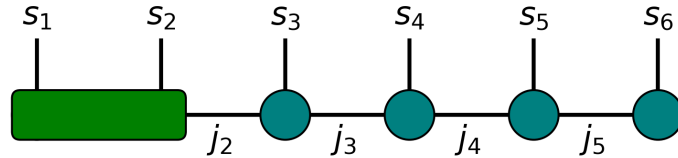


Figure 3.3.11. First step to of ML part. Contraction between first two tensors for interpolation process.

Ideally, the local tensor representing the contraction of the first two tensors should approximate the original tensor on entries $(s_1, s_2, j_2, \dots)$. However, at the

initial stage, this approximation is relatively poor. To improve it, we construct a corrected pair tensor by filling in the exact values of the function $f(s_1, s_2, j_2, \dots)$. We then compute the ID as shown below:



Figure 3.3.12. Compute ID between first two tensors and move to next two tensors.

After this step, the state of the MPS takes the following form:

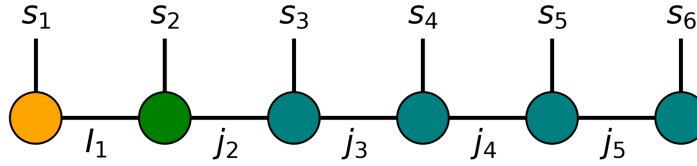


Figure 3.3.13. Formation of the MPS after first ID.

Next, we continue by merging the subsequent pair of tensors:

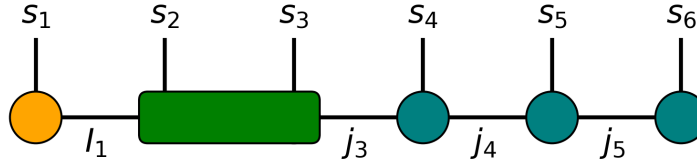


Figure 3.3.14. Second step to of ML part. Contraction between tensors two and three for interpolation process.

We again apply the ID to this contraction. After a few forward and backward sweeps, the quality of the interpolation rapidly improves and converges, as illustrated below:

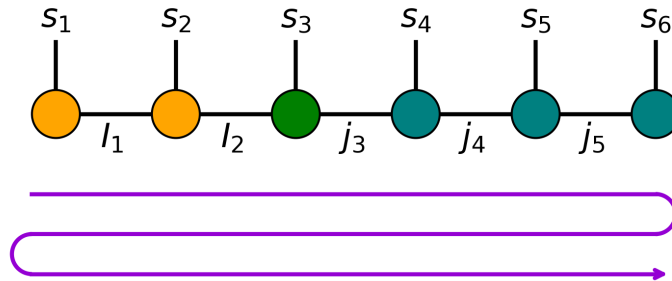


Figure 3.3.15. Forward and backward sweeping through the dataset to learn interpolation. This process updates the tensors sequentially, allowing the model to capture the underlying structure of the data.

TCI requires remarkably few function evaluations relative to the size of the full tensor. For a tensor with D dimensions and bond dimension χ , the algorithm performs approximately $D\chi^2$ function evaluations, significantly fewer than the 2^D entries of the full tensor.

Additional optimizations, such as rook pivoting, further reduce the computational cost by selectively evaluating only those rows and columns that contribute the largest residual errors.

Overall, TCI serves as a powerful bridge between classical numerical linear algebra and modern machine learning, providing an efficient and interpretable framework for learning and compressing high-dimensional data.

3.4 Applications of Tensor Networks

TNs were originally developed as a powerful tool for representing wavefunctions in many-body quantum systems, where they offered an efficient way to capture entanglement and correlations that would otherwise be intractable. While this application has long been the cornerstone of TNs methods, recent years have witnessed a surprising expansion of their scope. Beyond the quantum realm, TNs are now being explored in two exciting new directions: the representation of continuum functions and applications in machine learning. These emerging frontiers not only broaden the mathematical versatility of TNs but also reveal their potential as a unifying language across physics, mathematics, and data science.

3.4.1 Continuum Function

An intriguing application of TNs arises in the representation of continuous functions. The natural question we can ask is: can a continuous function be mapped to TNs, and if so, how can we encode it? More concretely, we would like to understand how to represent a function as a tensor, and what techniques allow us to input continuous variables into TNs. Of course, one has to keep in mind a crucial requirement: TNs are efficient only when the underlying structure exhibits low entanglement. Thus, even if such a mapping is possible, it is only meaningful for functions that can be represented with low entanglement.

To illustrate this idea, consider a one-dimensional function defined on the interval $[0, 1)$. We discretize the domain using a fine binary grid with spacing $(1/2^n)$. Each point on this grid can be expressed as a binary fraction

$$x \approx \frac{b_1}{2} + \frac{b_2}{2^2} + \frac{b_3}{2^3} + \cdots + \frac{b_n}{2^n} \approx 0.b_1b_2b_3 \cdots b_n, \quad (3.4.1)$$

where each $b_j \in \{0, 1\}$. On this discretized grid, the function values are given by

$$f\left(\frac{b_1}{2} + \frac{b_2}{2^2} + \cdots + \frac{b_n}{2^n}\right), \quad (3.4.2)$$

which can be naturally arranged into a tensor

$$F^{b_1b_2b_3 \cdots b_n}, \quad (3.4.3)$$

with entries corresponding to the sampled values $f(0.b_1b_2b_3 \cdots b_n)$. (see figure 3.4.1)



Figure 3.4.1. Definition of a continuous function using a collection of tensors. A general high-order tensor is reshaped into a tensor network, illustrating how the function can be represented and efficiently encoded via a tensor network structure.

This perspective connects directly to the idea of amplitude encoding in quantum computing: a function defined on a binary grid can be embedded into a quantum state of the form

$$|F\rangle = \sum_b F^{b_1b_2b_3 \cdots b_n} |b_1b_2b_3 \cdots b_n\rangle. \quad (3.4.4)$$

The essential question then becomes: for a given function $f(x)$, does the tensor F admit a low-rank TNs decomposition? In other words, does it correspond to a “low-entanglement” structure? If the answer is yes, then we can represent and manipulate such functions efficiently with TNs; otherwise, the cost becomes exponential.

Interestingly, many natural functions fall into a “classically simulable regime,” where the effective entanglement remains low. This makes TNs not only a practical tool for representing continuous functions in high-dimensional spaces, but also a foundation for a remarkably versatile class of algorithms. Beyond efficient representation, tensor-network methods enable the solution of partial differential equations, the integration of tensorized functions, and structured implementations of the quantum Fourier transform. They also support optimization strategies such as extremum finding without explicit gradients, as well as interpolation, extrapolation, and even exact sampling. These applications highlight both the mathematical elegance and the practical power of tensor-network formulations, which we explore in the following sections.

As a starting point, consider the case of PDEs. Even a simple example, such as the wave equation,

$$\frac{\partial^2}{\partial x^2} f(x) = -k^2 f(x), \quad (3.4.5)$$

can be naturally recast as a tensor-network equation (see Fig. 3.4.2). Here, the differential operator $\partial^2/\partial x^2$ is encoded as a MPO, transforming the PDE into an eigenvalue-like problem that can be solved efficiently using optimization algorithms such as DMRG. Importantly, DMRG is agnostic to the operator’s physical origin, whether it represents a quantum Hamiltonian or a differential operator, highlighting the unifying power of tensor-network approaches.

$$\frac{\partial^2}{\partial x^2} f(x) = -k^2 f(x)$$

Figure 3.4.2. Encoding a simple wave equation as a TNs: the differential operator $\frac{\partial^2}{\partial x^2}$ is expressed as a MPO, allowing the PDE to be solved using algorithms such as DMRG.

The application of TNs to partial differential equations (PDEs) has been investigated from several complementary perspectives. Khoromskij [87] laid the theoretical foundations of tensor numerical methods by demonstrating that many classes of PDE operators admit low-rank tensor-train representations, thereby mitigating the curse of dimensionality and enabling solvers with polynomial rather than exponential scaling. Building on this idea, Shinaoka et al. [110] introduced a multiscale space-time ansatz based on quantum tensor trains to efficiently represent correlation functions in quantum systems, achieving accurate treatment of both local and non-local features that are inherently PDE-like in structure. In a related but distinct direction, Lubasch et al. [111] combined TNs concepts with multigrid methods to propose a multigrid renormalization framework, where hierarchical coarse-graining yields scalable and rapidly convergent solvers for PDEs. Together, these works highlight the versatility of tensor-network approaches in addressing high-dimensional PDE problems, from theoretical foundations to quantum-inspired algorithms and hybrid multiscale methods.

Next, we consider how TNs can be used to efficiently perform integration over a function $f(x)$. Consider the integral of $f(x)$ over the interval $[0, 1]$, which can be approximated by a Riemann sum on a binary grid of spacing $1/2^n$:

$$\int_0^1 dx f(x) \approx \sum_{b_1 b_2 \dots b_n} \frac{1}{2^n} f\left(\frac{b_1}{2} + \frac{b_2}{2^2} + \dots + \frac{b_n}{2^n}\right). \quad (3.4.6)$$

Using the tensor representation of the function $F^{b_1 b_2 \dots b_n}$ introduced earlier, this sum can be expressed as

$$= \sum_{b_1 b_2 \dots b_n} \frac{1}{2^n} F^{b_1 b_2 b_3 \dots b_n}, \quad (3.4.7)$$

where each index of the tensor F is contracted with the vector $[1/2, 1/2]$. By performing a SVD on F , one can convert it into a MPS as illustrated in Fig. 3.4.3. In practice, the integration proceeds sequentially: the first tensor is contracted with the vector, producing an intermediate vector, which is then contracted with the next tensor, and so on, until the final contraction yields a scalar. This procedure effectively “decimates” the network from left to right (or right to left), and results in a significant computational advantage: the cost of integration scales linearly with the number of indices n as $\mathcal{O}(n\chi^2)$, or equivalently $\mathcal{O}(\chi^2 \log N)$ in terms of the total number of grid points $N = 2^n$, where χ is the bond dimension of the MPS.

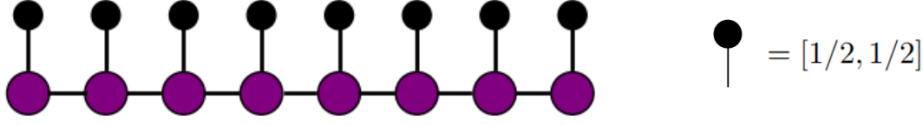


Figure 3.4.3. Integration of a tensorized function using a MPS. Each tensor index is contracted sequentially with a vector representing the grid weights $[1/2, 1/2]$, reducing the network step by step until the integral is obtained as a scalar.

Building on the tensorized integration approach, Khoromskij [112] introduced the quantics tensor approximation for high-dimensional tensors, providing an efficient framework for numerical integration over large-dimensional domains. By representing discretized functions as low-rank quantics tensors, this method drastically reduces the computational complexity, allowing integrals to be evaluated with cost scaling as $\mathcal{O}(d \log N \log \varepsilon^{-1})$, where d is the dimension, N the number of grid points per dimension, and ε the desired accuracy. This approach exploits the fact that many functions of practical interest, such as exponentials or polynomials on uniform grids, admit very low-rank quantics representations, enabling high-dimensional integration that would otherwise be intractable. In this way, TNs not only encode the function efficiently but also accelerate the evaluation of integrals in a systematic and scalable manner.

3.4.2 Quantum Fourier transform

Apart from continuous functions, we also encounter discrete functions, which can be represented within finite-dimensional vector spaces and conveniently mapped to TNs. In this discrete framework, one of the most fundamental mathematical tools is the Discrete Fourier Transform (DFT). The DFT provides a change of basis that expresses a sequence of values in terms of its frequency components. For a sequence $x = [x_0, x_1, \dots, x_{N-1}]$, the DFT is defined as

$$X_k = \sum_{n=0}^{N-1} x_n e^{-2\pi i k n / N}, \quad k = 0, 1, \dots, N-1. \quad (3.4.8)$$

As a simple example, if we consider $x = [1, 0, 0, 0]$ with $N = 4$, the result is $X = [1, 1, 1, 1]$, which illustrates that a localized impulse in the time domain corresponds to a uniform distribution across all frequencies.

The Quantum Fourier Transform (QFT) is the natural quantum analogue of the DFT. Instead of operating on classical sequences, it acts on the probability amplitudes of a quantum state. For an n -qubit state

$$|\psi\rangle = \sum_{x=0}^{N-1} a_x |x\rangle, \quad N = 2^n, \quad (3.4.9)$$

the QFT is defined by its action on the computational basis states,

$$f_k = \sum_{j=0}^{N-1} e^{2\pi i k j / N} f_j. \quad (3.4.10)$$

By linearity, this transformation extends to arbitrary superpositions. For example, in a two-qubit system ($N = 4$), applying the QFT to $|0\rangle$ yields

$$|0\rangle \mapsto \frac{1}{2}(|0\rangle + |1\rangle + |2\rangle + |3\rangle), \quad (3.4.11)$$

demonstrating that a single basis state transforms into an equal superposition over all possible states, closely mirroring the behavior of the DFT on a classical impulse sequence.

In this formulation, f_j represents the input quantum register, the exponential phase factors $\exp(2\pi i k j / N)$ define the unitary transformation block, and the output register encodes the transformed state (see Figure 3.4.4).

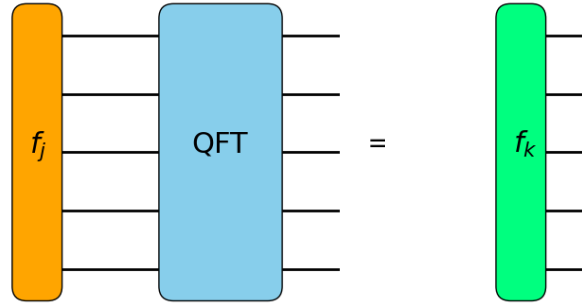


Figure 3.4.4. Quantum Fourier Transform (QFT) representation.

Importantly, the QFT block itself can be realized through different types of quantum circuits, depending on the chosen encoding strategy. Recent results by Chen et al. [205] show that the QFT exhibits surprisingly limited entangling power: aside from the bit-reversal step, its Schmidt coefficients decay exponentially across any bipartition, implying that the circuit generates only a constant amount of entanglement independent of system size. This structural property allows the QFT to be efficiently represented as a MPO, where each layer and column of the circuit can be viewed as a tensor contracted with its neighbors. Remarkably, a small and essentially constant bond dimension is sufficient to capture the QFT, enabling efficient simulation on MPS with computational resources scaling linearly in the number of qubits. This perspective highlights not only the structural simplicity of the QFT compared to other highly entangling circuits but also provides a natural bridge between quantum algorithms and TNs methods. (see Figure 3.4.5).

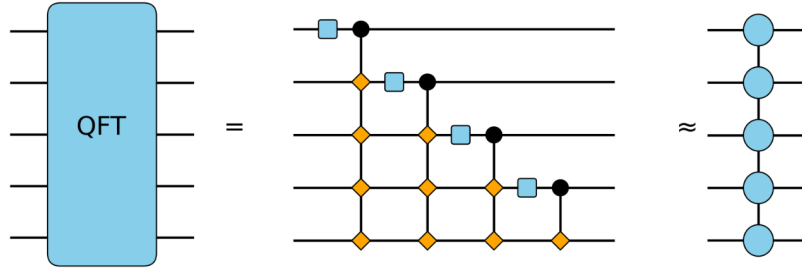


Figure 3.4.5. Mapping the QFT circuit to TNs representation MPO.

The QFT can be efficiently encoded within TNs frameworks, offering both a deeper theoretical understanding of its internal structure and practical advantages for classical simulation and algorithmic analysis.

3.4.3 Interpolation and Extrapolation

Interpolation can also be carried out efficiently within the tensor-network framework. As shown by García-Ripoll [206], the key idea is to represent a discretized multivariate function as a MPS, where the smoothness of the function ensures that the underlying representation has low entanglement and therefore remains compact. Instead of storing or computing function values on a full grid, the MPS provides a compressed description from which intermediate points can be reconstructed. In practice, one samples the function on a coarse set of points, builds the MPS approximation, and then uses the network to interpolate values at finer resolutions. This makes interpolation not only efficient in memory and computation but also naturally compatible with further tensorized operations such as differentiation, Fourier transforms, or integration.

Extrapolation has also been reimaged through tensor-network techniques. Lin and White [207] propose an entropy-based approach in which function samples are encoded into a tensor-network state, and the continuation of the sequence is determined by selecting the value that minimizes the entanglement entropy of the resulting representation. The guiding principle is that many smooth or structured functions admit low-entanglement tensor descriptions, so the correct extrapolation is the one that preserves this property. Iterating this procedure yields a predictive scheme that is particularly effective for functions with sharp features or non-analytic behavior, where classical linear prediction often fails.

3.4.4 Supervised Learning

In the supervised learning setting, TNs provide a principled framework for building expressive high-dimensional representations of data. One of the earliest and most influential works in this direction was carried out by Stoudenmire and Schwab [100]. Their approach begins by mapping each input variable, for instance, the intensity of an image pixel X_i , into a local feature vector, such as $[1, X_i]$ or $[\cos(X_i), \sin(X_i)]$. This step introduces a nonlinear feature expansion reminiscent of classical kernel

methods. The global feature map $\Phi(X)$ is then formed as the tensor product of all local feature vectors, and the decision function is defined through its contraction with a weight tensor W . This construction implicitly encodes an extremely high-order polynomial expansion, allowing the model to naturally capture complex correlations among input variables.

Because the weight tensor grows exponentially with the input dimension, it must be stored and optimized in compressed form. This is achieved by representing W as a MPS, where the label index is assigned to one of the MPS tensors without loss of generality. Training is then performed by minimizing a supervised cost function via gradient-based methods, often in combination with alternating local updates sweeps that exploit the MPS structure for computational efficiency over all N samples,

$$C = \frac{1}{N} \sum_{j=1}^N (f(x_j) - y_j)^2. \quad (3.4.12)$$

Here, N denotes the number of training examples, x_j the j -th input sample (e.g., j -th pixel in an image), $f(x_j)$ the scalar output of the tensor-network model for that sample, and y_j the corresponding target label.

This tensorized formulation has been shown to achieve competitive accuracy, notably surpassing 99% on the MNIST dataset, demonstrating that low-entanglement TNs can serve as effective machine learning models. At the same time, practical deployment requires relatively large bond dimensions (e.g., $\chi \approx 120$ for MNIST), and the computational cost scales with both bond dimension and dataset size, presenting challenges for scaling to more complex tasks.

To overcome the combinatorial explosion of explicit high-order outer products, Dilip et al. [208] encode classical images in an amplitude-style quantum state in which pixel locations are stored in $\approx \log N$ address qubits (the computational basis labels act collectively as binary indices) and pixel intensity in a color qubit. The resulting amplitude vector is then compressed into a tensor-train (matrix-product state, MPS) representation whose physical dimension is that of a qubit and whose bond dimension χ controls fidelity (and, via the MPS that map to circuit, circuit depth). By additionally splitting images into patches (to trade qubit count against depth) and by using a hardware-efficient sequential two-qubit circuit ansatz, they obtain a tunable, low-dimensional representation that can be implemented and trained on well-known hardware.

A related approach is presented in our recent project, where images are encoded directly as MPS and used in combination with a support vector machine for supervised image recognition. This methodology demonstrates how the MPS representation can be leveraged for practical classification tasks, and its implementation and results are discussed in detail in Chapter 8.

3.4.5 Gradient-Free Optimization in the Tensor Train Format

A notable application of TNs in optimization was introduced by Chertkov et al. [209], who developed a method to locate extrema (maximum or minimum values) of a function represented in the Tensor Train (TT) format. Unlike traditional approaches

that rely on gradient descent, their method is entirely gradient-free and instead exploits the block structure of the TT decomposition.

The key idea is to reinterpret the squared entries of the tensor as forming a probability distribution over all possible multi-indices. With this probabilistic view, searching for the extremum becomes equivalent to identifying the most “likely” index under this distribution. Instead of evaluating the full tensor, which would be computationally infeasible in high dimensions, the algorithm proceeds sequentially through the TT cores (blocks), keeping track of only the most promising candidates at each step. This procedure resembles a *beam search*: rather than exploring every combination of indices, the method prunes the search space aggressively, retaining only the top K candidates as it moves from one block to the next.

The optimization proceeds as follows:

1. First core: Compute the marginal probabilities over the indices of the first core. Select the top K values with the largest probabilities.
2. Subsequent cores: For each surviving candidate from the previous step, extend it by combining with all possibilities from the next core. Compute the updated probabilities, then prune the list again to the top K .
3. Final core: After the last core, only a small set of candidates remain. From these, the index corresponding to the largest (or smallest) value is chosen as the extremum.

This step-by-step strategy ensures that optimization is done *block by block* rather than on the entire tensor at once, making it computationally feasible and avoiding the need for gradients.

To illustrate, consider a tensor $Y(i_1, i_2, i_3)$ with three dimensions, each index taking values in $\{1, 2, 3\}$.

- Step 1 (Core 1): Compute marginal probabilities for i_1 . Suppose

$$P(i_1 = 1) = 0.2, \quad P(i_1 = 2) = 0.5, \quad P(i_1 = 3) = 0.3.$$

With $K = 2$, keep $i_1 = 2$ and $i_1 = 3$.

- Step 2 (Core 2): Extend these candidates with all values of i_2 . Suppose the best pairs are $(2, 2)$ and $(3, 3)$.
- Step 3 (Core 3): Extend further with all i_3 . Suppose the best full index is $(3, 3, 1)$.

Thus, the algorithm identifies $(3, 3, 1)$ as the most probable index, corresponding to the maximum entry of the tensor.

This method demonstrates that TNs are not only useful for compressing and representing high-dimensional objects, but also provide scalable optimization tools. By exploiting the block structure of TT and working sequentially rather than globally, it becomes possible to search for extrema in spaces that are otherwise computationally intractable. The approach illustrates how TNs can go beyond representation and play an active role in gradient-free optimization of high-dimensional functions.

3.5 Summary and Outlook

In this chapter, we reviewed the foundations, algorithms, and applications of TNs, emphasizing their power as a unifying framework for representing and manipulating high-dimensional data.

We began with the preliminaries and general principles, where Penrose graphical notation provided an intuitive visualization of tensors and their contractions. Key building blocks such as SVD, boundary conditions (open and periodic), translational invariance, and the area law were discussed as essential concepts for TNs methods. The MPS formalism was introduced in detail, including procedures for MPS–MPS contraction, evaluating local expectation values, summation of MPS, and working with MPO. The interplay between MPS and MPO was further explored through contractions, expectation values, and derivatives.

We then turned to TNs algorithms, focusing on their role in simulating quantum many-body systems and beyond. The DMRG was presented as a central method for determining ground states of strongly correlated systems. The TEBD algorithm was described for simulating quantum time evolution. In addition, we introduced TTRS as a generative modeling approach that extends the reach of TNs into modern data-driven tasks.

Finally, we addressed diverse applications of TNs across physics, mathematics, and machine learning. Topics included representing continuum functions, solving partial differential equations, performing high-dimensional integration, and implementing the quantum Fourier transform. TNs were also shown to be effective tools for interpolation and extrapolation, supervised learning tasks, and gradient-free optimization within the tensor train format.

Overall, this chapter demonstrated how TNs serve as both a conceptual and computational bridge between quantum physics, numerical analysis, and machine learning, providing scalable techniques for representing, compressing, and manipulating high-dimensional structures.

Chapter 4

Phase Diagram of Many-Body Quantum Models

Phase diagrams of quantum matters provide a powerful way to understand how many-body quantum systems change between different phases, from ordered to disordered states or from conventional to topological order. Mapping these diagrams is challenging, since exact analytical solutions exist only for a few simple cases. Classical simulations have therefore become essential tools not only to explore quantum phases and transitions, but also to benchmark emerging quantum computers. In this chapter, we review the main simulation approaches used in this context, including Order Parameters and phase indicators, Exact Diagonalization, Quantum Monte Carlo, and Tensor Networks methods, highlighting their strengths and limitations. We then turn to three representative models that have played a central role in shaping our understanding of quantum matter: the Axial Next-Nearest Neighbor Ising model, the Cluster-Ising model, and the Haldane spin-1 anisotropic chain. Each of these models illustrates different aspects of frustration, entanglement, and symmetry-protected order, while also serving as testbeds for advanced computational methods. Together, reviewing these methods and models provide a comprehensive framework for investigating the rich and complex landscape of quantum phases and their transitions.

4.1 Introduction

Studying the phase diagrams of many-body quantum systems is like exploring a vast and intricate landscape. These diagrams reveal how a system evolves from one phase to another, ordered to disordered, simple to topologically special, when interactions, external fields, or other parameters change. Understanding these behaviors is important not only for fundamental physics but also for designing new quantum materials, devices, and for benchmarking emerging quantum computers. As quantum systems grow more complex, knowing where and how these changes occur helps us predict behavior and guide experiments. In this chapter, we review key studies that introduced foundational many-body models, examined their phase behavior, and developed modern simulation techniques. By synthesizing these insights, we aim to provide a coherent picture of how different models behave, how phases are classified,

and how computational tools help map these complex quantum landscapes.

Simulations play a central role in this exploration. Interacting quantum particles exhibit emergent phenomena, topological order, entanglement, phase transitions, and complex dynamics, that are nearly impossible to solve exactly. In the early days, only a few simple models, like certain spin chains or integrable systems, could be fully understood. Today, however, simulations have become indispensable: they allow us to probe system sizes and parameter regimes far beyond analytical reach. Techniques such as order parameters [127–133] ED [134–142, 152–155], Monte Carlo methods [143–151, 156, 210, 211], and TNs provide quantitative insights into ground states, excitation spectra, and phase boundaries.

Another modern motivation for simulation comes from the rise of quantum computers. These devices promise to tackle problems beyond classical reach, including the simulation of complex many-body systems. But before we can fully trust their output, classical simulations remain essential to benchmark quantum devices, estimate errors, and test silicon algorithms. In this way, many-body simulations serve a dual role: they challenge quantum devices while providing a testing ground to demonstrate their capabilities. As Benedikt Fauseweh [212] emphasizes, simulating quantum systems is not just a potential application for the future; it is a key task shaping the development of quantum processors today. Even without full “quantum advantage,” current devices are already applied to model Hamiltonians, phase diagrams, and dynamical problems that push classical computation to its limits.

This chapter focuses on three models that have become central to the study of many-body quantum systems: the Axial Next-Nearest Neighbor Ising (ANNNI) model, the Cluster–Ising model, and the Haldane spin-1 anisotropic model. These models capture a wide range of fascinating quantum phenomena and provide a rich playground for exploring different phases. The ANNNI model, with its competing interactions, illustrates frustration and modulated ordering; the Cluster–Ising model highlights how non-local interactions can generate symmetry-protected topological phases; and the Haldane spin-1 anisotropic model exemplifies symmetry-protected topological order in integer spin chains.

The ANNNI model, introduced by Elliott [157], shows how simple spin interactions can produce surprisingly rich behavior. Classical studies using cluster variation, transfer matrices, and renormalization-group techniques revealed its complex thermal phase diagram [158–161]. Quantum extensions, analyzed through Trotter–Suzuki mapping, Monte Carlo [162], and ED [163, 213], uncovered ferromagnetic, antiphase, floating, and disordered phases. Modern numerical methods, including DMRG and infinite-MPS [165–167], and recent QML approaches [168, 169] provide high-precision maps of these phases, showing how even sparse data can reconstruct the global phase diagram.

The Cluster–Ising model offers a complementary perspective, combining conventional and topological order. Even in one dimension, it exhibits competition between antiferromagnetic order and cluster-type interactions, giving rise to highly entangled, symmetry-protected phases. Analytical studies [171–175] identified cluster and antiferromagnetic regimes and critical points beyond the Landau framework. Computational methods, including TEBD, ED, and DMRG [176–179], refined the phase diagrams, revealed dynamical signatures, and showed robustness under dis-

order. More recently, QML 2 and experimental simulations [180, 181] demonstrate that even sparse or noisy data can capture topological phases.

Finally, the Haldane spin-1 anisotropic model explores chains of spin-1 particles with competing interactions. Haldane’s 1983 conjecture [190, 191] predicted a finite gap, the Haldane gap, highlighting hidden symmetry-protected topological order. Introducing single-ion anisotropy produces a rich landscape of Néel, large-D, and Haldane phases [128], which early ED and QMC studies [192, 193] helped map. TNs methods, particularly DMRG [194, 195], revealed entanglement fingerprints of the Haldane phase and precisely located transitions. Even small uniaxial or rhombic perturbations can destabilize the Haldane state, giving rise to exotic phases like the y -Néel state [196, 197].

In this chapter, we aim to provide a comprehensive exploration of many-body quantum phase diagrams. We begin with section 4.2, Simulation Method Overview, presents the computational and analytical tools, order parameters, ED, QMC, and TNs, that underpin our analysis. We then examine the three model systems in detail: the ANNNI model 4.3, highlighting frustration and modulated phases; the Cluster-Ising model 4.4, showcasing conventional and topological order; and the Haldane spin-1 anisotropic model 4.5, illustrating symmetry-protected topological phases and entanglement signatures. Finally, the chapter concludes with an outlook, reflecting on the insights gained and pointing toward future directions in quantum phase identification and simulation 4.6. Together, these sections provide both the conceptual framework and technical tools needed to navigate the rich landscape of quantum phases.

4.2 Simulation Methods Overview

In this section, we provide an overview of the key numerical tools used to study many-body quantum systems. These methods allow us to extract phase boundaries, characterize order, and probe subtle quantum correlations that are otherwise inaccessible. We focus on four approaches that have proven particularly effective: order parameters and phase indicators, which reveal the presence and nature of different phases; ED, providing precise solutions for small systems; QMC, enabling statistical exploration of larger lattices; and TNs methods, which capture entanglement efficiently and give access to ground states and phase transitions. Each method brings unique strengths and limitations, which are reviewed separately.

4.2.1 Order Parameters and Phase Indicators

Order parameters provide the most fundamental and widely adopted tool for distinguishing quantum phases of matter at zero temperature. Within the framework of numerical simulations, they serve as direct probes of ground-state structure: by measuring an appropriate order parameter, one can determine whether the system resides in a trivial product phase, a symmetry-breaking ordered phase, or a nonlocal topological phase. In this sense, order parameters are the practical language through which a phase diagram is decoded. They are not only theoretical constructs but also concrete diagnostics that can be extracted from the wavefunctions produced by computational methods.

At the most general level, an order parameter is a quantity that takes a characteristic value inside one phase but vanishes (or changes qualitatively) upon entering another. By scanning order parameters across a range of Hamiltonian parameters, one can systematically delineate phase boundaries. The central role of order parameters is particularly evident in quantum spin chains, where competing exchange interactions and anisotropies give rise to a variety of ground states. Because different phases are defined either by broken symmetries or by hidden topological structures, no single observable suffices; instead, one must employ a hierarchy of order parameters, ranging from purely local to genuinely nonlocal correlators.

The simplest phases arise when the ground state reduces to a product of single-site states. These so-called trivial phases neither break symmetry nor host hidden long-range order, so conventional dipolar order parameters vanish. Instead, they can be identified through local quadrupolar expectation values, which indicate which single-site spin eigenstate is energetically selected. While not order parameters in the strict Landau sense, these quadrupolar quantities serve as practical diagnostics of trivial product states in numerical simulations [127–130].

Conventional ordered phases are characterized by local order parameters, such as staggered magnetization, which captures alternating spin alignment along a chosen axis. A nonzero value signals spontaneous breaking of spin-rotation and lattice-translation symmetries. Following the Landau paradigm, these order parameters vanish continuously at quantum phase transitions [131].

By contrast, some phases avoid detection by local observables altogether. In topological phases, the relevant order parameters are nonlocal and are commonly referred to as string order parameters, as they capture hidden correlations that local probes cannot detect. The Haldane phase, for instance, exhibits hidden long-range correlations that are invisible to conventional probes [132]. Here, a nonlocal string order parameter reveals the underlying topological pattern while filtering out local quantum fluctuations. Rather than indicating symmetry breaking, the string order encodes the topological character of the phase, which is protected by discrete symmetries such as inversion or time reversal. In numerical simulations, measuring string order provides a clear diagnostic for distinguishing the Haldane phase from both trivial and symmetry-breaking phases [127–129, 190].

With this hierarchy of indicators, the phase diagram can be mapped systematically. Along chosen parameter paths, quadrupolar values identify trivial phases, local order parameters diagnose symmetry-breaking phases along the corresponding spin axis, and string order reveals the topological phase. Critical points are located where these quantities vanish or change character, and results from different parameter cuts are combined to assemble the full two-dimensional phase diagram.

Example: Staggered Magnetization in the Ising Chain

Consider the one-dimensional pure Ising model without the transverse field; the central question is how to differentiate the ferromagnetic phase from the antiferromagnetic one.

$$H = J \sum_{j=1}^{N-1} S_j^z S_{j+1}^z, \quad (4.2.1)$$

where $S_j^z = \pm \frac{1}{2}$. The sign of the coupling constant J determines whether the ground state is antiferromagnetic ($J > 0$) or ferromagnetic ($J < 0$). For $J > 0$ and $N = 4$

sites with open boundaries, the energy is minimized when neighboring spins are antiparallel. The two degenerate ground states are

$$|\uparrow\downarrow\uparrow\downarrow\rangle, \quad |\downarrow\uparrow\downarrow\uparrow\rangle,$$

each with energy $E_0 = -\frac{3J}{4}$. The staggered magnetization, defined as

$$M_s = \frac{1}{N} \sum_{j=1}^N (-1)^j \langle S_j^z \rangle, \quad (4.2.2)$$

takes the values $M_s = \pm\frac{1}{2}$ depending on which of the two symmetry-related configurations the system occupies. Thus, M_s is a faithful order parameter for the antiferromagnetic phase. By contrast, for $J < 0$ the ground states are fully ferromagnetic,

$$|\uparrow\uparrow\uparrow\uparrow\rangle, \quad |\downarrow\downarrow\downarrow\downarrow\rangle,$$

with all spins aligned. In this case the staggered magnetization vanishes ($M_s = 0$).

This simple calculation highlights the diagnostic power of staggered magnetization. It is finite in the antiferromagnetic symmetry-breaking phase, zero in the ferromagnetic phase, and also vanishes in disordered paramagnetic phases. In this sense, M_s exemplifies the Landau paradigm of local order parameters as phase indicators. Beyond this case, numerous studies have demonstrated the broader examples of order parameters and nonlocal string order in characterizing quantum phases of matter [152–155].

4.2.2 Exact Diagonalization

ED is one of the most straightforward yet powerful numerical methods for solving quantum many-body problems. The basic principle is familiar from elementary quantum mechanics: one selects a suitable complete orthonormal basis spanning the system’s Hilbert space, constructs the Hamiltonian matrix in this basis, and then diagonalizes it fully or partially to obtain eigenvalues and eigenvectors. The term “exact” emphasizes that the solution involves no approximations, distinguishing ED from methods that yield approximate energies or wave functions. While often focused on ground-state properties, ED also provides direct access to the entire excitation spectrum and all eigenstates of the system, allowing detailed analysis of static and dynamic properties.

In quantum many-body physics, ED plays a crucial role by providing exact solutions for finite-size interacting systems, thus serving as an essential benchmark for more scalable but approximate techniques such as TNs methods and QMC simulations. This completeness enables the calculation of correlation functions, spectral properties, and entanglement measures without relying on perturbative assumptions. Consequently, ED serves as both a verification tool for novel algorithms and a primary method for exploring exact finite-size physics in regimes that are analytically intractable.

A fundamental limitation of ED is the exponential growth of the Hilbert space with system size. For a spin-half chain of length N , the Hilbert space dimension

scales as 2^N , rapidly making computations infeasible as N increases. Even on high-performance computers, accessible system sizes without exploiting symmetries typically max out around 32 spins. Hence, ED is most effective for small to intermediate system sizes and cannot directly access the thermodynamic limit.

One of the primary strategies to mitigate this limitation is exploiting symmetries of the Hamiltonian to reduce the effective Hilbert space dimension. Symmetries such as total spin conservation, translational invariance, or parity allow construction of symmetry-adapted bases that drastically reduce computational effort. Such symmetry considerations also enable analytical tools like the tower-of-states analysis, revealing characteristic signatures of ordered phases. Efficient implementation often requires specialized algorithms; for example, Lin tables provide an optimized indexing scheme for spin systems with conserved quantities, facilitating larger system simulations [134].

Because ED produces the full spectrum and eigenstates, it permits exact computation of correlation functions, entanglement entropy, spectral functions, and finite-size scaling. This makes ED a cornerstone technique in many-body physics, indispensable for studying exotic phases such as frustrated magnets, topological models, and strongly correlated fermionic or bosonic systems.

The development of scalable ED techniques traces back to seminal work by Lanczos [139], who introduced an iterative algorithm to efficiently approximate extremal eigenvalues by projecting the large Hamiltonian onto a smaller Krylov subspace. The resulting tridiagonal matrix can be diagonalized quickly, enabling practical ED on larger systems than previously possible. This foundational approach remains a workhorse for modern ED algorithms.

Subsequent studies have leveraged ED for various quantum spin models, expanding the accessible system sizes and physical insights. Parkinson and Bonner (1985) [140] employed Lanczos ED to investigate quantum-to-classical crossovers in antiferromagnetic Heisenberg chains under external fields, revealing finite-size effects and excitation spectrum features for spin values up to 2 and chain lengths up to 20 sites for spin- $\frac{1}{2}$. Similarly, Tang and Hirsch (1989) [141] studied the two-dimensional spin- $\frac{1}{2}$ Heisenberg antiferromagnet, demonstrating long-range magnetic order in finite systems without spontaneous symmetry breaking by analyzing the staggered magnetization and excitation spectra in lattices up to 26 spins.

In 1990, Lin's introduction of a sublattice-coding scheme [134] eliminated the need for costly hashing in Lanczos implementations, enabling ED of spin- $\frac{1}{2}$ systems up to 32 sites and thereby pushing the frontiers of accessible system sizes. This innovation improved the accuracy of finite-size extrapolations to the thermodynamic limit.

Despite its computational constraints, ED remains an essential benchmark. Noack and Manmana (2006) [135] review ED alongside renormalization group methods like the Numerical Renormalization Group (NRG) and DMRG, highlighting ED's unrivaled exactness on small lattices and its complementary role in quantum many-body studies.

More recently, techniques like shift-invert ED have been developed to access highly excited states in disordered spin chains of up to 26 sites [136], allowing investigations of many-body localization phenomena beyond ground-state physics. Jung et al. (2020) [142] provide a practical guide emphasizing the use of symmetries

to efficiently manage Hilbert space growth and study thermalization and dynamic properties of quantum systems.

In 2022, a comprehensive overview [137] reaffirmed ED’s importance as a reliable, unbiased method for solving quantum many-body Hamiltonians exactly on small systems, reinforcing its status as a fundamental numerical technique despite its exponential complexity bottleneck.

Finally, recent works [138] offer rich open-source libraries and computational tools that facilitate the implementation of ED, making it more accessible and efficient for researchers tackling complex quantum many-body problems.

4.2.3 Quantum Monte Carlo

Monte Carlo is a way to estimate values, often from very complex or high-dimensional problems, by using random sampling. The idea is to generate many random examples, calculate the result for each one, and then take the average. The method goes back to ideas like Buffon’s needle experiment, but it became a powerful modern tool at Los Alamos thanks to Ulam, Metropolis, von Neumann, and Fermi [156]. The key improvement was the use of Markov chains (the Metropolis algorithm [156]) to sample complicated probability distributions efficiently. Monte Carlo works well even when the problem has many variables, especially compared to traditional grid methods, and can be made more accurate using techniques like importance sampling.

QMC is a family of stochastic methods for solving the many-body Schrödinger equation by sampling configurations of the system rather than solving the equation directly. By exploring possible values of the wave function, its imaginary-time evolution, or related auxiliary variables, QMC fully incorporates particle–particle correlations that are often missed by simpler mean-field approaches [143]. From these sampled configurations, a broad range of physical properties can be calculated, including ground-state energies, pair-correlation functions, condensate fractions, momentum distributions, charge and spin structure factors, and, with additional techniques, excitation gaps.

For systems of bosons without certain types of frustration, QMC can give exact results for static properties with reasonable (polynomial) computational effort [144, 145]. For fermions, however, QMC often faces the sign problem, which makes exact calculations impractically slow [146]. In those cases, researchers either use carefully designed approximations (such as the fixed-node or fixed-phase methods) or accept that the cost will grow very quickly.

QMC methods can be grouped into two main families based on temperature. Zero-temperature QMC focuses on finding the ground-state properties of a quantum system, often using imaginary-time projection techniques to filter out excited states. Finite-temperature QMC, on the other hand, studies systems at nonzero temperatures, sampling thermal distributions to calculate properties like energy, specific heat, or correlation functions. Each family has multiple specific algorithms, tailored to either bosons or fermions and to different types of interactions. In the following paragraphs, we first present the zero-temperature QMC algorithms, which are designed to study ground-state properties. After that, we discuss the finite-temperature QMC methods, which focus on thermal properties of quantum systems.

Variational Monte Carlo (VMC) is a zero-temperature form of QMC) employed as the first stage in a project by Foulkes et al. [147]. In this method, a suitably parameterized trial wavefunction is optimized to minimize the expectation value of the system's Hamiltonian. The VMC approach provides an upper bound to the ground-state energy and captures essential correlation effects. Configurations are sampled using the Metropolis–Hastings algorithm so that they follow the probability distribution. This optimized trial wavefunction not only yields energetically accurate estimates but also defines high-quality nodal surfaces which are critical for subsequent projection-based methods.

Diffusion Monte Carlo (DMC) is another form of QMC at zero-temperature that projects out the exact ground-state component of the many-body wavefunction via imaginary-time evolution [143, 147]. DMC propagates an ensemble of “walkers” through drift, diffusion, and branching steps, with importance sampling guided by the trial wavefunction. To manage the fermionic sign problem, the fixed-node approximation is imposed using the nodal structure of the VMC-optimized wavefunction. This approach significantly improves the ground-state energy estimates and satisfies the zero-variance, zero-bias principle: as the trial wavefunction approaches the true eigenstate, both the variance and bias of the energy tend toward zero.

In the study by Foulkes et al. [147] over two-stage QMC strategy (VMC and DMC), authors demonstrate that although VMC and DMC simulations are computationally intensive, till 2001 they remain the only methods of accurately treating realistic continuum models of electrons in solids. The authors highlight how, over the preceding decade, these zero-temperature QMC techniques had matured enough to yield highly precise ground-state properties including energies and structural parameters for weakly correlated solids.

On the other side, we have the finite-temperature case, where one of the most prominent approaches is Path-Integral Monte Carlo (PIMC). In PIMC, the quantum partition function is reformulated as a sum over imaginary-time paths, which can be efficiently sampled using Monte Carlo techniques. For bosonic systems, this mapping to classical ring-polymers is numerically exact and naturally links macroscopic quantum phenomena such as superfluidity to the presence of long permutation cycles, quantified by winding numbers. For fermionic systems, however, PIMC faces the so-called fermion sign problem where sampled configurations can contribute with both positive and negative weights, leading to cancellations and an exponential growth of statistical noise with system size or inverse temperature. A common mitigation in PIMC is the restricted-path approximation, which constrains the sampled configurations using a trial density matrix, introducing a controlled bias [151].

This sign problem is not unique to PIMC but is a fundamental challenge in QMC simulations of fermions. Various strategies have been developed to control it. For instance, in Diffusion Monte Carlo, the fixed-node or fixed-phase approximation constrains the nodal surface of the wave function using a trial state [147]; in Auxiliary-Field QMC, the constrained-path or phaseless approach limits the evolution of Slater determinants based on a trial state to suppress sign or phase fluctuations [148, 149]; and in some models, special symmetry conditions, such as the half-filled Hubbard model on a bipartite lattice, remove the sign problem entirely, making the simulation numerically exact [150]. While these techniques can yield highly accurate results,

they are effective only for specific system geometries, fillings, or interaction forms, and no general solution exists for all cases.

4.2.4 Tensor Networks

TNs methods provide a powerful framework for simulating quantum many-body systems by efficiently capturing their entanglement structure. While a full discussion of these techniques, including matrix product states, matrix product operators, and their applications to one-dimensional and well-known algorithm like DMRG and TEBD, is presented in Chapter 3, here we simply note that TNs allow us to controll quantum many-body systems and access ground states and characterize phase transitions by QML methods with remarkable accuracy, making them an essential tool in the modern simulation toolbox.

4.3 Axial Next-Nearest-Neighbor Ising model

The ANNNI model occupies a special place in the history of condensed matter and statistical physics. It is one of those deceptively simple models that, once written down, revealed layers upon layers of complexity, frustration, and ordering phenomena. The story of the ANNNI model begins in the early 1960s, when Roger Elliott [157] first introduced it to capture modulated phases in rare-earth compounds. At the time, physicists were still trying to reconcile how relatively simple spin interactions could lead to the remarkably intricate magnetic patterns observed in experiments.

Two decades later, Michael Fisher and Walter Selke [160] gave the model both its formal name and its first systematic theoretical treatment. They considered an Ising chain where nearest neighbors prefer to align ferromagnetically, while next-nearest neighbors along a lattice axis prefer to anti-align. This simple competition lies at the heart of the model. It creates frustration: the system can never fully satisfy all its interaction energies simultaneously. Fisher and Selke showed that this tension gives rise to a cascade of ordered phases at low temperatures.

Through the 1980s, researchers extended this picture to higher dimensions and to more realistic finite-temperature scenarios. The two-dimensional ANNNI model, in particular, became a playground for testing theoretical methods. Approaches like the cluster variation method (Finel and de Fontaine) [158], transfer-matrix techniques (Grynberg and Ceva) [159], and renormalization-group arguments were all brought to bear. Each method provided a slightly different lens, but together they built a coherent picture: the ANNNI model exhibits an intricate dance of commensurate and incommensurate phases, with short-range correlations and dimensionality playing decisive roles. More recent computational advances, including high-precision transfer matrix calculations, have resolved ambiguities that lingered for decades, firmly establishing the existence and boundaries of floating incommensurate phases.

The model's richness became even more striking once external magnetic fields were introduced. Exact transfer-matrix studies in the early 1990s [161] showed that fields could dramatically reshape the topology of the phase diagram, stabilizing or suppressing phases that otherwise would not exist. Already by this point, the ANNNI model had secured its reputation as one of the classic frustrated spin systems, which is pretty simple in form, yet inexhaustible in consequences.

But the story did not stop at classical statistical mechanics. In the 1990s, the community began to ask: what happens when thermal fluctuations are replaced by quantum ones? In other words, what if spins in the ANNNI model are subjected to a transverse magnetic field, forcing them to fluctuate between up and down even at zero temperature?

The pioneering quantum studies took advantage of the Trotter–Suzuki mapping, which translates the one-dimensional quantum ANNNI chain into a two-dimensional classical Ising system. Monte Carlo simulations in this framework revealed a quantum phase diagram no less intricate than its thermal counterpart, with regions of ferromagnetism, antiphase order, floating incommensurate states, and disordered phases [162]. Shortly afterward, ED studies on finite chains confirmed and refined this picture, showing how quantum fluctuations can destabilize classical order and give rise to qualitatively new regimes [163]. The appearance of floating phases with algebraically decaying correlations highlighted how quantum mechanics reshapes the landscape of frustration.

By the 2010s, [213] numerical advances allowed for much larger system sizes and more refined techniques. Studies based on Lanczos and conjugate-gradient diagonalization pushed the chain lengths to two dozen spins, and the use of fidelity susceptibility, which is a measure of how rapidly the ground state changes under small parameter shifts, provided a powerful, order-parameter-free way of locating quantum phase transitions. These works not only confirmed the ferromagnetic, antiphase, and floating regions, but also hinted at an infinite hierarchy of modulated phases emerging in the thermodynamic limit.

In recent years, this story has entered a new phase, powered by computational advances. A particularly striking example comes from the work of Fumani et al. [170], who revisited the spin-half frustrated transverse-field Ising chain, essentially the quantum extension of the ANNNI model, with modern ED techniques. By carefully studying chains of up to 28 spins, they mapped out the phase diagram with unprecedented precision, using level-crossing analysis and spin-structure factor calculations to pinpoint subtle critical lines. What emerged was a clear view of commensurate and incommensurate phases, along with the quantum phase transitions that knit them together. In contrast to earlier mean-field or approximate approaches, this study demonstrated the power of exact numerics to uncover delicate behaviors that had long remained hidden.

If the early ED studies marked the first steps into the quantum landscape of the ANNNI model, the rise of TNs methods transformed the journey into something far deeper. With the development of the DMRG in the 1990s, theorists suddenly had a microscope powerful enough to look at ground states of frustrated spin chains with an accuracy and scale that had been unimaginable before. The ANNNI model, with its delicate interplay of ferromagnetism, frustration, and quantum fluctuations, quickly became one of the proving grounds for these techniques.

In 2006, Beccaria et al. [165] used high-precision DMRG to explore the low-frustration regime of the quantum ANNNI model. Working with chains of up to 80 spins, they carefully mapped the transition from ferromagnetic to paramagnetic order and traced the disorder line, where the subtle boundary marking the onset of incommensurate correlations. Their analysis confirmed a long-standing conjecture: the Peschel–Emery line coincides precisely with the disorder line. For a model that

had spawned decades of debate, this was a moment of clarity, made possible by the precision and numerical power of TNs.

The story deepened the following year, when the same group [166] turned their attention to the high-frustration regime. Pushing DMRG to chains as long as 710 spins, they provided the most compelling evidence yet for the existence of the elusive floating phase. This phase, characterized by algebraically decaying correlations and incommensurate order, had been suggested by field-theoretical arguments but was notoriously hard to pin down numerically. By combining entanglement entropy, modulation vectors, and excitation gap analyses, they not only confirmed its presence but also mapped its boundaries with quantitative rigor. What had long been a contested feature of the ANNNI phase diagram was now established on solid ground.

But TNs did not stop at finite systems. In 2011, Nagy [167] advanced the frontier further by employing iMPS together with the iTEBD algorithm. Instead of struggling with finite-size scaling, this approach worked directly in the thermodynamic limit, sidestepping size effects altogether. By tuning the bond dimension of the iMPS and exploiting finite-entanglement scaling, Nagy extracted critical exponents, central charges, and transition points with remarkable accuracy. Even the notoriously subtle Kosterlitz–Thouless transition, an infinite-order transition that had eluded earlier methods, became visible within this framework.

The ANNNI model’s journey did not end with TNs. In fact, the past few years have opened yet another chapter — one that blends physics with ideas from quantum information and machine learning. If DMRG and iMPS taught us how to see the ANNNI ground state with unprecedented clarity, then QML promises to teach us how to “learn” from it in ways that go beyond traditional calculation.

In 2023, Monaco et al. [169] took a bold step in this direction. They asked whether a QNNs could recognize the phases of the ANNNI model when given only very limited labeled data, which is the kind of challenge physicists routinely face when exploring non-integrable systems where analytic results are scarce. Their solution was to design a Quantum Convolutional Neural Network (QCNN), trained on ground states produced by VQE simulations. Remarkably, with training data drawn only from small, analytically tractable corners of the phase diagram, the QCNN was able to extrapolate across the full, non-integrable landscape, reconstructing phase boundaries with over 97% accuracy. This result felt like a breakthrough not only for the ANNNI model, but also for the broader hope that QML could uncover hidden order in complex quantum systems.

The following year, Cea et al. [168] extended this idea by combining supervised and unsupervised quantum learning strategies. Using high-quality ground states computed via DMRG, they built datasets for systems of up to 20 qubits and applied both QCNN and quantum anomaly detection methods. Once again, the supervised QCNN stood out: trained only on sparse, marginal points, it still managed to recover the global phase diagram of the ANNNI model with around 95% accuracy. The anomaly detection approach, while less precise, offered a complementary, unsupervised perspective. Taken together, these results highlight an emerging theme: even shallow QNNs, when properly trained, can serve as powerful phase classifiers, reducing the need for exhaustive data and opening a scalable path for analyzing non-integrable quantum systems.

These QML studies signal a new era for the ANNNI model. What began in

the 1960s as a simple Hamiltonian has now become a testing ground for the most modern approaches at the interface of quantum physics, computation, and data science. They remind us that the model's richness is not just in its phases, but also in how it continues to challenge and inspire new methods of exploration.

In line with this perspective, our study employs the Hamiltonian of the one-dimensional quantum ANNNI model at zero temperature. This formulation captures the essential ingredients, including a nearest-neighbor ferromagnetic interaction, a frustrating next-nearest-neighbor coupling, and a transverse magnetic field that injects quantum fluctuations,

$$H = -J \sum_{j=1}^{N-1} X_j X_{j+1} - k \sum_{j=1}^{N-2} X_j X_{j+2} - h \sum_{j=1}^N Z_j, \quad (4.3.1)$$

where the first term, involving nearest-neighbor coupling $-J \sum_j X_j X_{j+1}$, favors ferromagnetic alignment ($\uparrow\uparrow\uparrow\uparrow\uparrow$) along the z -axis and promotes uniform spin ordering when dominant. In contrast, the next-nearest-neighbor interaction $-k \sum_j X_j X_{j+2}$, with $k < 0$, introduces antiferromagnetic frustration, favoring periodic modulated patterns such as the antiphase ($\uparrow\uparrow\downarrow\uparrow\uparrow$). This competition between short- and longer-range interactions lies at the heart of the model's complex phase structure. Finally, the transverse-field term $-h \sum_j Z_j$ introduces quantum fluctuations by encouraging spin flips along the z -axis, destabilizing classical order and driving the system toward a quantum paramagnetic phase ($\downarrow \cdot \uparrow \cdot \downarrow \uparrow$). The balance among these three ingredients gives rise to the rich variety of zero-temperature quantum phases that continue to inspire both theoretical and computational innovation. In this study $j = 1$.

As reproduced in Figure 4.3.1 from Monaco et al. [169], the one-dimensional quantum ANNNI model hosts four principal phases: the ferromagnetic phase, the antiphase, the enigmatic floating phase, and the quantum paramagnetic phase. Each of these regions tells its own story about how spins organize or fail to organize under the subtle competition between k , and h .

Figure 4.3.2 (a) illustrates a one-dimensional spin-half chain in the trivial paramagnetic regime at $T = 0$. In this phase, the transverse field, which is acting along the z -direction, dominates over the x -axis interactions (J_1, J_2). Instead of locking into ordered patterns favored by the couplings, each spin is continually flip, aligning weakly along the transverse field. The result is a state where long-range order is lost: spins no longer point collectively up or down along x , but instead fluctuate in a way that gives the visual impression of a disordered array of arrows.

In the phase diagram (Figure 4.3.1), this corresponds to the paramagnetic phase, which emerges once the transverse field strength h becomes sufficiently large (roughly $h \gtrsim 1$ in natural units). Here, quantum fluctuations overwhelm the tendency of k to impose order, leaving behind a fully quantum-disordered ground state.

Figure 4.3.2 (b) depicts the ferromagnetic regime, which appears when the next-nearest-neighbor frustration is relatively weak and the transverse field is small. In this phase, the nearest-neighbor interaction along the X -axis dominates, encouraging all spins to align in the same direction. The result is a perfectly ordered state, where each spin points along X , forming a uniform pattern such as $(\rightarrow \rightarrow \rightarrow \rightarrow \rightarrow)$ or $(\leftarrow \leftarrow \leftarrow \leftarrow \leftarrow)$. Unlike the paramagnetic phase, where the transverse field dominates

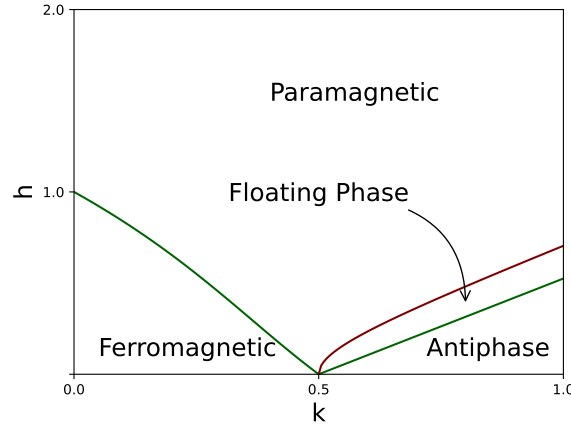


Figure 4.3.1. Phase diagram of the ANNNI model showing the paramagnetic, ferromagnetic, antiphase, and floating phases.

and disrupts ordering, here the spins collectively reinforce one another, producing long-range order. Another fascinating region of the phase diagram is the antiphase, which emerges when the next-nearest-neighbor interaction becomes dominant while the transverse field remains weak. Based on the Hamiltonian, the nearest-neighbor interaction would prefer all spins to align, as in the ferromagnet, but the next-nearest-neighbor term fights against this, favoring an alternating pattern over longer distances. When k wins this competition, the system settles into a compromise: the $\langle 2, 2 \rangle$ antiphase, where spins align in pairs before flipping, producing the repeating pattern $(\uparrow\uparrow\downarrow\downarrow\uparrow\uparrow\downarrow\downarrow)$. Figure 4.3.2 (c) illustrates this distinctive structure. Unlike the simple uniformity of the ferromagnet or the disorder of the paramagnet, the antiphase reflects a more subtle kind of order. Beyond the well-defined paramagnetic, ferromagnetic, and antiphase regimes lies the floating phase. Unlike the crisp order of the ferromagnet or antiphase, the floating phase does not lock into a single repeating pattern. Instead, it exhibits a kind of quasi-long-range order, where correlations persist over long distances but without a fixed periodicity enforced across the entire chain. In practice, the floating phase can manifest through several characteristic spin configurations, depending on the balance of k and h . Figures 4.3.2 (d-f) illustrate three typical examples of floating phases.

With the floating phase, the picture of the ANNNI model becomes complete. From the simple order of the ferromagnetic state, through the frustration-driven antiphase, and into the fluctuating landscape of the floating regime, we see how the competition between nearest- and next-nearest-neighbor couplings, together with the transverse field, weaves a remarkably rich phase diagram. Each phase tells its own story about order, frustration, and quantum fluctuations, and together they highlight why this model continues to attract so much attention. What makes the journey even more exciting is that, beyond traditional tools like DMRG and TNs, QML has now entered the scene, offering new ways to recognize and classify these phases even when data is scarce.

This is precisely where our work picks up. In the next chapters 5, 6, we build on

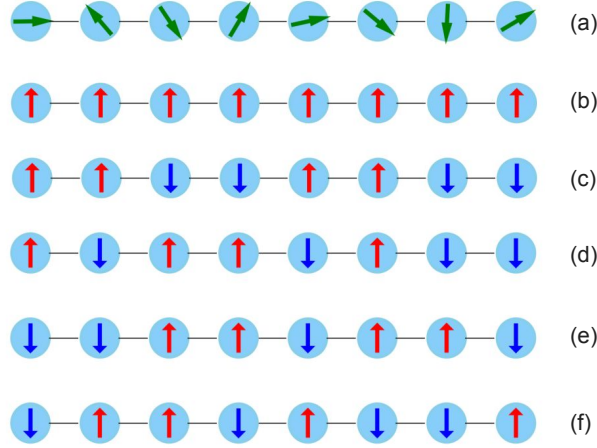


Figure 4.3.2. Illustration of representative spin- $\frac{1}{2}$ configurations in one dimension: (a) a paramagnetic chain; (b) a ferromagnetic chain; (c) the antiphase pattern corresponding to the $\langle 2, 2 \rangle$ structure; (d–f) examples of floating-phase configurations with different repeating patterns, including (d) $\langle 1^2 2 \rangle$, (e) $\langle 2^2 1 \rangle$, and (f) $\langle 1 2 1 \rangle$.

these insights and ask: can machine learning not only retrace known phase diagrams, like that of the ANNNI model, but also help us venture into Hamiltonians whose phases remain uncharted?

4.4 Cluster-Ising model

The cluster-Ising model has emerged as a remarkably rich playground for exploring the interplay between conventional order and more exotic, nonlocal structures in quantum matter. What makes this model so fascinating is that, even in its simplest one-dimensional form, it captures a competition between familiar antiferromagnetic order and less conventional cluster-type interactions that give rise to topological and highly entangled phases.

Some of the earliest work in this direction was carried out by Smacchia et al. [171] (2011), who used exact analytical methods to map out the fundamental phases of the model. They showed that the system supports two distinct regimes: a cluster phase characterized by nonlocal string order and multipartite entanglement, and an antiferromagnetic Ising phase marked by staggered magnetization. Interestingly, standard two-spin correlations are absent in both cases, meaning that only genuinely collective forms of entanglement survive. At the transition point, the system exhibits a novel type of criticality, effectively behaving like three decoupled Ising chains.

Around the same time, Son et al. [172] (2011) approached a closely related model from a different angle, demonstrating an exactly solvable transition between a symmetry-protected topological phase and a conventional antiferromagnet. Their work showed that this transition is continuous, self-dual, and beyond the reach of the traditional Landau framework. What set their study apart was the use of geometric entanglement, which turned out to be a particularly sensitive probe of the critical

point.

This theme of going beyond conventional order continued in the work of Montes et al. [173] (2012), who explored models where cluster interactions compete not just with Ising terms but also with XY-type couplings. By combining analytical tools like the quantum geometric tensor and Loschmidt echo, they revealed a far richer phase diagram, one that captures both static properties and the system’s highly nontrivial response to sudden quenches. This work highlighted how cluster–Ising physics is not just about equilibrium phases but also about dynamical behavior far from equilibrium.

Later, Giampaolo et al. [174] (2015) expanded the scope of exactly solvable cluster models even further, introducing families of spin chains with interactions spread over larger clusters. Remarkably, they found that the phase transition always occurs precisely when the Ising and cluster couplings balance, regardless of the cluster size. Even more striking was the discovery that the nature of the cluster phase depends on whether the cluster size is even or odd, which leading to nematic order in one case and symmetry-protected topological order in the other. This revealed how subtle structural differences in the model can completely reshape the type of order that emerges.

Analytical work has also shed light on the geometric and topological aspects of the model. Nie et al. [175] (2017), for example, studied the scaling of the geometric phase and showed how it encodes both conventional and topological transitions. By connecting this phase to a winding number in parameter space, they provided a powerful tool for diagnosing quantum criticality that goes well beyond the usual symmetry-breaking picture.

More recent studies have continued to deepen this understanding. A 2025 analysis (Ref. [182]) used Jordan–Wigner transformations to explore quantum correlations in finite cluster systems, uncovering sharp entanglement singularities and nontrivial behavior near degeneracy points. These results emphasized how different correlation measures can serve as fingerprints of structural changes in the ground state. In parallel, another 2025 study (Ref. [183]) addressed the robustness of subsystem symmetry-protected topological order under strong decoherence. Surprisingly, they found that this order can survive even at maximal noise levels, though they also uncovered subtleties in entanglement diagnostics, such as spurious signals in the entanglement negativity.

So far, we have focused on the analytical progress that shaped our understanding of the cluster–Ising model. Yet, analytical solvability has its limits, and computational studies have played an equally important role in revealing the full richness of this model, especially when exploring non-equilibrium dynamics, large-scale systems, and connections to quantum simulation.

One of the earliest motivations for computational work can be traced back to Kay et al. [184] (2005), who derived effective three-body spin Hamiltonians from cold atoms in triangular optical lattices. Although primarily analytical, their work highlighted how such models naturally host symmetry-protected topological phases and unconventional entanglement structures, thereby setting the stage for later numerical explorations of criticality and phase diagrams in multi-spin systems. Building on this foundation, Chen et al. [185] (2010) examined cluster Hamiltonians through the lens of quantum information theory. Their analysis of string order

and nonlocal entanglement measures revealed the robustness of symmetry-protected phases and gave a precise diagnostic framework for tracking transitions between topological and trivial regimes.

A decade of computational studies then pushed these insights much further. Using powerful numerical techniques such as TEBD and ED, researchers [186] in 2015 mapped out the quantum phase diagram of generalized cluster models and analyzed their dynamics under slow parameter sweeps. A striking discovery was that quasi-particles generated at critical points can leave behind spatially periodic “fingerprints” in both correlations and entanglement, which is an elegant bridge between static topological order and dynamical response. This line of inquiry was deepened by Ohta et al. [214] (2016), who refined the phase diagram and studied the entanglement spectrum in detail. Their simulations revealed that topological signatures appear not only in the ground state but also persist dynamically, with oscillatory entanglement features reflecting the presence of Bogoliubov quasi-particles. These works firmly established computational modeling as a window into how topological order manifests and evolves far beyond equilibrium.

Later studies expanded the scope of computational approaches in new and unexpected directions. Jeyaretnam et al. [187] (2021) uncovered quantum many-body scars within a symmetry-protected phase, special eigenstates that defy thermalization and maintain long-lived coherence deep in the bulk of the system. Their results challenged the idea that topological features are confined to ground states, showing instead that they can survive at finite energies. Around the same time, Verga [188] (2023) explored Floquet-driven cluster chains and identified long-lived nonthermal states with distinct entanglement patterns, further enriching the dynamical phase landscape.

The most recent developments have pushed these ideas onto real quantum hardware and large-scale simulations. Shen et al. [180] (2025) demonstrated that quantum processors, aided by techniques like quantum imaginary-time evolution and noise mitigation, can already simulate Symmetry Protected Topological (SPT) transitions in the cluster–Ising model with striking accuracy. Their results revealed hallmark features such as edge modes and entanglement spectral gaps, proving that experimental platforms are catching up with theory and computation. In parallel, Yu et al. [181] (2025) showed that even measurement-only quantum circuits, without any unitary dynamics, can support robust gapless SPT phases. Using massive simulations of Clifford circuits, they constructed dynamical phase diagrams and uncovered entirely new routes to realizing topological phases.

Having discussed both the analytical and computational studies more broadly, we now turn our attention to a more specialized yet remarkably powerful tool: TNs methods. These approaches, particularly MPS and their extensions, have become indispensable for probing the entanglement structure and topological features of cluster–Ising models, offering both conceptual clarity and numerical precision.

The foundations of this perspective were laid by Pérez-García et al. [176] (2008), who established a rigorous link between symmetries and string order in one-dimensional systems through the formalism of Matrix Product States combined with group representation theory. Their work provided a universal classification of SPT phases, showing how hidden nonlocal orders can be systematically understood and characterized. Building on this, Skrøvseth and Bartlett [177] (2009) applied

MPS as a variational tool to explore the robustness of cluster-state ground states against perturbations from Ising interactions and transverse fields. By simulating spin chains of up to forty sites, they mapped out how entanglement length and string correlations evolve under perturbations, thereby identifying when cluster states remain useful for measurement-based quantum computing and when their computational potential breaks down.

As TNs methods matured, researchers began addressing more complex scenarios, such as disordered systems. Calvanese Strinati et al. [178] (2017) combined analytical reasoning with MPS simulations to test whether nonlocal string order can survive in the presence of disorder that respects system symmetries. Their results were striking: the transition between antiferromagnetic and string-ordered phases remained robust, confirming that SPT order is not fragile but can persist even under realistic imperfections. More recently, Morral-Yeps et al. [179] (2023) pushed these ideas into the domain of random quantum dynamics. Using stabilizer formalism for Clifford circuits and MPS-based TEBD simulations for Haar-random circuits, they showed that SPT order continues to emerge in non-equilibrium, highly randomized settings, provided the symmetries are preserved. This demonstrates that measurement-induced SPT phases are not confined to fine-tuned conditions but can arise broadly across diverse dynamical regimes.

Having explored analytical, computational, and tensor-network approaches, it is natural to ask how modern QML methods can contribute to our understanding of the Cluster-Ising model. A notable step in this direction was taken by Cong et al. [36] (2019), who introduced QCNN as a scalable way to identify quantum phases of matter using near-term quantum devices. Inspired by classical convolutional architectures but rooted in ideas from MERA and quantum error correction, their QCNN framework efficiently detects subtle entanglement patterns and topological features that often escape conventional diagnostics. Applied to the Cluster-Ising model, the QCNN was trained on only a small set of representative states yet succeeded in reconstructing the full phase diagram, distinguishing symmetry-protected, symmetry-broken, and trivial paramagnetic phases with remarkable accuracy, even in systems as large as 45 spins. This demonstrates how machine learning tools can uncover global quantum features from local training, complementing the analytical and numerical approaches discussed earlier.

In line with this perspective, our study employs the Hamiltonian of the Cluster-Ising model, which combines a three-spin cluster interaction, a two-spin coupling term, and a transverse field,

$$H = -J \sum_{j=1}^{N-2} Z_j X_{j+1} Z_{j+2} - h_2 \sum_{j=1}^{N-1} X_j X_{j+1} - h_1 \sum_{j=1}^N X_j, \quad (4.4.1)$$

where X_j and Z_j are the standard Pauli operators acting on site j , and N is the total number of spins in the chain. This Hamiltonian captures the competition between three distinct contributions, each of which plays a key role in shaping the quantum phases of the system.

The first term, $-J \sum_j Z_j X_{j+1} Z_{j+2}$, is the three-spin cluster interaction. It generates a SPT phase, characterized by nonlocal string order and long-range entanglement. In this phase, the correlations cannot be captured by conventional two-

spin observables, and the system exhibits topological edge modes at its boundaries, making it fundamentally distinct from ordinary magnetic order. For our study, we set $J = 1$ as the energy scale.

The second term, $-h_2 \sum X_j X_{j+1}$, is a two-spin coupling that competes with the cluster interaction. When h_2 is small, the cluster term dominates only partially, and the system exhibits antiferromagnetic order along the x -direction, with spins alternating in a staggered pattern. As h_2 increases to intermediate values, the three-body cluster term regains relative importance, giving rise to the SPT phase. Here, long-range entanglement and topological order emerge, and the system can host edge modes protected by symmetry. When h_2 becomes large, the two-body interaction overcomes the cluster correlations, and the system enters a paramagnetic phase dominated by the transverse field-like behavior, where spins tend to align along the x -direction independently, suppressing both antiferromagnetic and topological order.

Finally, the last term, $-h_1 \sum X_j$, represents a uniform transverse field applied to every spin. This term acts to polarize spins along the x -axis, enhancing the paramagnetic tendencies when strong and competing with both antiferromagnetic and cluster-induced correlations. By adjusting the relative strengths of h_1 and h_2 , the model allows exploration of the full phase diagram, revealing the delicate balance between conventional magnetic order, symmetry-protected topological order, and paramagnetic behavior.

The phase diagram shown in Figure 4.4.1 is reproduced from Cong et al. [189]. By applying QCNN to a small set of representative quantum states, they were able to reconstruct the full phase diagram, accurately identifying the antiferromagnetic, SPT, and paramagnetic phases. Remarkably, the QCNN captures subtle changes in entanglement that signal phase transitions, even in regions where conventional order parameters fail to reveal the underlying quantum correlations.

Figure 4.4.2 (a) illustrates a one-dimensional spin-half chain in the trivial paramagnetic regime at $T = 0$. In this phase, the transverse field applied along the x -direction dominates the system, pulling each spin to weakly align with the field. This gives the visual impression of a random array of arrows along the chain, reflecting the absence of long-range order.

As discussed by Guo et al. [130], once the non-local string order reveals the SPT phase and staggered magnetization captures the antiferromagnetic phase, what remains is the trivial paramagnetic regime. Here, no symmetry is broken and no hidden order survives, leaving a product-like state where conventional order parameters vanish. Instead, the phase can be recognized through simple local expectation values, which act as practical diagnostics of its uncorrelated, field-aligned nature.

In the antiferromagnetic regime, spins in the chain tend to align in an alternating up-down pattern along the chain 4.4.2 (b). Physically, this pattern arises because the system favors antiparallel alignment to reduce the interaction energy, creating a state with long-range magnetic order throughout the chain.

In the SPT regime, the spin chain develops a different structure compared to conventional phases. Spins in the bulk form highly entangled singlet-like pairs, weaving long-range correlations throughout the chain, while the spins at the two ends remain essentially free and unpaired. This creates a pattern where the interior

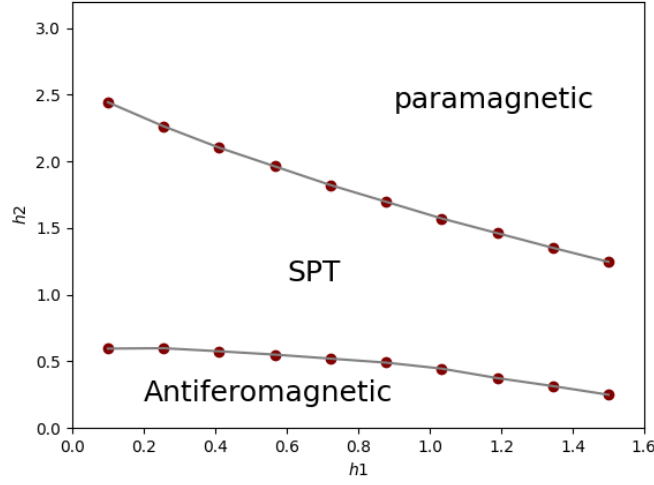


Figure 4.4.1. Phase diagram for Cluster-Ising model, showing the antiferromagnetic and paramagnetic phases as locally ordered phases, and the symmetry-protected topological (SPT) phase as a non-local phase.

of the chain is tightly connected through quantum entanglement, yet the edge spins retain independent, low-energy degrees of freedom. (see figure 4.4.2 (c)) Such a structure often emerges when a chain that previously showed antiferromagnetic tendencies is driven into a topological phase by cluster interactions, replacing local staggered order with a global, non-local pattern protected by symmetry.

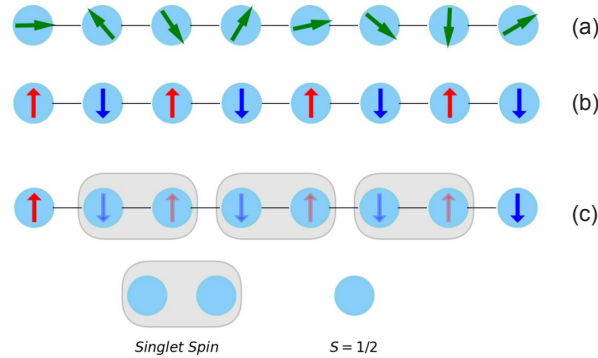


Figure 4.4.2. (a) One-dimensional spin- $\frac{1}{2}$ paramagnetic chain. (b) One-dimensional spin- $\frac{1}{2}$ antiferromagnetic chain with staggered order. (c) SPT chain, where non-local spin-singlet pairs are indicated by the shaded boxes.

Lastly, we looked at the Cluster-Ising model through the lens of QML in chapters 5, 6, and 7. Using both supervised and unsupervised models trained on quantum states from DMRG, we found that the phases can be identified accurately even when only small parts of the system are measured. This shows that reliable phase

classification is possible with limited data. Moreover, the unsupervised approach opens the door to exploring Hamiltonians whose phase diagrams are not yet known.

4.5 Haldane Anisotropic Spin Integer model

The Haldane anisotropic spin integer model in our study refers to a one-dimensional quantum spin system composed of spin-1 particles with anisotropic interactions. When Duncan Haldane first put forward his bold conjecture in 1983, he challenged a long-standing intuition in quantum magnetism. Up to that point, physicists had grown accustomed to thinking that all spin chains should behave alike in the low-energy limit. But Haldane's work revealed a striking dichotomy: integer-spin chains open a gap, while half-integer chains remain gapless [190,191]. That prediction did not just surprise the community, in fact it reshaped our understanding of quantum matter. The so-called Haldane gap was more than a technical detail; it hinted at a new kind of order, one invisible to traditional probes, that would later be recognized as a SPT phase.

Of course, real materials and models rarely sit in such a clean, isotropic limit. As theorists began to ask: what happens if we perturb the chain?. Adding single-ion anisotropy to the spin-1 Heisenberg chain opened up new possibilities. Depending on the strength and direction of this anisotropy, the system could settle into distinct phases: the antiferromagnetic Néel order, the so-called large-D state, or the elusive Haldane state in between. Langari and collaborators mapped this terrain by using concepts like ground-state fidelity and renormalization ideas, showing how the invisible threads of non-local order can still track the passage between topological and trivial phases [128].

But the story did not end there. Pinpointing these transitions required sharper tools. ED, combined with conformal field theory, gave the first quantitative foothold, even clarifying long-standing mismatches with bosonization predictions [192]. Pushing to larger system sizes, QMC simulations and series expansions refined the picture further, yielding precise estimates for critical points, excitation gaps, and spectral features [193].

The real breakthrough, however, came with TNs methods. With the rise of the DMRG, the Haldane phase could finally be seen through the lens of quantum entanglement. Pollmann and colleagues showed that its hidden topological character manifests as a characteristic double degeneracy in the entanglement spectrum, which is a fingerprint of SPT order protected by symmetries [194]. Later refinements of DMRG achieved astonishing precision, resolving Gaussian critical points to four significant figures and determining critical exponents with remarkable confidence [195].

Even today, the Haldane model continues to unfold new surprises. Subtle uniaxial and rhombic anisotropies can destabilize the delicate Haldane order and push the system into new territory, such as the exotic g -Néel state [196,197]. In our case study, we build directly on this line of work: the Hamiltonian we consider follows the formulation of [196], providing a natural framework to explore how topology, entanglement, and symmetry weave together across the competing quantum phases

$$H = J \sum_{j=0}^{L-1} \mathbf{S}_j \cdot \mathbf{S}_{j+1} + D \sum_{j=0}^L (S_j^z)^2 + E \sum_{j=0}^L [(S_j^x)^2 - (S_j^y)^2], \quad (4.5.1)$$

where the spin-1 Hamiltonian, defined in terms of spin operators $\mathbf{S}_j = (S_j^x, S_j^y, S_j^z)^\top$ at each site j , captures key features of quantum magnetism through the interplay of exchange interactions and anisotropies. The first term corresponds to the nearest-neighbor Heisenberg exchange, with coupling strength J . For the spin-1 case with antiferromagnetic coupling ($J > 0$), this term favors short-range singlet correlations.

The second term, $D \sum_j (S_j^z)^2$, is known as the uniaxial single-ion anisotropy. Its role is to break the full spin-rotation symmetry ($SU(2)$) of the Heisenberg exchange interaction by selecting the z -axis as a special direction. In practice, this term acts differently on the three local spin states of an $S = 1$ ion. Since $(S^z)^2 |m_z\rangle = m_z^2 |m_z\rangle$, the energy contribution from this anisotropy is:

$$E(m_z = 0) = 0, \quad E(m_z = \pm 1) = D. \quad (4.5.2)$$

Thus, the anisotropy directly controls which local spin states are energetically preferred. For large positive D , the $|m_z = \pm 1\rangle$ states are penalized by an energy cost of $+D$, while $|m_z = 0\rangle$ remains at zero energy. To minimize energy, the system selects $|m_z = 0\rangle$ on every site. This produces the so-called Large- D phase, a trivial product state in which all spins are “frozen” in the $|0\rangle$ configuration. It is a gapped, non-magnetic phase with neither dipolar order nor topological character. While for negative D , the situation is reversed. The $|m_z = \pm 1\rangle$ states are now energetically favored, while $|m_z = 0\rangle$ becomes costly. In this regime, each spin prefers to occupy $|+1\rangle$ or $|-1\rangle$. When the antiferromagnetic exchange $J > 0$ is also active, the system minimizes its total energy by arranging neighboring sites in alternating $|+1\rangle$ and $|-1\rangle$ states. This ordering corresponds to a Néel phase along the z -axis, where spins alternate up and down in a staggered pattern [193, 197, 198].

The final term, called Rhombic single-ion anisotropy, can be equivalently expressed as

$$E \sum [(S_j^x)^2 - (S_j^y)^2] = \frac{E}{2} \sum [(S_j^+)^2 + (S_j^-)^2], \quad (4.5.3)$$

showing explicitly that it couples $|m_z = \pm 1\rangle$ states. This term mixes these states into eigenstates aligned preferentially along the x - or y -axis. For $E > 0$, the system favors $|m_x = 0\rangle$, while for $E < 0$, it favors $|m_y = 0\rangle$. In extended chains, these tendencies give rise to trivial quadrupolar product phases (“Large- E_x ” and “Large- E_y ”), and in combination with strong easy-axis exchange may induce transverse Néel order along x or y [196].

The quantum phase diagram of the spin-1 anisotropic Haldane chain, shown in Fig.4.5.1 which is reproduced from Ref. [196], includes SPT phases, symmetry-breaking Néel phases, and trivial disordered phases. This phase diagram is derived from numerical investigations employing the DMRG technique. In their study, first they were using DMRG to identify different quantum phases, then classify states using conserved quantum numbers such as the parity of magnetization, spatial inversion, and time-reversal symmetry. The phase boundaries and critical points are subsequently determined with high precision through analysis of the entanglement

spectrum, which is computed from the reduced density matrix derived from the system's ground state wavefunction.

Crucially, their work demonstrates that phase boundaries in this system are not characterized by conventional local order parameters, but rather by the structure of the entanglement spectrum and crossings of energy levels distinguished by symmetry quantum numbers. The degeneracy structure in the entanglement spectrum plays a key role: the Haldane phase exhibits a characteristic fourfold degeneracy associated with the hidden $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetry; the Néel phases (e.g., x -, y -, or z -Néel) show twofold degeneracy due to spontaneous $\mathbb{Z}_2$ symmetry breaking; and the trivial phases (such as large- D , large- E_x , and large- E_y) are identified by a non-degenerate (single-fold) entanglement spectrum [190, 199, 200, 202].

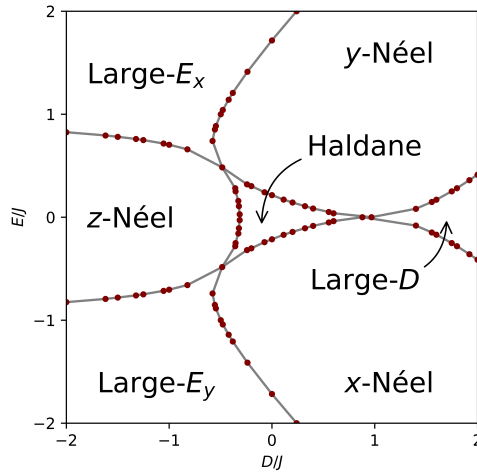


Figure 4.5.1. Phase diagram for Haldan Anisotropic spin-1, showing trivial phases (including large E_x , E_y , and D), Néel-ordered phases (including (x , y - and z -Néel)), and SPT (Haldane) phases.

Quantum phases in Haldane Anisotropic are characterized not only by their energy gaps and correlation behaviors, but more fundamentally by how they relate to symmetry breaking.

Figure 4.5.2 illustrates the antiferromagnetic coupling between neighboring units in an $S = 1$ Haldane chain. In a classical spin framework, a one-dimensional chain of $S = 1/2$ spins with antiferromagnetic interactions typically forms a Néel-ordered state, exhibiting an alternating pattern of spin orientations: $\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow$.

Indeed, the system exhibits more complex quantum behavior than a classical picture suggests because of spin singlet [190]. In figure 4.5.2 spins from neighboring units align in opposing directions $S = 0$ to form singlet spin states, $(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)/\sqrt{2}$, as indicated by the gray boxes in figure. These singlet spins give rise to a nonlocal entangled ground state that is a hallmark of the Haldane phase.

Each blue sphere represents an individual $S = 1/2$ spin. Pairs of such spins belonging to the same unit cell (enclosed by white boxes) are aligned in the same direction, forming an effective spin-1 object. These triplet states are described by the basis states $|\uparrow\uparrow\rangle$, $1/\sqrt{2}(|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle)$, and $|\downarrow\downarrow\rangle$. This configuration realizes a SPT

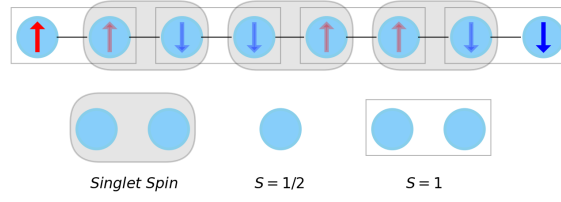


Figure 4.5.2. One-dimensional spin-1 Haldane chain, illustrating the formation of spin-1 sites from two spin- $\frac{1}{2}$ particles and the nonlocal spin-singlet pairs shown by shaded boxes.

phase, characterized by a gapped bulk and topologically protected localized edge states with effective spin- $\frac{1}{2}$ at the boundaries [201,202].

Beyond the Haldane phase at the center of the phase diagram, there exist three distinct regions characterized by Néel order along the spin axes specifically the x , y , and z directions. A Néel state reflects antiferromagnetic order, where neighboring spins are aligned in opposite directions. For instance, Néel order along the z -axis manifests as a spin configuration like $\uparrow\downarrow\uparrow\downarrow$, which explicitly breaks the full spin-rotation symmetry $SU(2)$ down to a residual $U(1)$ symmetry about the z -axis. In this configuration, the state remains invariant under rotations around the z -axis, but not around the x or y axes which indicating spontaneous symmetry breaking in those directions.

Crucially, Néel order can also develop along the transverse directions of spin space. By rotating the spin basis, one can realize a Néel state along the x -axis, represented by alternating spin states $\rightarrow\leftarrow\rightarrow\leftarrow$, where each spin is a coherent superposition of up and down spins: $|\rightarrow\rangle = (|\uparrow\rangle + |\downarrow\rangle)/\sqrt{2}$. Similarly, Néel order along the y -axis involves alternating states such as $|\odot\rangle$ and $|\oslash\rangle$, which are complex combinations of $|\uparrow\rangle$ and $|\downarrow\rangle$ that encode spin orientations in the transverse plane. In each case, the emergence of directional Néel order spontaneously breaks $SU(2)$ symmetry, preserving only rotations around the ordering axis. (See figure 4.5.3)

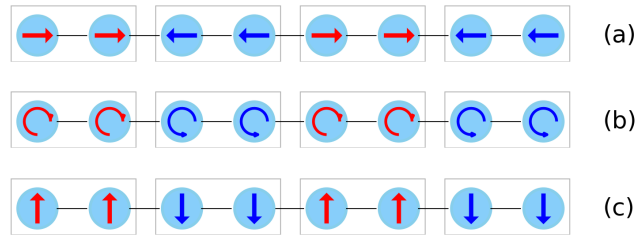


Figure 4.5.3. Standard $S = 1$ Néel states aligned along the (a) X-axis, (b) Y-axis, and (c) Z-axis, illustrating antiferromagnetic spin ordering in each direction.

Furthermore, the competition between D and E anisotropies stabilizes three trivial product phases: Large- D , Large- E_x , and Large- E_y . Each suppresses different spin components and leads to quadrupolar or paramagnetic states, adiabatically

connected to simple product configurations. Despite their differences, they are symmetry-related through spin-axis permutations, reflecting the rotational symmetry of the (D, E) phase diagram [155, 196, 202].

Finally, we also explored the anisotropic spin-1 Haldane chain from a QML perspective in chapter 7. By training supervised models on quantum states obtained with DMRG, we showed that phase recognition remains accurate even when only reduced subsystems are used, thanks to the information retained through Schmidt decomposition. This result demonstrates that reliable classification can be achieved with limited data, offering a data-efficient route to studying quantum phases when full system access is computationally or experimentally challenging.

4.6 Summary and Outlook

This chapter has presented a comprehensive study of the phase diagrams of representative many-body quantum models, with emphasis on both theoretical perspectives and computational approaches.

We began with an overview of the principal simulation methods employed in the analysis of quantum phases. The discussion included the role of order parameters and phase indicators in characterizing phase transitions, as well as a review of standard numerical techniques such as ED, QMC, and TNs methods, outlining their respective advantages and limitations.

The subsequent sections focused on three paradigmatic models. For the ANNNI model, we reviewed the existing theoretical formulations and computational investigations, with particular attention to studies employing TNs techniques and QML to resolve the complex structure of its phase diagram.

For the cluster Ising model, we provided a review of both theoretical and numerical studies, highlighting the role of TNs and machine learning approaches in extending the understanding of this model. Particular emphasis was placed on the analysis of symmetry-protected topological phases, reflecting the importance of the model in the study of topological order.

Finally, the Haldane anisotropic spin-1 model was examined in detail. We discussed prior theoretical and numerical work, again incorporating TNs and QML studies, with a focus on the characterization of its Haldane phases.

In summary, this chapter has provided a comprehensive review of the literature on many-body quantum models, with particular emphasis on the ANNNI, cluster Ising, and Haldane anisotropic spin-1 models. By surveying studies based on traditional numerical methods alongside more recent developments in TNs and QML. Collectively, these works reveal the progress made in understanding their phase diagrams, while also pointing to the open challenges and opportunities for future research in the exploration of quantum phases.

Chapter 5

Learning to Classify Quantum Phases of Matter with a Few Measurements

The work presented in this chapter forms the basis of a paper published in *Quantum Science and Technology*, whose authors are (in order) Mehran Khosrojerdi, Jason L. Pereira, Alessandro Cuccoli, and Leonardo Banchi [215].

This chapter begins with a brief introduction, followed by a clear statement of the problem. We then present the methodology and discuss the results, focusing on the ANNNI and Cluster–Ising models.

5.1 Introduction

Identifying phases of quantum matter is a complex task [216, 217] that requires the extraction of possibly non-local features from the ground states of quantum many-particle systems. While phases described by the Landau–Ginzburg paradigm can often be identified using simple local order parameters, more exotic quantum phases lacking such local descriptors require more sophisticated tools.

Given the recent remarkable success of machine learning methods to extract general laws from a few known examples [218], many studies applied machine learning techniques to learn quantum phases of matter [36, 168, 169, 219–223] – see [218] for more references. Among the different learning methods that can be applied to “quantum data” [49], such as the ground states of quantum many-particle systems, kernel methods stand out for their interpretability, since learning consists in finding a decision hyperplane in a Hilbert space [47, 49, 51]. The problem of classifying quantum phases of matter with kernel methods has been recently considered in Refs. [224, 225].

In a quantum phase classification problem, the state we want to classify is the ground state of a parametrised Hamiltonian with a specific form. For some parameter values we may have full (classical) knowledge of the phases. In many physically relevant scenarios, this comes from our ability to classically simulate the true ground state of the Hamiltonian, e.g., using MPS or other TNs methods [71, 226]. This suggests a machine learning approach where training is purely classical, performed

on states within the known parameter region, with the ultimate aim of learning a general rule (e.g., an observable or a measurement) that can be applied to classify ground states unseen during training.

One physically motivated reason to *test* a machine learning model with different ground states may be that classical approximation methods are not accurate in that parameter regime. A central result of our analysis is that, if such ground state may be prepared in a quantum device such as quantum simulators [227], e.g., by cooling a quantum many-body system, the classification of new ground states, unseen during training, can be obtained with single-qubit measurements. This result is derived by pairing kernel methods, tensor network techniques, generalization bounds [228, 229], and the certification strategy from Ref. [230]. Our algorithm is hybrid, since training is performed classically, while testing new states can be done in different ways, e.g., in quantum devices. Since single-qubit measurements are routinely performed in quantum simulators, an universal quantum computer is never required. Nonetheless, the same approach can also be applied to different situations, e.g., when we have classical knowledge of the test ground states or when they are approximated in a quantum computer via parametric quantum circuits [231].

5.2 Problem description

We are interested in getting an approximation of the phase diagram for the ground state of a quantum many-particle Hamiltonian $H(\mathbf{x})$ that depends on classical external parameters $\mathbf{x} = (x_1, x_2, \dots)$. We focus on settings satisfying two main assumptions: i) the ground state $|\psi(\mathbf{x})\rangle$ of $H(\mathbf{x})$ can be obtained with sufficiently high precision, for certain values of $\mathbf{x}$, using classical methods based on tensor networks [226]; ii) within the parameter subspace where the above methods are accurate, there is a (usually smaller) subspace $\mathcal{S}$ where the phase of $|\psi(\mathbf{x})\rangle$ is known for all $\mathbf{x} \in \mathcal{S}$. The task is to use our knowledge from subregion $\mathcal{S}$ to construct an observable Λ that allows us to predict the phase of a ground state. If we are given copies of a state $|\psi(\mathbf{x})\rangle$ for unknown $\mathbf{x} \in \mathcal{S}$ (e.g., as the outcome of an experiment), we can evaluate the expectation value of Λ to determine its phase (in-distribution generalization). Alternatively, for a test point outside the known subregion ($\mathbf{x} \notin \mathcal{S}$), we can use Λ to learn the phase diagram outside $\mathcal{S}$ (out-of-distribution generalization) with the help of a quantum computer or simulator, e.g., a cold-atom experiment, able to efficiently find the ground state when $\mathbf{x} \notin \mathcal{S}$.

To construct Λ we use a supervised learning approach, where we first build a training set $\{|\psi(\mathbf{x}_n)\rangle, y_n\}$, for different choices of $\mathbf{x}_n$ from $\mathcal{S}$, with our best approximation of the true ground state of $H(\mathbf{x}_n)$, $|\psi(\mathbf{x}_n)\rangle$, and its known phase y_n . The index n is a discrete index labelling the M training pairs. To simplify the presentation, we use the same notation $|\psi(\mathbf{x})\rangle$ for the true ground state and our best approximation of it. In choosing the training algorithm we considered two desirable properties: i) it should work efficiently with both classical (e.g., tensor network) and shallow quantum circuit approximations of many-body quantum states whose Hilbert space is exponentially large, and ii) it should give, as an output, a quantity that can be efficiently measured on a quantum device. Because of those requirements, we decided to focus on kernel methods [49].

For any two points $\mathbf{x}_n$ and $\mathbf{x}_m$, the kernel is defined as

$$k(\mathbf{x}_n, \mathbf{x}_m) = \text{Tr}[\rho(\mathbf{x}_n)\rho(\mathbf{x}_m)] = |\langle \psi(\mathbf{x}_n) | \psi(\mathbf{x}_m) \rangle|^2, \quad (5.2.1)$$

where $\rho(\mathbf{x}) = |\psi(\mathbf{x})\rangle\langle\psi(\mathbf{x})|$. Both training and testing in kernel methods only require the above kernel matrix, and never use the full ground states, which live in an exponentially large Hilbert space. As such, it can be easily extended to settings involving many-particle systems, provided that the inner product (5.2.1) can be efficiently computed, as with tensor networks, shallow quantum circuits, or a combination of both [230, 232, 233]. Another, more physical, reason to focus on kernel methods for classifying phases of matter is that, for nearby points $k(\mathbf{x}, \mathbf{x} + \mathbf{x})$, the kernel reduces to the fidelity susceptibility, which has been extensively used to predict phase transitions [234–236]. Therefore, it is reasonable to expect that the kernel entries (5.2.1) contain all the information about the quantum phases.

In kernel methods, training consists in finding a decision function $y = f(\mathbf{x})$ that can infer the label y (namely the quantum phase) given a certain input $\mathbf{x}$. The training algorithm involves a convex optimization routine, which always reaches the global optimum, and whose outcome is a weight vector $\boldsymbol{\alpha}$ of real numbers, together with the baseline $b \in \mathbb{R}$. From $\boldsymbol{\alpha}$ and b , the algorithm constructs the *decision function* as a “kernel expansion of the data” $\sum_n \alpha_n y_n k(\mathbf{x}_n, \mathbf{x}) + b$, where $\mathbf{x}_n$ are the points used during training, and $\mathbf{x}$ is a new point, possibly not belonging to $\mathcal{S}$, that we want to test. The crucial observation, already noted in [49], is that the kernel expansion can also be expressed as the expectation value of an observable Λ over the state $\rho(\mathbf{x})$. For instance, in the simplest case of binary classification, with two phases labeled as $y = \pm 1$, the model prediction is expressed as

$$y_{\text{predicted}} = \text{sign} \left(\sum_n y_n \alpha_n k(\mathbf{x}_n, \mathbf{x}) + b \right) = \quad (5.2.2)$$

$$= \text{sign} (\langle \psi(\mathbf{x}) | \Lambda | \psi(\mathbf{x}) \rangle + b), \quad (5.2.3)$$

where in the second line we have defined the observable

$$\Lambda = \sum_n y_n \alpha_n \rho(\mathbf{x}_n), \quad (5.2.4)$$

as a weighted combination of density matrices. In other words, kernel methods allow for the construction of a *decision operator* Λ that, being Hermitian, can be measured in a quantum device capable of preparing the state $|\psi(\mathbf{x})\rangle$ even for those values of $\mathbf{x}$ for which the true phase is unknown. Decision functions with multiple labels are mapped into multiple binary decisions, each with a different decision observable. For numerical results, as a classical training algorithm, we focus on SVM [47], which can be computed efficiently using available numerical libraries [48].

We now discuss the evaluation of the kernel entries (5.2.1). For training and in-distribution generalization, or in any other case when the true states are replaced by their tensor network approximations, inner product (5.2.1) can be computed efficiently on classical hardware [71]. When the true states are approximated via a shallow quantum circuit, it can be computed using different methods such as the swap test or inversion test [49]. For out-of-distribution generalization where the

overlap is hybrid, between a *classical* and a quantum state, single-qubit measurement suffice [230], provided each amplitude of the classical state can be computed efficiently. When the classical state is represented as an MPS, each amplitude requires $\mathcal{O}(N)$ operations, where N is the number of qubits. Alternatively, each state $\rho(\mathbf{x}_n)$ can be mapped into a *shallow* quantum circuit [232, 233], which can be implemented in a quantum device without exponential overhead.

Given the semi-analytic structure of Λ , we can also express the variance as

$$\text{Var}(\Lambda)_{\rho(\mathbf{x})} = \sum_{nm} \alpha_n \alpha_m y_n y_m \left(\text{Tr}[\rho(\mathbf{x}_n) \rho(\mathbf{x}_m) \rho(\mathbf{x})] - \text{Tr}[\rho(\mathbf{x}) \rho(\mathbf{x}_n)] \text{Tr}[\rho(\mathbf{x}) \rho(\mathbf{x}_m)] \right), \quad (5.2.5)$$

which can be classically estimated efficiently for the in-distribution case ($\mathbf{x} \in \mathcal{S}$), allowing the learner to infer the number of measurements to be performed in the quantum device to achieve the desired precision. From Hölder's inequality, we get the bound $\text{Var}(\Lambda) \leq (\sum_n \alpha_n)^2 \leq (CN_S)^2$, where N_S is the number of support vectors and C is the regularization parameter – typically $C = 1$. Clearly $N_S \leq M$, where M is the number of training data. Therefore, the number of measurement shots to be performed scales at most quadratically in the number of training data. We will numerically show, using the generalization bounds from [228, 229], that, for the in-distribution case, the number of training data to reach a desired accuracy scales linearly with the number of qubits. When this is the case, the overall complexity of our method scales polynomially with the number of qubits. When paired with [230], the whole measurement will require a polynomial number of single-qubit measurements.

5.3 Handling symmetries

Symmetries may introduce degeneracies in the ground state. Consider the simplest case with two degenerate ground states labelled by $|\psi_{\pm}(\mathbf{x})\rangle$. For instance, if $H(\mathbf{x})$ commutes with the parity operator, as in one of our numerical results, $\pm$ may label the parity, and $|\psi_{\pm}(\mathbf{x})\rangle$ will be orthogonal, having two different parities. If the symmetry is not enforced by the diagonalization procedure, the ground state appears as a superposition

$$|\psi_c(\mathbf{x})\rangle = \alpha_+ |\psi_+(\mathbf{x})\rangle + \alpha_- |\psi_-(\mathbf{x})\rangle, \quad (5.3.1)$$

with possibly random coefficients $\alpha_{\pm}$. Those coefficients represent a problem for kernel methods, as they introduce arbitrariness in the kernel entries (5.2.1), which might become random. To avoid this, during training we compute $|\psi_{\pm}(\mathbf{x})\rangle$, e.g., by implementing the symmetry in the tensor network [237] and insert both states into the training set. During test on the other hand, we assume we have less control and we do not implement any of those strategies. Therefore, each test ground state may appear as a possibly random combination of different states. This will be the case in ground states created by quantum devices, unless symmetries are explicitly enforced [238]. Nonetheless, at test the model should have learnt to separately classify the states with different parity, and so it should now be able to classify even states of the form (5.3.1).

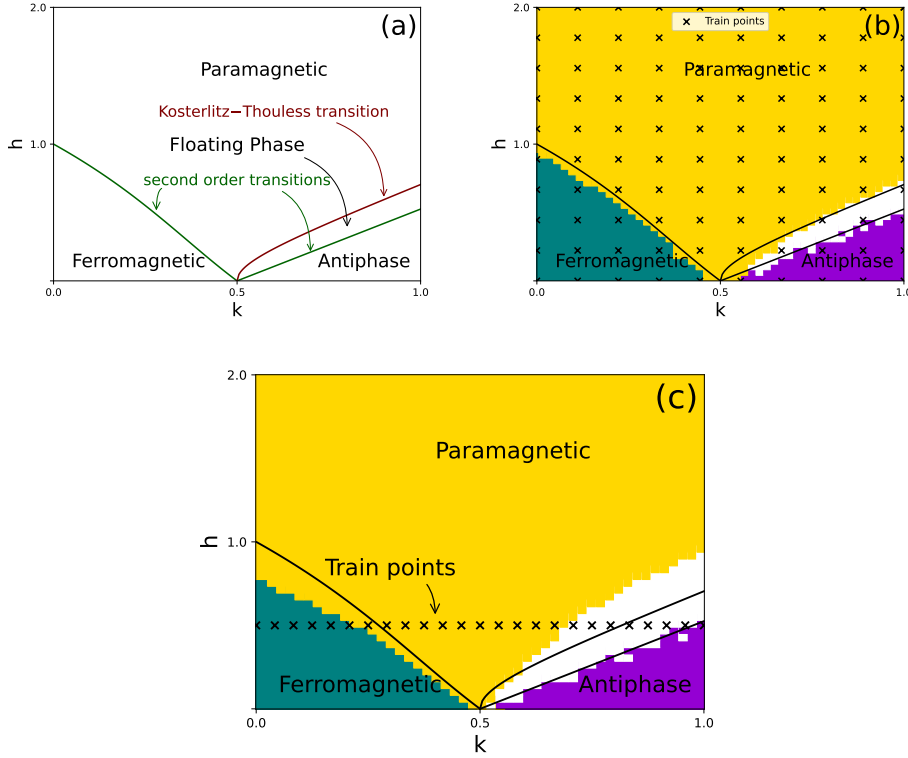


Figure 5.2.1. (a) Expected phase diagram of the ANNNI model in the thermodynamic limit. (b-c) Predicted phase diagram for the ANNNI Hamiltonian (5.4.1) with $N = 50$ spins, where the ground state is simulated using tensor network methods with a small bond dimension (20). The training points are marked with a black cross pints. In (b) the training point are uniformly scattered, while in (c) they are located on the line $h = 0.25$.

5.4 Results

5.4.1 Axial Next-Nearest-Neighbor Ising model

As a first example, we focus on the one-dimensional ANNNI model [157, 160, 162, 239], which describes N spin- $\frac{1}{2}$ particles, with the Hamiltonian

$$H = - \sum_{j=1}^{N-1} \sigma_j^x \sigma_{j+1}^x + k \sum_{j=1}^{N-2} \sigma_j^x \sigma_{j+2}^x - h \sum_{j=1}^N \sigma_j^z, \quad (5.4.1)$$

where the two parameters $\mathbf{x} = (k, h)$ are both assumed to be positive. The above model displays a global $\mathbb{Z}_2$ symmetry, as $[H, P] = 0$, where $P = \bigotimes_n \sigma_n^z$ is the parity operator. Therefore, energy eigenstates can be degenerate and come as pairs with different parities.

The Hamiltonian (5.4.1) models ferromagnetic interactions between the x -components of nearest neighbouring spins (NN), and antiferromagnetic interactions (with relative strength k) between the same components of next-nearest neighbouring spins (NNN), together with the action of a magnetic field of strength h along the z direction. For $k = 0$ the model is exactly solvable using fermionization [240], while for $h = 0$ the

model is effectively classical, as the Hamiltonian is diagonal in the $\mathbf{x}$ basis. For $k > 0$, the system is frustrated, as the first term favours spin configurations with all spins aligned along the $\mathbf{x}$ axis, while the second term favours configurations where next-nearest spins are anti-aligned.¹

The phase diagram of the above model, shown in Fig. 5.2.1(a), has been characterised with different methods [162, 166, 170, 241–243], and the possibility of using tensor network techniques to compute its ground state was analyzed in [166, 167]. For small h and small k , the nearest neighbour interaction dominates, and the model displays ferromagnetic order; for sufficiently large h the model is in a paramagnetic phase, while for large k and suitably small h the next-nearest neighbor interaction dominates and the model displays the so-called antiphase (AP), where pairs of aligned spins alternate pointing in opposite direction. Between the AP and the paramagnetic phase the model displays a floating phase, where correlation functions $\langle \sigma_j^x \sigma_{j+l}^x \rangle - \langle \sigma_j^x \rangle^2$ decay algebraically rather than exponentially, and the Hamiltonian becomes gapless in the thermodynamic limit $N \rightarrow \infty$.

Aside from these general properties, the phase diagrams obtained with different methods do not always agree [164, 244]: Indeed, while there is a general consensus over the different phases schematically depicted in Fig. 5.2.1(a), some works split the paramagnetic phase into two different phases [170], while others found the floating phase over an extended region. For the phase diagram of Fig. 5.2.1, the transition lines between the different phases have been found using different perturbative and numerical methods to approximately be [166, 168]

$$h_I(k) \simeq \frac{1-k}{k} \left(1 - \sqrt{\frac{1-3k+4k^2}{1-k}} \right), \quad (5.4.2a)$$

$$h_{KT}(k) \simeq 1.05 \sqrt{(k-0.5)(k-0.1)}, \quad (5.4.2b)$$

$$h_{AP}(k) \simeq 1.05 (k-0.5). \quad (5.4.2c)$$

Our numerical results for the ANNNI model are shown in Fig. 5.2.1 for a chain with $N = 50$ spins. The ground states were obtained using the DMRG algorithm implemented in the `quimb` library [245]. In spite of the small bond dimension (20), the predicted phase diagram is in good agreement with the theoretical prediction, shown in Fig. 5.2.1, with differences only near the phase transition points. These differences might be due to finite size effects, as the phase diagram in Fig. 5.2.1 is expected in the thermodynamic limit $N \rightarrow \infty$, and due to the finite bond dimension, as close to the phase transition point the entanglement area law breaks down [246] and a larger bond dimension might be needed. This highlights a use case in which a classically learned observable could be measured in a shallow quantum circuit for ground states near the transition point in order to locate the phase boundary with greater precision than is possible purely classically.

¹Actually, the sign of the nearest neighbors interaction is not really relevant, as the Hamiltonian can be mapped into itself by changing the NN ferromagnetic interaction into an anti-ferromagnetic one, while rotating by an angle π the spins on the even sites by a $\mathbb{Z}_2$ transformation $\bigotimes_{n \text{ even}} \sigma_n^z$, explicitly proving that the frustrated nature of the model does not depend on the sign of the NN interaction, as two spin configurations cannot display exact anti-ferromagnetic order between next-nearest neighbours either when nearest neighbours are aligned or when they are anti-aligned.

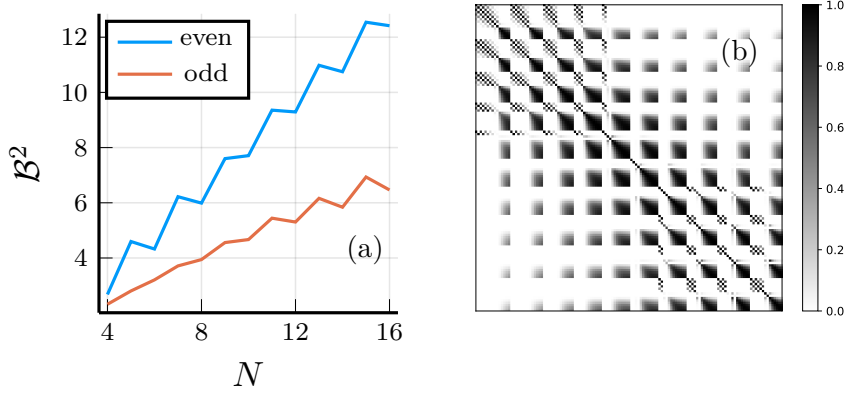


Figure 5.4.1. (a) Scaling of the generalization error bound for the ground state of the ANNNI model with $h \in [0, 2]$, $k \in [0, 1]$, as a function of the number of qubits N , and for the two different parity sectors (even and odd). (b) Kernel matrix entries for the training states shown in Fig. 5.2.1.

In Fig. 5.2.1 we trained the model in two different ways, and then tested over the same parameter region. In panel (b), the training points are uniformly spaced (in-distribution), while in panel (c) they are chosen along a line that passes through all four different phases (out-of-distribution). Prediction in (b) is simpler, as the model has to interpolate over the training points, while in (c) it is more complex as the model has to extrapolate, resulting in a lower accuracy. Nonetheless, in both cases the model displays remarkable generalization abilities given the limited amount of data.

The ability of the training algorithm to generalize, namely to predict the phase of ground states outside of the training set, can be explained using the language of Ref. [228, 229] when the training and test data belong to the same distribution. Indeed, the generalization error, namely the difference between an optimal classifier with full knowledge of the data distribution and an empirical classifier that only has access to M samples from that distribution, is upper bounded by $\mathcal{B}/\sqrt{M}$ where $\mathcal{B}$ depends on the mutual information between the parameter space and the set of possible ground states. The in-distribution generalization error is small as long as $M \gg \mathcal{B}^2$. The values of $\mathcal{B}^2$ are shown in Fig. 5.4.1(a), where we show that $\mathcal{B}^2$ increases at most linearly with the number of qubits. Therefore, in the worst case, at most a number of training points that scales linearly with the number of qubits is sufficient to ensure a good generalization, in spite of the exponentially growing Hilbert space. In practice though, as discussed in [229], because of the regularization term, the above bound might be loose and the model might be able to generalize even with less data. The good generalization capabilities are due to the “clustering” of the ground states within the different phases, which is evident from the kernel matrix entries shown in Fig. 5.4.1(b). In that figure, the “tiled” subregions are due to the degeneracy of the ground states, which can be labeled by the parity and are mutually orthogonal.

When training and test data belong to different distributions, as in Fig. 5.2.1(c), we need to employ out-of-distribution bounds, which have been discussed in the

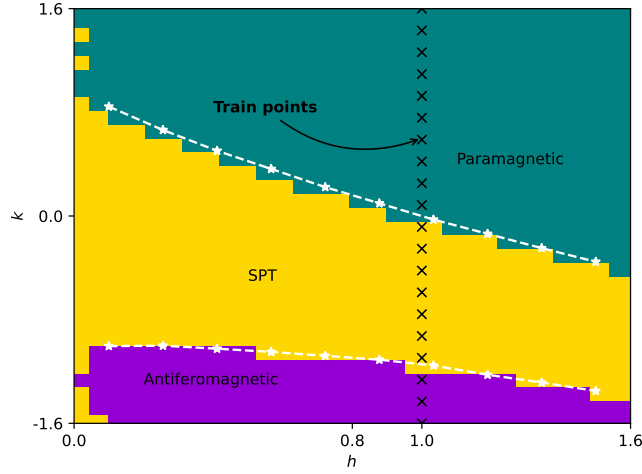


Figure 5.4.2. Predicted phase diagram for the Hamiltonian (5.4.3) with $N = 41$ spins, where the ground state is simulated using tensor network methods with a bond dimension 150. The training points, marked with black crosses, are located along the line $h = 1$. The phase transition lines (dashed white) are estimated by interpolating the numerically observed points (white asterisks) from Ref. [36].

quantum case only for different settings [63].

5.4.2 Cluster-Ising Model

As a second model, we focus on the one-dimensional SPT spin model described by the Hamiltonian [36]

$$H = - \sum_{j=1}^{N-2} \sigma_j^x \sigma_{j+1}^z \sigma_{j+2}^x - k \sum_{j=1}^{N-1} \sigma_j^z \sigma_{j+1}^z - h \sum_{j=1}^N \sigma_j^z, \quad (5.4.3)$$

with parameters $\mathbf{x} = (h, k)$. When $k = 0$ the model is exactly solvable via the Jordan-Wigner transformation [240]. Such a Hamiltonian has a $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry generated by the operators $\hat{X}_{\text{even(odd)}} = \prod_{i \in \text{even(odd)}} \sigma_i^z$. The ground state of H displays, as a function of h and k , a paramagnetic phase, an antiferromagnetic phase and a $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry-protected topological (SPT) phase. The latter can be detected by non-zero string order parameters $S_{ab} = \sigma_a^x \prod_{a < i < b} (\sigma_i^z) \sigma_b^x$ or with a quantum convolutional neural network circuit [36].

To train the model, we fix $N = 41$ and provide just the known information from Refs. [36, 247] that the model belongs to the SPT phase for $h = 1$ and $-1.15 \leq k \leq 0$, to the paramagnetic phase for $k > 0$ and to the antiferromagnetic phase for $k < -1.15$. The 50 training points are displayed with black crosses in Fig. 5.4.2. We then test the model over the entire phase space shown in Fig. 5.4.2, discretized as a 30×30 grid. For comparison, we also draw the phase transition lines obtained in [36]. From the results shown in Fig. 5.4.2 we see that the SVM classifier is able to accurately recognise all three phases, generalizing the given information from the states along the $h = 1$ line. Imperfections around the phase transition lines are possibly due to finite size effects or due to the small bond-dimension in the tensor network. Misclassifications around the $h = 0$ line are comparable to those obtained with other

methods [36, 247] and may be exacerbated by the fact that the states around $h = 0$ might be qualitatively quite different from the training states along $h = 1$.

5.5 Conclusions

We studied the classification of quantum phases of matter using a combination of classical and quantum techniques. We studied how to use knowledge from part of a phase diagram to populate other, possibly unknown, parts of the phase diagram. The result of our learning approach is a set of observables that can be measured in quantum devices, with a number of measurements that scale polynomially in the system size. We studied different models with rather rich phase diagrams and showed that our classifier is able to both interpolate and extrapolate the information from the training set to predict the quantum phase for new sets of parameters.

Chapter 6

Unsupervised Learning to Recognize Quantum Phases of Matter

The work presented in this chapter forms the basis of a paper currently available on arXiv (2510.14742) and under review for publication in Quantum Science and Technology, authored by (in order) Mehran Khosrojerdi, Alessandro Cuccoli, Paola Verrucchi, and Leonardo Banchi [248].

This chapter begins with a brief introduction, followed by a detailed explanation of the feature map. We then present the cluster method as an unsupervised model and discuss the results, focusing on the ANNI and Cluster-Ising models.

6.1 Introduction

Quantum many-body systems can exhibit phase-transitions also at $T = 0$ due to purely quantum fluctuations [216], making it possible for Ground States to feature different collective properties for infinitesimally different values of the hamiltonian parameters. Studying the Quantum Phase Diagrams (QPD) outlined by these transitions, as well as the quantum phases themselves, is of interest not only in the realm of many-body physics but also in quantum information processing, whose operation happens at temperatures as near as possible to $T = 0$. On the other hand, drawing the QPD of quantum many-body systems is a hard task, unless one considers very simple models such as, for instance, the Ising chain in a transverse field. In fact, analytical methods are of limited help in studying models of realistic systems, while established numerical techniques demand huge computational resources, that become unaffordable as the size of the models grows, as required by an evaluation of non-local quantities that characterize different phases, such as long-range correlations and concurrence, or multipartite entanglement.

In this framework, approaches based on hybrid quantum-classical machine-learning methods provide some alternatives. One possibility comes from quantum-phase classification based on training on labeled data, where some prior theoretical or numerical understanding of the phase-diagram is necessary, at least in a limited parameter region. This approach matches a supervised learning task, where the

phases are known for some particular parameter values that are used to train the classification algorithm. The algorithm is then used to explore larger areas of the parameter space and label the corresponding phase [36, 168, 169, 215, 218–223, 249]. Quantum states clustering, instead, uses an unsupervised learning strategy, whose goal is to identify similarities among ground states, given no prior knowledge of the QPD, and hence without pre-existing labeling of the different phases. Unlike classification, which is typically applied as a validation tool, clustering can be used to explore novel quantum systems, revealing hidden structures in the data, and uncovering emergent phases of matter without relying on predefined categories [250–253]. Needless to say, clustering quantum states to unveil a phase-diagram is a challenging task, due to the inherently non-local nature of quantum correlations and entanglement [216, 217]: in fact, when the Hilbert space dimension grows, identifying features for an effective clustering is by itself a difficult task. On the other hand, some unsupervised methods [52, 211, 254–256], even if acting on classical data-sets, can turn out to be suitable to spot similarities between elements of large Hilbert spaces, providing an interpretable framework for clustering quantum states based on their genuinely quantum properties.

Scope of this work is to develop a method for quantum-states clustering, precise enough to match QPD of parameterized Hamiltonians whose ground state can be determined and represented via TNs methods, such as the DMRG [80, 90, 257]. Since our method uses a “kernel” between two Hamiltonian parameter sets given by the fidelity between the corresponding ground states, it can be extended to work directly with ground states approximated in quantum devices, since the fidelity between states can be experimentally estimated using different methods such as the SWAP test or the Hadamard test, see e.g. [258].

Our procedure goes as follows: be $\hat{H}(\mathbf{x})$ a one-dimensional Hamiltonian that fulfills the above requirements, with $\mathbf{x} = (x_1, x_2, \dots, x_q)$ the set of q parameters upon which it depends. We take four steps for predicting the QPD with the desired insight into different quantum phases.

1. Generate D sets of Hamiltonian parameters $\{\mathbf{x}_i = (x_{1i}, x_{2i}, \dots, x_{iq})\}$ according to some rule (e.g. uniformly in the parameter space), with $i = 1, \dots, D$. This set defines our “classical data”, where each datum is an element of $\mathbb{R}^q$ and a point in the phase-diagram.
2. Represent the Hamiltonian $\hat{H}(\mathbf{x}_i)$ as a MPO for each datum $\mathbf{x}_i$ and find its ground state $|\text{GS}(\mathbf{x}_i)\rangle$.
3. Define the *kernel* between two data $\mathbf{x}_i$ and $\mathbf{x}_j$ as $K_{ij} = |\langle \text{GS}(\mathbf{x}_i) | \text{GS}(\mathbf{x}_j) \rangle|^2$, namely as the fidelity between two ground states $|\text{GS}(\mathbf{x}_i)\rangle$ and $|\text{GS}(\mathbf{x}_j)\rangle$. In the language of machine learning, this corresponds to defining the *feature space* as the $\mathcal{O}(4^N)$ -dimensional space of density matrices $|\text{GS}(\mathbf{x}_i)\rangle\langle\text{GS}(\mathbf{x}_i)|$, where N is the number of qubits in $\hat{H}(\mathbf{x})$. Note that the dimensionality of the feature space is much larger than that of the dataset, which is one of the key-element of any machine-learning procedure for clustering, as it reflects the richer characterization offered by features w.r.t. data.
4. Perform a kernel clustering algorithm, which only depends on the kernel matrix

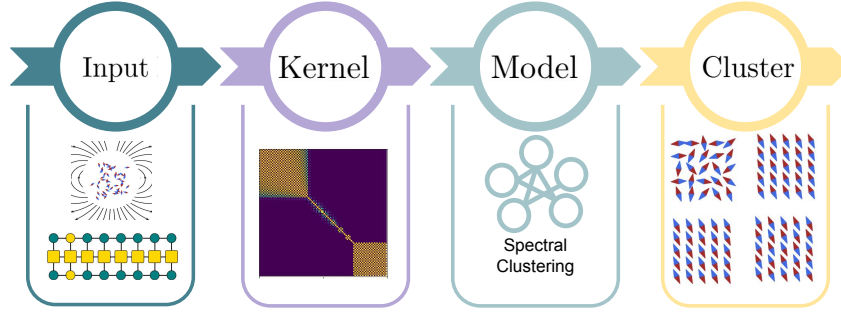


Figure 6.1.1. Outline of the clustering procedure for quantum ground states, as described in the introduction. The process begins with randomly producing ground states of a target Hamiltonian, either using classical approximations (e.g. TNs), a quantum hardware with physical spins, or a quantum computer. The kernel is then estimated either in hardware or via tensor network techniques. Next, a kernel-driven clustering model groups the ground states based on their correlation structure, capturing quantum similarities, including entanglement properties. Finally, we present the resulting clusters.

K , so as to group ground states according to the similarity amongst their quantum features.

Notice that, although not explicitly considered in this paper, steps 2. and 3. can be replaced with:

- 2b. Approximate each ground state $|\text{GS}(\mathbf{x}_i)\rangle$ in a quantum device, e.g. using a parametric quantum algorithm and the VQE [259] or more recent versions [260].
- 3b. Compute the kernel $K_{ij} = |\langle \text{GS}(\mathbf{x}_i) | \text{GS}(\mathbf{x}_j) \rangle|^2$ using the quantum hardware, e.g. via SWAP test or alternative methods [49].

In any case, possible approximations must be considered. With TNs the approximation follows from the finite value of the bond dimension, which we denote by χ . On the other hand, when performing experiments in a quantum hardware, the approximations follow from the poor convergence of VQE, the imperfect estimation of the kernel (due to shot noise), and the imperfect quantum gates and measurements. Be that as it may, the end effect is a noisy estimate of the kernel matrix. In this paper, we deliberately use TNs with small χ , so as to assess the performances of our clustering technique under imperfect data.

A schematic view of the overall process is provided in Fig. 6.1.1. In order to benchmark the reliability of the obtained QPD we will demand that, within the domain where DMRG remains accurate, there exists at least one subregion of the parameter-space in which the phase to which the ground state belongs is already known and possibly characterized.

The remainder of this article is organized as follows. Section 6.2 introduces the dataset construction. Section 6.3 describes the kernel-based unsupervised model and the mathematical methods used to determine the optimal number of clusters. Section 6.4 presents benchmark studies on the Axial Next-Nearest-Neighbor Ising (ANNNI) and Cluster-Ising models. Finally, Section 6.5 summarizes the findings and highlights the key contributions of this work.

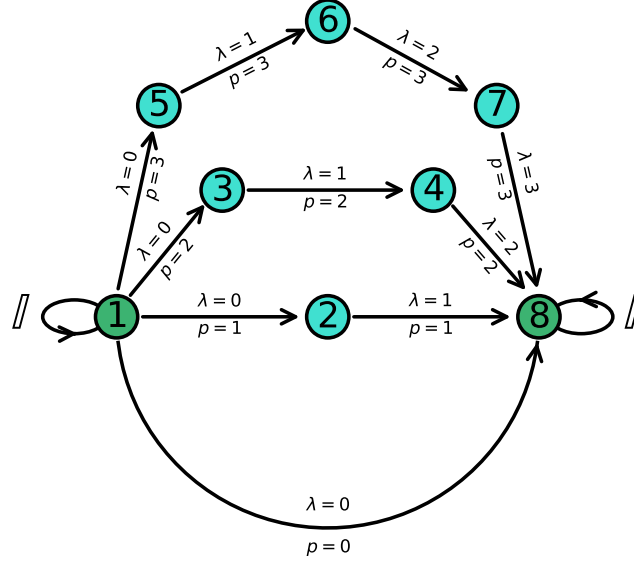


Figure 6.2.1. Graphical representation of a typical local Hamiltonian, illustrating how the automaton structure translates into the explicit MPO representation.

6.2 From classical data to quantum features

We consider one-dimensional spin- $\frac{1}{2}$ models, with short enough interaction-range for DMRG to be efficient in numerically determine their ground state [36, 63, 162, 166, 167, 215, 240, 241, 261]; these models often exhibit nontrivial QPD that we explore by moving around in their parameter space, which is $\mathbb{R}^q$. As explained in the introduction, each point in this space is a datum $\mathbf{x}_i = (x_{i1}, x_{i2}, \dots, x_{iq})$, in one-to-one correspondence with the Hamiltonian $\hat{H}(\mathbf{x}_i)$ and its ground state $|GS(\mathbf{x}_i)\rangle$. We wander erratically in this parameter-space, randomly generating D data $\{\mathbf{x}_i\}$, $i = 1, \dots, D$, assuming no prior knowledge about the different phases, let alone the QPD.

Specifically, we will consider model-Hamiltonians $\hat{H}$ that can be decomposed as $\hat{H} = \sum_{j=1}^N \hat{H}_j$, where each local Hamiltonian $\hat{H}_j$ involves a small number of sites near j , thus acting not trivially only on the subspace of a limited number of spins, so that a compact and computationally efficient representation in terms of MPO is possible. In fact, for Hamiltonians defined on an open chain, it can be shown that

$$\hat{H} = \hat{\ell}_1 \hat{W}_2 \hat{W}_3 \cdots \hat{W}_{N-1} \hat{r}_N, \quad (6.2.1)$$

where each matrix element $(\hat{W}_j)_{lm}$ is an operator acting on the j th spin Hilbert space and the boundary terms are such that $\hat{\ell}_j$ and $\hat{r}_j$ are the first row and the last column of the $\hat{W}_j$ matrix, respectively. The multiplication in Eq. (6.2.1) corresponds to a matrix product in the representation space and a tensor product in the physical one [262]; the specific structure of the operator-matrix $\hat{W}$ is not identifiable at first sight, but can be obtained with the help of a graphical representation borrowed from finite-state automata, as we explain below.

Be d the distance (expressed in number of lattice sites) between the furthest interacting spins: terms in the Hamiltonian involve at most $d+1$ single-spin operators.

We assume that the local Hamiltonian $\hat{H}_j$ can be written as

$$\hat{H}_j = \sum_{p=0}^d \bigotimes_{\lambda=0}^p {}^p\hat{h}_{j+\lambda}, \quad (6.2.2)$$

where the index in brackets indicates that the single-site operator ${}^p\hat{h}_{(j+\lambda)}$ acts upon the system sitting at site $j + \lambda$. For instance, if the model includes next to next-nearest neighbours interactions, it is $d = 3$ and we can expect tensor products of at most $d + 1 = 4$ single-spin operators in each $\hat{H}_j$. A graph of numbered nodes and labelled links can be drawn from Eq. (6.2.2) as shown in Fig.(6.2.1): $d + 1 = 4$ paths spout from node 1, each path formed by an increasing number of links, from 1 to $d + 1 = 4$, and each ending in the last node, numbered $d(d + 1)/2 + 2 = 8$. Different paths stand for different types of interaction, with the number of links equal to the number of single-site spin operators involved. Referring to Eq. (6.2.2) this means that paths are labelled by the index p . Paths with increasing number of links are drawn from bottom to top so that the lowest refers to local terms, the second one to nearest-neighbours interaction, the third one to next-nearest neighbours interaction, and so on. Links are labeled by the pair (p, λ) , with p identifying the path to which they belong, and λ counting their position w.r.t. the first node, so that the first link of each path has $\lambda = 0$, the second one $\lambda = 1$ and so on, until the $(d + 1)(d + 2)/2 = 10$ th link, having $\lambda = d = 3$ (that only appears in the $d + 1 = 4$ th path). Finally, nodes are numbered in increasing order from bottom to top and left to right.

Once the graph is drawn, the operator-matrices in Eq. (6.2.1) have dimension equal to the number of nodes, i.e. $d(d + 1)/2 + 2 = 8$, with elements $(\hat{W}_j)_{lm}$ different from zero if there exists a link (p, λ) connecting nodes l and m , in which case it is $(\hat{W}_j)_{lm} = {}^p\hat{h}_j$. Specific examples of such graphs, together with their corresponding operator matrices, are shown in Fig. 6.2.1 and Equation 6.2.3, respectively

$$\hat{W}_j = \begin{pmatrix} \mathbb{I} & {}^1_0\hat{h}_j & {}^2_0\hat{h}_j & 0 & {}^3_0\hat{h}_j & 0 & 0 & {}^0_0\hat{h}_j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & {}^1_1\hat{h}_j \\ 0 & 0 & 0 & {}^2_1\hat{h}_j & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & {}^2_2\hat{h}_j \\ 0 & 0 & 0 & 0 & 0 & {}^3_1\hat{h}_j & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & {}^3_2\hat{h}_j & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & {}^3_3\hat{h}_j \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbb{I} \end{pmatrix}. \quad (6.2.3)$$

The graphical representation encodes the same operator structure as the matrix in Eq. (6.2.3), but in a more compact and intuitive form. Each edge (green circles) in the graph corresponds to a possible transition weighted by a local operator, while the adjacency-like structure (blue circles) directly translates into the block entries of the operator matrix. In this way, the finite-state automaton view and the matrix formulation provide two complementary perspectives of the same MPO construction.

MPO representations allow for an adequate implementation of variational algorithms such as the DMRG, whose numerical efficiency relies on the compactness of such representation. Within the algorithm, an initial state is iteratively modified via

a unitary process that depends on the parameters $(\mathbf{x}_i)$ so as to minimize its energy. Without entering into the details of this well established method, for which we refer the reader to Ref. [71], we here mention the relevance of the bond dimension χ . Its logarithm gauges the number of spins that we expect to be significantly entangled in the ground state of our chain, not to be mistaken for the range of the interaction or the size of the $\hat{W}_j$ matrices. Could one numerically deal with a bond dimension as large as $\mathcal{O}(2^N)$, then the algorithm would be exact. However, as this is not possible for large N , the algorithm involves a well-thought-out adjustment of the bond dimension. In our case, for instance, we ask for *i*) a reasonable computational time for producing as many ground states as necessary for feeding our unsupervised classification, and *ii*) a knowledge of their structure precise enough to obtain a QPD that makes sense and can be understood in terms of qualitative features of ground states in different phases.

6.3 Clustering

Let us now focus on clustering approaches: these are all aimed at grouping objects according to some sort of similarity principle, that can be the most diverse depending on the specific goal of the clustering itself. The key concept of the process is that of *affinity* between two data, as measured by some quantity whose domain is not the space where the data are defined, but rather the so-called feature space. This quantity must be defined for all possible pairs of the dataset upon which the clustering is due, and then organized in what is usually called affinity matrix, or *kernel*.

We choose to measure the pairwise affinity between any two data $\mathbf{x}_i, \mathbf{x}_j$ via the square fidelity between the corresponding ground states, as obtained by MPO and DMRG,

$$K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j) = |\langle \text{GS}(\mathbf{x}_i) | \text{GS}(\mathbf{x}_j) \rangle|^2, \quad (6.3.1)$$

organized in the $D \times D$ real positive matrix, that is our Kernel. Notice that the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathbb{R}^q$ has nothing to do with the actual value of K_{ij} , which is rather related with the distance between $|\text{GS}(\mathbf{x}_i)\rangle$ and $|\text{GS}(\mathbf{x}_j)\rangle$ as induced by the inner-product in the 2^N -dimensional Hilbert space of our chain.

Once the Kernel is numerically obtained, we can proceed with the clustering process; amongst the various techniques available for the scope, spectral clustering via k-means algorithm emerges as a suitable choice, and this is how it goes: first a “similarity” graph is generated from the quantum Kernel, in such a way that indices i, j of its matrix-elements identify nodes i and j of the graph, and edges are drawn if K_{ij} is larger than a threshold value $\tau < 1$ chosen at will; the graph Laplacian L is then determined, via $L_{ij} = A_{ij} - D_{ij}$, with $A_{ij} = 1$ if i and j are connected and zero otherwise, while $D_{ij} = \delta_{ij}d_i$ where d_i is the degree of the i th node, i.e. the number of nodes with which it is connected (A_{ij} and D_{ij} are the elements of the so called adjacency and degree matrices, respectively). Once the eigendecomposition of L is performed, providing D eigenvalues and eigenvectors, a number $c < D$ is chosen, to be the desired number of clusters into which grouping the data. Then, the c eigenvectors corresponding to the c largest eigenvalues are retained, to become the columns of a $D \times c$ matrix, Δ , whose rows are the D elements to be finally

clustered, while elements of each row are the c features to be used for clustering. Notice that each row Δ_{i*} is in objective relation with a datum $\mathbf{x}_i$ via the row-index i , meaning that grouping the rows effectively amount to cluster the original data.

The sheer clustering is then performed via the standard k-means algorithm, whose goal is that of grouping the D rows of the above matrix Δ into c different clusters, based on the c features selected when ignoring all but the c largest eigenvalues of the Laplacian. Reasons for performing the spectral analysis of the Laplacian rather than the Kernel itself reside in the particular way L retains properties of the Kernel that are relevant for clustering (whether $K(i, j)$ is larger than τ or not), ignoring useless information (the specific value of each $K(i, j)$).

Without entering the details of the k-means algorithm, for which we refer the reader to the literature [53], we recall that c is a crucial and yet arbitrary number and ask ourselves how to properly choose it. In fact, if c is too near to D we have an essentially useless clustering, with just a few data in each cluster and no significant computational advantage. On the other hand, too low a c might imply the loss of resolution and information in our classification process, computationally fast but in vain.

There are several ways to determine the optimal number of clusters c in k-means clustering, amongst which the *Elbow Method* [57, 58] and the *Silhouette method* [56], that belong to different classes of approaches, often referred to as within-cluster and between-clusters, respectively, to represent their different logic. In fact Elbow Method provides for plotting the Within-Cluster Sum of Squares (WCSS), a distance metric that measures the spread of data points within each cluster, as a function of c , so as to find where WCSS stops being strongly dependent on c itself, which is where the “elbow” is placed in the WCSS curve that reminds of a bended arm as c increases. After the “elbow” point further increases in c lead to marginal reductions in WCSS, indicating that adding more clusters provides little additional value and may even result in overfitting [55, 263]. Unfortunately, finding the elbow point is not enough, as some sort of validation is necessary to avoid dealing with a truly optimal c and not with the result of some arbitrary fitting. To this aim we consider the *Silhouette Method*, that assesses the quality of a c -clustering by evaluating the so called silhouette width

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}, \quad (6.3.2)$$

where i refers to one of the elements to be clustered, $a(i)$ is the average distance from i to all other elements in the same cluster, and $b(i)$ is the minimum average distance from i to elements in other clusters, with the distance being defined by the same metric employed in the clustering procedure. In this paper all distances are evaluated starting from the quantum kernel matrix.

The silhouette width ranges from -1 to 1 , with values close to 1 indicating well-clustered elements, and nearly-zero values suggesting that the element could equally well belong to another cluster. The average silhouette width across all elements is used to determine the best c ; the largest average silhouette width indicates the most suitable number of clusters. This method is particularly effective as it balances within-cluster tightness and separation from other clusters, providing a clear metric for evaluating clustering performance [56, 264]. Ultimately, a high silhouette score

for the value of c in the elbow region confirms that the clusters are well-separated, thereby validating the selection of the elbow point as an appropriate choice.

An effective way of visualizing the quality of the clustering as obtained for a certain value of c is the so called silhouette plot, examples of which are shown in Figs. 6.4.2(b) and 6.4.4(b). The plot is drawn as follows: to each element to be clustered, say $\mathbf{x}_i$, is associated a horizontal bar whose length is proportional to the silhouette score $s(i)$ defined in Eq. (6.3.2).

Negative values are shown as bars to the left of the vertical axis. Bars are grouped in bands according to the cluster assigned to their respective elements and piled bottom-up for increasing $s(i)$ within each band. The order in which different bands are displayed is arbitrary. The vertical width of each band visualizes the number of elements assigned to the respective cluster by the spectral algorithm. A vertical line indicating the average silhouette score is also shown: a band with a large number of bars exceeding such average indicates a well identified cluster, while bars to the left suggests the corresponding elements have been assigned to the wrong cluster.

6.4 Results

6.4.1 Axial Next-Nearest-Neighbor Ising Model

Our first case study is the ANNNI chain of N spin- $\frac{1}{2}$ particles [157, 160, 162, 239], defined by the Hamiltonian

$$H = J \left[- \sum_{j=1}^{N-1} \sigma_j^x \sigma_{j+1}^x + k \sum_{j=1}^{N-2} \sigma_j^x \sigma_{j+2}^x - h \sum_{j=1}^N \sigma_j^z \right], \quad (6.4.1)$$

where σ_j^x and σ_j^z are the Pauli operators acting on the spin at site j . We set the overall energy scale $J = 1$ and take k and h positive so that, besides the ferromagnetic nearest-neighbour (NN) exchange along the x direction, described by the first term in Eq. (6.4.1), there is an antiferromagnetic interaction between next-nearest-neighbours (NNN), gauged by the value of k , and a uniform magnetic field h pointing in the positive z direction. When $h = 0$ the system behaves classically, as all terms in the Hamiltonian commute with each other. When $k = 0$ the model reduces to the celebrated Ising model in a transverse field, exactly solvable via fermionization and extensively studied since the '70th of the last century – see Ref. [240] and references therein for a review. For $k > 0$ the system exhibits frustration, as the first term tends to align all spins along the x -axis while the second one promotes antialignment of NNN. The phase diagram of the ANNNI model has been explored using various analytical and numerical approaches [162, 166, 239, 241, 243, 265], and the possible use of tensor network techniques to determine its ground state has been investigated in Refs. [166, 167, 215, 253].

For small values of h and k , the NN interaction is the dominant term, leading to a ferromagnetic phase. As h increases beyond a critical threshold the system enters a paramagnetic phase where all spins get a positive z -component while pointing in random directions on the xy plane. When k is large and h stays sufficiently small the NNN interaction takes over, stabilizing the so-called antiphase order, in which

spins align along x in pairs, with alternate sign from one pair to the other. Between the antiphase and paramagnetic phases, the system enters a floating phase and, in the thermodynamic limit $N \rightarrow \infty$, becomes gapless, indicating the emergence of a critical behavior.

Although the general structure of the QPD is well established, there persist discrepancies in the identification of phase boundaries as obtained by different methods [164, 244]. In particular, while the primary phases depicted in Fig. 6.4.2c are widely recognized, some studies propose that the paramagnetic phase may split into two distinct regions [239], while others suggest the existence of an extended floating phase, with evidence reported in various studies [162, 164, 266]. In Fig. 6.4.2 (c) we mark with black lines the phase transitions as previously obtained by perturbative and numerical techniques [166, 168].

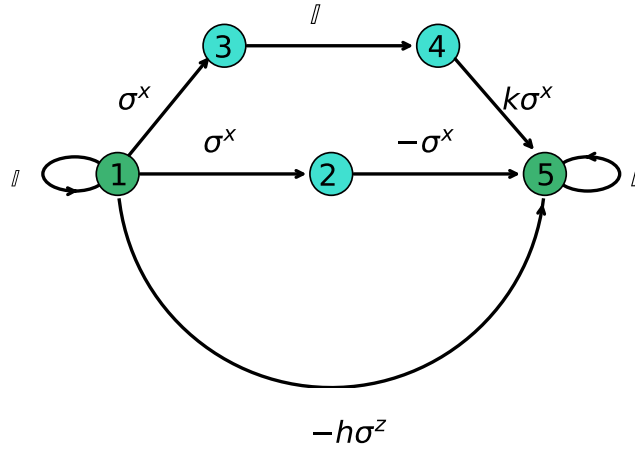


Figure 6.4.1. Local interaction in the MPO formalism for ANNNI model using finite-state automata graphical representation.

As for this work, referring to Eq. (6.2.2) the ANNNI hamiltonian (6.4.1) has $d = 2$, ${}^0h_j = h\sigma_j^z$, $\otimes_{\lambda=0}^1 \lambda h_{j+\lambda} = -\sigma_j^x \sigma_{j+1}^x$, and $\otimes_{\lambda=0}^2 h_{j+\lambda} = k\sigma_j^x \sigma_{j+2}^x$; using the prescription given in Sec. 6.2 we draw the corresponding graph (see Fig. 6.4.1) to which it corresponds the MPO

$$\hat{W} = \begin{pmatrix} \mathbb{I} & \sigma^x & \sigma^x & 0 & -h\sigma^z \\ 0 & 0 & 0 & 0 & -\sigma^x \\ 0 & 0 & 0 & \mathbb{I} & 0 \\ 0 & 0 & 0 & 0 & k\sigma^x \\ 0 & 0 & 0 & 0 & \mathbb{I} \end{pmatrix}. \quad (6.4.2)$$

Our results for a chain of $N = 51$ spins, sampling 30×30 pairs (k, h) , are shown in Fig. 6.4.2. The ground states were obtained using the DMRG algorithm, as implemented in the `quimb` library [245]. In order to choose the optimal c for the spectral clustering we have used the silhouette method: Fig. 6.4.2a shows $s(i)$ as a function of the number of clusters c : the largest value is seen for $c = 4$, suggesting that four clusters might provide a well-separated grouping of the data. Moreover, the elbow method applied to WCSS is shown in the inset: an evident inflection point

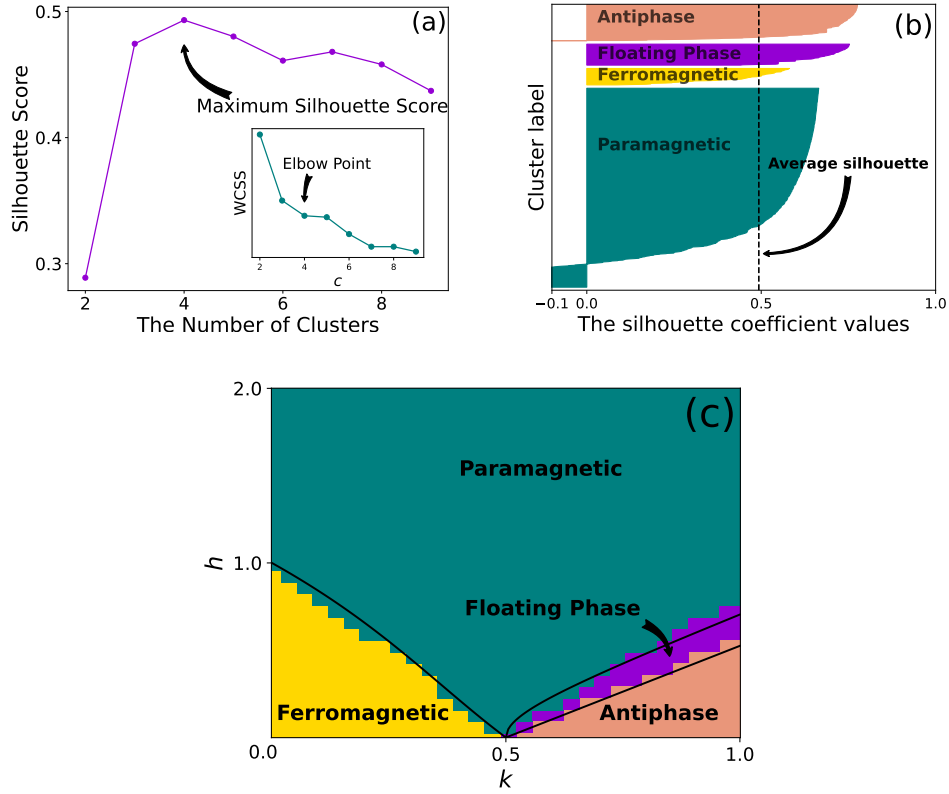


Figure 6.4.2. Results for the ANNNI Hamiltonian (6.4.1) with $N = 51$ spins, where the ground states are simulated with a bond dimension $\chi = 20$. (a) Average silhouette score as a function of the number of clusters c ; the inset shows the elbow point, identified by a significant change in the WCSS dependence on c . (b) Silhouette plot for $c = 4$, as determined by Fig. 6.4.2a. (c) The predicted phase diagram.

is observed at $c = 4$. Fig. 6.4.2b shows the silhouette-plot for $c = 4$, with the dashed vertical line marking the average silhouette score: the plot confirms the effectiveness of spectral clustering with $c = 4$, though some elements in the paramagnetic phase have negative $s(i)$, indicating misclustering. The clusters corresponding to the ferromagnetic and floating phases have smaller but entirely positive silhouette scores, suggesting well-defined boundaries. The antiphase and the paramagnetic phase, though, despite having relatively high silhouette scores, show variations in width, pointing to fluctuations in phase boundaries.

Fig. 6.4.2c shows our QPD and the previously predicted boundaries [166,168], indicated by the black lines. Discrepancies are seen near the phase transitions, which may be attributed either to finite-size effects, or to the DMRG bond dimension χ that we have set to 20. In fact, near transitions the entanglement area law breaks down [246], potentially requiring a larger bond dimension to accurately capture the correlations of the system.

6.4.2 Cluster-Ising Model with a symmetry-protected topological phase

Amongst the many different phases possibly featured by QPD of many-body systems, the so called SPT ones are particularly intriguing. An example of such phases can be found in the renowned Haldane chain, where the protecting symmetry is the discrete $\mathbb{Z}_2 \times \mathbb{Z}_2$ one. This same symmetry, realized via π rotations around the x -axis of all spins along the chain, is also featured by the spin- $\frac{1}{2}$ chain with hamiltonian

$$H = J \left[- \sum_{j=1}^{N-2} \sigma_j^z \sigma_{j+1}^x \sigma_{j+2}^z - h_2 \sum_{j=1}^{N-1} \sigma_j^x \sigma_{j+1}^x - h_1 \sum_{j=1}^N \sigma_j^x \right], \quad (6.4.3)$$

where the first term encodes the 1D cluster interaction, h_2 represents an Ising coupling, h_1 is a transverse field, and $J > 0$ sets the overall energy scale, whose value does not enter the QPD. For dominant cluster interaction, i.e. small h_1 and h_2 , and in the presence of the protecting $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry, the ground state of open chains realizes a short-range entangled, gapped SPT phase characterized by spin-1/2 edge modes, nonlocal string order, and two-fold entanglement-spectrum degeneracy [171, 177, 214, 267]. These features remain robust as long as the system is gapped; they disappear only when the bulk gap closes at a phase transition [171, 178, 267]. Competing with this cluster phase are the paramagnetic and antiferromagnetic phases, reached respectively by increasing h_1 or h_2 , with critical behavior that departs from the conventional Ising universality class. Recent studies have further evidenced the presence of this SPT phase through quantum convolutional neural network circuit [36, 268] and by a supervised learning approach [215].

In what follows we put our method to the test by applying it to the model (6.4.3), with $J = 1$ and $h_{1,2}$ free to take positive or negative values. Referring to Eq. (6.2.2), it is $d = 2$, ${}_0^0 h_j = -h_1 \sigma_j^x$, $\otimes_{\lambda=0}^1 {}_1^1 h_{j+\lambda} = -h_2 \sigma_j^x \sigma_{j+1}^x$, and $\otimes_{\lambda=0}^2 {}_2^2 h_{j+\lambda} = -\sigma_j^z \sigma_{j+1}^x \sigma_{j+2}^z$; using the prescription given in Sec. 6.2 we draw the corresponding graph, see Fig. 6.4.3, to which it corresponds the MPO

$$\hat{W} = \begin{pmatrix} \mathbb{I} & \sigma^x & \sigma^z & 0 & -h_1 \sigma^x \\ 0 & 0 & 0 & 0 & -h_2 \sigma^x \\ 0 & 0 & 0 & \sigma^x & 0 \\ 0 & 0 & 0 & 0 & -\sigma^z \\ 0 & 0 & 0 & 0 & \mathbb{I} \end{pmatrix}. \quad (6.4.4)$$

Fig. 6.4.4 presents our numerical results for a chain of $N = 51$ spins, sampling 30×30 pairs (h_1, h_2) . We use the silhouette method to determine the number of phases, i.e. the optimal number of clusters, c . The silhouette score as a function of c is shown in Fig. 6.4.4 (a), with the highest score at $c = 3$. The WCSS vs. c shown in the inset confirms this result, according to the Elbow method. The silhouette plot is shown in Fig. 6.4.4 (b), where the dashed vertical line marks the average silhouette score. The results demonstrate that phase boundaries are identified even when the elusive SPT phase is involved, as silhouette scores are mostly positive. However, some misclusterings are observed in the paramagnetic and antiferromagnetic phases, indicated by few negative silhouette scores. Finally, the QPD for the model (6.4.3) is shown in Fig. 6.4.4 (c). Its structure closely matches the expected one, according

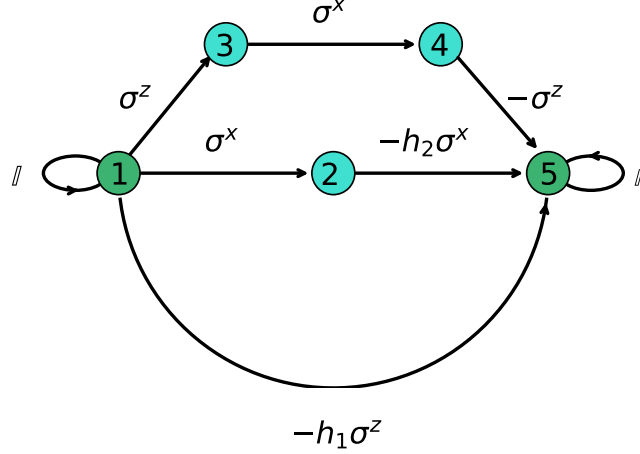


Figure 6.4.3. Local interaction in the MPO formalism for Cluster-Ising model using finite-state automata graphical representation.

to the above mentioned previous results [36, 268], represented by the black dashed lines. Minor discrepancies near the transition lines may arise from finite-size effects or limitations in the DMRG bond dimension used.

6.5 Conclusion and Discussion

We explored the use of classical unsupervised machine learning methods for studying QPD of quantum many-body systems. We found that kernel-based spectral clustering methods, where the different clusters define the different quantum phases, are effective when fed by kernels of genuine quantum origin. Importantly, the number of phases is discovered by the algorithm itself.

Two specific models of spin- $\frac{1}{2}$ chains, the ANNNI and the Cluster-Ising models, the latter of which featuring a topological phase, are used to benchmark our results. The bond dimension χ is a free parameter that defines the noise in the dataset, in the language of machine learning. For the ANNNI model, a moderate bond dimension ($\chi = 20$) is sufficient, whereas the Cluster-Ising model requires a relatively larger bond dimension ($\chi = 70$). It is relevant that the clustering approach successfully detects complex phases, such as those lacking a local order parameter, demonstrating its ability to capture nontrivial topological features directly from quantum-generated datasets.

Our findings show that integrating classical unsupervised machine learning models with quantum-generated datasets can provide physicists with a powerful tool for recognizing and classifying quantum phases of matter. Our work paves the way to the study of systems with completely unknown phase diagrams, without any prior labeling. Amongst these, spin liquids [269, 270], where conventional theoretical or numerical guidance is limited, and spin glasses [271], represent a promising avenue for future exploration, offering an opportunity to extend our methods to even more challenging quantum phases.

Another future direction is to replace our quantum kernels, estimated via TNs,

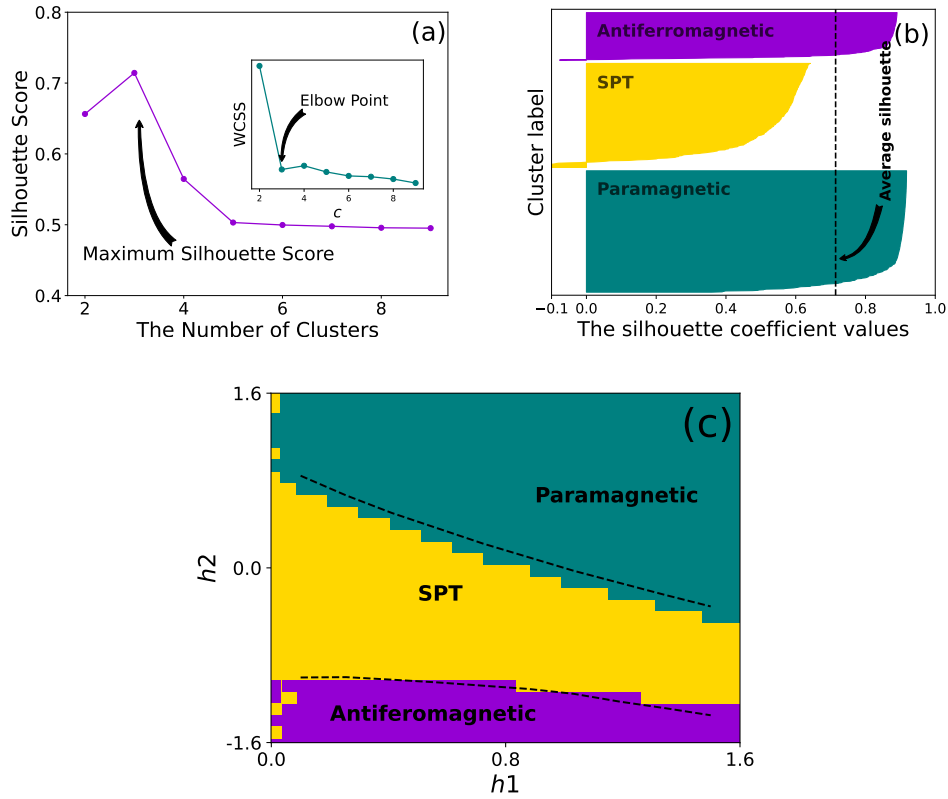


Figure 6.4.4. Results for the Cluster-Ising Hamiltonian (6.4.3) with $N = 51$ spins, where the ground states are simulated with a bond dimension $\chi = 70$. (a) Average silhouette score as a function of the number of clusters c ; the inset shows the elbow point, identified by a significant change in the WCSS dependence on c . (b) Silhouette plot for $c = 3$, as determined by Fig. 6.4.4 (a). (c) The predicted phase diagram.

with kernels directly reconstructed from experimental data, possibly using classical shadow techniques [272] or other randomized measurements [273].

Chapter 7

Quantum Phase Recognition from Reduced Subsystems

The work presented in this chapter forms the basis of a paper currently in preparation, authored by Mehran Khosrojerdi, Sougato Bose, Alessandro Cuccoli, Paola Verrucchi, Abolfazl Bayat, and Leonardo Banchi in that order.

This chapter begins with a brief introduction, followed by the methodology. We then discuss the results, focusing on Cluster–Ising models and Haldane Anisotropic Spin 1.

7.1 Introduction

Conventional mapping for quantum phase diagrams in one-dimensional spin systems has traditionally relied on a familiar set of tools. Researchers use exact diagonalization [198] and large-scale density-matrix renormalization group [197] to get ground states and low-energy spectra; conformal field theory and level spectroscopy to identify universality classes [196]; and entanglement-entropy scaling to extract central charges [274, 275]. When sign problems allow, quantum Monte Carlo and stochastic analytic continuation give access to dynamical features [193]. These techniques have enabled precise maps of canonical models, such as the spin-1 Haldane chain with single-ion anisotropies and the spin-1/2 Cluster–Ising chain with competing three-spin cluster and Ising interactions, where they reveal key transitions and phase structures.

Framing these results conceptually, Landau’s paradigm provides the starting point, where phases that break symmetries are classified through local order parameters, such as Néel or Ising magnetizations. But SPT phases resist this description, since they lack any symmetry-breaking local order. In such cases, nonlocal string orders and entanglement-based measures become essential. In Cluster–Ising models, cluster-type string orders and entanglement measures play a similar role, distinguishing the SPT phase from Ising order [171, 214, 267]. More generally, matrix product state diagnostics can extract projective representations of symmetries, offering a complete framework for identifying SPT order. Foundational work further established the equivalence between string order and the presence of a protecting local symmetry within finitely correlated states [276].

However, relying on order parameters and string orders has important weaknesses. First, Landau diagnostics cannot distinguish an SPT from a trivial symmetric phase and indeed there is no symmetry-allowed local order parameter separating them [276]. Second, string orders only make sense if the protecting symmetry is exactly preserved by the Hamiltonian, the boundaries, and the measurement. In spin-1 chains, for instance, adding a rhombic anisotropy reshapes the symmetry and certain string-order components can vanish or become ambiguous, even though the SPT itself survives. In such cases, one must turn to entanglement signatures like spectral degeneracies [196, 277]. Third, string orders are fragile to noise and experimental imperfections and even small errors can drive them to zero, while the underlying topological distinctions remain visible in other ways [276, 278]. All of these highlight the need for complementary approaches.

This is where recent advances in both classical and QML intersect with quantum many-body physics. Early studies demonstrated that classical supervised learning can successfully classify quantum phases directly from raw data. For example, Carrasquilla and Melko [222] showed that neural networks could detect phase transitions in spin systems without prior knowledge of order parameters. Building on this, classical methods such as SVM and kernel approaches have been employed to handle high-dimensional features, providing scalable and interpretable classifiers [47, 49]. More recently, Huang et al. [219] introduced “classical shadows” to efficiently predict ground-state properties across Hamiltonians, offering rigorous guarantees for ML efficiency in many-body systems.

Quantum computing opens the door to even more powerful approaches. QCNN leverage hierarchical feature extraction to identify both conventional and topological phases, including SPT phases [36, 168]. Hybrid classical-quantum frameworks combine variational circuits or quantum simulations with neural network classifiers to reduce resource requirements while retaining accuracy [215, 221, 223, 279, 280]. Importantly, these quantum-assisted methods remain effective even under moderate symmetry-breaking noise or invertible symmetry-preserving errors, precisely where conventional string-order diagnostics fail. They can classify phases from partial measurements or reduced subsystem data, demonstrating practical feasibility for near-term devices.

Tensor-network-based methods provide further enhance this synergy, making them compatible with machine learning pipelines. Mossi et al. [281] proposed an matrix product state framework capable of simultaneous classification and generation, showing that compressed or partial representations still retain sufficient information for phase recognition. Similarly, Tibaldi et al. [253] applied both supervised and unsupervised ML to reconstruct topological phase diagrams from single-particle correlations. These works collectively highlight that even partial or reduced subsystem data can encode global phase information, motivating approaches like ours that perform classification using reduced density matrices.

In this work, we propose a framework for quantum phase recognition based on reduced subsystems. Our motivation is to provide a method that makes the study of quantum phase diagrams in many-body systems more accessible, particularly from an experimental perspective. In realistic settings, experimentalists often have access only to a limited part of the system, which naturally suggests performing a partial trace over some degrees of freedom and analyzing the reduced density matrix (RDM)

of the remaining subsystem.

This approach is particularly well-suited to entangled quantum states, since the act of tracing out subsystems does not erase correlations entirely but rather encodes nontrivial information into the RDM. As a result, the reduced subsystem retains signatures of the global phase, effectively allowing one to infer properties of the whole system from measurements on a smaller part. This feature is especially appealing for experiments, as it means that local measurements can carry information about global order.

Building on this insight, we construct reduced density matrices from partial MPS and use them to define similarity kernels that serve as input to a SVM classifier. We demonstrate the effectiveness of this method on two paradigmatic models: the Cluster Ising model, which hosts symmetry-protected topological phases, and the Haldane anisotropic spin-1 chain, a canonical model for topological order. Our results show that reliable phase recognition can be achieved using only reduced subsystems, highlighting both the practical feasibility and conceptual significance of this approach.

In this work, we focus on phase recognition in gapped regimes, while the characterization of gapless critical points remains outside the scope of this study.

Our findings demonstrate that reduced subsystems can be used effectively for quantum phase recognition. Even small subsystems, which encode correlations over only a few sites, are sufficient to distinguish between different phases in the Cluster Ising and Haldane models. This establishes the feasibility of phase recognition from partial data while reducing computational demands. In addition, our results provide further evidence for the robustness of SPT phases across these two distinct models. From an experimental standpoint, this approach is especially appealing, as it enables reliable phase detection without requiring extensive or costly measurement setups.

The remainder of this article is structured as follows. Section 7.2 presents the methodology, detailing the construction of the dataset and the machine learning framework employed. Section 7.3 reports and analyzes the results obtained from our approach. Finally, Section 7.4 summarizes the findings and highlights the main achievements and insights of this work.

7.2 Methodology

Since the focus of this study lies in QML, the methodology is primarily centered around machine learning techniques. Similar to conventional machine learning tasks, our framework involves a dataset and a model, where the objective is to train the model using the available dataset. In this context, the dataset is generated from quantum many-body systems, while the learning model employed is a classical supervised machine learning algorithm.

In this section we first detail the procedure used to generate the quantum datasets, including the physical models and numerical techniques involved. Subsequently, we discuss how quantum-state-derived data can be encoded and utilized within classical machine learning frameworks. Finally, we describe the specific classical supervised learning algorithm adopted in this study and the rationale for its selection.

In the context of quantum phase recognition, it is essential that the dataset

captures quantum features capable of distinguishing between different phases of matter. Such features may include order parameters [222, 282], correlation functions [283], entanglement entropy [196, 284, 285], or full many-body wavefunctions [215]. In this study, we focus on using ground-state wavefunctions as the primary source of information for machine learning. These wavefunctions are rich in physical information and inherently encode the relevant phase-specific properties of the quantum system.

To construct such a dataset, we require access to accurate ground states of quantum many-body Hamiltonians. However, due to the exponential scaling of the Hilbert space with system size, exact diagonalization becomes infeasible for systems beyond a modest number of sites. To address this limitation, we employ tensor network methods, specifically, the MPS formalism, which offers an efficient and scalable representation of quantum states in one-dimensional systems. In this framework, the many-body wavefunction is expressed as a sequence of low-rank tensors, enabling us to approximate ground states with high accuracy and significantly reduced computational cost.

The ground states are obtained using the DMRG algorithm, a variational method that iteratively optimizes the MPS representation. This process involves sweeping back and forth along the chain of particles, updating tensors locally to minimize the total energy. Each sweep refines the approximation of the ground state, ultimately yielding an optimized MPS corresponding to the lowest-energy eigenstate of the Hamiltonian.

In the next step, we construct the density matrices corresponding to the optimized ground-state wavefunctions. Rather than working with the full density matrix of the entire system, we perform a partial trace to obtain reduced density matrices for subsystems of interest. In this setting, the one-dimensional chain of particles is naturally treated as a bipartite system, allowing us to trace out part of the chain and focus on the reduced state of a specific region. These reduced density matrices capture local quantum correlations and entanglement properties, which are often crucial for identifying and distinguishing different quantum phases.

We now turn to a more technical and numerical description of the dataset generation procedure. Let us consider a classical parameter vector $\mathbf{x}$, which represents tunable couplings or interaction strengths within the quantum system, such as nearest-neighbor or next-nearest-neighbor interactions. By systematically varying $\mathbf{x}$ across a predefined range, we construct a set of Hamiltonians $\mathcal{H}(\mathbf{x})$, each corresponding to a specific point in the parameter space and thus to a distinct region of the quantum phase diagram.

Each Hamiltonian is then mapped into the tensor network framework via its MPO representation. These MPOs are used as input to the DMRG algorithm, which is executed independently for each Hamiltonian. The DMRG procedure iteratively minimizes the energy of the system and returns an optimized ground state $\psi(\mathbf{x})$ in the form of a MPS, serving as a highly accurate approximation of the corresponding lowest-energy eigenstate.

Next, we bring each optimized MPS into canonical form so that the tensors around the cut define an orthonormal Schmidt basis. By left- and right-orthonormalizing the MPS tensors, the state can be expressed in its Schmidt decomposition:

$$|\psi(\mathbf{x}_n)\rangle = \sum_{\alpha=1}^{\chi} s_{\alpha} |A_{\alpha}\rangle \otimes |B_{\alpha}\rangle, \quad (7.2.1)$$

where $|A_{\alpha}\rangle$ are orthonormal states on sites $1, \dots, L$, $|B_{\alpha}\rangle$ are orthonormal on sites $L+1, \dots, N$, s_{α} are the Schmidt coefficients, and χ is the bond dimension at the cut between sites L and $L+1$ (see figure 7.2.1 (a)). The sum over α is necessary because the bond index takes multiple values ($\alpha = 1, \dots, \chi$), with each α representing a correlated pair of left and right states. The superposition over all α builds the entanglement across the cut.

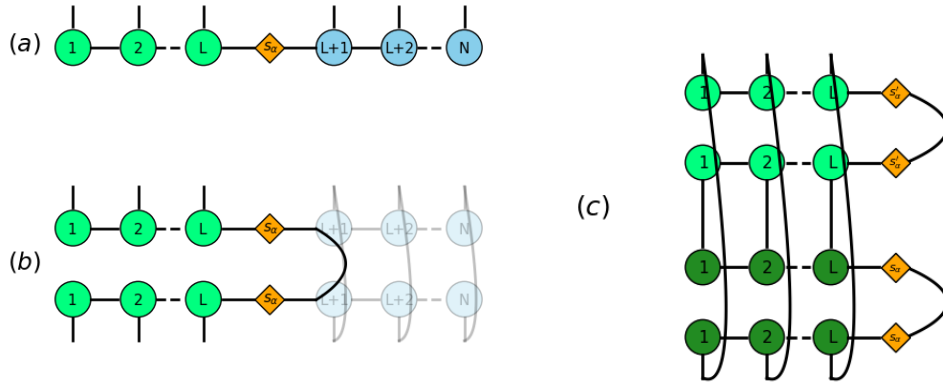


Figure 7.2.1. (a) Bipartite MPS in canonical form. The chain is split into left (A) and right (B) blocks, with the bond index α representing the Schmidt decomposition $|\psi\rangle = \sum_{\alpha} s_{\alpha} |L_{\alpha}\rangle \otimes |R_{\alpha}\rangle$ and encoding the entanglement across the cut. (b) Reduced density matrix ρ_A of the left block A , obtained by tracing out the right block B in the MPS Schmidt decomposition. Eigenvalues s_{α}^2 encode the entanglement between the two blocks. (c) Quantum fidelity kernel between reduced density matrices $\rho_A(\mathbf{x}_n)$ and $\rho_A(\mathbf{x}_m)$, measuring the similarity of MPS ground states via $|\langle\psi_A(\mathbf{x}_n)|\psi_A(\mathbf{x}_m)\rangle|^2$.

From this Schmidt decomposition, the reduced density matrix of the left block A is obtained by tracing out the right block B :

$$\rho_A(\mathbf{x}_n) = \text{Tr}_B [|\psi(\mathbf{x}_n)\rangle \langle\psi(\mathbf{x}_n)|] = \sum_{\alpha=1}^{\chi} s_{\alpha}^2 |A_{\alpha}\rangle \langle A_{\alpha}|, \quad (7.2.2)$$

This expression shows that the eigenvalues of ρ_A are precisely the squared Schmidt coefficients s_{α}^2 , while the corresponding eigenvectors are the left Schmidt states $|L_{\alpha}\rangle$ (see figure 7.2.1 (b)). Consequently, ρ_A compactly encodes the entanglement and correlation information across the bipartition, preserving the essential local and non-local features of the state relevant for phase classification.

Given two such reduced density matrices, $\rho_A(\mathbf{x}_n)$ and $\rho_A(\mathbf{x}_m)$, derived from optimized ground-state MPS wavefunctions $\psi(\mathbf{x}_n)$ and $\psi(\mathbf{x}_m)$, we quantify their similarity using a kernel function based on quantum-fidelity. This quantum-fidelity kernel is expressed in equation 7.2.3 and illustrated in figure 7.2.1 (c).

$$\begin{aligned} K(\mathbf{x}_n, \mathbf{x}_m) &= \text{Tr}[\rho_A(\mathbf{x}_n) \rho_A(\mathbf{x}_m)] \\ &= \sum_{\alpha, \alpha'} s_{\alpha}^2(\mathbf{x}_n) s_{\alpha'}^2(\mathbf{x}_m) |\langle A_{\alpha}(\mathbf{x}_n) | A_{\alpha'}(\mathbf{x}_m) \rangle|^2, \end{aligned} \quad (7.2.3)$$

This expression explicitly shows how the entanglement encoded in the Schmidt coefficients s_α and $s_{\alpha'}$ contributes to the similarity measure where the overlaps of the Schmidt states determine the kernel value. As a result, the kernel captures correlations across the bipartition, providing a principled, entanglement-aware measure of quantum state similarity.

This kernel serves as the central metric for quantifying the overlap between quantum states and is used as input to classical kernel-based learning algorithms.

It is important to note that the reduced density matrices ρ_A generally represent mixed states, even when derived from pure global MPS wavefunctions. As a consequence, the diagonal elements of the kernel, $K(\mathbf{x}_n, \mathbf{x}_n) = \text{Tr}[\rho_A(\mathbf{x}_n)^2]$, are typically less than one, reflecting the partial loss of information due to tracing out subsystem B . For instance, in maximally entangled bipartitions, these self-overlaps can start below unity (e.g., 0.92 in one of the study cases), in contrast to pure states where the kernel diagonal reaches exactly one. This property emphasizes that the kernel captures similarity between mixed subsystem states, rather than idealized pure states, and provides a nuanced measure of entanglement-preserving correlations in reduced subsystems.

Thanks to our previous work on quantum phase recognition using full many-body wavefunctions, we now extend our analysis to reduced wavefunction representations. In our supervised learning setup, we employ the SSVM, a well-established kernel-based classification method that constructs decision boundaries in high-dimensional feature spaces where the SVM aims to find an optimal hyperplane in this Hilbert space that separates different quantum phases [47]. The classification is governed by the decision function

$$f(\mathbf{x}) = \text{Tr}[W\rho_A(\mathbf{x})] + b, \quad (7.2.4)$$

where W is an operator defining the hyperplane and b is a bias term. The predicted class is given by the sign of $f(\mathbf{x})$.

Training the SVM involves solving a convex optimization problem in its dual form:

$$\arg \max_{0 \leq \alpha_i < C} \left[\sum_{i=1}^M \alpha_i - \frac{1}{2} \sum_{i,j=1}^M y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \right], \quad (7.2.5)$$

subject to the constraint $\sum_{i=1}^M \alpha_i y_i = 0$, where $(\mathbf{x}_i, y_i)$ are the training examples and C is a regularization hyperparameter. After solving for the Lagrange multipliers α_i , the observable W and bias b can be reconstructed analytically. The final classifier uses the decision rule

$$y = \text{sign} \left[\sum_{i=1}^M \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}) + b \right], \quad (7.2.6)$$

which allows efficient phase prediction of unseen data.

7.3 Results

7.3.1 Cluster-Ising Model

The Hamiltonian of this model including the cluster, coupling, and transverse field terms can be written as

$$\mathcal{H} = - \sum_i (JZ_{i-1}X_iZ_{i+1} + h_1X_i + h_2X_iX_{i+1}), \quad (7.3.1)$$

where X and Z denote the Pauli operators on site i , J and h_2 are interaction strengths, and h_1 is a transverse field. The first term corresponds to the three-spin cluster ZXZ interaction, which generates entangled states and underpins the SPT phase. The second term is a transverse field along the x -direction, which competes with the interactions and can drive quantum phase transitions. The final term represents a nearest-neighbor XX interaction, contributing to coherent spin dynamics and supporting topological order.

This Hamiltonian possesses several key symmetries, including time-reversal, particle-hole, and lattice translation invariance. These symmetries protect the topological phases, stabilize edge modes, and are responsible for the presence of non-ergodic excited states, commonly referred to as quantum many-body scars. The interplay of cluster and XX interactions, together with the transverse field, leads to a rich phase diagram featuring various topological and symmetry-breaking phases. Notably, these systems exhibit long-lived coherence in certain excited states, geometric-phase scaling linked to the band structure, and nontrivial quench dynamics, reflecting the complex interaction between topology, symmetry, and dynamics in quantum many-body systems [173, 175, 187].

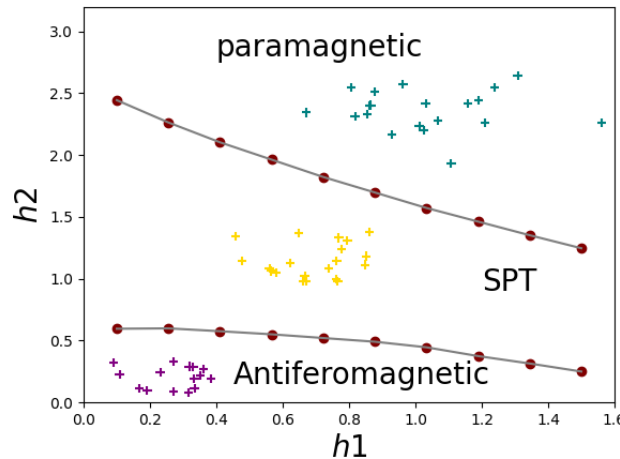


Figure 7.3.1. Phase diagram of the Cluster-Ising model. Colored plus signs indicate the training points scattered across the phase diagram, representing the corresponding phase labels.

The phase diagram in Figure 7.3.1 is adapted from the study by Cong et al. [36],

where two complementary approaches were employed for quantum phase recognition. In the first approach, the authors combined numerical simulations with an analytically designed QCNN circuit. Ground states of the spin chain were obtained using infinite-DMRG, and phase boundaries were located by identifying peaks in the second derivative of the ground-state energy with respect to the Hamiltonian parameters. These transition points were then compared with the output of the QCNN applied to simulated finite-size ground states. The resulting phase diagram shows both the “true” iDMRG boundaries and the QCNN’s classification probabilities, illustrating that the QCNN architecture alone can successfully reproduce the correct phase structure without any training.

In the second approach, the authors investigated the performance of a trained QCNN. Starting from a random initialization, the network was trained only on a small subset of analytically solvable points along the $h_2 = 0$ line. Once trained, the QCNN was used to classify states across the entire two-dimensional parameter space, producing a phase diagram that was then benchmarked against iDMRG results. Remarkably, the trained QCNN generalized from this limited training set to accurately reconstruct the global phase diagram, demonstrating its efficiency and scalability for quantum phase recognition, even with sparse data.

In our previous supervised learning project [215], we studied this model in the finite-size setting. Using DMRG to generate ground states and training with only a few measurements sampled along a line $h_1 = 1$ that crosses all phases, we showed that the kernel-based approaches is likewise capable of identifying quantum phases of matter, and the same phase diagram structure re-emerges. Furthermore, in an unsupervised framework 6, by considering only optimized states and applying spectral clustering, we once again recovered the correct pattern of quantum phases confirming the robustness of this phase diagram across distinct machine learning approaches.

The colored plus signs in Figure 7.3.1 indicate the training points, which are randomly distributed across the phase diagram and labeled according to their corresponding quantum phases. Each of these points initially represents the ground state of the full spin chain. To construct the kernel for the SVM model, we compute reduced density matrices from these ground states by tracing out a subset of the system and keeping only a small number of sites. The resulting reduced density matrices are then used to build the training kernel.

For testing, we evaluate the model on the full set of ground states sampled across the entire phase diagram, where each pixel corresponds to a different ground state. As in training, we construct the test kernel from reduced density matrices, obtained by partial tracing over the full-chain ground states. A key advantage of this approach is that, regardless of the original system size in the training or testing set, the reduced density matrices have the same fixed dimension. This ensures consistency in kernel construction and allows phase classification to be performed efficiently across different system sizes.

Figure 7.3.2 analyze the phase diagram of the cluster-Ising model with an additional coupling term and transverse field by performing a partial trace over a 51-site chain. Training and testing are carried out on the full system, while classification relies on reduced density matrices obtained by keeping only the central and edge sites which are depicted in first and second row, respectively. By systematically

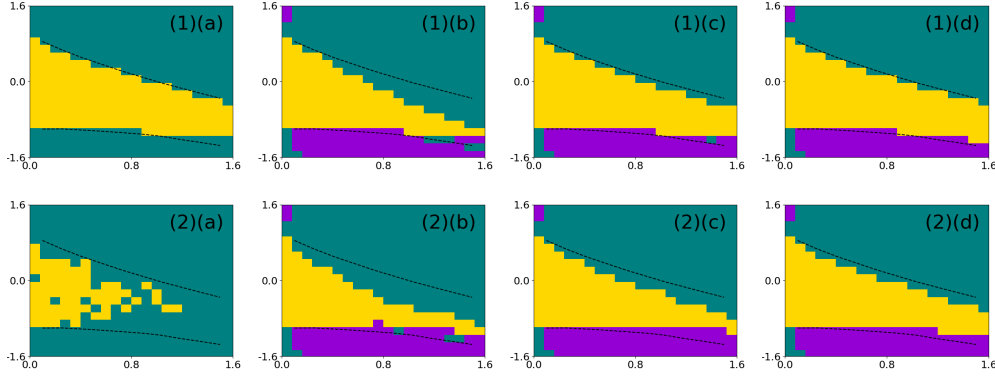


Figure 7.3.2. Panels (1)(a–d) and (2)(a–d) illustrate the effect of keeping an increasing number of particles while tracing out the rest of the system. In Panels (1)(a–d), particles are kept from the **central sites**: (a) 1 site, (b) 2 sites, (c) 3 sites, and (d) 4 sites. In Panels (2)(a–d), particles are kept from the **edge sites**: (a) 1 site, (b) 2 sites, (c) 3 sites, and (d) 4 sites. Prior to the tracing-out process, both the **training** and **test sets** consist of **51 sites** each.

varying the number of kept sites, we assess how much information about the phases survives under strong reduction.

First row of figure 7.3.2 tells us when only a single spin is kept, the paramagnetic phase remains clearly identifiable, while the topological phase can still be distinguished, but the antiferromagnetic phase is absent. With two spins, the accuracy of the topological phase decreases, and the antiferromagnetic phase remains unresolved. Keeping three spins restores the full phase diagram in a way that the antiferromagnetic order becomes visible, and the topological phase appears with much higher accuracy, while the paramagnetic phase continues to show the greatest stability. With four spins, all three phases are robustly resolved.

Second row of figure 7.3.2 shows that when only a single spin is kept, the paramagnetic phase remains clearly identifiable and dominant. In contrast, the topological phase becomes less distinct and noisier compared to the first row, while the antiferromagnetic phase is absent. With two spins kept, the topological phase is captured more accurately, though the antiferromagnetic phase is still unresolved. Keeping three spins restores the overall phase diagram and the antiferromagnetic order emerges, the topological phase is recognized with higher accuracy, and the paramagnetic phase continues to show the greatest stability—although some misclassification occurs between the topological and paramagnetic phases. Finally, with four spins, all three phases are robustly resolved, and our results indicate that the accuracy continues to improve as more sites are preserved.

In next case, our goal is to reduce the computational cost during training, which is also reasonable from an experimental perspective. Figure 7.3.3 present results for the cluster-Ising model, where a partial trace is performed over a 31-site chain during training and tested on a 51-site chain. While training and testing are conducted on the full system, the classification itself is based on reduced density matrices obtained by keeping only the central or edge sites, which are illustrated in first and second row, respectively. By systematically varying the number of preserved sites, we evaluate

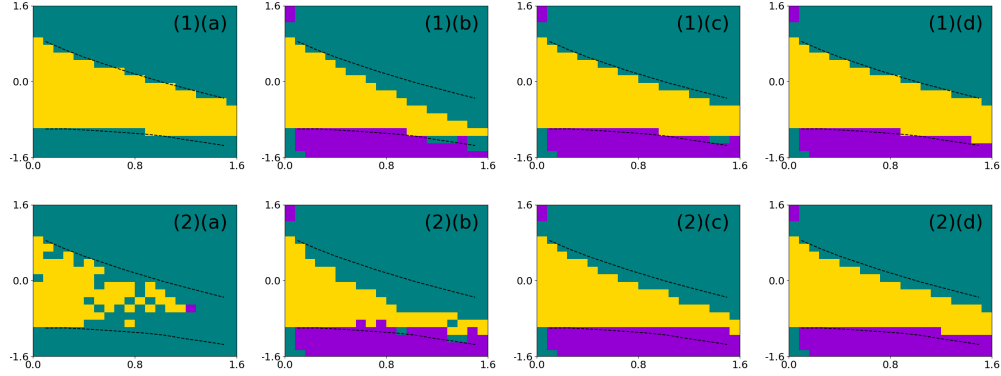


Figure 7.3.3. Panels (1)(a–d) and (2)(a–d) illustrate the effect of keeping an increasing number of particles while tracing out the rest of the system. In Panels (1)(a–d), particles are kept from the **central sites**: (a) 1 site, (b) 2 sites, (c) 3 sites, and (d) 4 sites. In Panels (2)(a–d), particles are kept from the **edge sites**: (a) 1 site, (b) 2 sites, (c) 3 sites, and (d) 4 sites. Prior to the tracing-out process, the **training set** consists of **31 sites**, while the **test set** consists of **51 sites**.

how much phase information remains under strong reduction.

First row of figure 7.3.3 exhibits behavior very similar to that of first row of figure 7.3.2. Even with a reduced number of sites in the training set, the overall behavior remains largely consistent.

Second row of figure 7.3.3 shows the same conditions. Our comparison reveals that it exhibits behavior similar to figure 7.3.2, indicating once again that we can reduce the size of the training set, which is quite advantageous.

The patterns observed in figures 7.3.2 and 7.3.3 demonstrate that the paramagnetic phase remains the most robust under partial trace, consistently retaining its stability even when a substantial reduction of sites is applied. This observation is further supported by previous findings [286], where the re-entrance and persistence of the paramagnetic phase were highlighted in the steady-state phase diagram obtained via a cluster approach. For the antiferromagnetic phase, we find in all cases that it emerges only when at least three sites are kept. This reflects the intrinsic nature of the cluster–Ising model, where the interplay of three-body cluster interactions and Ising antiferromagnetic couplings governs the ordering. Capturing antiferromagnetic order requires resolving the alternating spin alignment, which cannot be detected with just one or two sites but becomes visible once three or more spins are preserved. Our observation is consistent with [171], which emphasized that the onset of antiferromagnetic correlations in the cluster–Ising model depends on the ability to probe beyond local, single-site information and to resolve the staggered structure across multiple sites.

The SPT phase is a particularly interesting aspect of our research. Remarkably, the SPT phase appears in all cases, even when only a single site is kept. It is important to emphasize that here we perform a partial trace over a long chain, keeping only one site while tracing out all others. This differs from training the model on just one site from the start, as the partial trace indirectly encodes non-local correlations [287,288]. These correlations are reflected in the structure of the reduced

density matrix. As system parameters vary, the single-site reduced states change in a way that mirrors the phase transitions of the underlying many-body system. In short, through the partial trace, the single-site reduced density matrix inherits information from the larger entangled system. In contrast, when the model is trained on a single site in isolation, there is no entanglement, no correlations, and thus no nontrivial phase information to extract.

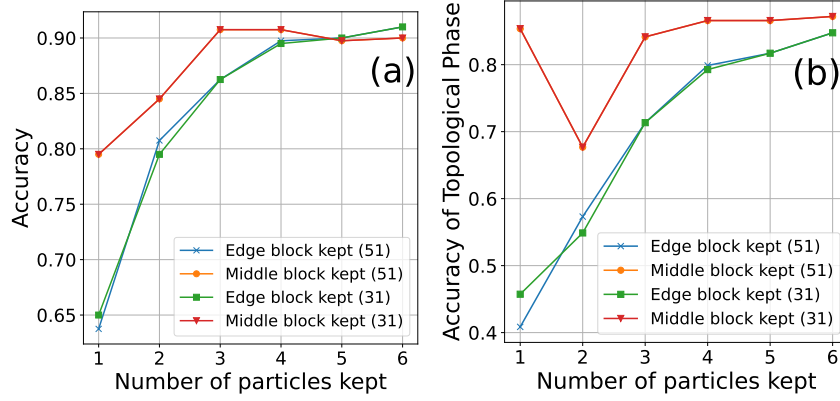


Figure 7.3.4. (a) Accuracy of phase diagram versus number of keeping sites. (b) Accuracy of topological phase versus number of keeping sites

Figure 7.3.4 (a) illustrates the accuracy of the entire phase diagram. It shows that keeping sites from the middle of the chain yields better results than keeping sites from the edges. After three sites, the accuracy converges, and both cases achieve similar performance. Additionally, it is evident that using 31 or 51 sites produces the same level of accuracy.

Figure 7.3.4 (b) shows the accuracy of the SPT phase. A similar trend is observed here. Keeping sites from the middle of the chain yields better results than keeping sites from the edges. Both 31 and 51 sites achieve comparable accuracy, and convergence occurs after approximately six sites are kept.

7.3.2 Haldane Anisotropic Spin 1

The second Hamiltonian that we do study over that is anisotropic Haldane Spin one, which is described by

$$\mathcal{H} = J \sum_{i=0}^{L-1} \mathbf{S}_i \cdot \mathbf{S}_{i+1} + D \sum_{i=0}^L (Z_i)^2 + E \sum_{i=0}^L [(X_i)^2 - (Y_i)^2], \quad (7.3.2)$$

where in terms of spin operators $\mathbf{S}_i = (X_i, Y_i, Z_i)^\top$ at each site i , captures key features of quantum magnetism through the interplay of exchange interactions and anisotropies. The first term describes the Heisenberg interaction between neighboring sites with coupling strength J , while the additional single-ion anisotropy terms which $D(Z_i)^2$ for uniaxial anisotropy along the z -axis and $E[(X_i)^2 - (Y_i)^2]$ for rhombic anisotropy in the x - y plane reflect the effects of crystal fields and spin-orbit coupling.

Anisotropic interactions are central to understanding the ground-state properties and quantum phase transitions in such system. In this context, the term anisotropic refers to the direction-dependent nature of spin-spin interactions, which removes the spin rotational symmetry ($SU(2)$) of the isotropic Heisenberg model by making the interactions direction-dependent. Specifically, the exchange coupling along the z -axis differs from that in the x - y plane, introducing a preferred direction and leading to a richer phase structure. These anisotropic terms significantly affect the stability of the Haldane phase and are crucial for describing topological quantum phase transitions.

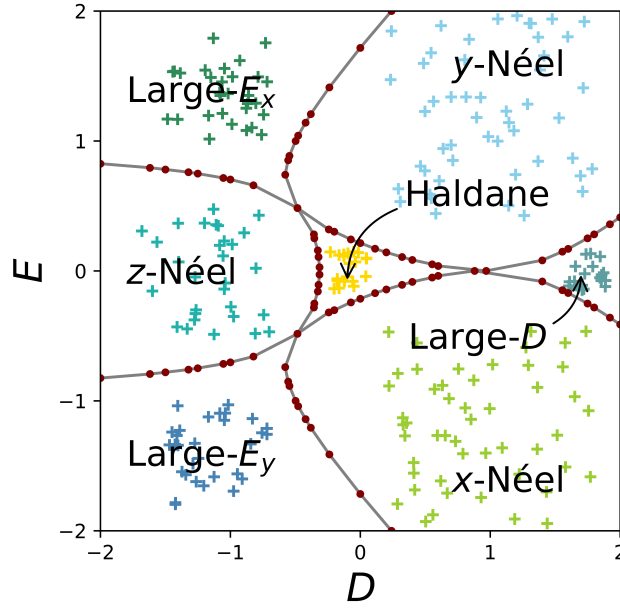


Figure 7.3.5. Phase diagram for Haldan Anisotropic spin-1 model

The quantum phase of this Hamiltonian, shown in Fig.7.3.5 and presented in [196], includes SPT phases, symmetry-breaking Néel phases, and trivial disordered phases. This phase diagram is derived from numerical investigations employing the DMRG technique. In their study, first they were using DMRG to identify different quantum phases, then classify states using conserved quantum numbers such as the parity of magnetization, spatial inversion, and time-reversal symmetry. The phase boundaries and critical points are subsequently determined with high precision through analysis of the entanglement spectrum, which is computed from the reduced density matrix derived from the system's ground state wavefunction. Crucially, their work demonstrates that phase boundaries in this system are not characterized by conventional local order parameters, but rather by the structure of the entanglement spectrum and crossings of energy levels distinguished by symmetry quantum numbers. The degeneracy structure in the entanglement spectrum plays a key role: the Haldane phase exhibits a characteristic fourfold degeneracy associated with the hidden $\mathbb{Z}_2 \otimes \mathbb{Z}_2$ symmetry; the Néel phases (e.g., x -, y -, or z -Néel) show twofold degeneracy due to spontaneous $\mathbb{Z}_2$ symmetry breaking; and the trivial phases (such as large- D , large- E_x , and large- E_y) are identified by a non-degenerate (single-fold) entanglement spectrum. Quantum phases in Haldane Anisotropic are

characterized not only by their energy gaps and correlation behaviors, but more fundamentally by how they relate to symmetry breaking.

As with the dataset preparation for the cluster-Ising model in figure 7.3.5, training points that are randomly scattered across the phase diagram are marked by plus signs and labeled by their corresponding quantum phase. Each of these points initially represents the optimized ground states of the full chain. We compute the reduced density matrix from these optimized states and we are just keeping only a few numbers of sites. Then we are reconstructing the training kernel. For testing, we evaluate the model on the full set of ground states sampled across the entire phase diagram, where each pixel corresponds to a different ground state. As in training, we construct the test kernel from reduced density matrices, obtained by partial tracing over the full-chain ground states. A key advantage of this approach is that, regardless of the original system size in the training or testing set, the reduced density matrices have the same fixed dimension. This ensures consistency in kernel construction and allows phase classification to be performed efficiently across different system sizes.

The kernels are then employed to train and test the SVM model. In this study, we examine two scenarios. In the first, the training and test sets are constructed from one-dimensional chains of equal length. In the second, the model is trained on a shorter chain and tested on a longer one.

We begin with the equal-length case. Figure 7.3.6 present the phase diagram of the anisotropic spin-1 Haldane chain, obtained by performing a partial trace on a 51-site system. Training and testing are performed on the complete system, while classification is based on reduced density matrices that keep only the central and edge sites, respectively. By progressively increasing the number of kept sites, we evaluate how much phase information persists.

First row of figure 7.3.6 shows the central-site case and interestingly, here the Haldane phase achieves remarkably high accuracy even when only a single site is kept. The trivial phases also display consistently high accuracy, whereas the Néel phases improve progressively as the number of retained sites increases.

Second row of figure 7.3.6 illustrates the edge-site case. As the number of kept sites increases, the prediction accuracy for the Haldane phase steadily improves. A similar trend is observed for the Néel phases, where larger numbers of kept sites yield a more accurate phase diagram. In contrast, the trivial phase remains consistently robust, maintaining high accuracy across all cases.

We now turn to the second scenario, where the training and testing chain lengths differ. Figure 7.3.7 illustrate this case, with training performed on a 31-site chain and testing on a 51-site chain. As before, classification is based on reduced density matrices obtained by keeping only central or edge sites. By gradually increasing the number of preserved sites, we again evaluate the persistence of phase information and the extent to which the model trained on shorter chains can generalize to longer ones. Interestingly, results coming out from this scenario is similar to the equal-length training and testing scenario.

First row of figure 7.3.7 illustrates the central-site case. Interestingly, the Haldane phase attains remarkably high accuracy even when only a single site is retained. The trivial phases likewise maintain consistently high accuracy, while the Néel phases show a steady improvement as the number of retained sites increases.

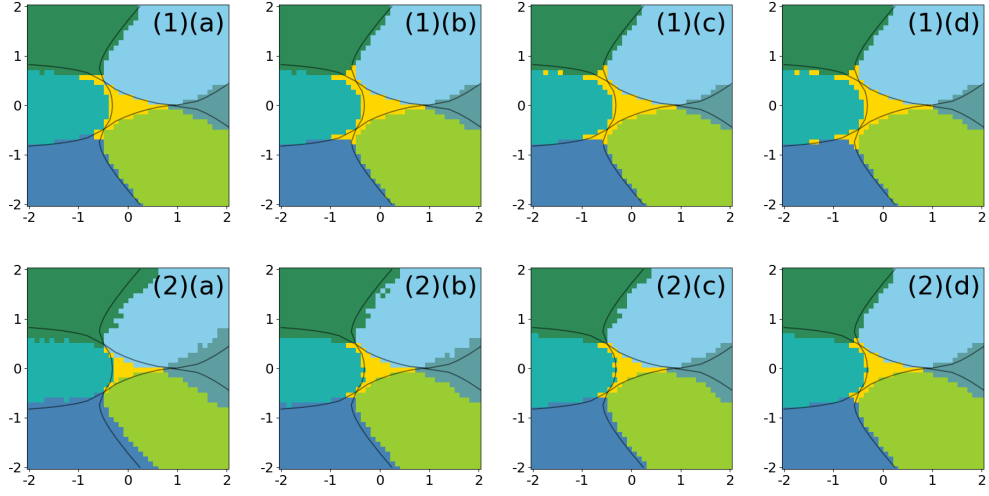


Figure 7.3.6. Panels (1)(a–d) and (2)(a–d) illustrate the effect of keeping an increasing number of particles while tracing out the rest of the system. In Panels (1)(a–d), particles are kept from the **central sites**: (a) 1 site, (b) 2 sites, (c) 3 sites, and (d) 4 sites. In Panels (2)(a–d), particles are kept from the **edge sites**: (a) 1 site, (b) 2 sites, (c) 3 sites, and (d) 4 sites. Prior to the tracing-out process, both the **training** and **test sets** consist of **51 sites** each.

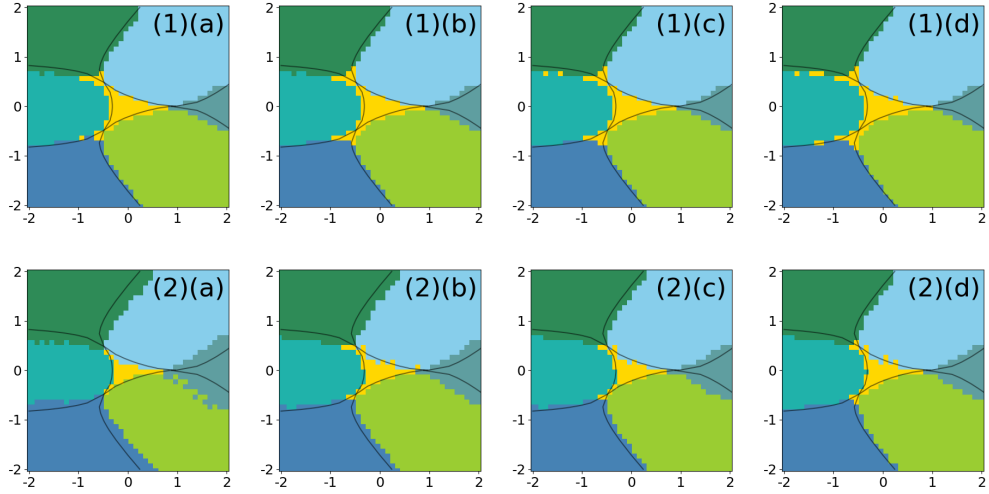


Figure 7.3.7. Panels (1)(a–d) and (2)(a–d) illustrate the effect of keeping an increasing number of particles while tracing out the rest of the system. In Panels (1)(a–d), particles are kept from the **central sites**: (a) 1 site, (b) 2 sites, (c) 3 sites, and (d) 4 sites. In Panels (2)(a–d), particles are kept from the **edge sites**: (a) 1 site, (b) 2 sites, (c) 3 sites, and (d) 4 sites. Prior to the tracing-out process, the **training set** consists of **31 sites**, while the **test set** consists of **51 sites**.

Similarly, the edge-site case yields comparable results, as shown in second row of figure 7.3.7. Here, the prediction accuracy for the Haldane phase steadily increases with the number of kept sites. The Néel phases follow the same trend, with larger subsets of sites leading to more accurate phase diagrams. In contrast, the trivial phase remains robust throughout, consistently achieving high accuracy across all configurations.

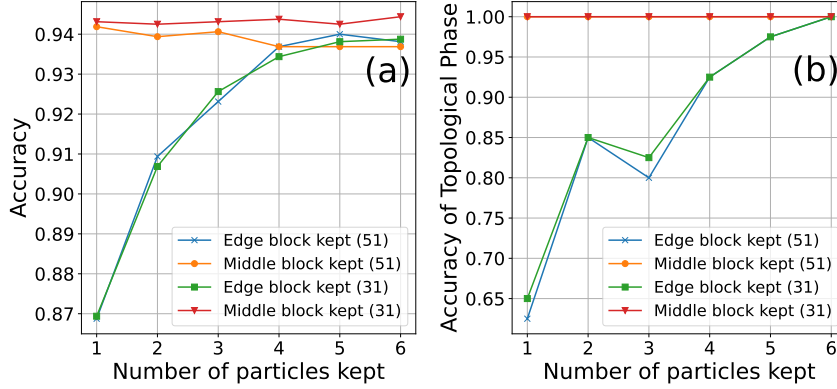


Figure 7.3.8. (a) Accuracy of phase diagram versus number of keeping sites. (b) Accuracy of topological phase versus number of keeping sites

Figure 7.3.8 (a) presents the overall accuracy of the phase diagram. The results reveal that keeping sites from the middle of the chain yields higher accuracy compared to keeping sites from the edges. Beyond four sites, however, the accuracy converges, and both approaches achieve comparable performance. Moreover, the analysis shows that systems of length 31 and 51 produce virtually identical accuracy, with their results overlapping almost perfectly.

Figure 7.3.8 (b) shows the accuracy of the topological phase. A similar trend is observed here. Keeping sites from the middle of the chain provides the best performance, reaching perfect accuracy even when only a single site is considered, and remaining at 100% as more sites are added. In contrast, for the edge-site case, accuracy improves progressively with the number of kept sites. Both the 31-site and 51-site systems exhibit comparable behavior, with convergence achieved after approximately six sites are kept.

7.4 Conclusion and Discussion

In summary, we focus on applying machine learning techniques to phase diagrams that are already well established experimentally and theoretically, rather than locating critical points or determining precise phase boundaries of quantum matter. Our results confirm strong agreement with theoretical predictions while extending the analysis to larger system sizes that are computationally accessible. By leveraging reduced (partial) density matrices, we can probe order parameters and study the phase structure of quantum matter. This approach offers practical advantages for experimentalists, who can work with a limited number of measurement devices,

and for machine learning specialists, by significantly reducing computational cost. Importantly, our findings reaffirm the remarkable stability of SPT phases, which consistently emerge even under partial-trace analysis.

A particularly striking result of this work is that even extreme reductions, down to a single site in some cases, retain enough information to reliably reconstruct phase behavior. This surprising effectiveness arises from the correlations encoded in the Schmidt coefficients between subsystems. By keeping one subsystem, we implicitly capture information about the complementary subsystem, allowing the kernel-based learning approach to detect the phase diagram with high fidelity. This demonstrates that partial-trace representations can preserve the essential physics of quantum phases, providing both a computationally efficient and experimentally feasible route to study strongly correlated and highly entangled systems.

Overall, our study emphasizes how partial density matrices can be used to study order parameters and phase diagrams in quantum matter, while confirming the robustness and entanglement structure of SPT phases. It highlights a powerful synergy between quantum mechanics and machine learning, where correlations in reduced representations enable accurate predictions from minimal data.

Chapter 8

Analysis of Quantum Image Representations for Supervised Classification

The research presented in this chapter serves as the foundation of a paper published in AVS Quantum Science, authored by (in order) Marco Parigi, Mehran Khosrojerdi, Filippo Caruso, and Leonardo Banchi.

This chapter begins with a brief introduction, followed by an explanation of Quantum Image Processing approaches. We then present the methodology and discuss the results, focusing on the Fashion MNIST and MNIST datasets.

8.1 Introduction

QIP is a research field of Quantum Information Science that attempts to overcome the limitations of classical computers, harnessing the peculiar properties of quantum mechanical systems, e.g. *quantum superposition* and *entanglement*, to provide a more efficient way to store, manipulate, and extract visual information from digital images [289, 290].

On a classical computer, a black and white digital image with $N = 2^n \times 2^n$ pixels is represented as a matrix and encoded by N bits. In contrast, the number of qubits needed to store the same image on a quantum device can be $\mathcal{O}(\log N)$. We therefore obtain an *exponential* reduction in the computational space resources to store images on quantum processors [290]. Furthermore, quantum operations [291, 292] such as Fourier, Hadamard, and Haar wavelet transforms, which are usually included as subroutines in image processing tasks, provide *exponential* speed-up over their classical counterparts [290, 293, 294], which is nonetheless challenged by state preparation costs [295].

Reduced storage requirements can also provide higher accuracies in machine learning applications. Indeed, there is a tight connection between learning and compression, which has been explored in both classical deep learning [296] and quantum learning settings [228, 229]. This connection is quite intuitive: even humans learn new topics not just by merely memorizing all the “training data” rather than by extracting and compressing *relevant* information from training sources, such as books.

By exploiting the superposition principle, quantum states can compress classical data, not only images [297], with exponentially fewer resources without lose accuracy. Moreover, the advantage still persists when some loss or noise is tolerated, paving the way to machine learning applications [298, 299]. Recent studies have demonstrated that hybrid quantum–classical architectures can leverage these compression principles for image data, leading to promising results in image classification performance on NISQ devices [248, 300].

The Quantum Image Representation (QImR) is a subarea of QIP that focuses on image processing tasks and how well they can be performed on quantum hardware. In a classical computer, digital images are defined as matrices of numbers representing the discrete color or intensity values present in every image pixel. In other words, an image is an object that carries two pieces of information: the position and the color of each pixel. Over the years, many QImRs have been proposed that employ different models and techniques to encode the intensity and position information of pixels [289, 301].

A general quantum state that encodes color and position information for each pixel a digital image I has the following form:

$$|I\rangle = \frac{1}{\sqrt{N}} \sum_{P=0}^{N-1} |C_P\rangle \otimes |P\rangle, \quad (8.1.1)$$

where N is the number of pixels, and the two quantum states, $|C_P\rangle$ and $|P\rangle$, respectively encode the color and the position of a pixel.

Various models for representing quantum images have been investigated and implemented recently on NISQ devices. Das et al. [302] notably realized a quantum pattern recognition protocol on a real Quantum Processing Unit (QPU) [303] choosing the QPIE [290]. Furthermore, Geng et al. [304] explored the possibilities of implementing QImRs on superconducting and trapped-ion quantum computers, and successfully implemented a 2×2 FRQI [305] image.

In addition to these approaches, tensor network based quantum image representations have shown strong potential for compressing and representing image data in a way that is naturally compatible with quantum circuit architectures. In fact, Stoudenmire et al. [100] demonstrated how quantum-inspired tensor networks can be applied effectively to supervised learning tasks, especially image classification, by representing data with reduced dimensionality while preserving key features.

In this work, we consider four representations for grayscale images that incorporate the classical image information differently, namely: i) TNR [306], ii) FRQI [305]; iii) NEQR [307] and iv) QPIE [290]. Table 8.1 summarizes the most important parameters and characteristics of the four encoding methods.

The rest of this chapter is organized as follows. We first describe how these representations encode the color and position of pixels, discussing their advantages and drawbacks. Then we investigate the trade-off between accuracy and memory in binary classification and finally discuss a few applications for learning to classify images.

Quantum Image Representation	Color encoding	Qubit resource	Complexity
TNR	Tensor	$2n + 1$	–
FRQI	1 Angle	$2n + 1$	$\mathcal{O}(2^{4n})$
NEQR	Basis States	$2n + q$	$\mathcal{O}(qn2^{2n})$
QPIE	Amplitudes	$2n$	$\mathcal{O}(2^n)$

Table 8.1. Comparison of key parameters and characteristics of four quantum image representations, TNR, FRQI, NEQR, and QPIE, for encoding images of size $2^n \times 2^n$ with grayscale values 2^q .

8.2 Quantum Image Processing

8.2.1 Tensor Network Representation

In quantum mechanics, a *ket* is a mathematical construct used to represent states in a Hilbert space. This concept can be applied to images by mapping pixel values into a structured quantum format. Using a block-structured addressing method, pixels of an image can be organized in a way that reflects their spatial relationships, allowing them to be represented as quantum state with real amplitudes, here called *real-kets*. This transformation provides a foundation for analyzing images using quantum-inspired techniques. Each pixel in an image can be represented using a certain number of classical bits. For instance, in an 8-bit grayscale image each pixel is a integer between 0 (black) and white (255). An image with $2^n \times 2^n$ pixels can be mathematically represented as a real-ket as follows [306]

$$|\psi_{2^n \times 2^n}\rangle = \sum_{i_1=1}^4 \cdots \sum_{i_n=1}^4 c_{i_n, \dots, i_1} |i_n \cdots i_1\rangle, \quad (8.2.1)$$

where $c_{i_n, \dots, i_1}$ stores the pixel values, demonstrating how the image is encoded in a structured quantum state. As an example for better understanding is shown in 8.2.1 for $n = 2$. In other terms, an image is iteratively reduced in four larger blocks, each indexed with an integer $i = 1, \dots, 4$. This reduction is performed n times until each block is composed only of a single pixel. The position of each pixel can hence be expressed using n integers $1 \dots 4$. In the example of 8.2.1, each sub-block is assigned an index based on its position: $i_2 = 1$ corresponds to the upper-left pixel, $i_2 = 2$ to the upper-right, $i_2 = 3$ to the lower-left, and $i_2 = 4$ to the lower-right. This labeling scheme provides a structured way to organize pixel positions within the block. To specify which sub-block is being addressed within the larger structure, we introduce an additional label, i_1 , following the same convention used for the inner sub-blocks. This hierarchical labeling ensures a consistent and scalable representation of pixel organization.

Images with varying textures and structures exhibit different levels of entanglement, capturing the complexity of pixel correlations. This formulation enables the use of mathematical expressions to describe an image as a quantum state, where tensors encode the relationships between pixels. Once the image is mapped to

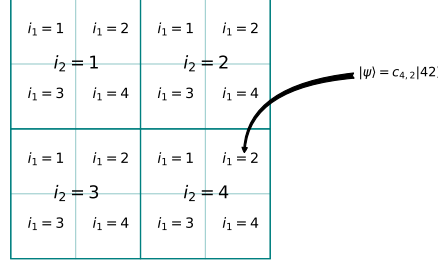


Figure 8.2.1. Example real-ket indexing for an image with $2^2 \times 2^2$ pixels

a quantum state, a particularly useful framework for compressing it is via TNs methods or, specifically, MPS. A MPS provides a compact way to encode an image by organizing pixel information into an array of tensors that efficiently capture entanglement between different regions. The MPS representation can be mathematically expressed as

$$|\psi_{\Gamma}\rangle = \sum_{\{i_j\}, \{\lambda_j\}} \Gamma_{\lambda_1 \lambda_2}^{(1) i_1} \Gamma_{\lambda_2 \lambda_3}^{(2) i_2} \cdots \Gamma_{\lambda_n \lambda_1}^{(n) i_n} |i_n, \cdots i_2, i_1\rangle, \quad (8.2.2)$$

where $\Gamma^{(j)}$ are tensors with real elements. The advantage of MPS lies in its ability to represent images with low entanglement using fewer parameters, making it an efficient tool for compression and structured encoding. Since the entanglement in MPS corresponds to classical pixel correlations, it offers a natural bridge between quantum mechanics and traditional image processing. To further illustrate the concept of MPS at this stage, we refer to Figure 8.2.2. This figure visually demonstrates how blocks and sub-blocks are interconnected and how a Tree Tensor Network (TTN) in one dimension can be formulated as an MPS representation, providing a structured approach to encoding pixel relationships of a typical image.

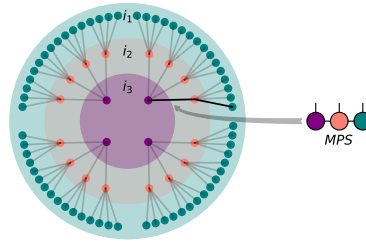


Figure 8.2.2. A typical image in the framework of tensor networks

By leveraging real-kets and MPS, images can be encoded efficiently and the structured nature of these representations allows for a deeper understanding of how images store information.

8.2.2 Flexible Representation of Quantum Images

In 2010, Le et al. [305] proposed Flexible Representation of Quantum Images (FRQI) that uses quantum superposition to encode a classical image in the form of a normalized quantum state. More precisely, inspired by the pixel representation for images in a classical computers, the FRQI encodes a $2^n \times 2^n$ gray-scale image in a quantum state as follow:

$$|I(\theta)\rangle = \frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} |c(\theta_i)\rangle \otimes |i\rangle, \quad (8.2.3)$$

where the color-state,

$$|c(\theta_i)\rangle = \cos \theta_i |0\rangle + \sin \theta_i |1\rangle, \quad (8.2.4)$$

with $|0\rangle$ and $|1\rangle$ are the two-dimensional quantum states of the computational basis, $|i\rangle$ is a sequence of $2n$ qubits storing the position of the pixel, and $\theta_i \in [0, \frac{\pi}{2}]$ is the angle that encodes the color of the i -th pixel. An example of a 2×2 FRQI image is illustrated in 8.2.3. The FRQI requires $2n + 1$ qubits to represent an image on a quantum device, exponentially reducing the space resources needed in the classical case. Specifically, $2n$ qubits encode the positions of the pixels, while a single qubit stores the grayscale information of all pixels. These are commonly referred to as position qubits and color qubit, respectively. Furthermore, this model provides a series of efficient color [308] and fast geometric transformations [309]. In addition to image processing tasks, the FRQI encoding has also been employed in non-visual domains. For instance, it has recently been used in biological applications, such as DNA sequencing, by mapping genomic information into the FRQI quantum image representation [310].

The two main FRQI drawbacks are image retrieval and the depth of the quantum circuit. The first is due to the color encoding of each pixel that is in the probability amplitudes of a qubit, and consequently an infinite number of measurements are necessary to exactly reconstruct the initial image. Therefore, the original classical image can only be retrieved probabilistically. The second significant drawback is that the preparation of a FRQI state is based on the polynomial preparation theorem [305] and has a computational complexity of $\mathcal{O}(2^{4n})$. The implementation of a FRQI circuit begins by applying Hadamard gates to the position qubits to create a uniform superposition encoding the position of pixels. Then, for each pixel, a multi-controlled rotation gate is applied to the color qubit, where the rotation angle θ depends on the intensity of the pixel. However, the last drawback has recently been overcome by Nasr et al. [311] who introduced Enhanced Flexible Representation of Quantum Images (EFRQI) reducing the computational complexity to $\mathcal{O}(2n2^{2n})$.

θ_0 00	θ_1 01
θ_2 10	θ_3 11

Figure 8.2.3. A 2×2 FRQI image. The angles θ_i , with $i = 0, 1, 2, 3$, at the center of each pixel encode the grayscale intensity of the corresponding pixels in the angle representation, while the 2-bit strings in the lower-right corner identify the positions in the image in binary representation.

8.2.3 Novel Enhanced Quantum Representation

The NEQR representation has been proposed by Zhang et al. [307] to overcome the limitations of FRQI. More precisely, to improve image retrieval, the NEQR model encodes the digital image in a superposition of two entangled qubit sequences storing grayscale and position information. Formally, a $2^n \times 2^n$ image with the intensity range 2^q (where q equals 8 for 256 intensity gray levels) is stored in a NEQR state of the form:

$$|I\rangle = \frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} |f(y, x)\rangle \otimes |yx\rangle, \quad (8.2.5)$$

where $f(y, x) \in [0, 2^q - 1]$ is a q -bit sequence $C_{yx}^0 \dots C_{yx}^{q-2} C_{yx}^{q-1}$ encoding the gray-scale value of yx pixel:

$$f(y, x) = C_{yx}^0, C_{yx}^1, \dots, C_{yx}^{q-2}, C_{yx}^{q-1}, \quad C_{yx}^k \in [0, 1],$$

and position states,

$$|yx\rangle = |y_0 y_1 \dots y_{n-1}\rangle \otimes |x_0 x_1 \dots x_{n-1}\rangle.$$

The NEQR requires $2n + q$ qubits to encode image in a quantum computer. In particular, $2n$ qubits are used to encode the pixel positions and q qubits to store the grayscale intensity values. An example of a 2×2 NEQR image is shown in 8.2.4. Because NEQR uses different basis states of qubits, the original classical image can be accurately retrieved through quantum measurements. In addition, some complex color operations, such as partial color operations, can be performed [307]. Moreover, NEQR achieves a quadratic speedup in quantum image preparation with respect to FRQI taking a computational complexity of $\mathcal{O}(2qn2^{2n})$. The implementation of a NEQR circuit begins by applying Hadamard gates to the position qubits, creating a uniform superposition of all pixel positions. Then, multi-controlled NOT quantum gates are applied to encode the grayscale intensity information into the q color

qubits, conditioned on the corresponding pixel positions. Recently, Enhanced Novel Enhanced Quantum Representation (ENEQR) has been proposed by Nasr et al. [311] that requires computational complexity $\mathcal{O}(2n2^{2n})$.

11110000 00	01000100 01
10010100 10	01001001 11

Figure 8.2.4. A 2×2 NEQR image. The 8-bit strings at the center of each pixel and the 2-bit strings in the lower-right corner encode, respectively, the gray intensity and the positions of the pixels in the image.

8.2.4 Quantum Probability Image Encoding

Quantum Probability Image Encoding (QPIE), proposed by Yao et al. [290] uses the probability amplitudes of pure quantum states to store the grayscale values of pixels. Formally, given a $2^n \times 2^n$ grayscale digital image, the QPIE state is the following:

$$|I\rangle = \sum_{i=0}^{2^{2n}-1} c_i |i\rangle, \quad c_i = \frac{I_i}{\sqrt{\sum I_i^2}}, \quad (8.2.6)$$

where I_i is the color intensity of i -th pixel, and c_i is the normalized intensity so that the squared sum of all the probabilities amplitudes is equal to 1. An example of the QPIE image is illustrated in 8.2.5. For the representation on a quantum computer of a grayscale image of size $2^n \times 2^n$, the QPIE only uses $2n$ qubits. This model therefore requires the least number of qubits compared to the FRQI and NEQR. However, because pixel values are stored in the state amplitude, QPIE has the same drawback as FRQI about retrieval of the original image from a finite number of measurements. The preparation of a n -qubit pure quantum state to a specific probability distribution computational complexity $\mathcal{O}(2^n)$ [258, 312]. Implementing a QPIE circuit is equivalent to performing amplitude embedding. The method described by Möttönen et al. provides an efficient way to construct such circuits using uniformly controlled rotations [313].

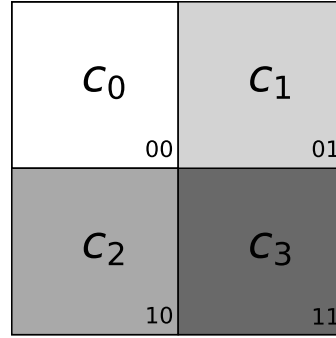


Figure 8.2.5. A 2×2 QPIE image. The amplitudes c_i , with $i = 0, 1, 2, 3$, at the center of each pixel encode the gray intensity of the corresponding pixels, while the 2-bit strings in the lower-right corner label the pixel positions in the image in binary representation.

8.3 Classical and Quantum Kernel Methods

In supervised Machine Learning (ML) problems, kernel methods [314] are a collection of algorithms commonly used for classification and regression tasks. The core idea of the kernel method is based on the use of the *feature maps* $\phi : X \rightarrow F$ that represent the original data points from the input space X to a higher dimensional feature space F , where classification among classes becomes more simple. Kernel methods avoid explicit calculation of the new representation of the point coordinates in the new space F and simply compute the inner products between two data points via a bi-variate function called *kernel* $K : X \times X \rightarrow \mathbb{R}$ [314]. Formally, let us consider a classification problem and let $D = \{(x_i, y_i)\}_{i=1}^N$ be the dataset, where $\mathbf{x}_i$ is the data points and y_i their corresponding labels. Then, given a feature map $\phi : X \rightarrow F$, a kernel function K is written as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle, \quad (8.3.1)$$

where $\langle \cdot, \cdot \rangle$ is the inner product that must satisfy the Mercer condition of positive semi-definiteness [46, 315]

$$\sum_{i=1}^N \sum_{j=1}^N K(\mathbf{x}_i, \mathbf{x}_j) c_i c_j \geq 0 \quad (8.3.2)$$

for all choices of real numbers $(c_1, \dots, c_n)$. This procedure is known as *kernel trick* and allows us to also work when the data points $\mathbf{x}$ are not linearly separable. The choice typically depends on the characteristics of the data and the task. A straightforward example is the linear kernel, which is given by the formula:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j. \quad (8.3.3)$$

SVM is well-known kernel based algorithm [48, 316]. The kernel method in SVM is a powerful technique that enables effective classification by mapping data into a

higher-dimensional space, allowing for the identification of an optimal separating hyperplane even in cases where linear separation is infeasible. More precisely, SVM commonly used in ML classification problems with the objective of finding the optimal hyperplane by maximizing the margin between the closest data points of opposing classes. The kernel trick is crucial for constructing the optimal hyperplane by solving the dual optimization problem, which maximizes the function:

$$L_D(\alpha) = \sum_{i=1}^t \alpha_i - \frac{1}{2} \sum_{i,j=1}^t y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (8.3.4)$$

with constraints $\sum_{i=1}^t \alpha_i y_i = 0$ and $\alpha_i \geq 0$ for all i . The solution is given by a nonnegative vector $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_t)$. The optimal solution $\boldsymbol{\alpha}^*$ is subsequently employed to construct the classifier:

$$m(\mathbf{s}) = \text{sign} \left(\sum_{i=1}^t y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{s}) + b \right), \quad (8.3.5)$$

where $\mathbf{s}$ is a datum of test set [11] and b is bias.

In the context of quantum computing, the kernel is constructed using quantum circuits to enhance the computational efficiency of classification using SVM [11, 215]. Specifically, the quantum kernel estimation involves calculating the overlap between quantum states corresponding to the training data points [258]. To start we consider zero states $|00 \dots 0\rangle$ of N -dimensional Hilbert space $\mathcal{H}$, and then we apply an operator $U_\phi(\mathbf{x})$ over these states and map the classical data point $\mathbf{x}$ to a vector in Hilbert space. More precisely, the quantum state that encodes data point $\mathbf{x}$ is given by:

$$|\phi(\mathbf{x})\rangle = U_\phi(\mathbf{x}) |00 \dots 0\rangle, \quad (8.3.6)$$

and the quantum kernel between two different pure quantum state is computed by:

$$\begin{aligned} K(\mathbf{x}_i, \mathbf{x}_j) &= |\langle 0 \dots 00 | U_\phi^\dagger(\mathbf{x}_i) U_\phi(\mathbf{x}_j) |00 \dots 0\rangle|^2 \\ &= |\langle \phi(\mathbf{x}_i) | \phi(\mathbf{x}_j) \rangle|^2, \end{aligned} \quad (8.3.7)$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two classical data points.

8.3.1 Compression-accuracy trade-offs

Before presenting numerical results, we first discuss some theoretical arguments that directly follow from the information theoretic analysis of Refs. [228, 229]. First of all, from the kernel it is possible to compute the average information content of an ensemble of quantum states. For instance, let $\{|\phi(\mathbf{x}_i)\rangle\}_{i=1}^N$ be the ensemble of training states, which can also be expressed as a density matrix

$$\rho = \frac{1}{N} \sum_{i=1}^N |\phi(\mathbf{x}_i)\rangle \langle \phi(\mathbf{x}_i)|. \quad (8.3.8)$$

The information content of such an ensemble, which is directly linked to its compressibility, can be quantified using either Von Neumann or Rényi entropies

$S_\gamma(\rho) = \frac{1}{1-\gamma} \log_2 \text{Tr}[\rho^\gamma]$. The latter can also be expressed [228] as $S_\gamma(\rho) = S_\gamma(K)$ where the “density matrix” K has elements $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)/N$ and satisfies the usual properties: $K \geq 0$ and $\text{Tr } K = 1$. Thanks to such result, the entropy of the ensemble (8.3.8) is equal to the entropy of the normalized kernel matrix (8.3.7), which does not directly depend on the dimensionality of the quantum Hilbert space, though it is upper bounded by it. While the number of qubits in the embedding (8.1.1) quantifies the memory requirement of a single image, the entropy quantifies the information content of an ensemble. Large entropy is possible when the kernel matrix is close to an identity matrix, namely when the offdiagonal elements become small, which happens when the different states $|\phi(\mathbf{x}_i)\rangle, |\phi(\mathbf{x}_j)\rangle$ become almost orthogonal for $i \neq j$. Therefore, the non-orthogonality of the quantum embedding states, which comes from storing the different pixels via quantum superposition, is the central ingredient behind the enhanced memory storage.

From these considerations we can draw a few conclusions, assuming a link between the entropic quantifiers with different α . On one hand, smaller entropy guarantees better learning, since the generalization error is linked to $S_2(\rho)$ [228]. On the other hand, due to Holevo’s theorem [317], $S_1(\rho)$ provides a limit to the amount of information about the embedded images that can be retrieved via quantum measurements, so a smaller entropy means that the training images are more “hidden”. Nonetheless, this is not necessarily a problem in learning settings, where the task of the algorithm is not to retrieve the image $\mathbf{x}_i$ but rather its label y_i . As long as the quantum state embedding has enough information about the label y_i , accurate prediction is still possible.

8.4 Results

Simulations are performed using PennyLane [318] and Scikit-Learn [319], which are open-source software for quantum and classical machine learning, respectively. In particular, PennyLane is used to implement quantum circuits that embed classical images into quantum states, while Scikit-Learn is employed for the SVM-based classification. In this study, we adopt the MNIST and Fashion MNIST [320] dataset whose samples are illustrated in 8.4.1. The size of each image is reduced to 16×16 to encode them into QImRs. This is performed using the module `cv2.INTER_AREA` of OpenCV [321]. The color of each pixel is resized to $[0, 255]$. Furthermore, the training and testing sets consist of 10^3 and 10^2 images, respectively.

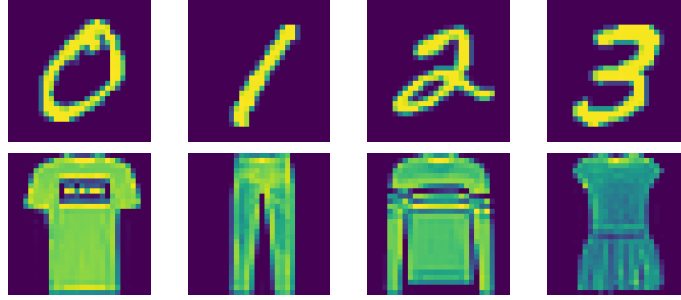


Figure 8.4.1. Image samples of MNIST (top row) and Fashion MNIST (bottom row) datasets.

We perform binary classification using SVM model in the noiseless case. A summary describing the simulation parameters is reported in Table 8.2.

Component	Description
Datasets	MNIST and Fashion-MNIST datasets, each consisting of 10^3 training and 10^2 test images per binary classification task (class 0 vs. classes 1–9).
Data Preprocessing	Grayscale images resized to 16×16 pixels and the intensity is between $[0, 255]$. Model performance is evaluated using k-fold cross-validation.
Model	Support Vector Machine (SVM) with a precomputed kernel, implemented using the <code>scikit-learn</code> library.

Table 8.2. Implementation details for SVM-based binary classification using quantum image representations kernel and classical linear kernel.

Given two classical digital images I and I' encoded on the pure quantum states $|I\rangle$ and $|I'\rangle$, respectively, the kernel operator K used in ideal case is given by:

$$K_{I,I'} = |\langle I|I'\rangle|^2, \quad (8.4.1)$$

where $\langle \cdot | \cdot \rangle$ denote the inner product. In the following we calculate QImRs-based quantum kernels used in the simulations via SVM model.

The inner product between two MPS states 8.2.2 reads,

$$\langle \psi_\Gamma | \psi_{\Gamma'} \rangle = \sum_{\{i_k\}, \{\lambda_k\}, \{\beta_k\}} \prod_k \Gamma_{\lambda_k \lambda_{k+1}}^{(k) i_k^*} \Gamma_{\beta_k \beta_{k+1}}'^{(k) i_k} \quad (8.4.2)$$

and can be computed efficiently using standard contraction techniques [322] Defining:

$$E_k = \sum_{i_k} \Gamma_{\lambda_k \lambda_{k+1}}^{(k) i_k^*} \Gamma_{\beta_k \beta_{k+1}}'^{(k) i_k}, \quad k = 1, 2, \dots, n, \quad (8.4.3)$$

the TNR quantum kernel between two MPS states can be written in compact form as:

$$|\text{Tr}(E_1 E_2 \cdots E_n)|^2. \quad (8.4.4)$$

Given two FRQI states 8.2.3, the inner product between them is the following:

$$\begin{aligned} \langle I(\theta) | I(\theta') \rangle &= \frac{1}{2^{2n}} \sum_{i,j=0}^{2^{2n}-1} \langle c(\theta_i) | c(\theta_j) \rangle \otimes \langle i | j \rangle \\ &= \frac{1}{2^{2n}} \sum_{i,j=0}^{2^{2n}-1} (\cos \theta_i \cos \theta'_i + \sin \theta_i \sin \theta'_i) \\ &= \frac{1}{2^{2n}} \sum_{i=0}^{2^{2n}-1} \cos(\theta_i - \theta'_i), \end{aligned}$$

where we use that the orthonormal property of the basis state, i.e. $\langle i | j \rangle = \delta_{i,j}$. Thus, the FRQI quantum kernel is given by:

$$K_{I,I'} = \left| \frac{1}{2^{2n}} \sum_{i=0}^{2^{2n}-1} \cos(\theta_i - \theta'_i) \right|^2. \quad (8.4.5)$$

For two NEQR state 8.2.5 the inner product reads:

$$\begin{aligned} \langle I | I' \rangle &= \frac{1}{2^{2n}} \sum_{y,y',x,x'=0}^{2^n-1} \langle f'(y,x) | f'(y',x') \rangle \otimes \langle yx | y'x' \rangle \\ &= \frac{1}{2^{2n}} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} \delta_{f,f'} \delta_{yx,y'x'}, \end{aligned}$$

where in the last equality we use the orthonormal property of the basis states of the same state space. Then, a NEQR quantum kernel is equal to:

$$K_{I,I'} = \left| \frac{1}{2^{2n}} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} \delta_{f,f'} \delta_{yx,y'x'} \right|^2. \quad (8.4.6)$$

Finally, let us consider the inner product between two QPIE states 8.2.6 is the following:

$$\begin{aligned} \langle I | I' \rangle &= \sum_{i,j=0}^{2^{2n}-1} c_i c'_j \langle i | j \rangle \\ &= \sum_{i=0}^{2^{2n}-1} c_i c'_i. \end{aligned}$$

Accordingly, the QPIE quantum kernel is given by:

$$K_{I,I'} = \left| \sum_{i=0}^{2^{2n}-1} c_i c'_i \right|^2. \quad (8.4.7)$$

QImRs	Quantum State	Quantum Kernel
TNR	$\sum_{i's} \sum_{\lambda's} \Gamma_{\lambda_1 \lambda_2}^{(1)i_1} \Gamma_{\lambda_2 \lambda_3}^{(2)i_2} \cdots \Gamma_{\lambda_n \lambda_1}^{(n)i_n} i_n \cdots i_2, i_1\rangle$	$ \text{Tr}(E_1 E_2 \cdots E_n) ^2$
FRQI	$\frac{1}{2^n} \sum_{i=0}^{2^{2n}-1} (\cos \theta_i 0\rangle + \sin \theta_i 1\rangle) \otimes i\rangle$	$\left \frac{1}{2^{2n}} \sum_{i=0}^{2^{2n}-1} \cos(\theta_i - \theta'_i) \right ^2$
NEQR	$\frac{1}{2^n} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} f(y, x)\rangle \otimes yx\rangle$	$\left \frac{1}{2^{2n}} \sum_{y=0}^{2^n-1} \sum_{x=0}^{2^n-1} \delta_{f,f'} \delta_{yx,y'x'} \right ^2$
QPIE	$\sum_{i=0}^{2^{2n}-1} c_i i\rangle$	$\left \sum_{i=0}^{2^{2n}-1} c_i c'_i \right ^2$

Table 8.3. We report four different quantum image representations with their quantum states and kernels for $2^n \times 2^n$ grayscale images.

A summary of QImR-based quantum kernels used in the simulations by the SVM model are reported in 8.3.

Kernels	Storage Cost	Accuracy MNIST	Accuracy Fashion MNIST
TNR	9 qubits	0.99 ± 0.04	0.95 ± 0.02
FRQI	9 qubits	0.99 ± 0.00	0.95 ± 0.02
NEQR	16 qubits	0.99 ± 0.00	0.94 ± 0.02
QPIE	8 qubits	0.99 ± 0.00	0.96 ± 0.02
Linear	2 048 bits	0.99 ± 0.00	0.96 ± 0.02

Table 8.4. We report the mean and SEM of the accuracy for different quantum and classical kernels for a binary classification between the pair of classes: $(0, 1), (0, 2), \dots, (0, 9)$ of MNIST and Fashion MNIST datasets. The bond dimension for the TNR is equal to 6. Additionally, the storage cost for a 16×16 grayscale image is reported in terms of qubits (quantum case) and bits (classical case). Quantum kernels on average perform binary classification with the same accuracy as the classical linear kernel, but with exponentially lower storage cost.

Successively, to investigate the compression performance of the four QImRs, we compute the Gram matrices of 100 training images for two classes 0 and 1, and show them in 8.4.2. In fact, the elements of the Gram are the overlap between two quantum states that encode the information of two classical images. In particular, the elements of the Gram matrix take values in the interval $[0, 1]$, where 1 indicates that the two quantum states completely overlap, while 0 do not overlap. The different off-diagonal colorings of the four Gram matrices represent the overlap value between two quantum states encoding different images. From 8.4.2, we see that FRQI and QPIE perform a high compression of the classical image information with respect to NEQR in the encoding procedure. In fact, the most off-diagonal elements of FRQI and QPIE Gram matrices are in the range $[0.8, 1]$, while for NEQR case they are in the interval $[0, 0.4]$. The TNR Gram matrix shows an intermediate level of compression with respect to FRQI, QPIE, and NEQR, and its elements are in the range $[0.4, 0.7]$.

We computed the mean and Standard Error on the Mean (SEM) of the accuracy for binary classification between the pair of classes: $(0, 1), (0, 2), \dots, (0, 9)$. The results are reported in 8.4. The accuracy is obtained by the `accuracy_score` function

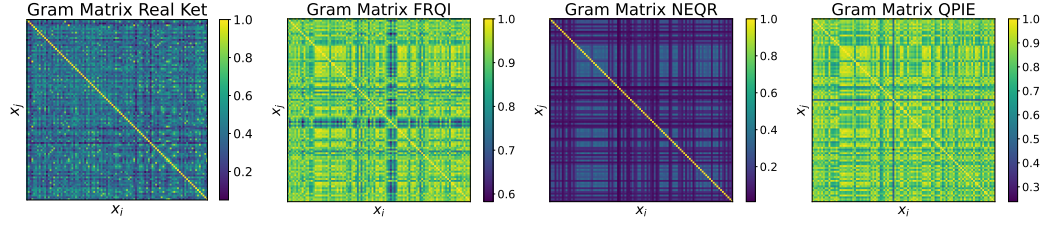


Figure 8.4.2. The Gram matrices of 100 training images x_i from Fashion MNIST for the quantum kernels: TNR, FRQI, NEQR and QPIE, respectively from left to right. Classes 0 and 1 have been selected. The elements on the diagonal are the kernel of a quantum state $|I\rangle$ with itself, while the elements off the diagonal are the kernel between two different quantum states $|I\rangle$ and $|I'\rangle$. The different off-diagonal coloring of the four Gram matrices, describing the overlap value of two quantum states, gives information on how the different QImRs compress the classical information into quantum states.

of Scikit-Learn [319] library and is computed by equation:

$$\text{accuracy}(y, \hat{y}) = \frac{1}{N} \sum_{i=0}^{N-1} 1(\hat{y}_i = y_i), \quad (8.4.8)$$

where y_i and $\hat{y}_i$ are, respectively, true and the predicted value of the i -sample, N is the size of the dataset, and $1(\cdot)$ is the indicator function. Our findings show that QImR-based kernels achieve average accuracy comparable to classical linear kernel on both datasets, while requiring exponentially fewer computational resources to store the image data. The results are reported in 8.4. For the TNR, the SVM model's classification accuracy is evaluated across different bond dimensions for both the MNIST and Fashion-MNIST datasets, as illustrated in 8.4.3. The x-axis represents the bond dimension, which governs the level of entanglement captured by the MPS, while the y-axis shows the classification accuracy (%) achieved using the MPS-encoded kernel. The plot suggests that at low bond dimensions, accuracy is limited due to insufficient encoding capacity. However, as the bond dimension increases, the accuracy improves, indicating that the MPS representation captures richer image features, enhancing classification performance. The model attains its peak accuracy of [99.4%] on the MNIST dataset and [95.4%] on the Fashion-MNIST dataset at a bond dimension of four, indicating that moderate bond dimensions are sufficient to capture the essential image correlations. Overall, the results show that the average accuracy for MNIST is consistently higher than for Fashion-MNIST, as expected due to the simpler visual structure of the MNIST digits. Moreover, for MNIST, the accuracy converges once the bond dimension reaches four, whereas for Fashion-MNIST, convergence occurs only after the bond dimension exceeds six.

8.5 Discussion

In this work, we compare different quantum image representation techniques for grayscale digital image to qualitatively analyze the degree of compression when classical image information is encoded in a quantum state and to study the trade-off between accuracy and memory for binary classification problems using kernel methods.

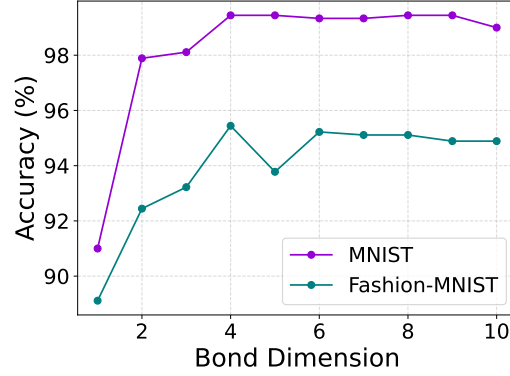


Figure 8.4.3. Classification accuracy of the SVM model as a function of bond dimension

In order to investigate how the four QImRs compress the classical information of an image, we compute their Gram matrices whose elements are the fidelity (overlap) between two quantum encoding states. We find that with the FRQI and QPIE representation, the quantum states are very close to each other after the encoding procedure, while NEQR shows the opposite scenario. In other words, when the classical information of the images is encoded in quantum systems, FRQI and QPIE performs higher compression than NEQR. For TNR encoding, instead we have an intermediate scenario between FRQI, QPIE and NEQR.

Subsequently, we perform image binary classification using quantum kernels based on QImR and compare their performance with the classical linear kernel. We find that the mean accuracy of quantum kernels is comparable to the classical one, but quantum kernels require exponentially fewer computational resources to store the image than their classical counterpart. However, we highlight that to date the best known protocols for loading an exact representation of the data into an n -qubit state require a number of quantum gates equal to $\mathcal{O}(2^n)$, which predominates the complexity of quantum algorithms and compromises its potential quantum advantage [295]. An efficient loading of the classical data into a quantum state is still an open question. A future research direction could be to extend the study to RGB images and compare them with other types of classical kernels. Finally, another interesting perspective could be the possibility of studying the accuracy performance of QImR kernels when they are implemented on quantum hardware and noise effects occur.

Bibliography

- [1] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, 2015.
- [2] Nimish Mishra, Manik Kapil, Hemant Rakesh, Amit Anand, Nilima Mishra, Aakash Warke, Soumya Sarkar, Sanchayan Dutta, Sabhyata Gupta, Aditya Prasad Dash, et al. Quantum machine learning: A review and current status. *Data Management, Analytics and Innovation: Proceedings of ICDMAI 2020, Volume 2*, pages 101–145, 2021.
- [3] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.
- [4] Nathan Wiebe, Ashish Kapoor, and Krysta M Svore. Quantum deep learning. *arXiv preprint arXiv:1412.3489*, 2014.
- [5] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.
- [6] Mikko Möttönen, Juha J Vartiainen, Ville Bergholm, and Martti M Salomaa. Quantum circuits for general multiqubit gates. *Physical review letters*, 93(13):130502, 2004.
- [7] Martin Plesch and Časlav Brukner. Quantum-state preparation with universal gate decompositions. *Physical Review A—Atomic, Molecular, and Optical Physics*, 83(3):032302, 2011.
- [8] Phillip Kaye and Michele Mosca. Quantum networks for generating arbitrary quantum states. In *International Conference on Quantum Information*, page PB28. Optica Publishing Group, 2001.
- [9] Dan Ventura and Tony Martinez. Quantum associative memory. *Information sciences*, 124(1-4):273–296, 2000.
- [10] Antonio Tudisco, Andrea Marchesin, Maurizio Zamboni, Mariagrazia Graziano, and Giovanna Turvani. Evaluating angle and amplitude encoding strategies for variational quantum machine learning: their impact on model’s accuracy. *arXiv preprint arXiv:2508.00768*, 2025.

- [11] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.
- [12] Michael J Bremner, Bin Cheng, and Zhengfeng Ji. Instantaneous quantum polynomial-time sampling and verifiable quantum advantage: Stabilizer scheme and classical security. *PRX Quantum*, 6(2):020315, 2025.
- [13] Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622*, 2020.
- [14] Berta Casas and Alba Cervera-Lierta. Multidimensional fourier series with quantum circuits. *Physical Review A*, 107(6):062612, 2023.
- [15] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.
- [16] Matthias C Caro, Elies Gil-Fuster, Johannes Jakob Meyer, Jens Eisert, and Ryan Sweke. Encoding-dependent generalization bounds for parametrized quantum circuits. *Quantum*, 5:582, 2021.
- [17] Chih-Chieh Chen, Masaya Watabe, Kodai Shiba, Masaru Sogabe, Katsuyoshi Sakamoto, and Tomah Sogabe. On the expressibility and overfitting of quantum circuit learning. *ACM Transactions on Quantum Computing*, 2(2):1–24, 2021.
- [18] Jonas Kübler, Simon Buchholz, and Bernhard Schölkopf. The inductive bias of quantum kernels. *Advances in Neural Information Processing Systems*, 34:12661–12673, 2021.
- [19] Supanut Thanasilp, Samson Wang, Marco Cerezo, and Zoë Holmes. Exponential concentration in quantum kernel methods. *Nature communications*, 15(1):5200, 2024.
- [20] Abdulkadir Canatar, Evan Peters, Cengiz Pehlevan, Stefan M Wild, and Ruslan Shaydulin. Bandwidth enables generalization in quantum kernel models. *arXiv preprint arXiv:2206.06686*, 2022.
- [21] Valentin Heyraud, Zejian Li, Zakari Denis, Alexandre Le Boité, and Cristiano Ciuti. Noisy quantum kernel machines. *Physical Review A*, 106(5):052421, 2022.
- [22] Yabo Wang, Bo Qi, Xin Wang, Tongliang Liu, and Daoyi Dong. Power characterization of noisy quantum kernels. *IEEE Transactions on Neural Networks and Learning Systems*, 2025.
- [23] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021.

- [24] Tobias Haug and MS Kim. Generalization of quantum machine learning models using quantum fisher information metric. *Physical Review Letters*, 133(5):050603, 2024.
- [25] Matthias C Caro, Hsin-Yuan Huang, Marco Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, and Patrick J Coles. Generalization in quantum machine learning from few training data. *Nature communications*, 13(1):4919, 2022.
- [26] Sofiene Jerbi, Lukas J Fiderer, Hendrik Poulsen Nautrup, Jonas M Kübler, Hans J Briegel, and Vedran Dunjko. Quantum machine learning beyond kernel methods. *Nature Communications*, 14(1):1–8, 2023.
- [27] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17(9):1013–1017, 2021.
- [28] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439(1907):553–558, 1992.
- [29] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.
- [30] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O’Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):4213, 2014.
- [31] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *nature*, 549(7671):242–246, 2017.
- [32] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.
- [33] Kostas Blekos, Dean Brand, Andrea Ceschini, Chiao-Hui Chou, Rui-Hao Li, Komal Pandya, and Alessandro Summer. A review on quantum approximate optimization algorithm and its variants. *Physics Reports*, 1068:1–66, 2024.
- [34] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.
- [35] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv preprint arXiv:1802.06002*, 2018.
- [36] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.

- [37] Kerstin Beer, Dmytro Bondarenko, Terry Farrelly, Tobias J Osborne, Robert Salzmänn, Daniel Scheiermann, and Ramona Wolf. Training deep quantum neural networks. *Nature communications*, 11(1):808, 2020.
- [38] Jonathan Romero, Jonathan P Olson, and Alan Aspuru-Guzik. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, 2(4):045001, 2017.
- [39] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Quantum generative adversarial networks for learning and loading random distributions. *npj Quantum Information*, 5(1):103, 2019.
- [40] Jennifer R Glick, Tanvi P Gujarati, Antonio D Corcoles, Youngseok Kim, Abhinav Kandala, Jay M Gambetta, and Kristan Temme. Covariant quantum kernels for data with group structure. *Nature Physics*, 20(3):479–483, 2024.
- [41] Jiaqi Yang, Wei Xie, and Xiaohua Xu. Stability and generalization of quantum neural networks. *arXiv preprint arXiv:2501.12737*, 2025.
- [42] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Information-theoretic bounds on quantum advantage in machine learning. *Physical Review Letters*, 126(19):190505, 2021.
- [43] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature Communications*, 12:2631, 2021.
- [44] Sofiene Jerbi, Joe Gibbs, Manuel S Rudolph, Matthias C Caro, Patrick J Coles, Hsin-Yuan Huang, and Zoë Holmes. The power and limitations of learning quantum dynamics incoherently. *arXiv preprint arXiv:2303.12834*, 2023.
- [45] Matthias C Caro, Tom Gur, Cambyse Rouzé, Daniel Stilck Franca, and Sathyawageeswar Subramanian. Information-theoretic generalization bounds for learning from quantum data. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 775–839. PMLR, 2024.
- [46] James Mercer. Xvi. functions of positive and negative type, and their connection the theory of integral equations. *Philosophical transactions of the royal society of London. Series A, containing papers of a mathematical or physical character*, 209(441-458):415–446, 1909.
- [47] Nello Cristianini and John Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.
- [48] Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.
- [49] Maria Schuld. Supervised quantum machine learning models are kernel methods. *arXiv preprint arXiv:2101.11020*, 2021.

- [50] Gian Gentinetta, David Sutter, Christa Zoufal, Bryce Fuller, and Stefan Woerner. Quantum kernel alignment with stochastic gradient descent. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 1, pages 256–262. IEEE, 2023.
- [51] Giuseppe Montalbano and Leonardo Banchi. Quantum adversarial learning for kernel methods. *Quantum Machine Intelligence*, 7(1):15, 2025.
- [52] Geoffrey Schiebinger, Martin J. Wainwright, and Bin Yu. The geometry of kernelized spectral clustering. *The Annals of Statistics*, 43(2):819 – 846, 2015.
- [53] Abiodun M Ikotun, Absalom E Ezugwu, Laith Abualigah, Belal Abuhaija, and Jia Heming. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. *Information Sciences*, 622:178–210, 2023.
- [54] Duc Truong Pham, Stefan S Dimov, and Chi D Nguyen. Selection of k in k-means clustering. *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 219(1):103–119, 2005.
- [55] Trupti M Kodinariya, Prashant R Makwana, et al. Review on determining number of cluster in k-means clustering. *International Journal*, 1(6):90–95, 2013.
- [56] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [57] Adeiza James Onumanyi, Daisy Nkele Molokomme, Sherrin John Isaac, and Adnan M Abu-Mahfouz. Autoelbow: An automatic elbow detection method for estimating the number of clusters in a dataset. *Applied Sciences*, 12(15):7515, 2022.
- [58] Congming Shi, Bingtao Wei, Shoulin Wei, Wen Wang, Hai Liu, and Jialei Liu. A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm. *EURASIP journal on wireless communications and networking*, 2021(1):31, 2021.
- [59] Hao-Chung Cheng, Min-Hsiu Hsieh, and Ping-Cheng Yeh. The learnability of unknown quantum measurements. *arXiv preprint arXiv:1501.00559*, 2015.
- [60] Mohsen Heidari and Wojciech Szpankowski. New bounds on quantum sample complexity of measurement classes. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 1515–1520. IEEE, 2024.
- [61] Abram Magner and Arun Padakandla. Fat shattering, joint measurability, and pac learnability of povm hypothesis classes: A. magner, a. padakandla. *Quantum Information Processing*, 23(10):337, 2024.

- [62] Joe Gibbs, Zoë Holmes, Matthias C Caro, Nicholas Ezzell, Hsin-Yuan Huang, Lukasz Cincio, Andrew T Sornborger, and Patrick J Coles. Dynamical simulation via quantum machine learning with provable generalization. *Physical Review Research*, 6(1):013241, 2024.
- [63] Matthias C Caro, Hsin-Yuan Huang, Nicholas Ezzell, Joe Gibbs, Andrew T Sornborger, Lukasz Cincio, Patrick J Coles, and Zoë Holmes. Out-of-distribution generalization for learning quantum dynamics. *Nature Communications*, 14(1):3751, 2023.
- [64] Shima Bab Hadiashar, Ashwin Nayak, and Pulkit Sinha. Optimal lower bounds for quantum learning via information theory. *IEEE Transactions on Information Theory*, 70(3):1876–1896, 2023.
- [65] Patrick Huembeli and Alexandre Dauphin. Characterizing the loss landscape of variational quantum circuits. *Quantum Science and Technology*, 6(2):025011, 2021.
- [66] Abram Magner and Arun Padakandla. Empirical risk minimization and uniform convergence for probabilistically observed and quantum measurement hypothesis classes. In *2024 IEEE International Symposium on Information Theory (ISIT)*, pages 563–568. IEEE, 2024.
- [67] Yuxuan Du, Zhuozhuo Tu, Xiao Yuan, and Dacheng Tao. Efficient measure for the expressivity of variational quantum algorithms. *Physical Review Letters*, 128(8):080506, 2022.
- [68] Casper Gyurik, Vedran Dunjko, et al. Structural risk minimization for quantum linear classifiers. *Quantum*, 7:893, 2023.
- [69] Guangxi Li, Ruilin Ye, Xuanqiang Zhao, and Xin Wang. Concentration of data encoding in parameterized quantum circuits. *Advances in Neural Information Processing Systems*, 35:19456–19469, 2022.
- [70] Xin Wang, Yabo Wang, Bo Qi, and Rebing Wu. Limitations of amplitude encoding on quantum classification. *arXiv preprint arXiv:2503.01545*, 2025.
- [71] Ulrich Schollwöck. The density-matrix renormalization group in the age of matrix product states. *Annals of physics*, 326(1):96–192, 2011.
- [72] Ulrich Schollwöck. The density-matrix renormalization group. *Reviews of modern physics*, 77(1):259–315, 2005.
- [73] Guifré Vidal. Efficient simulation of one-dimensional quantum many-body systems. *Physical review letters*, 93(4):040502, 2004.
- [74] Jutho Haegeman, Christian Lubich, Ivan Oseledets, Bart Vandereycken, and Frank Verstraete. Unifying time evolution and optimization with matrix product states. *Physical Review B*, 94(16):165116, 2016.

- [75] Gregory M Crosswhite, Andrew C Doherty, and Guifré Vidal. Applying matrix product operators to model systems with long-range interactions. *Physical Review B—Condensed Matter and Materials Physics*, 78(3):035116, 2008.
- [76] G. Vidal. Classical simulation of infinite-size quantum lattice systems in one spatial dimension. *Phys. Rev. Lett.*, 98:070201, Feb 2007.
- [77] J Ignacio Cirac and Frank Verstraete. Renormalization and tensor product states in spin chains and lattices. *Journal of physics a: mathematical and theoretical*, 42(50):504004, 2009.
- [78] Roger Penrose et al. Applications of negative dimensional tensors. *Combinatorial mathematics and its applications*, 1(221-244):3, 1971.
- [79] Chris Camaño, Ethan N Epperly, and Joel A Tropp. Successive randomized compression: A randomized algorithm for the compressed mpo-mps product. *arXiv preprint arXiv:2504.06475*, 2025.
- [80] David Perez-Garcia, Frank Verstraete, Michael M Wolf, and J Ignacio Cirac. Matrix product state representations. *arXiv preprint quant-ph/0608197*, 2006.
- [81] Jutho Haegeman and Frank Verstraete. Diagonalizing transfer matrices and matrix product operators: A medley of exact and computational methods. *Annual Review of Condensed Matter Physics*, 8(1):355–406, 2017.
- [82] Xiongjie Yu, David Pekker, and Bryan K Clark. Finding matrix product state representations of highly excited eigenstates of many-body localized hamiltonians. *Physical review letters*, 118(1):017201, 2017.
- [83] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.
- [84] Thomas Barthel, Jianfeng Lu, and Gero Friesecke. On the closedness and geometry of tensor network state sets. *Letters in Mathematical Physics*, 112(4):72, 2022.
- [85] Peter Pippian, Steven R White, and Hans Gerd Evertz. Efficient matrix-product state method for periodic boundary conditions. *Physical Review B—Condensed Matter and Materials Physics*, 81(8):081103, 2010.
- [86] B Pirvu, Frank Verstraete, and G Vidal. Exploiting translational invariance in matrix product state simulations of spin chains with periodic boundary conditions. *Physical Review B—Condensed Matter and Materials Physics*, 83(12):125104, 2011.
- [87] Boris N Khoromskij. Tensor numerical methods for high-dimensional pdes: Basic theory and initial applications. *arXiv preprint arXiv:1408.4053*, 2014.
- [88] M. Fannes, B. Nachtergaele, and R. F. Werner. Finitely correlated states on quantum spin chains. *Communications in Mathematical Physics*, 144(3):443 – 490, 1992.

- [89] J Ignacio Cirac, David Perez-Garcia, Norbert Schuch, and Frank Verstraete. Matrix product states and projected entangled pair states: Concepts, symmetries, theorems. *Reviews of Modern Physics*, 93(4):045003, 2021.
- [90] Román Orús. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of physics*, 349:117–158, 2014.
- [91] Frank Verstraete, Valentin Murg, and J Ignacio Cirac. Matrix product states, projected entangled pair states, and variational renormalization group methods for quantum spin systems. *Advances in physics*, 57(2):143–224, 2008.
- [92] 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.
- [93] Glen Evenbly and Guifré Vidal. Tensor network states and geometry. *Journal of Statistical Physics*, 145(4):891–918, 2011.
- [94] 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.
- [95] Ho N Phien, Guifré Vidal, and Ian P McCulloch. Infinite boundary conditions for matrix product state calculations. *Physical Review B—Condensed Matter and Materials Physics*, 86(24):245107, 2012.
- [96] F Alexander Wolf, Ian P McCulloch, and Ulrich Schollwöck. Solving nonequilibrium dynamical mean-field theory using matrix product states. *Physical Review B*, 90(23):235131, 2014.
- [97] Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical review letters*, 91(14):147902, 2003.
- [98] YoonHaeng Hur, Jeremy G Hoskins, Michael Lindsey, E Miles Stoudenmire, and Yuehaw Khoo. Generative modeling via tensor train sketching. *Applied and Computational Harmonic Analysis*, 67:101575, 2023.
- [99] Yifan Peng, Yian Chen, E Miles Stoudenmire, and Yuehaw Khoo. Generative modeling via hierarchical tensor sketching. *arXiv preprint arXiv:2304.05305*, 2023.
- [100] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. *Advances in neural information processing systems*, 29, 2016.
- [101] Valentin Zauner-Stauber, Laurens Vanderstraeten, Matthew T Fishman, Frank Verstraete, and Jutho Haegeman. Variational optimization algorithms for uniform matrix product states. *Physical Review B*, 97(4):045145, 2018.
- [102] Sergey Dolgov and Dmitry Savostyanov. Parallel cross interpolation for high-precision calculation of high-dimensional integrals. *Computer Physics Communications*, 246:106869, 2020.

- [103] Ivan Oseledets and Eugene Tyrtyshnikov. Tt-cross approximation for multidimensional arrays. *Linear Algebra and its Applications*, 432(1):70–88, 2010.
- [104] Dmitry V Savostyanov. Quasioptimality of maximum-volume cross interpolation of tensors. *Linear Algebra and its Applications*, 458:217–244, 2014.
- [105] Yurriel Núñez Fernández, Matthieu Jeannin, Philipp T Dumitrescu, Thomas Kloss, Jason Kaye, Olivier Parcollet, and Xavier Waintal. Learning feynman diagrams with tensor trains. *Physical Review X*, 12(4):041018, 2022.
- [106] Suhan Shetty, Teguh Lembono, Tobias Loew, and Sylvain Calinon. Tensor train for global optimization problems in robotics. *The International Journal of Robotics Research*, 43(6):811–839, 2024.
- [107] Sebastian Keller, Michele Dolfi, Matthias Troyer, and Markus Reiher. An efficient matrix product operator representation of the quantum chemical hamiltonian. *The Journal of chemical physics*, 143(24), 2015.
- [108] Guifre Vidal, José Ignacio Latorre, Enrique Rico, and Alexei Kitaev. Entanglement in quantum critical phenomena. *Physical review letters*, 90(22):227902, 2003.
- [109] EM Stoudenmire and Steven R White. Minimally entangled typical thermal state algorithms. *New Journal of Physics*, 12(5):055026, 2010.
- [110] Hiroshi Shinaoka, Markus Wallerberger, Yuta Murakami, Kosuke Nogaki, Rihito Sakurai, Philipp Werner, and Anna Kauch. Multiscale space-time ansatz for correlation functions of quantum systems based on quantics tensor trains. *Physical Review X*, 13(2):021015, 2023.
- [111] Michael Lubasch, Pierre Moinier, and Dieter Jaksch. Multigrid renormalization. *Journal of Computational Physics*, 372:587–602, 2018.
- [112] Boris N Khoromskij. $O(d \log n)$ -quantics approximation of n -d tensors in high-dimensional numerical modeling. *Constructive Approximation*, 34(2):257–280, 2011.
- [113] Steven R White. Density matrix formulation for quantum renormalization groups. *Physical review letters*, 69(19):2863, 1992.
- [114] Steven R White. Density-matrix algorithms for quantum renormalization groups. *Physical review b*, 48(14):10345, 1993.
- [115] Ian P McCulloch. From density-matrix renormalization group to matrix product states. *Journal of Statistical Mechanics: Theory and Experiment*, 2007(10):P10014, 2007.
- [116] Alberto Baiardi and Markus Reiher. The density matrix renormalization group in chemistry and molecular physics: Recent developments and new challenges. *The Journal of Chemical Physics*, 152(4), 2020.

- [117] Weitang Li, Jiajun Ren, and Zhigang Shuai. Numerical assessment for accuracy and gpu acceleration of td-dmrg time evolution schemes. *The Journal of Chemical Physics*, 152(2), 2020.
- [118] Jiajun Ren, Weitang Li, Tong Jiang, Yuanheng Wang, and Zhigang Shuai. Time-dependent density matrix renormalization group method for quantum dynamics in complex systems. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 12(6):e1614, 2022.
- [119] C Karrasch, JH Bardarson, and JE Moore. Reducing the numerical effort of finite-temperature density matrix renormalization group calculations. *New Journal of Physics*, 15(8):083031, 2013.
- [120] Dey Dayasindhu, Maiti Debasmita, and Kumar Manoranjan. An efficient density matrix renormalization group algorithm for chains with periodic boundary condition. *Papers in physics*, 8(2):01–07, 2016.
- [121] Andrew John Daley, Corinna Kollath, Ulrich Schollwöck, and Guifré Vidal. Time-dependent density-matrix renormalization-group using adaptive effective hilbert spaces. *Journal of Statistical Mechanics: Theory and Experiment*, 2004(04):P04005, 2004.
- [122] Frank Verstraete, Diego Porras, and J Ignacio Cirac. Density matrix renormalization group and periodic boundary conditions: A quantum information perspective. *Physical review letters*, 93(22):227205, 2004.
- [123] Tao Xiang, Jizhong Lou, and Zhaobin Su. Two-dimensional algorithm of the density-matrix renormalization group. *Physical Review B*, 64(10):104414, 2001.
- [124] Örs Legeza and Jenő Sólyom. Quantum data compression, quantum information generation, and the density-matrix renormalization-group method. *Physical Review B—Condensed Matter and Materials Physics*, 70(20):205118, 2004.
- [125] Roman Orus and Guifre Vidal. Infinite time-evolving block decimation algorithm beyond unitary evolution. *Physical Review B—Condensed Matter and Materials Physics*, 78(15):155117, 2008.
- [126] Johannes Hauschild and Frank Pollmann. Efficient numerical simulations with tensor networks: Tensor network python (tenpy). *SciPost Physics Lecture Notes*, page 005, 2018.
- [127] Yohei Fuji, Frank Pollmann, and Masaki Oshikawa. Distinct trivial phases protected by a point-group symmetry in quantum spin chains. *Physical Review Letters*, 114(17):177204, 2015.
- [128] A Langari, F Pollmann, and M Siahatgar. Ground-state fidelity of the spin-1 heisenberg chain with single ion anisotropy: quantum renormalization group and exact diagonalization approaches. *Journal of Physics: Condensed Matter*, 25(40):406002, 2013.

- [129] Ian Affleck, Tom Kennedy, Elliott H Lieb, and Hal Tasaki. Rigorous results on valence-bond ground states in antiferromagnets. *Physical review letters*, 59(7):799, 1987.
- [130] Zheng-Xin Guo, Xue-Jia Yu, Xi-Dan Hu, and Zhi Li. Emergent phase transitions in a cluster ising model with dissipation. *Physical Review A*, 105(5):053311, 2022.
- [131] Lev Davidovich Landau et al. On the theory of phase transitions. *Zh. eksp. teor. Fiz*, 7(19-32):926, 1937.
- [132] Pimonpan Sompet, Sarah Hirthé, Dominik Bourgund, Thomas Chalopin, Julian Bibo, Joannis Koepsell, Petar Bojović, Ruben Verresen, Frank Pollmann, Guillaume Salomon, et al. Realizing the symmetry-protected haldane phase in fermi–hubbard ladders. *Nature*, 606(7914):484–488, 2022.
- [133] F Duncan M Haldane. Nonlinear field theory of large-spin heisenberg antiferromagnets: semiclassically quantized solitons of the one-dimensional easy-axis néel state. *Physical review letters*, 50(15):1153, 1983.
- [134] HQ Lin. Exact diagonalization of quantum-spin models. *Physical Review B*, 42(10):6561, 1990.
- [135] Reinhard M Noack and Salvatore R Manmana. Diagonalization-and numerical renormalization-group-based methods for interacting quantum systems. In *AIP Conference Proceedings*, volume 789, pages 93–163. American Institute of Physics, 2005.
- [136] Francesca Pietracaprina, Nicolas Macé, David J Luitz, and Fabien Alet. Shift-invert diagonalization of large many-body localizing spin chains. *SciPost Physics*, 5(5):045, 2018.
- [137] Ester Pagčs Fontanella. Exact diagonalization of the one-dimensional bose-hubbard model. -, 2022.
- [138] Alexander Wietek, Luke Staszewski, Martin Ulaga, Paul L Ebert, Hannes Karlsson, Siddhartha Sarkar, Henry Shackleton, Aritra Sinha, and Rafael D Soares. Xdiag: Exact diagonalization for quantum many-body systems. *arXiv preprint arXiv:2505.02901*, 2025.
- [139] Cornelius Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Journal of research of the National Bureau of Standards*, 45(4):255–282, 1950.
- [140] JB Parkinson and Jill C Bonner. Spin chains in a field: Crossover from quantum to classical behavior. *Physical Review B*, 32(7):4703, 1985.
- [141] Sanyee Tang and JE Hirsch. Long-range order without broken symmetry: Two-dimensional heisenberg antiferromagnet at zero temperature. *Physical Review B*, 39(7):4548, 1989.

- [142] Jung-Hoon Jung and Jae Dong Noh. Guide to exact diagonalization study of quantum thermalization. *Journal of the Korean Physical Society*, 76(8):670–683, 2020.
- [143] David Ceperley and Berni Alder. Quantum monte carlo. *Science*, 231(4738):555–560, 1986.
- [144] M Rossi, M Nava, L Reatto, and DE Galli. Exact ground state monte carlo method for bosons without importance sampling. *The Journal of chemical physics*, 131(15), 2009.
- [145] Gabriele Spada, Stefano Giorgini, and Sebastiano Pilati. Path-integral monte carlo worm algorithm for bose systems with periodic boundary conditions. *Condensed Matter*, 7(2):30, 2022.
- [146] Matthias Troyer and Uwe-Jens Wiese. Computational complexity and fundamental limitations to fermionic quantum monte carlo simulations. *Physical review letters*, 94(17):170201, 2005.
- [147] William MC Foulkes, Lubos Mitas, RJ Needs, and Guna Rajagopal. Quantum monte carlo simulations of solids. *Reviews of Modern Physics*, 73(1):33, 2001.
- [148] Shiwei Zhang, Joseph Carlson, and James E Gubernatis. Constrained path monte carlo method for fermion ground states. *Physical Review B*, 55(12):7464, 1997.
- [149] Shiwei Zhang and Henry Krakauer. Quantum monte carlo method using phase-free random walks with slater determinants. *Physical review letters*, 90(13):136401, 2003.
- [150] EY Loh Jr, JE Gubernatis, RT Scalettar, SR White, DJ Scalapino, and RL Sugar. Sign problem in the numerical simulation of many-electron systems. *Physical Review B*, 41(13):9301, 1990.
- [151] J Shumway and DM Ceperley. Path integral monte carlo simulations for fermion systems: Pairing in the electron-hole plasma. *Le Journal de Physique IV*, 10(PR5):Pr5–3, 2000.
- [152] Filiberto Ares, MA Rajabpour, and Jacopo Viti. Exact full counting statistics for the staggered magnetization and the domain walls in the xy spin chain. *Physical Review E*, 103(4):042107, 2021.
- [153] Ansgar Kalz and Gennady Y Chitov. Topological floating phase in a spatially anisotropic frustrated ising model. *Physical Review B—Condensed Matter and Materials Physics*, 88(1):014415, 2013.
- [154] Zheng-Cheng Gu and Xiao-Gang Wen. Tensor-entanglement-filtering renormalization approach and symmetry-protected topological order. *Phys. Rev. B*, 80:155131, Oct 2009.
- [155] Yu-Chin Tzeng, Hsiang-Hsuan Hung, Yung-Chung Chen, and Min-Fong Yang. Fidelity approach to gaussian transitions. *Phys. Rev. A*, 77:062321, Jun 2008.

- [156] Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21(6):1087–1092, 1953.
- [157] R. J. Elliott. Phenomenological discussion of magnetic ordering in the heavy rare-earth metals. *Phys. Rev.*, 124:346–353, Oct 1961.
- [158] Alphonse Finel and Didier de Fontaine. The two-dimensional annni model in the cvm approximation. *Journal of Statistical Physics*, 43(3):645–661, 1986.
- [159] Marcelo D. Grynberg and Horacio Ceva. Finite-size-scaling estimation of critical exponents in the two-dimensional axial next-nearest-neighbor ising model. *Phys. Rev. B*, 38:9172–9177, Nov 1988.
- [160] Michael E. Fisher and Walter Selke. Infinitely many commensurate phases in a simple ising model. *Phys. Rev. Lett.*, 44:1502–1505, Jun 1980.
- [161] N Bhattacharyya and S Dasgupta. Statistical mechanics of the 1d ferromagnetic annni chain under an external field: revisited. *Journal of Physics A: Mathematical and General*, 24(16):3927, aug 1991.
- [162] C. M. Arizmendi, A. H. Rizzo, L. N. Epele, and C. A. García Canal. Phase diagram of the annni model in the hamiltonian limit. *Zeitschrift für Physik B Condensed Matter*, 83(2):273–276, 1991.
- [163] P. Sen, S. Chakraborty, Subinay Dasgupta, and B. K. Chakrabarti. Numerical estimate of the phase diagram of finite annni chains in transverse field. *Zeitschrift für Physik B Condensed Matter*, 88(3):333–338, 1992.
- [164] OF de Alcantara Bonfim, B Boechat, and J Florencio. Quantum fidelity approach to the ground-state properties of the one-dimensional axial next-nearest-neighbor ising model in a transverse field. *Physical Review E*, 96(4):042140, 2017.
- [165] Matteo Beccaria, Massimo Campostrini, and Alessandra Feo. Density-matrix renormalization-group study of the disorder line in the quantum axial next-nearest-neighbor ising model. *Phys. Rev. B*, 73:052402, Feb 2006.
- [166] Matteo Beccaria, Massimo Campostrini, and Alessandra Feo. Evidence for a floating phase of the transverse annni model at high frustration. *Physical Review B*, 76(9):094410, 2007.
- [167] Adam Nagy. Exploring phase transitions by finite-entanglement scaling of mps in the 1d annni model. *New Journal of Physics*, 13(2):023015, 2011.
- [168] M Cea, M Grossi, S Monaco, E Rico, L Tagliacozzo, and S Vallecorsa. Exploring the phase diagram of the quantum one-dimensional annni model. *arXiv preprint arXiv:2402.11022*, 2024.
- [169] Saverio Monaco, Oriel Kiss, Antonio Mandarino, Sofia Vallecorsa, and Michele Grossi. Quantum phase detection generalization from marginal quantum neural network models. *Physical Review B*, 107(8):L081105, 2023.

- [170] F Khastehdel Fumani, S Nemati, and S MahdaviFar. Quantum critical lines in the ground state phase diagram of spin-1/2 frustrated transverse-field ising chains. *Annalen der Physik*, 533(2):2000384, 2021.
- [171] Pietro Smacchia, Luigi Amico, Paolo Facchi, Rosario Fazio, Giuseppe Florio, Saverio Pascazio, and Vlatko Vedral. Statistical mechanics of the cluster ising model. *Physical Review A—Atomic, Molecular, and Optical Physics*, 84(2):022304, 2011.
- [172] W. Son, L. Amico, R. Fazio, A. Hamma, S. Pascazio, and V. Vedral. Quantum phase transition between cluster and antiferromagnetic states. *Europhysics Letters*, 95(5):50001, aug 2011.
- [173] Sebastián Montes and Alioscia Hamma. Phase diagram and quench dynamics of the cluster- xy spin chain. *Phys. Rev. E*, 86:021101, Aug 2012.
- [174] S. M. Giampaolo and B. C. Hiesmayr. Topological and nematic ordered phases in many-body cluster-ising models. *Phys. Rev. A*, 92:012306, Jul 2015.
- [175] Wei Nie, Feng Mei, Luigi Amico, and Leong Chuan Kwek. Scaling of geometric phase versus band structure in cluster-ising models. *Physical Review E*, 96(2):020106, 2017.
- [176] D. Pérez-García, M. M. Wolf, M. Sanz, F. Verstraete, and J. I. Cirac. String order and symmetries in quantum spin lattices. *Phys. Rev. Lett.*, 100:167202, Apr 2008.
- [177] Stein Olav Skrøvseth and Stephen D. Bartlett. Phase transitions and localizable entanglement in cluster-state spin chains with ising couplings and local fields. *Phys. Rev. A*, 80:022316, Aug 2009.
- [178] Marcello Calvanese Strinati, Davide Rossini, Rosario Fazio, and Angelo Russomanno. Resilience of hidden order to symmetry-preserving disorder. *Physical Review B*, 96(21):214206, 2017.
- [179] Raúl Morral-Yepes, Frank Pollmann, and Izabella Lovas. Detecting and stabilizing measurement-induced symmetry-protected topological phases in generalized cluster models. *Phys. Rev. B*, 108:224304, Dec 2023.
- [180] Ruizhe Shen, Tianqi Chen, Bo Yang, Yin Zhong, and Ching Hua Lee. Robust simulations of many-body symmetry-protected topological phase transitions on a quantum processor. *arXiv preprint arXiv:2503.08776*, 2025.
- [181] Xue-Jia Yu, Sheng Yang, Shuo Liu, Hai-Qing Lin, and Shao-Kai Jian. Gapless symmetry-protected topological states in measurement-only circuits. *arXiv preprint arXiv:2501.03851*, 2025.
- [182] Sadaf F and V Subrahmanyam. Quantum correlations in a cluster spin model with three-spin interactions. *Journal of Physics A: Mathematical and Theoretical*, 58(13):135301, mar 2025.

- [183] Seunghun Lee and Eun-Gook Moon. Robust mixed-state cluster states and spurious topological entanglement negativity. *Phys. Rev. B*, pages –, Jul 2025.
- [184] A. Kay, D. K. K. Lee, J. K. Pachos, M. B. Plenio, M. E. Reuter, and E. Rico. Quantum information and triangular optical lattices. *Optics and Spectroscopy*, 99(3):339–356, 2005.
- [185] Yi-Xin Chen, Sheng-Wen Li, and Zhi Yin. Quantum correlations in a clusterlike system. *Phys. Rev. A*, 82:052320, Nov 2010.
- [186] Takumi Ohta, Shu Tanaka, Ippei Danshita, and Keisuke Totsuka. Phase diagram and sweep dynamics of a one-dimensional generalized cluster model. *Journal of the Physical Society of Japan*, 84(6):063001, 2015.
- [187] Jared Jeyaretnam, Jonas Richter, and Arijeet Pal. Quantum scars and bulk coherence in a symmetry-protected topological phase. *Physical Review B*, 104(1):014424, 2021.
- [188] Alberto D. Verga. Entanglement dynamics and phase transitions of the floquet cluster spin chain. *Phys. Rev. B*, 107:085116, Feb 2023.
- [189] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.
- [190] F. D. M. Haldane. Nonlinear field theory of large-spin heisenberg antiferromagnets: Semiclassically quantized solitons of the one-dimensional easy-axis néel state. *Phys. Rev. Lett.*, 50:1153–1156, Apr 1983.
- [191] FDM Haldane. Ground state properties of antiferromagnetic chains with unrestricted spin: Integer spin chains as realisations of the o (3) non-linear sigma model. *arXiv preprint arXiv:1612.00076*, 2016.
- [192] Shuichi Shiraishi and Kiyohide Nomura. Multicritical point of an $s=1$ xxz chain with single-ion anisotropy. *Physical Review B*, 111(2):024406, 2025.
- [193] A Fabricio Albuquerque, Chris J Hamer, and Jaan Oitmaa. Quantum phase diagram and excitations for the one-dimensional $s=1$ heisenberg antiferromagnet with single-ion anisotropy. *Physical Review B—Condensed Matter and Materials Physics*, 79(5):054412, 2009.
- [194] Frank Pollmann, Ari M. Turner, Erez Berg, and Masaki Oshikawa. Entanglement spectrum of a topological phase in one dimension. *Phys. Rev. B*, 81:064439, Feb 2010.
- [195] Shijie Hu, Bruce Normand, Xiaoqun Wang, and Lu Yu. Accurate determination of the gaussian transition in spin-1 chains with single-ion anisotropy. *Physical Review B—Condensed Matter and Materials Physics*, 84(22):220402, 2011.
- [196] Yu-Chin Tzeng, Hiroaki Onishi, Tsuyoshi Okubo, and Ying-Jer Kao. Quantum phase transitions driven by rhombic-type single-ion anisotropy in the $s=1$ haldane chain. *Phys. Rev. B*, 96:060404, Aug 2017.

- [197] Satoshi Ejima and Holger Fehske. Comparative density-matrix renormalization group study of symmetry-protected topological phases in spin-1 chain and bose-hubbard models. *Physical Review B*, 91(4):045121, 2015.
- [198] Wei Chen, Kazuo Hida, and BC Sanctuary. Ground-state phase diagram of $s=1$ xxz chains with uniaxial single-ion-type anisotropy. *Physical Review B*, 67(10):104401, 2003.
- [199] Tom Kennedy and Hal Tasaki. Hidden symmetry breaking and the haldane phase in $s=1$ quantum spin chains. *Communications in mathematical physics*, 147:431–484, 1992.
- [200] Tom Kennedy and Hal Tasaki. Hidden $z_2 \times z_2$ symmetry breaking in haldane-gap antiferromagnets. *Phys. Rev. B*, 45:304–307, Jan 1992.
- [201] P. Jelínek. Quantum spin chains go organic. *Nature Chemistry*, 15(1):12–13, 2023.
- [202] Jie Ren, Wen-Long You, and Andrzej M. Oleś. Quantum phase transitions in a spin-1 antiferromagnetic chain with long-range interactions and modulated single-ion anisotropy. *Phys. Rev. B*, 102:024425, Jul 2020.
- [203] X. G. Wen. Topological Orders in Rigid States. *International Journal of Modern Physics B*, 4(2):239–271, January 1990.
- [204] Jacob Jordan, Roman Orús, Guifre Vidal, Frank Verstraete, and J Ignacio Cirac. Classical simulation of infinite-size quantum lattice systems in two spatial dimensions. *Physical review letters*, 101(25):250602, 2008.
- [205] Jielun Chen, EM Stoudenmire, and Steven R White. Quantum fourier transform has small entanglement. *PRX Quantum*, 4(4):040318, 2023.
- [206] Juan José García-Ripoll. Quantum-inspired algorithms for multivariate analysis: from interpolation to partial differential equations. *Quantum*, 5:431, 2021.
- [207] Lambert Lin and Steven R White. A novel method of function extrapolation inspired by techniques in low-entangled many-body physics. *arXiv preprint arXiv:2308.09001*, 2023.
- [208] Rohit Dilip, Yu-Jie Liu, Adam Smith, and Frank Pollmann. Data compression for quantum machine learning. *Physical Review Research*, 4(4):043007, 2022.
- [209] Andrei Chertkov, Gleb Ryzhakov, Georgii Novikov, and Ivan Oseledets. Optimization of functions given in the tensor train format. *arXiv preprint arXiv:2209.14808*, 2022.
- [210] N. Metropolis and S Ulam. The monte carlo method. *Journal of the American Statistical Association*, 44(247):335–341, 1949.
- [211] Yoshua Bengio, Olivier Delalleau, Nicolas Le Roux, Jean-François Paiement, Pascal Vincent, and Marie Ouimet. Learning eigenfunctions links spectral embedding and kernel pca. *Neural Computation*, 16(10):2197–2219, 10 2004.

- [212] Benedikt Fauseweh. Quantum many-body simulations on digital quantum computers: State-of-the-art and future challenges. *Nature Communications*, 15(1):2123, 2024.
- [213] O. F. de Alcantara Bonfim, B. Boechat, and J. Florencio. Quantum fidelity approach to the ground-state properties of the one-dimensional axial next-nearest-neighbor ising model in a transverse field. *Phys. Rev. E*, 96:042140, Oct 2017.
- [214] Takumi Ohta, Shu Tanaka, Ippei Danshita, and Keisuke Totsuka. Topological and dynamical properties of a generalized cluster model in one dimension. *Physical Review B*, 93(16):165423, 2016.
- [215] Mehran Khosrojerdi, Jason L Pereira, Alessandro Cuccoli, and Leonardo Banchi. Learning to classify quantum phases of matter with a few measurements. *Quantum Science and Technology*, 10(2):025006, jan 2025.
- [216] Subir Sachdev. Quantum phase transitions. *Physics world*, 12(4):33, 1999.
- [217] Jean Zinn-Justin. *Phase transitions and renormalization group*. Oxford University Press, 2007.
- [218] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. Machine learning and the physical sciences. *Reviews of Modern Physics*, 91(4):045002, 2019.
- [219] Hsin-Yuan Huang, Richard Kueng, Giacomo Torlai, Victor V Albert, and John Preskill. Provably efficient machine learning for quantum many-body problems. *Science*, 377(6613):eabk3333, 2022.
- [220] Xiao-Yu Dong, Frank Pollmann, Xue-Feng Zhang, et al. Machine learning of quantum phase transitions. *Physical Review B*, 99(12):121104, 2019.
- [221] AV Uvarov, AS Kardashin, and Jacob D Biamonte. Machine learning phase transitions with a quantum processor. *Physical Review A*, 102(1):012415, 2020.
- [222] Juan Carrasquilla and Roger G Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, 2017.
- [223] Qingyu Li, Yuhuan Huang, Xiaokai Hou, Ying Li, Xiaoting Wang, and Abolfazl Bayat. Ensemble-learning error mitigation for variational quantum shallow-circuit classifiers. *Physical Review Research*, 6(1):013027, 2024.
- [224] Teresa Sancho-Lorente, Juan Román-Roche, and David Zueco. Quantum kernels to learn the phases of quantum matter. *Physical Review A*, 105(4):042432, 2022.
- [225] Yusen Wu, Bujiao Wu, Jingbo Wang, and Xiao Yuan. Quantum phase recognition via quantum kernel methods. *Quantum*, 7:981, 2023.
- [226] Mari Carmen Bañuls. Tensor network algorithms: A route map. *Annual Review of Condensed Matter Physics*, 14:173–191, 2023.

- [227] Ehud Altman, Kenneth R Brown, Giuseppe Carleo, Lincoln D Carr, Eugene Demler, Cheng Chin, Brian DeMarco, Sophia E Economou, Mark A Eriksson, Kai-Mei C Fu, et al. Quantum simulators: Architectures and opportunities. *PRX quantum*, 2(1):017003, 2021.
- [228] Leonardo Banchi, Jason Pereira, and Stefano Pirandola. Generalization in quantum machine learning: A quantum information standpoint. *PRX Quantum*, 2(4):040321, 2021.
- [229] Leonardo Banchi, Jason Luke Pereira, Sharu Theresa Jose, and Osvaldo Simeone. Statistical complexity of quantum learning. *Advanced Quantum Technologies*, page 2300311, 2024.
- [230] Hsin-Yuan Huang, John Preskill, and Mehdi Soleimanifar. Certifying almost all quantum states with few single-qubit measurements. *arXiv preprint arXiv:2404.07281*, 2024.
- [231] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermann Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022.
- [232] Marcus Cramer, Martin B Plenio, Steven T Flammia, Rolando Somma, David Gross, Stephen D Bartlett, Olivier Landon-Cardinal, David Poulin, and Yi-Kai Liu. Efficient quantum state tomography. *Nature communications*, 1(1):149, 2010.
- [233] Manuel S Rudolph, Jacob Miller, Danial Motlagh, Jing Chen, Atithi Acharya, and Alejandro Perdomo-Ortiz. Synergistic pretraining of parametrized quantum circuits via tensor networks. *Nature Communications*, 14(1):8367, 2023.
- [234] Paolo Zanardi and Nikola Paunković. Ground state overlap and quantum phase transitions. *Physical Review E*, 74(3):031123, 2006.
- [235] Lorenzo Campos Venuti and Paolo Zanardi. Quantum critical scaling of the geometric tensors. *Physical review letters*, 99(9):095701, 2007.
- [236] Leonardo Banchi, Paolo Giorda, and Paolo Zanardi. Quantum information-geometry of dissipative quantum phase transitions. *Physical Review E*, 89(2):022102, 2014.
- [237] Sukhwinder Singh, Robert NC Pfeifer, and Guifré Vidal. Tensor network decompositions in the presence of a global symmetry. *Physical Review A—Atomic, Molecular, and Optical Physics*, 82(5):050301, 2010.
- [238] Quynh T. Nguyen, Louis Schatzki, Paolo Braccia, Michael Ragone, Patrick J. Coles, Frédéric Sauvage, Martín Larocca, and M. Cerezo. Theory for equivariant quantum neural networks. *PRX Quantum*, 5:020328, May 2024.
- [239] F. Khastehdel Fumani, S. Nemati, and S. MahdaviFar. Quantum critical lines in the ground state phase diagram of spin-1/2 frustrated transverse-field ising chains. *Annalen der Physik*, 533(2):2000384, 2021.

- [240] Glen Bigan Mbeng, Angelo Russomanno, and Giuseppe E Santoro. The quantum ising chain for beginners. *SciPost Physics Lecture Notes*, page 082, 2024.
- [241] Anjan Kumar Chandra and Subinay Dasgupta. Floating phase in the one-dimensional transverse axial next-nearest-neighbor ising model. *Physical Review E*, 75(2):021105, 2007.
- [242] Sei Suzuki, Jun-ichi Inoue, and Bikas K. Chakrabarti. *Quantum Ising Phases and Transitions in Transverse Ising Models*, volume 862 of *Lecture Notes in Physics*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
- [243] Amit Dutta, Gabriel Aeppli, Bikas K Chakrabarti, Uma Divakaran, Thomas F Rosenbaum, and Diptiman Sen. *Quantum phase transitions in transverse field spin models: from statistical physics to quantum information*. Cambridge University Press, 2015.
- [244] Somayyeh Nemati, Fatemeh Khastehdel Fumani, and Saeed MahdaviFar. Comment on “quantum fidelity approach to the ground-state properties of the one-dimensional axial next-nearest-neighbor ising model in a transverse field”. *Physical Review E*, 102(1):016101, 2020.
- [245] Johnnie Gray. quimb: A python package for quantum information and many-body calculations. *Journal of Open Source Software*, 3(29):819, 2018.
- [246] Jens Eisert, Marcus Cramer, and Martin B Plenio. Colloquium: Area laws for the entanglement entropy. *Reviews of modern physics*, 82(1):277, 2010.
- [247] Akira Sone, Akira Tanji, and Naoki Yamamoto. Quantum inception score. *arXiv preprint arXiv:2311.12163*, 2024.
- [248] Mehran Khosrojerdi, Alessandro Cuccoli, Paola Verrucchi, and Leonardo Banchi. Unsupervised learning to recognize quantum phases of matter. *arXiv preprint arXiv:2510.14742*, 2025.
- [249] Marco Parigi, Mehran Khosrojerdi, Filippo Caruso, and Leonardo Banchi. Supervised quantum image processing. *arXiv preprint arXiv:2507.22039*, 2025.
- [250] Gael Sentís, Alex Monras, Ramon Muñoz-Tapia, John Calsamiglia, and Emilio Bagan. Unsupervised classification of quantum data. *Physical Review X*, 9(4):041029, 2019.
- [251] Yiwei Chen, Yu Pan, Guofeng Zhang, and Shuming Cheng. Detecting quantum entanglement with unsupervised learning. *Quantum Science and Technology*, 7(1):015005, 2021.
- [252] Yuan Yang, Zheng-Zhi Sun, Shi-Ju Ran, and Gang Su. Visualizing quantum phases and identifying quantum phase transitions by nonlinear dimensional reduction. *Physical Review B*, 103(7):075106, 2021.

- [253] Simone Tibaldi, Giuseppe Magnifico, Davide Vodola, and Elisa Ercolessi. Unsupervised and supervised learning of interacting topological phases from single-particle correlation functions. *SciPost Physics*, 14(1):005, 2023.
- [254] Andrew Ng, Michael Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 14, 2001.
- [255] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17:395–416, 2007.
- [256] Zhixin Zhou and Arash A.Amini. Analysis of spectral clustering algorithms for community detection: the general bipartite setting. *Journal of Machine Learning Research*, 20(47):1–47, 2019.
- [257] Shi-Ju Ran, Emanuele Tirrito, Cheng Peng, Xi Chen, Luca Tagliacozzo, Gang Su, and Maciej Lewenstein. *Tensor network contractions: methods and applications to quantum many-body systems*. Springer Nature, 2020.
- [258] Maria Schuld and Francesco Petruccione. *Machine learning with quantum computers*. Springer, 2021.
- [259] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H Booth, et al. The variational quantum eigensolver: a review of methods and best practices. *Physics Reports*, 986:1–128, 2022.
- [260] Harper R Grimsley, Sophia E Economou, Edwin Barnes, and Nicholas J Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature communications*, 10(1):3007, 2019.
- [261] Eytan Barouch, Barry M. McCoy, and Max Dresden. Statistical mechanics of the XY model. i. *Phys. Rev. A*, 2:1075–1092, Sep 1970.
- [262] Daniel E Parker, Xiangyu Cao, and Michael P Zaletel. Local matrix product operators: Canonical form, compression, and control theory. *Physical Review B*, 102(3):035147, 2020.
- [263] Purnima Bholowalia and Arvind Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.
- [264] Mehrnoosh Monshizadeh, Vikramajeet Khatri, Raimo Kantola, and Zheng Yan. A deep density based and self-determining clustering approach to label unknown traffic. *Journal of Network and Computer Applications*, 207:103513, 2022.
- [265] Sei Suzuki, Jun-ichi Inoue, and Bikas K Chakrabarti. *Quantum Ising phases and transitions in transverse Ising models*, volume 862. Springer, 2012.

- [266] P Sen, S Chakraborty, Subinay Dasgupta, and BK Chakrabarti. Numerical estimate of the phase diagram of finite annni chains in transverse field. *Zeitschrift für Physik B Condensed Matter*, 88(3):333–338, 1992.
- [267] Wonmin Son, Luigi Amico, Rosario Fazio, Alioscia Hamma, Saverio Pascazio, and Vlatko Vedral. Quantum phase transition between cluster and antiferromagnetic states. *Europhysics Letters*, 95(5):50001, 2011.
- [268] Johannes Herrmann, Sergi Masot Lima, Ants Remm, Petr Zapletal, Nathan A McMahon, Colin Scarato, François Swiadek, Christian Kraglund Andersen, Christoph Hellings, Sebastian Krinner, et al. Realizing quantum convolutional neural networks on a superconducting quantum processor to recognize quantum phases. *Nature communications*, 13(1):4144, 2022.
- [269] Yi Zhou, Kazushi Kanoda, and Tai-Kai Ng. Quantum spin liquid states. *Reviews of Modern Physics*, 89(2):025003, 2017.
- [270] Jinsheng Wen, Shun-Li Yu, Shiyang Li, Weiqiang Yu, and Jian-Xin Li. Experimental identification of quantum spin liquids. *npj Quantum Materials*, 4(1):12, 2019.
- [271] Kurt Binder and A Peter Young. Spin glasses: Experimental facts, theoretical concepts, and open questions. *Reviews of Modern physics*, 58(4):801, 1986.
- [272] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.
- [273] Paweł Cieśliński, Satoya Imai, Jan Dziwior, Otfried Gühne, Lukas Knips, Wiesław Laskowski, Jasmin Meinecke, Tomasz Paterek, and Tamás Vértesi. Analysing quantum systems with randomised measurements. *Physics Reports*, 1095:1–48, 2024.
- [274] Yu-Chin Tzeng and Min-Fong Yang. Scaling properties of fidelity in the spin-1 anisotropic model. *Physical Review A—Atomic, Molecular, and Optical Physics*, 77(1):012311, 2008.
- [275] Xue-Jia Yu and Wei-Lin Li. Quantum phase transition between topologically distinct quantum critical points. *arXiv preprint arXiv:2403.03716*, 2024.
- [276] Frank Pollmann and Ari M Turner. Detection of symmetry-protected topological phases in one dimension. *Physical Review B—Condensed Matter and Materials Physics*, 86(12):125441, 2012.
- [277] Jie Ren, Yimin Wang, and Wen-Long You. Quantum phase transitions in spin-1 xxz chains with rhombic single-ion anisotropy. *Physical Review A*, 97(4):042318, 2018.
- [278] David Pérez-García, Michael M Wolf, Mikel Sanz, Frank Verstraete, and J Ignacio Cirac. String order and symmetries in quantum spin lattices. *Physical review letters*, 100(16):167202, 2008.

- [279] Weicheng Ye, Shuwei Liu, Shiyu Zhou, and Yijian Zou. Universal quantum phase classification on quantum computers from machine learning. *arXiv preprint arXiv:2508.04774*, 2025.
- [280] Ruiyang Zhou, Saubhik Sarkar, Sougato Bose, and Abolfazl Bayat. Enhanced image classification via hybridizing quantum dynamics with classical neural networks. *arXiv preprint arXiv:2507.13587*, 2025.
- [281] Alex Mossi, Bojan Žunkovič, and Kyriakos Flouris. A matrix product state model for simultaneous classification and generation. *Quantum Machine Intelligence*, 7(1):48, 2025.
- [282] Nicolas Sadoune, Giuliano Giudici, Ke Liu, and Lode Pollet. Unsupervised interpretable learning of phases from many-qubit systems. *Phys. Rev. Res.*, 5:013082, Feb 2023.
- [283] Ming-Chiang Chung, Guang-Yu Huang, Ian P. McCulloch, and Yuan-Hong Tsai. Deep learning of phase transitions for quantum spin chains from correlation aspects. *Phys. Rev. B*, 107:214451, Jun 2023.
- [284] Michael Levin and Xiao-Gang Wen. Detecting topological order in a ground state wave function. *Physical review letters*, 96(11):110405, 2006.
- [285] Alexei Kitaev and John Preskill. Topological entanglement entropy. *Physical review letters*, 96(11):110404, 2006.
- [286] Jiasen Jin, Alberto Biella, Oscar Viyuela, Leonardo Mazza, Jonathan Keeling, Rosario Fazio, and Davide Rossini. Cluster mean-field approach to the steady-state phase diagram of dissipative spin systems. *Physical Review X*, 6(3):031011, 2016.
- [287] V Zauner, D Draxler, Laurens Vanderstraeten, Jutho Haegeman, and Frank Verstraete. Symmetry breaking and the geometry of reduced density matrices. *New Journal of Physics*, 18(11):113033, 2016.
- [288] Christopher L Henley and Hitesh J Changlani. Density-matrix based numerical methods for discovering order and correlations in interacting systems. *Journal of Statistical Mechanics: Theory and Experiment*, 2014(11):P11002, 2014.
- [289] Fei Yan, Abdullah M. Ilyasu, and Salvador E. Venegas-Andraca. A survey of quantum image representations. *Quantum Information Processing*, 15(1):1–35, Jan 2016.
- [290] Xi-Wei Yao, Hengyan Wang, Zeyang Liao, Ming-Cheng Chen, Jian Pan, Jun Li, Kechao Zhang, Xingcheng Lin, Zhehui Wang, Zhihuang Luo, Wenqiang Zheng, Jianzhong Li, Meisheng Zhao, Xinhua Peng, and Dieter Suter. Quantum image processing and its application to edge detection: Theory and experiment. *Phys. Rev. X*, 7:031041, Sep 2017.
- [291] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

- [292] Y. S. Weinstein, M. A. Pravia, E. M. Fortunato, S. Lloyd, and D. G. Cory. Implementation of the quantum fourier transform. *Phys. Rev. Lett.*, 86:1889–1891, Feb 2001.
- [293] Peter Hoyer. Efficient quantum transforms, 1997.
- [294] Amir Fijany and Colin P. Williams. Quantum wavelet transforms: Fast algorithms and complete circuits. In Colin P. Williams, editor, *Quantum Computing and Quantum Communications*, pages 10–33, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.
- [295] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, 2015.
- [296] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pages 1–5. Ieee, 2015.
- [297] Thomas J Elliott, Chengran Yang, Felix C Binder, Andrew JP Garner, Jayne Thompson, and Mile Gu. Extreme dimensionality reduction with quantum modeling. *Physical Review Letters*, 125(26):260501, 2020.
- [298] Leonardo Banchi. Accuracy vs memory advantage in the quantum simulation of stochastic processes. *Machine Learning: Science and Technology*, 5(2):025036, 2024.
- [299] Chengran Yang, Marta Florido-Llinàs, Mile Gu, and Thomas J Elliott. Dimension reduction in quantum sampling of stochastic processes. *npj Quantum Information*, 11(1):34, 2025.
- [300] Donovan Slabbert and Francesco Petruccione. Classical-quantum approach to image classification: Autoencoders and quantum svms. *AVS Quantum Science*, 7(2):023804, 06 2025.
- [301] Marina Lisnichenko and Stanislav Protasov. Quantum image representation: a review. *Quantum Machine Intelligence*, 5(1):2, Dec 2022.
- [302] Sreetama Das, Jingfu Zhang, Stefano Martina, Dieter Suter, and Filippo Caruso. Quantum pattern recognition on real quantum processing units. *Quantum Machine Intelligence*, 5(1):16, Apr 2023.
- [303] Quantum processing unit. <https://www.ibm.com/think/topics/qpu>. Accessed: 2025-10-14.
- [304] Alexander Geng, Ali Moghiseh, Claudia Redenbach, and Katja Schladitz. Quantum image processing on real superconducting and trapped-ion based quantum computers. *tm - Technisches Messen*, 90(7-8):445–454, 2023.
- [305] Phuc Q. Le, Fangyan Dong, and Kaoru Hirota. A flexible representation of quantum images for polynomial preparation, image compression, and processing operations. *Quantum Information Processing*, 10(1):63–84, Feb 2011.
- [306] Jose I. Latorre. Image compression and entanglement, 2005.

- [307] Yi Zhang, Kai Lu, Yinghui Gao, and Mo Wang. Neqr: a novel enhanced quantum representation of digital images. *Quantum Information Processing*, 12(8):2833–2860, Aug 2013.
- [308] Phuc Le, Abdullah Ilyasu, Fangyan Dong, and Kaoru Hirota. Efficient color transformations on quantum images. *Journal of Advanced Computational Intelligence and Intelligent Informatics*, 15:698–706, 05 2011.
- [309] Phuc Le, Abdullah Ilyasu, Dong Fangyan, and Hirota Kaoru. Fast geometric transformations on quantum images. *IAENG International Journal of Applied Mathematics*, 40, 08 2010.
- [310] Büsra Kösoğlu-Kind, Robert Loredó, Michele Grossi, Christian Bernecker, Jody M Burks, and Rüdiger Buchkremer. A biological sequence comparison algorithm using quantum computers. *Scientific reports*, 13(1):14552, 2023.
- [311] Norhan Nasr, Ahmed Younes, and Ashraf Elsayed. Efficient representations of digital images on quantum computers. *Multimedia Tools and Applications*, 80(25):34019–34034, Oct 2021.
- [312] Sahel Ashhab. Quantum state preparation protocol for encoding classical data into the amplitudes of a quantum information processing register’s wave function. *Phys. Rev. Res.*, 4:013091, Feb 2022.
- [313] 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, September 2005.
- [314] Konstantinos Koutroumbas and Sergios Theodoridis. *Pattern recognition*. Academic Press, 2008.
- [315] Mehryar Mohri. Foundations of machine learning, 2018.
- [316] Corinna Cortes. Support-vector networks. *Machine Learning*, 1995.
- [317] Alexander S Holevo. *Probabilistic and statistical aspects of quantum theory*, volume 1. Springer Science & Business Media, 2011.
- [318] PennyLane. <https://pennylane.ai>. Accessed: 2025-10-14.
- [319] Scikit-learn. <https://scikit-learn.org>. Accessed: 2025-10-14.
- [320] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.
- [321] Opencv. <https://docs.opencv.org>. Accessed: 2025-10-14.
- [322] Johnnie Gray and Stefanos Kourtis. Hyper-optimized tensor network contraction. *Quantum*, 5:410, 2021.



Finanziato
dall'Unione europea
NextGenerationEU



Ministero
dell'Università
e della Ricerca



Italiadomani
PIANO NAZIONALE
DI RIPRESA E RESILIENZA



UNIVERSITÀ
DEGLI STUDI
FIRENZE

La borsa di dottorato è stata finanziata con risorse dell'Unione europea-*NextGeneration EU*
Piano Nazionale di Ripresa e Resilienza Missione 4 – Componente 2 – Investimento 1.3 –
CUP B83C22004940006