PhD-FSTM-2025-081

The Faculty of Science, Technology and Medicine

DISSERTATION

Defence held on 08/07/2025 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG
EN INFORMATIQUE

by

Yangjie XU

Born on 18 June 1994 in Jiangsu (China)

Quantum Machine Learning: Diverse Perspectives on Application Scenarios

Dissertation Defence Committee:

Dr. Radu STATE, Dissertation Supervisor
Professor, SnT, University of Luxembourg

Dr. Raphaël FRANK, Chairman
Assistant Professor, SnT, University of Luxembourg

Dr. Hui HUANG
Research Scientist, SnT, University of Luxembourg

Dr. Rémi BADONNEL
Professor, TELECOM Nancy Engineering School, University of Lorraine, France

Dr. Kai LI
Senior Research Scientist, CISTER Research Unit at ISEP/IPP, Lisbon, Portugal

Affidavit

I hereby confirm that the PhD thesis entitled “Quantum Machine Learning: Diverse Perspectives on Application Scenarios ” has been written independently and without any other sources than cited.

Luxembourg, ____________________ ______________________________

Yangjie XU

Abstract

In recent years, quantum computing, as a potential way to break the bottleneck of Moore's Law, has gradually shown its unique advantages in the field of machine learning. Not only can it bring exponential or polynomial acceleration in some specific tasks, it also provides entirely new ways of representing and processing information. Focusing on the core theme of application and method innovation of quantum machine learning in structured data, this thesis systematically studies the theoretical models and practical applications of quantum computing in many sub-fields such as agricultural remote sensing, graph neural networks, and hyperdimensional computing, and explores the adaptability, scalability, and effectiveness of quantum models in real-world tasks.

Firstly, in the cropland classification and yield prediction task, this study identifies that traditional neural networks tend to underperform when facing imbalanced and small-sample remote sensing data, often misclassifying minority crop types. To address this, we propose a quantum-classical hybrid recognition framework that integrates quantum feature mapping with classical neural networks. After incorporating quantum feature encoding, we observe a significant improvement in classification performance, especially in cases involving subtle field boundaries and heterogeneous textures. The quantum-enhanced model demonstrates superior accuracy, robustness, and generalization compared to its classical counterpart, validating the advantage of leveraging quantum representations in real-world agricultural datasets.

The second part focuses on the learning of graph structure data and proposes to embed the Continuous Time Quantum Walk (CTQW) into the Graph Neural Network (GNN) architecture. This method not only preserves the global interference characteristic of quantum states but also realizes the efficient modeling of graph topology through the spectral graph methods. Experiments show that the proposed method is superior to many classical graph learning models in tasks such as node classification and graph structure inference.

The third and fourth components of this thesis collectively explore quantum approaches to hyperdimensional computing (HDC), establishing a progressive transition from native quantum-state-based implementation to quantum-enhanced HDC strategies. Initially, we investigate the intrinsic mathematical correspondence between quantum states and high-dimensional hypervectors, and propose a novel quantum-state-based high-dimensional encoding method. This method is realized through quantum circuits capable of generating hypervectors with controllable entropy, thereby implementing essential HDC operations, such as classification and memory reconstruction, on real quantum hardware. This marks the first successful migration of core HDC mechanisms into a fully quantum-native framework.

Building upon this foundation, we further propose the first quantum-enhanced HDC strategy, which does not rely solely on quantum-native implementations but rather utilizes key quantum features to boost the efficiency and expressivity of conventional HDC systems. Specifically, we introduce mechanisms such as quantum-reference-state-based binding, using density matrices to represent superclass states, and quantum state tomography for robust feature extraction. These innovations significantly enhance the resilience and performance of HDC under high-noise and high-dimensional sensing environments. The resulting models achieve superior accuracy and robustness across a variety of multi-class classification tasks, demonstrating the practical benefits of integrating quantum computation into the broader HDC paradigm.

In this thesis, experimental validation is carried out in multiple real and simulation environments, including Qiskit and Paddle Quantum platforms, and a complete closed loop was achieved from theoretical derivation, quantum circuit design to end-to-end experimental evaluation. The results show that quantum methods can expand the expressive power of classical models and become an important tool to deal with large-scale, structured data problems in the future. The work in this thesis provides methodological support for quantum machine learning from theory to application, and also expands the research boundaries of high-dimensional computation and graph learning in the quantum background.

Dedicated to my beloved wife, parents, and friends.

Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor, Prof. Radu State, for providing unlimited support for my work. My co-author Dr. Hui Huang has discussed with us countless times. Our thoughts collided and gave me a lot of inspiration. My CET Member Prof. FRANK Raphael gave me a lot of very valuable suggestions during the CET process.

Secondly, I want to thank my parents. In a nutshell, without you all, there would be no me.

Next, I would like to thank my lover, Ying Shen. From being my girlfriend to my wife, she has given me unlimited strength. Finally, I would like to thank my friends. Zhaoxiang Shen, Tong Wei, Xinlin Wang, Yidi Yang, Xin Wang, Lujun Li, and so on. The ranking is not in any particular order.

Stay Hungry, Stay Foolish.

Steve Jobs

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Motivation

1.1.1 General Motivation

With the rapid growth of data scale and the increasing complexity of problem structures, machine learning has become the core technology for understanding the world, assisting decision making, and intelligent reasoning [1–4]. However, traditional machine learning methods rely on classical computing resources and gradually expose their inherent limitations in typical challenge scenarios such as high-dimensional, structured, and small samples. In particular, the performance of traditional models has reached the limit in terms of the tradeoff between expressive power, modeling efficiency and generalization ability.

At the same time, quantum computing [5], as a new computational model based on the principles of quantum mechanics, is providing breakthrough possibilities for machine learning. The superposition, entanglement, and interference of qubits have led to state space expressiveness far beyond that of classical models, and have inspired an emerging cross-cutting research field called Quantum Machine Learning (QML) [6, 7].

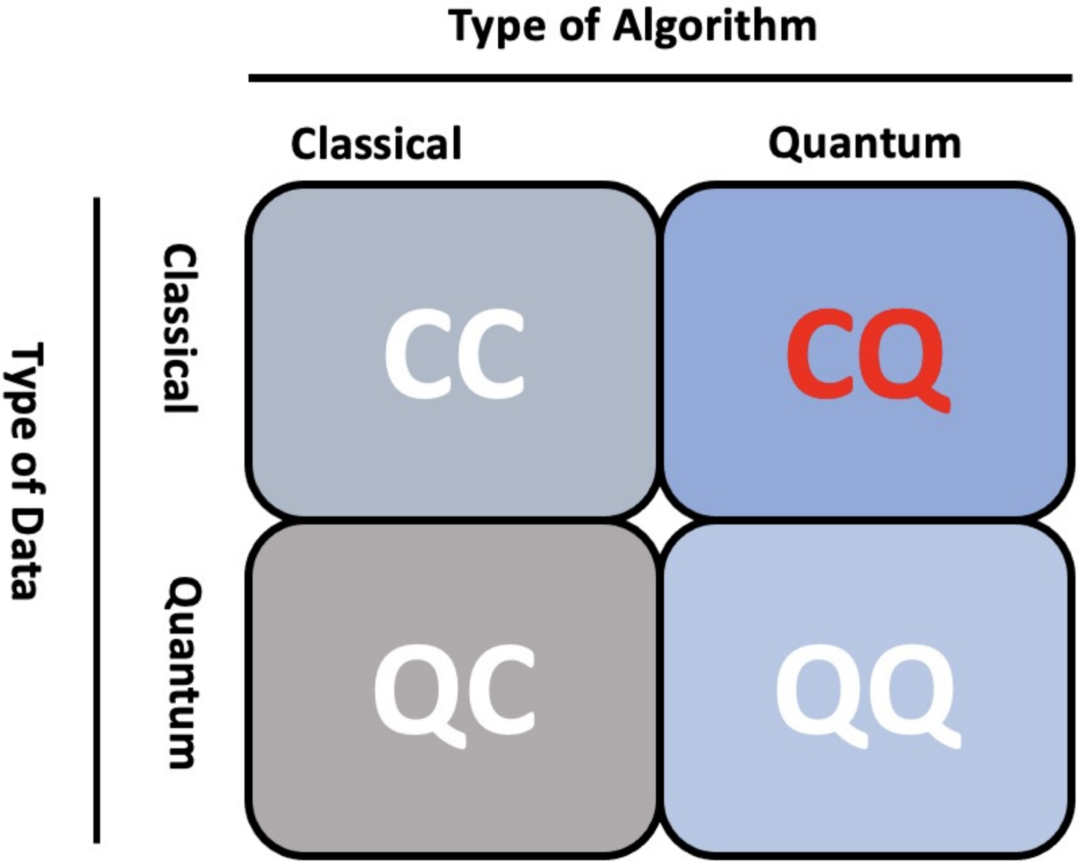


Figure 1.1: Taxonomy of learning paradigms based on data and algorithm types

Figure 1.1 shows a classification method of computing, Despite the great success of traditional machine learning over the past decade or so, its modeling mechanisms are mainly based on real-valued vector spaces and deterministic optimization paths. Such models face bottlenecks in the following areas, which are the key to quantum models:

- **Limited Respresentational Space:** Classical models operate in linerly or polynomially expanding feature space. In contrast, a quantum system with n qubits inhabits a complex Hilbert Space [8] of dimension 2^n, offering exponential representational capacity suitable for encoding high-order statistical dependencies and abstract patterns.

- **Restricted Feature Mapping:** Classical models often rely on fixed nuclei or linear projections. Quantum models can encode inputs through parameterized quantum feature maps to achieve nonlinear and structure-aware quantum state space transformations.

- **Model Efficiency and Parameters Scalability:** Classical deep learning models typically involve millions of parameters and require large labeled datasets to train. Such a high number of parameters not only requires significant computational resources, but also increases the risk of overfitting and instability in resource-poor environments. In contrast, quantum models (especially those based on variable component subcircuits) can express rich classes of functions using compact, parametric efficient architectures. Their high-dimensional state coding enables efficient learning with fewer trainable parameters and less data.

- **Compact Representations and Latent Expressivity:** Classical models explicitly represent features in large vector Spaces, often requiring deep layers to capture complex dependencies. Quantum model utilization of the exponentially large Hilbert space of an n-qubit system allows complex correlations to be embedded in a compact underlying state, potentially reducing the depth and training overhead of the model while maintaining high expressiveness.

Therefore, we attempt to use the algorithm of quantum computing to enhance the method of machine learning. Figure 1.1 shows a classification method of computing paradigms based on data types and algorithm types. The horizontal axis represents the type of algorithm adopted, which can be a classical algorithm or a quantum algorithm; The vertical axis represents the type of input data, which can be classical data or quantum data. Thus, four combination methods are formed: CC (classical data + classical algorithm), CQ (classical data + quantum algorithm), QC (quantum data + classical algorithm), and QQ (quantum data + quantum algorithm). At present, quantum data is not widespread, and real-world data still remains at traditional levels such as text, images, and videos. We want to use quantum algorithms to accelerate the processing of massive amounts of data. Therefore, the position where this work is located is the CQ region, that is, quantum algorithms are introduced on the basis of classical data for processing. This paradigm has significant practical significance. On the one hand, it retains the existing basic framework of data processing. On the other hand, it enhances the expressive ability and computational efficiency through Quantum mechanisms, and is particularly suitable for the current NISQ (Noisy Intermediate-Scale Quantum) era with limited quantum hardware resources.

1.1.2 Problem-Driven Motivation

Starting from four specific and typical practical problems, this paper deeply analyzes the inherent bottlenecks of traditional machine learning methods in these problems, and clearly puts forward the specific advantages and breakthrough points of quantum machine learning methods.

Quantum Machine Learning for Cropland classification and Yield Prediction

With the development of remote sensing technology and the improvement of data collection capabilities, the remote sensing data involved in the tasks of farmland classification and yield prediction continue to increase in dimension and complexity. This high-dimensional and multimodal data structure poses higher requirements for the feature expression ability of the model. However, classical machine learning methods (especially deep learning) performed well when the amount of data grew initially. But as the dimensions increased and the data structure became more complex, their expressive ability in the feature space was gradually limited, making it difficult for the model to further mine potential information. Furthermore, in practical applications of remote sensing data, there is often a significant class imbalance phenomenon. Traditional models are prone to prediction bias when dealing with a few classes, making it difficult to ensure the stability and generalization ability of the overall classification performance.

In response to the above problems, this study proposes to map remote sensing data into the exponentially dimensional quantum Hilbert space and utilize the quantum feature mapping method to mine deeper higher-order structure information. By means of the interference and entanglement mechanism of quantum states, the model can model complex distributions more effectively in high-dimensional Spaces and achieve more sensitive expression and discrimination capabilities for imbalanced samples, thereby breaking through the performance bottleneck of classical models in high-dimensional and imbalanced data scenarios.

Quantum Walk on Graph-structured Data

Graph structure data is widely used in social networks, biological networks, knowledge graph, traffic networks and other practical applications [9, 10]. Although the classical Graph Neural Network (GNN) has been widely used, there are still a series of deep problems and bottlenecks:

- The information propagation mechanism of classical GNN is highly dependent on the aggregation of local neighborhoods, and it is difficult to capture the global dependence and long-range interaction in the graph structure.
- As the network structure becomes deeper, the feature of classical GNN tends to be too smooth, which reduces the feature differentiation between different nodes.
- When the scale of graph structure increases, the number of parameters and computational complexity of the classical model increase rapidly, and the efficiency of the model decreases significantly.

In order to solve the above problems, the Continuous Time Quantum Walk (CTQW) mechanism is introduced in this study. By utilizing the essential features of quantum superposition and interference, CTQW naturally has a globally efficient information transmission mechanism, and can simultaneously consider the long-term correlation between multiple nodes. CTQW has a stronger global representation ability, and can capture richer and higher-order graph topological structure information with fewer parameters and lower model complexity, effectively overcoming the above bottleneck problems of classical models.

Quantum Learning for HDC

Classical Hyperdimensional Computing (HDC) realizes information representation and cognitive operation by using high-dimensional random vectors, which has good fault tolerance, robustness and small sample learning characteristics. However, as the complexity of practical tasks increases, classic HDC faces obvious bottlenecks and limitations:

- The expression ability of classical random vectors is limited: traditional methods rely on classical pseudorandom number generators to generate high-dimensional vectors, and the richness of vector randomness and entropy is limited, thus limiting the ability of models to distinguish fine-grained features.
- High computational complexity and low efficiency: When the classic HDC combines bind and superclass operations, the computational complexity of vector space

increases rapidly with the increase of dimensions, which seriously affects the deployment efficiency of the model.

- Feature representation ability: It is difficult for classical linear combination methods to capture high-order nonlinear associations in data, resulting in insufficient classification generalization ability, especially in task scenarios with high noise interference or class similarity.

To overcome the above bottleneck, this study proposes a Quantum-enhanced HDC framework, which includes the following two key steps. Firstly, by introducing quantum state as the encoding carrier of high-dimensional hypervector, information representation is realized by using the natural exponential dimensional advantage of quantum system. This coding mechanism based on quantum states provides far more randomness and information entropy than classical random vectors, and fundamentally breaks the bottleneck of classical HDC model in expressing complex information modes. Secondly, quantum mechanics is further used to realize bind operation and superclass representation in classical HDC, which significantly improves the computational efficiency and expression ability of the model in high-dimensional vector space. Specifically, through quantum gate operation and quantum superposition mechanism, binding operation in classical HDC is realized in a more compact and efficient way, making the representation of classes more differentiated and robust.

The quantum-enhanced HDC framework proposed in this study has not only been systematically implemented and verified in classical simulation environments (such as IBM Qiskit's Aer Simulator), but also successfully deployed in real Quantum computing devices (IBM Quantum Processing Unit (QPU)). The feasibility and application potential of the method are further verified.

Therefore, the motivation of this study is to use quantum state space and quantum mechanism to break through the bottlenecks of classical HDC in high-dimensional representation, randomness and computational efficiency, and significantly improve the generalization performance and practical application ability of HDC model in complex classification tasks through more efficient and expressive quantum enhancement methods.

1.2 Research Objectives

In view of the bottleneck problem of traditional machine learning model proposed in the above research motivation, and the potential advantages of quantum machine learning in the field of remote sensing classification of farmland, graph structure learning and ultra-high dimensional computing, this doctoral thesis aims to achieve the following specific research objectives.

1.2.1 Cropland Classification and Yield Prediction Method Based on Quantum Feature Mapping

Aiming at the bottleneck that the performance of traditional remote sensing models tends to be saturated with the increase of data volume and unbalanced data, the objectives of this research are as follows:

- To design a parametric quantum feature mapping method for remote sensing data.
- To construct a quantum-classical hybrid classification and regression framework to explore the ability of quantum feature space to express complex features of remote sensing data.
- To verify the performance improvement effect of quantum feature mapping in remote sensing small sample data scenario, and conduct a quantitative comparative analysis with traditional deep learning models.

1.2.2 A Graph Learning Model Based on Continuous-time Quantum Random Walk

This study aims to explore how to introduce the Continuous QuantumWalk (CTQW) mechanism into graph representation learning to enhance the model's ability to express graph structure information and its reasoning performance. Although the classical Graph Neural Network (GNN) has made significant progress in the node classification task, there are still problems such as limited expressive ability, excessive smoothing and insufficient local aggregation of information. This study attempts to introduce quantum random walk dynamics to capture richer global relationships and interference patterns in graph structures and break through the limitations of existing methods.

The specific objectives of this work are as follows:

- To construct a continuous-time quantum walk operator on graph structures by leveraging the graph Laplacian as the Hamiltonian governing quantum evolution.
- To analyze the spectral and dynamical properties of CTQW on various types of graphs, and compare their expressive power and convergence behavior with classical diffusion processes and discrete quantum walks.

- To evaluate the proposed model on various graph learning tasks, such as node classification, using both synthetic and real-world datasets, and compare its performance with state-of-the-art GNN baselines.

1.2.3 High Dimensional information coding and computing framework based on quantum states

Aiming at the problems such as limited expression ability and insufficient computational efficiency of the classical HDC model. To overcome the bottlenecks in randomness, expressiveness and computational efficiency of classic HDC, the objectives of this study are as follows:

- To propose, for the first time, a theoretical framework and method for directly encoding classical hypervector information into quantum state high-dimensional space.
- To explore the efficient representation mechanism of higher-order patterns and nonlinear correlations within the quantum state Hilbert space.
- To implement the quantum-based hyperdimensional coding method in a simulation environment and systematically analyze its randomness, entropy, and information representation capacity.
- To design an efficient supervector generation method based on quantum state space and quantum mechanisms for realizing binding operations and superclass representation in classical HDC.
- To implement the proposed method end-to-end on both the IBM Quantum Aer Simulator and real quantum processing units (IBM QPU).
- To compare the quantum-enhanced HDC method with its classical counterpart and verify its advantages in classification accuracy and computational efficiency.

1.3 Contributions

This thesis proposes a series innovative theoretical and methodological contributions around the practical applications of quantum machine learning in cropland remote sensing classification and yield prediction, graph structure data learning, and hyperdimensional computing tasks. The specific contributions are as follows.

1.3.1 Cropland Classification and Yield Classification

Aiming at the bottleneck that the traditional remote sensing machine learning tends to be saturated with more data and more complex models, this paper proposes and implements a framework for remote sensing cropland classification and yield prediction based on quantum feature mapping.

- **Theoretical contributions:** This paper systematically analyzes the features representation bottleneck of traditional deep learning models in remote sensing classification tasks. The mathematical model and theoretical basis of the quantum feature mapping method are proposed, and the mathematical properties of nonlinear feature mapping in quantum Hilbert space and the potential advantages of high-dimensional feature representations are clarified.

- **Method contributions:** A parametric quantum feature mapping circuit structure for remote sensing data is designed for the first time, and a concrete implementation scheme is given. A remote sensing classification method based on variable component subclassifier (VQC) is proposed, and a complete process of classical and quantum hybrid remote sensing cropland classification and yield prediction framework is realized.

- **Experimental contributions:** The proposed classification method of quantum feature mapping is first implemented and strictly verified on real remote sensing data. We conduct systematic experiments on field classification (classification task) and yield prediction (regression task). Through accurate comparison with classical deep learning models (such as CNN), the results clearly show that under certain data conditions, the quantum-classical hybrid method is significantly better than the traditional method in performance, which effectively validates the practical feasibility and application value of the theoretical hypothesis.

1.3.2 CTQW-GraphSAGE:

A new graph structure learning model incorporating continuous time quantum random walks is proposed and implemented. Aiming at the bottleneck of GraphSAGE in capturing long-range dependency and global topology information, the specific contributions of this part include:

- **Theoretical contributions:** The validity of continuous time quantum random walk in graph structure data learning is explicitly proposed and theoretically analyzed. The mathematical basis and potential advantages of quantum random walk in graph topology representations are systematically explained. The essential advantage of quantum random walk in capturing topological information in graphs is deeply analyzed, and the theoretical feasibility of introducing quantum mechanism into graph learning is further illustrated.

- **Method contributions:** A new graph structure learning model combining quantum random walk mechanism and classical GNN information propagation mechanism for the first time, and a clear quantum-classical hybrid architecture is designed.

- **Experimental contributions:** The proposed quantum graph learning method is implemented and verified on graph datasets. Compared with the traditional GNN models, it is proved that the CTQW mechanism can improve the performance of the node classification task.

1.3.3 Quantum State based Hyperdimensional Computing

In order to break the bottleneck of classical hyperdimension computing (HDC) in randomness, entropy expression and highdimensional feature coding, this study makes the following original contributions:

- **Theoretical contributions:** A new theory using the natural highdimensional space of quantum state itself as a highdimensional vector representation is proposed for the first time, and the mathematical characteristics and advantages of the highdimensional feature representation of quantum Hilbert space are clarified. The randomness of quantum state space and its theoretical advantages in representing complex patterns and nonlinear correlation are analyzed and defined for the first time.

- **Method contributions:** A method of constructing supervector based on quantum state space is proposed, and a mapping strategy of super-high-dimensional classical data vector to quantum state space is given. An efficient quantum state representation mechanism is designed to make information encoding more efficient and expressive, and break through the limitations of classical random vector generation.

- **Experimental contributions:** The experimental results show that the quantum state coding method has significant advantages over the classical HDC in randomness and nonlinear mode representation, which provides a preliminary experimental basis for the quantum implementation of hyperdimensional computing.

1.3.4 Quantum-enhanced Hyperdimensional Computing

- **Theoretical contributions:** The theoretical hypothesis and basic framework of classical HDC core operation by quantum mechanism are proposed for the first time, and the mathematical principle and potential advantages of binding and superclass construction based on quantum operation are clearly explained. The theoretical feasibility and application prospect of quantum mechanism in improving the expression capability of HDC and reducing the computational complexity are analyzed.

- **Method contributions:** A set of binding and bundling methods based on quantum circuits is designed and implemented for the first time. A clear and efficient quantum enhancement strategy is proposed, which significantly improves the comprehensive performance of the classical HDC model in terms of expression performance and computational efficiency.

- **Experimental contributions:** The proposed quantum-enhanced HDC framework is fully implemented on IBM Qiskit Aer simulation platform and real quantum processing unit (QPU) for the first time. Through systematic comparison experiments, the proposed method has significant advantages over classical HDC in classification accuracy, computational complexity and generalization ability, and further proves its application potential in practical tasks.

Paper Publication and Under Review

Published and Accepted

This dissertation contains the following papers published as the first author. The appearances of these papers are as follows:

- **Yangjie Xu**, Hui Huang, and Radu State. *Cropland Quantum Learning: A Hybrid Quantum-Classical Neural Network for Cropland Classification* published in 2024 IEEE 3rd International Conference on Computing and Machine Intelligence is included in Chapter 3

- **Yangjie Xu**, Hui Huang, and Radu State. *CTQW-GraphSAGE: Trainabel Continuous-Time Quantum Walk On Graph* published in ICANN 2024 - 33rd International Conference on Artificial Neural Networks, Proceedings is included in Chapter 5

- **Yangjie Xu**, Hui Huang, and Radu State. *Quantum-StateHD: Leveraging Random Quantum States for Hyperdimensional Learning* accepted by The 35rd International Joint Conference on Neural Network is includes in Chapter 6

Under Review

- **Yangjie Xu**, Hui Huang, and Radu State. *A Quantum-Classical Hybrid Feature Mapping for Yield Prediction* submitted to 32nd International Conference on Neural Information Processing is included in Chapter 4

- **Yangjie Xu**, Hui Huang, and Radu State. *QeHDC: Hyperdimensional Computing based on Quantum-enhanced binding and SuperClass Construction* being submitting to The 40th Annual AAAI Conference on Artificial Intelligence is included in Chapter 7

1.4 Dissertation Organization

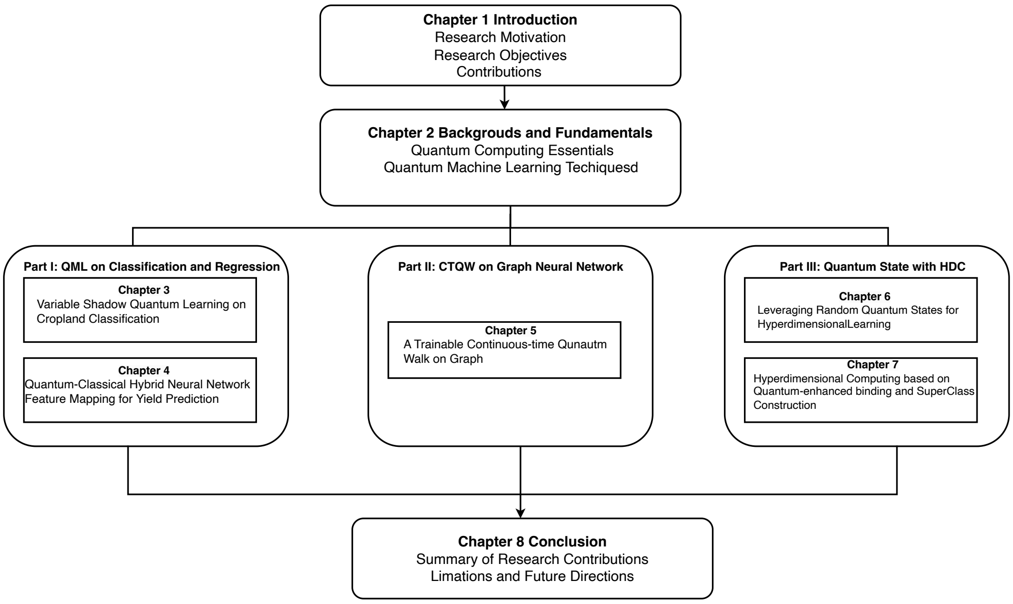

Chapter 1 Introduction
Research Motivation
Research Objectives
Contributions
Chapter 2 Backgrouds and Fundamentals
Quantum Computing Essentials
Quantum Machine Learning Techiquesd
Part I: QML on Classification and Regression
Chapter 3
Variable Shadow Quantum Learning on Cropland Classification
Chapter 4
Quantum-Classical Hybrid Neural Network Feature Mapping for Yield Prediction
Part II: CTQW on Graph Neural Network
Chapter 5
A Trainable Continuous-time Qunautm Walk on Graph
Part III: Quantum State with HDC
Chapter 6
Leveraging Random Quantum States for HyperdimensionalLearning
Chapter 7
Hyperdimensional Computing based on Quantum-enhanced binding and SuperClass Construction
Chapter 8 Conclusion
Summary of Research Contributions
Limations and Future Directions

Figure 1.2: Organization of This Dissertation

In order to express the contribution clearly and completely, the structure of the dissertation is shown in the Figure 1.2 the thesis is organized as follows:

Part I: Background and Fundamentals

- **Chapter 2** introduces the fundamental concepts and theoretical backgrounds required for the study, covering key quantum computing principles such as qubits, superposition, entanglement, quantum measurements, and quantum circuits, along with state-of-the-art quantum machine learning techniques including quantum feature mappings, variational quantum algorithms, and quantum random walks.

Part II: Quantum Machine Learning on Cropland Classification and Yield Prediction

- **Chapter 3** proposes using Variable Shadow Quantum Learning method specifically tailored for cropland classification tasks. This chapter thoroughly demonstrates how quantum feature mapping can effectively project classical remote sensing data into higher-dimensional quantum Hilbert spaces, thereby overcoming the intrinsic representational limitations of classical deep learning models. Extensive experiments on real-world remote sensing datasets confirm the superiority of the proposed quantum method compared to traditional approaches.
- Building on the insights from classification, Chapter 4 further advances toward quantum-enhanced yield prediction—a regression task requiring richer temporal and spectral data modeling. This transition from classification to regression reflects not only an increasing task complexity but also a broader generalization capability of quantum models. Experiments on progressively complex datasets validate the scalability and robustness of the proposed quantum approaches across different agricultural scenarios.

Part III: Continuous-time Quanutm Random Walk on Graph

- **Chapter 4** presents an innovative graph learning framework combining Continuous-Time Quantum Walks (CTQW) with classical Graph Neural Networks (GNNs). This chapter elaborates on how quantum random walks can efficiently encode graph topology and dependencies, addressing critical limitations such as over-smoothing issues encountered in classical GNNs. Comprehensive experiments on benchmark graph datasets highlight the superior performance of the proposed quantum-enhanced approach for node classification tasks.

Part IV: Quantum and Quantum-enhanced Hyperdimensional Computing

- **Chapter 5** further advances the thesis by introducing a Quantum State-based Hyperdimensional Computing framework. Here, quantum states are proposed as native high-dimensional vector encodings, significantly enhancing the expressivity, randomness, and computational efficiency of classical HDC models. Experimental results systematically verify the advantages of quantum-state encoding mechanisms in representing complex, nonlinear information patterns.

- **Chapter 6** develops a Quantum-enhanced Hyperdimensional Computing method by leveraging quantum mechanisms to implement core HDC operations such as binding and bundling. This chapter provides a detailed theoretical foundation and practical implementations conducted on IBM quantum platforms (both Aer Simulator and real Quantum Processing Units). Rigorous experimental comparisons clearly demonstrate that the quantum-enhanced framework surpasses classical HDC methods in terms of classification accuracy, computational efficiency, and generalization capability, thus validating the practical potential of quantum computing in realistic machine learning scenarios.

Finally, **Chapter 7** concludes the thesis by summarizing the key theoretical and practical contributions across all chapters. It also discusses the limitations of the current work and outlines potential future directions, including the extension of quantum learning methods to multi-task remote sensing scenarios, the development of more scalable quantum graph learning architectures, the deployment of quantum-enhanced HDC in broader applications, and the exploration of engineering pathways for real-world quantum machine learning systems.

Chapter 2

Backgrouds and Fundamentals

Contents

2.1 Quantum Computing Essentials

2.1.1 Qubits and Multi-Qubits Systems

At the core of quantum computing lies the concept of the **quantum bit**, or qubit, which servers as the fundamental unit of quantum information. Unlike classical bits can only exit in one of two states—0 or 1, a qubit can exit in a **superposition** of both states simultaneously. Mathematically, a qubit is represented as a unit vector in a 2-dimensional complex Hilbert Space:

$$|\psi\rangle = \alpha\,|0\rangle + \beta|1\rangle, \text{ where } \alpha,\, \beta \in \mathbb{C}, \quad |\alpha|^2 + |\beta|^2 = 1 \tag{2.1}$$

Here α and β are the standard computational basis states, and α and β are complex-valued **probablity amplitude**. The squared magnitudes $|\alpha|^2$ and $|\beta|^2$ determine the probabilities of the qubit collapsing to the corresponding basis state upon measurement. In addition to the linear combination of basis vector expression, a convenient parametrization of a pure state using spherical coordinates is:

$$|\psi\rangle = cos(\frac{\theta}{2})\,|0\rangle + e^{i\theta} sin(\frac{\theta}{2})\,|1\rangle\ , \text{ where } \theta \in [0, 2\pi] \text{ and } \psi \in [0, 2\pi] \tag{2.2}$$

This form ensures the state is normalized and directly maps to a poitn on the Bloch sphere. For example:

- $\theta = 0$ gives the $|0\rangle$
- $\theta = \pi$ gives the $|1\rangle$
- $\theta = \pi/2$ gives the equally weighted stated $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$
- $\theta = \pi/2, \phi = \pi$ gives $\frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$, etc.

Such arbitrary superposition states are routinely initialized and manipulated in quantum algorithms and serve as inputs to variational quantum circuits and quantum feature maps The superposition principle allows qubits to hold more information than classical bits, making them suitable for exponentially larger state spaces. A single qubit state can be visualized as a point on the Bloch sphere, a unit sphere in three-dimensional space, with the north and south poles representing the classical states $|0\rangle$ and $|1\rangle$, and all other points representing superpositions. This geometric representation is widely used to understand quantum gates as rotations on the sphere.

When working with multiple qubits, the total quantum system is described by the tensor product of the individual qubit states. For instance,a two-qubit system is represented as:

$$|\psi\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \tag{2.3}$$

if each qubit is in a superpositon, their joint state lives in a composite-Hilbert space of dimension 2^n, where n is the number of qubits. For example:

$$|\psi\rangle = (\alpha\,|0\rangle + \beta|1\rangle) \otimes (\gamma\,|0\rangle + \delta\,|1\rangle) \tag{2.4}$$

This tensor structure enables the representation of entangled states, a uniquely quantum phenomenon where the overall state cannnot be factored into individual qubit states. A famous example is the **Bell State**:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{2.5}$$

One of the foundational reasons quantum computing is believed to have the potential for exponential speedups lies in the exponential growth of its representational state space. In classical computing, an n-qubit register can only represent one of 2^n possible configurations at a time. For example, a 3-qubit classical register may store either $000, 001, \ldots\ldots$ or 1111. In contrast, a quantum system with n qubits can exist in a superposition of all 2^n basis states simultaneously. That is, the full state of an n-qubit system can be written as :

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle, \quad \text{where } \alpha_i \in \mathbb{C}, \quad \sum_{i=0}^{2^n-1} |\alpha_i\rangle^2 = 1. \tag{2.6}$$

each basis state $|i\rangle$ corresponds to one of the 2^n classical bitstrings of length n, and α_i is the complex probability amplitude associated with that configuration. The set of all possible quantum states thus forms a Hilbert space of dimension 2^n. For instance, a 3-qubit system can be in a state like:

$$|\psi\rangle = \frac{1}{2}|000\rangle + \frac{i}{2}|011\rangle + \frac{1}{2}|101\rangle - \frac{1}{2}|111\rangle \tag{2.7}$$

This state represents a combination of multiple states and can be processed and evolved as a whole through a unified quantum operation, such as the unitary transformation of units. The exponential growth of this type of state space enables quantum systems to compress, express, and process in parallel high-dimensional, structured and highly combinable information with far fewer resources than classical systems, demonstrating potential theoretical advantages in the following tasks:

- Quantum Search (e.g., Crover's algorithm)
- Quantum simulation of physical systems
- Quantum-enhanced machine learning, where high-dimension feature space can be naturally encoded in quantum states.

However, it should also be noted that the measurement of quantum systems will cause the collapse of states, making it impossible to directly extract all amplitudes α_i, which means we cannot "read the entire quantum state". Therefore, quantum algorithms need to be carefully designed to indirectly obtain useful information by using interference effects and expected value extraction strategies. This is precisely the reason why variable component sub-algorithms and hybrid quantum-classical learning models are widely used nowadays.

2.1.2 Quantum Gates and Quantum Circuits

Quantum computation proceeds through the manipulation of quantum states using **quantum gates**, which are reversible, linear, and unitary operations. Unlike classical logic gates (e.g., AND, OR), which are often irreversible, all quantum gates correspond to unitary matrices, satisfying $U^\dagger U = I$. These gates serve as the fundamental building blocks of quantum circuits.

In real-world implementations, idealized pure quantum states are difficult to maintain due to decoherence, gate noise, and hardware limitations. This results in mixed states, especially in near-term quantum devices (NISQ era). Therefore, current research, including this thesis, often employs hybrid quantum-classical algorithms that are robust to hardware imperfections and implementable on simulators like IBM Aer or real backends via IBM Qiskit

Single-Qubit Gates

Single-qubit gates act on a single qubit and correspond to 2×2 unitary matrices. Some of the most commonly used gates include:

- **Pauli Gates:**

  $$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

  The X gates act like a quantum version of the classical NOT operation. The Y and Z gates apply complex phase inversions, flipping or rotating states along different axes of the Bloch sphere.

- **Hadamard Gate (H):** used to create equal superposition states.

  $$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$$

  The Hadamard gate maps basis states to the equatorial plane of the Bloch sphere and is essential for interference-based quantum algorithms.

- Phase Gates (S and T):

  $$S = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \quad T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

These gates introduce global or conditional phase shifts and are critical in quantum Fourier transforms and universal gate construction.

- Rotation Gates (parameterized Gates):

$$R_x(\theta) = e^{-i\theta X/2}, \quad R_y(\theta), \quad R_z(\theta)$$

Rotation gates enable continuous and differentiable transformations on qubit states and are widely used in variational quantum algorithms and quantum neural networks.

Multi-Qubits Gates: Entanglement and Control Logic

Multi-qubit gates act on two or more qubits, enabling entanglement and conditional operations. They are essential for building expressive and globally correlated quantum circuits.

- **CNOT (Controlled-NOT) Gate**:

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

This gate flips the target qubit if the control qubit is in the state $|1\rangle$. For example:

$$(H \otimes I)|00\rangle \xrightarrow{\text{CNOT}} \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

This operation creates maximally entangled Bell state.

- **Controlled-Z (CZ) Gate**: The CZ gate applies a Z rotation to the target qubit only when the control qubit is in the state $|1\rangle$. It plays an important role in cluster-state quantum computing and quantum graph algorithms.

Universal Gate sets and Circuit Decomposition

Quantum circuits are constructed by sequencing quantum gates in time. Any quantum operation can be approximated using a universal gate set, such as: $\{H, T, CNOT\}$, $\{R_x, R_z, CNOT\}$.

The spatial complexity of a circuit depends on the number of qubits, and the temporal complexity on its depth. Parameterized quantum circuits (PQCs) form the backbone of many quantum learning models. They typically consist of:

- Layers of parameterized rotation gates,
- Entangling layers (e.g., CNOT or CZ),
- Measurement and classical optimization via loss minimization.

These circuits support:

- Variational Quantum Classifiers (VQC),
- Quantum Neural Networks (QNN),
- Quantum Feature Maps in kernel-based learning.

They are central to this thesis's contributions in cropland classification, quantum-enhanced hyperdimensional computing, and graph-based quantum models.

Parameterized Quantum Circuits for Learning Tasks

In variational quantum algorithms (VQAs), circuits are constructed using parameterized gates such as $R_y(\theta_i)$, and trained using classical optimization. These circuits typically involve:

- Repeated layers of rotation gates,
- Entanglement layers using CNOT or CZ gates,
- Final measurement and loss function evaluation.

Such architectures are referred to as:

- Variational Quantum Circuits (VQC),
- Quantum Neural Networks (QNNs),
- Quantum Feature Maps.

These structures form the foundation of quantum models for classification, kernel learning, and representation learning. They are also central to the experiments presented in this thesis for cropland classification, high-dimensional state encoding, and graph learning.

2.1.3 Quantum Measurement and Probabilistic Behavior

Measurement is a fundamental operation in quantum mechanics and plays a central role in all quantum computing processes. Unlike classical systems where the state of a system can be queried deterministically, quantum measurements are inherently *probabilistic*, resulting in **state collapse** and irreversible information extraction.

Measurement in the Computational Basis

Consider a qubit in an arbitrary pure state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad \text{where } \alpha, \beta \in \mathbb{C},\ |\alpha|^2 + |\beta|^2 = 1.$$

When measured in the standard computational basis $\{|0\rangle, |1\rangle\}$, the state collapses to one of the basis vectors:

- Outcome 0 occurs with probability $P(0) = |\alpha|^2$;
- Outcome 1 occurs with probability $P(1) = |\beta|^2$.

After the measurement, the superposition is destroyed, and the qubit resides in the observed state. This irreversible transformation is known as *wavefunction collapse*.

General Projective Measurement

More generally, measurement is defined via a set of Hermitian projection operators $\{P_i\}$ satisfying:

$$\sum_i P_i = I, \quad P_i = P_i^\dagger = P_i^2.$$

For a state $|\psi\rangle$, the probability of observing outcome i is:

$$P(i) = \langle\psi|P_i|\psi\rangle,$$

and the post-measurement state becomes:

$$|\psi'\rangle = \frac{P_i|\psi\rangle}{\sqrt{P(i)}}.$$

For qubits, standard measurement corresponds to:

$$P_0 = |0\rangle\langle 0| = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad P_1 = |1\rangle\langle 1| = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

Measurement in Alternate Bases

Quantum measurement is not restricted to the computational basis. For example, measurement in the Hadamard (X) basis uses the states:

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad |-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle).$$

This is implemented by first applying a unitary transformation (e.g., Hadamard gate), then measuring in the standard basis. Such flexibility is essential for quantum algorithms like teleportation, entanglement detection, and kernel estimation.

Expectation Values and Observables

In variational quantum algorithms, the output is often the expected value of an observable $\mathcal{O}$:

$$\langle \mathcal{O} \rangle = \langle \psi | \mathcal{O} | \psi \rangle.$$

This is estimated via repeated measurements and statistical averaging. Observables are typically decomposed into Pauli operators (e.g., $Z_1 Z_2$), and measurements are performed in the corresponding bases.

Sampling and NISQ Limitations

On near-term quantum devices, measurement results are obtained through repeated sampling (shots). For N runs of a circuit, the outcome distribution is estimated as:

$$P(0) \approx \frac{N_0}{N}, \quad P(1) \approx \frac{N_1}{N}.$$

This sampling-based estimation introduces variance. Hardware noise, decoherence, and readout errors further degrade precision. Practical solutions include:

- Readout error mitigation;
- Calibration and measurement unfolding;
- Shadow tomography and classical post-processing.

Measurement and Irreversibility

A defining feature of quantum measurement is its irreversibility. Once a state is measured, the original superposition is lost and cannot be recovered. This makes measurement a non-trivial resource in quantum algorithm design.

For quantum machine learning, this constraint requires:

- Efficient extraction of task-relevant statistics;
- Minimization of destructive measurements during training;
- Careful design of readout strategies in classification and regression.

Measurement protocols thus shape both the theoretical and practical landscape of quantum learning models.

2.1.4 Quantum Computing Models and Platforms

Quantum computing can be achieved through various theoretical models and can be implemented on multiple hardware platforms. Understanding these models and current actual systems is crucial for designing effective quantum algorithms and deploying them to actual quantum processors.

Quantum Computing Models

During the development of quantum computing, a variety of different computing models have gradually emerged, and each model reveals the potential capabilities of quantum computing from different perspectives. Among them, the most mainstream and widely used one is the "Quantum Circuit Model", also known as the "Gate Model". In this model, quantum computing is achieved by applying a series of unitary quantum gate operations with sequential arrangements to the initial qubit states. Each quantum gate corresponds to a reversible linear transformation, and the calculation result is ultimately obtained by measuring the qubits. Overall, a quantum circuit can be represented by an overall unitary matrix U. Its role as the input state $|\psi_{\text{in}}\rangle$ mapped to the output state $|\psi\rangle_{\text{out}} = U\,|\psi_{\text{in}}\rangle$. At present, most classical quantum algorithms, such as the Shor factorization algorithm, the Grover search algorithm, and various variable-component sub-algorithms, are constructed and implemented based on this model.

In addition to the gate model, "Measurement-Based Quantum Computing (MBQC)" provides another computing paradigm. In MBQC, a highly entangled resource state (such as a cluster state) is first prepared, and then the entire computing process is advanced by conducting a series of adaptive measurements on each qubit. Unlike the explicit unitary operations applied in the gate model, MBQC relies entirely on the selection of measurement sequences and measurement bases to control the calculation process. This mode proves that even without explicit quantum gate operations, general quantum computing can be achieved merely through measurement itself. Although MBQC is not yet widely used in practical engineering applications at present, it has important theoretical significance for the design of quantum error correction and fault-tolerant computing architectures.

Another important Quantum Computing model is Adiabatic Quantum Computing (AQC) and its engineering implementation form, Quantum Annealing. In the adiabatic model, the solution of the problem is encoded in the ground state of a target Hamiltonian. By allowing a quantum system to slowly evolve from a simple initial Hamiltonian of a known ground state to a complex target Hamiltonian, according to the adiabatic theorem, as long as the evolution process is slow enough, the system will remain in the ground state, thereby obtaining the solution to the problem. Although the adiabatic model is theoretically equivalent to the gate model, actual quantum annealing equipment (such as D-Wave systems) can only approximately implement this process and is mainly applied in solving combinatorial optimization problems, rather than having general quantum computing capabilities.

In the current era known as ”NISQ (Noisy Intermediate-Scale Quantum)”, due to the limited scale of quantum devices and the high noise level, Variational and Hybrid Quantum-Classical Models have become the mainstream choice for practical applications. Under this paradigm, quantum computing is only used to perform shallow parameterized quantum circuits, and the circuit parameters are iteratively updated by the classical optimizer after the measurement output. This hybrid structure greatly reduces the requirements for quantum hardware, enhances the robustness of the algorithm to noise, and also opens up broad space for applications such as quantum machine learning, quantum chemical computing, and quantum optimization. Typical examples include variable component feature solvers (VQE), quantum Approximation optimization algorithms (QAOA), and various quantum-enhanced machine learning frameworks.

Overall, different quantum computing models demonstrate different dimensions and possibilities of quantum information processing. From the gate model that emphasizes the operation sequence of single quantum gates, to the measurement-driven MBQC, to the adiabatic model that utilizes the physical evolution path, and to the variational model that integrates classical optimization strategies, these theoretical systems provide a rich toolbox for the design of quantum algorithms. In this paper, mainly based on the ”quantum circuit model”, combined with the variational method, we use the superconducting qubit platform to implement specific quantum-enhanced machine learning tasks and conduct experimental verification on actual NISQ hardware.

2.2 Quantum Machine Learning Techniques

This section describes some techniques of quantum machine learning and serves as a bridge connecting the fundamental theories of quantum with specific research tasks. It requires a systematic introduction to the current major QML methods, including their principles, applications, and feasibility on NISQ devices.

2.2.1 Quantum Feature Mapping and Hilbert Space Embeddings

In many Quantum Machine Learning (QML) methods, a core idea is to utilize the high-dimensional characteristics of the Hilbert space where the quantum states are located to map classical data to the quantum space for processing. This process is similar to the feature mapping idea used in the classical kernel method, that is, to enhance separability by projecting the input data into a high-dimensional space. However, quantum feature mapping may far exceed classical methods in terms of expressive power due to its involvement in exponential spatial structures and non-classical state interference mechanisms.

Quantum Feature Mapping

Let $x \in \mathbb{R}^d$ denote a classical input sample. A quantum feature map aims to encode x into a quantum state $|\phi(x)\rangle \in \mathcal{H}$, where $\mathcal{H}$ is a high-dimensional Hibert space associated with n qubits: $x \mapsto |\phi(x)\rangle = U_{\phi(x)} |0\rangle^{\otimes n}$

Among them, $U_{\phi(x)}$ is a parameterized quantum circuit that depends on a input x, and $|0\rangle^{\otimes n}$ is the all-zero initial state. The quantum state $|\phi(x)\rangle$ can encode complex non-linear relationships between features due to the entangling and interference properties of the quantum circuit. some common choices for quantum feature encoding include:

- Angle encoding: $x_i \mapsto R_y(x_i)$ or $R_z(x_i)$
- Basis encoding: Binary values mapped directly to computational basis states.
- Amplitude encoding: Embeds x into the amplitude vector of a state, requiring normalization
- Entanglement-based embeddings: Incorporates correlations using multi-qubit interactions

The design of the quantum encoding circuit $U_{\phi(x)}$ directly affects the representational capacity and trainability of the model.

Inner Products as Quantum Kernels

One important application of quantum feature mapping in quantum machine learning is to construct quantum kernel functions, whose core idea is to measure the similarity between samples by using the inner product between quantum states. Specifically, for the two input samples x and $x^{'}$ They are respectively encoded into quantum states through feature mapping $|\phi(x)\rangle$ and $|\phi(x^{'})\rangle$. Their similarity is expressed by the square of their inner product as $k(x,x') = |\langle\phi(x)|\phi(x')\rangle|^2$. This quantum kernel function can be used as the input of kernel machine learning models (such as support vector machines or kernel regression), replacing classical kernels (such as Gaussian kernels, polynomial kernels, etc.). Compared with classical methods, quantum kernel functions have potential computational advantages: when quantum states construction circuits $U_{\phi(x)}$ are ifficult to simulate by classical computation, the corresponding kernel functions are also difficult to calculate efficiently by classical methods, which may show quantum advantages in complex data distributions. In practical operation, this type of quantum inner product can be estimated on NISQ hardware through SWAP tests, interference circuits or quantum overlap estimation circuits, and it is a key component of multiple current quantum classification models.

Advantages and Challenges

Quantum feature mapping has the following potential advantages:

- Exponential spatial representation capability: n qubits can be spanned 2^n dimensional space.
- Natural nonlinear modeling: Capturing high-order interactions through entanglement and polymorphic interference.
- Hardware-friendliness: Some coding circuits have a simple structure and are suitable for implementation on NISQ devices.

But there are also challenges:

- Coding circuit design problem: How to balance representation ability and trainability;
- Measurement noise and sampling error: Affecting the accuracy of kernel estimation;
- Plain phenomenon (Barren Plateaus) : Partial circuits cause gradients to disappear, hindering training.

Applications in Classification and Regression tasks

Quantum feature mapping has achieved initial application effects in multiple tasks:

- Quantum support vector classification: Used for the classification of high-dimensional data such as images and time series;
- Quantum kernel regression: Conducting function approximation and prediction modeling;
- Unsupervised learning: Used in methods such as quantum k-means and quantum spectral clustering.

In several works of this thesis, quantum feature mapping constitutes the basic module of the modeling process, including the remote sensing image classification task and the quantum ultra-high-dimensional coding representation task, etc., aiming to effectively embed structured data into the high-dimensional quantum state space.

2.2.2 Variational Quantum Algorithms and Quantum Neural Networks

Variational quantum algorithms (VQAs) represent a significant breakthrough for practical quantum computing, particularly in the current noisy intermediate-scale quantum (NISQ) era. These algorithms integrate quantum hardware with classical optimization routines, providing a flexible framework for addressing computational tasks such as combinational optimization, quantum chemistry simulations, and machine learning.

A typical variational quantum algorithm employs a *parameterized quantum circuit (PQC)*, also referred to as an *ansatz*, whose parameters are optimized iteratively using classical optimization algorithms. The goal of optimization is to minimize (or maximize) a cost function defined as the expectation value of a quantum observable measured from the quantum circuit.

Mathematically, given a parameterized circuit $U(\boldsymbol{\theta})$, with parameters $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_m)$, the cost function is defined as:

$$C(\boldsymbol{\theta}) = \langle 0| \, U^\dagger(\boldsymbol{\theta}) \, \mathcal{O} \, U(\boldsymbol{\theta}) \, |0\rangle \,,$$

where $\mathcal{O}$ is a Hermitian operator representing the problem at hand. For example, in Variational Quantum Eigensolvers (VQE), $\mathcal{O}$ typically corresponds to the Hamiltonian of the physical system under study.

Quantum Neural Networks

Quantum neural networks (QNNs) are a special subclass of variational quantum circuits tailored specifically for machine learning tasks. Analogous to classical neural networks, QNNs are organized into layers comprising parameterized gates and entanglement operations.

Formally, the state after applying L layers of quantum gates can be expressed as:

$$|\psi(\boldsymbol{x},\boldsymbol{\theta})\rangle = U_L(\boldsymbol{\theta}_L)\, U_{L-1}(\boldsymbol{\theta}_{L-1}) \dots U_1(\boldsymbol{\theta}_1)\; |\phi(\boldsymbol{x})\rangle\,,$$

where $|\phi(\boldsymbol{x})\rangle$ encodes classical data $\boldsymbol{x}$ into a quantum state, and $\boldsymbol{\theta}_l$ represents parameters associated with the l-th layer.

The output of a QNN is typically extracted by measuring certain qubits in the computational basis, obtaining expectation values or probability distributions, which are subsequently processed by classical post-processing methods, such as logistic regression or softmax classifiers.

Optimization and Training of Variational Circuits

Training variational quantum circuits is inherently a hybrid quantum-classical procedure. Circuit parameters $\boldsymbol{\theta}$ are optimized classically, employing gradient-based methods (e.g., stochastic gradient descent, Adam) or gradient-free methods (e.g., SPSA, Nelder-Mead).

Gradients for variational circuits can be efficiently computed via the *parameter-shift rule*, specifically designed for quantum circuits:

$$\frac{\partial C(\boldsymbol{\theta})}{\partial \theta_i} = \frac{1}{2}\left[C\left(\boldsymbol{\theta} + \frac{\pi}{2}\mathbf{e}_i\right) - C\left(\boldsymbol{\theta} - \frac{\pi}{2}\mathbf{e}_i\right)\right],$$

where $\mathbf{e}_i$ is the unit vector in parameter space corresponding to parameter θ_i.

Despite their promise, variational algorithms frequently encounter challenges such as barren plateaus (regions with vanishing gradients), local minima, and optimization landscapes distorted by hardware noise. Advanced methods, including layer-wise initialization, adaptive circuit architectures, and error mitigation strategies, are actively investigated to address these issues.

Advantages and Limitations

Variational quantum algorithms and QNNs offer notable advantages:

- **Hardware-friendly implementation**: Typically involves shallow circuits compatible with current NISQ devices.
- **Adaptive expressivity**: Flexible circuit structures capable of modeling complex data distributions.
- **Noise resilience**: Hybrid quantum-classical frameworks partially mitigate hardware imperfections.

Nevertheless, practical limitations exist:

- **Optimization difficulty**: Complex landscapes and barren plateaus complicate training.
- **High sampling cost**: Large numbers of shots required for accurate expectation value estimation.
- **Lack of theoretical guarantees**: Limited rigorous analysis demonstrating quantum advantage.

Applications in Quantum Machine Learning

Quantum neural networks have been explored in various learning tasks, including:

- Quantum-enhanced classifiers and regression models;
- Quantum autoencoders for compression and feature learning;
- Quantum generative adversarial networks (QGANs) for generating data distributions.

Within the context of this thesis, variational quantum circuits and quantum neural networks constitute foundational architectures for quantum-enhanced classification, hyperdimensional encoding, and graph learning tasks. Their ability to efficiently represent complex correlations within high-dimensional quantum states is systematically validated through experimental evaluations conducted on IBM quantum processors.

2.2.3 Quantum Platforms

With the wide application of quantum technology, a large number of quantum platforms and communities have emerged worldwide. We have summarized the following quantum platforms in the table 2.1.

PennyLane, developed by Xanadu, is a hybrid quantum–classical ML framework that natively supports differentiable quantum circuits and seamlessly integrates with PyTorch, TensorFlow, and JAX. It offers plugin-based backends—including Lightning and Autograd simulators as well as access to IBM Q, Rigetti, IonQ, and Xanadu's own Borealis photonic processor—enabling automatic differentiation for training variational quantum algorithms, though it can face performance bottlenecks for very large circuits and some hardware plugins require separate agreements or fees. Qiskit Machine Learning, part of IBM's Qiskit ecosystem, provides prebuilt models such as QSVM, QCNN, and QGAN, allows direct execution on IBM Quantum's superconducting processors and simulators via the IBM Quantum Cloud, and supports quantum kernel methods, but remains closely tied to IBM infrastructure with limited cross-vendor portability. TensorFlow Quantum, built by Google on top of Cirq, embeds quantum layers into the Keras API and leverages TensorFlow's autodiff and distributed training for large-scale simulation, though it currently supports only simulation backends and lacks direct hardware integration. Paddle Quantum, built on the PaddlePaddle deep learning platform, offers state-vector, density-matrix, and sampling simulators with distributed computing capabilities and a suite of variational algorithms, yet does not yet provide direct connections to hardware. MindQuantum, from Huawei, emphasizes high-performance state-vector and tensor-network simulations with automatic differentiation via MindSpore integration—optimized for large-circuit emulation—but likewise remains simulation-only at present. QuTiP, a community-driven library for open quantum systems and dynamics, features comprehensive solvers for Lindblad master equations, Bloch equations, and quantum-optics scenarios, making it ideal for foundational QML prototyping and quantum-chemistry research, but does not support differentiable circuits or hardware execution. Cirq, Google's low-level framework, offers precise gate-level programming, noise modeling, circuit optimization tools, and (upon application) access to Google's Sycamore and Bristlecone processors, yet lacks high-level ML abstractions and publicly available device endpoints. Strawberry Fields, another Xanadu project, specializes in continuous-variable photonic quantum computing with Gaussian and Fock-state simulators, integrates as a PennyLane backend for photonic QML models, and aligns with Xanadu's Borealis hardware trials, though real-device access remains in an early testing phase.

For the most mature hardware integration and ML workflows, PennyLane and Qiskit Machine Learning are top choices; for deep foundational simulations and system dynamics, QuTiP and Cirq excel; and for differentiable algorithm development within specific deep-learning ecosystems, TensorFlow Quantum, Paddle Quantum, and MindQuantum each provide targeted simulation capabilities, while Strawberry Fields addresses the unique needs of continuous-variable photonic research. For our work, the platforms we use are mostly Paddle Quantum and IBM Qiskit. The main reason why we chose these two platforms is that Paddle Quantum provides high-performance distributed simulation and seamless integration with the existing deep learning pipeline, while Qiskit can directly access real quantum hardware and has a complete algorithm library.

Table 2.1: Comparison of popular quantum machine learning platforms

Platform	**Organization**	**Language**	**Highlights**
PennyLane	Xanadu	Python	One of the most popular hybrid quantum-classical ML frameworks; integrates with PyTorch and TensorFlow; suitable for VQE, QNN, and QML research.
Qiskit Machine Learning	IBM	Python	A dedicated module of Qiskit for quantum ML; supports variational classifiers, quantum kernels, QNNs; compatible with real IBM quantum hardware and simulators.
TensorFlow Quantum (TFQ)	Google	Python	Built on top of Cirq; integrates quantum circuits with TensorFlow; supports gradient-based training of quantum models.
Paddle Quantum	Baidu	Python	Based on the PaddlePaddle ecosystem; supports a variety of quantum algorithms and seamless integration with PaddlePaddle's deep learning stack.
MindQuantum	Huawei	Python	Provides efficient circuit simulation and hybrid training; supports automatic differentiation; designed for quantum-classical co-optimization.
QuTiP	Community	Python	Focused on simulating open quantum systems and quantum dynamics; useful for foundational QML model prototyping.
Cirq	Google	Python	A low-level quantum programming framework; forms the backend of TFQ; good for customizing circuits and quantum logic.
Strawberry Fields	Xanadu	Python	Designed for continuous-variable (CV) quantum computing; can be integrated with PennyLane for photonic quantum ML models.

Chapter 3

Cropland Quantum Learning on Cropland Classification

Contents

3.1 Overview

Accurate cropland classification is important for sustainable land management and agricultural planning. Remote sensing data, such as the Normalized Difference Vegetation Index (NDVI) and Enhanced Vegetation Index (EVI), have proven instrumental in vegetation monitoring. Traditional machine learning and deep learning show remarkable results in land classification using remote sensing data. However, deep learning requires large models, extensive parameters, and substantial training resources when facing an immense volume of data or high-dimensional features. To address the above issues, this section proposes Cropland Quantum Learning (CQL), a quantum-classical hybrid method that utilizes quantum machine learning to extract features from geospatial information is proposed in this chapter. It integrates with a single-layer fully connected classifier to locate cultivation regions of a given target crop. We conduct comprehensive experiments to demonstrate the effectiveness of our proposed method on NDVI and EVI datasets. The results show that the proposed CQL can achieve performance comparable to traditional deep learning while significantly reducing the number of model parameters. In some datasets, it even achieves better results in multiple metrics. This study provides a new solution for quantum-enhanced approaches in geospatial analysis.

3.1.1 Problem Statement

Cropland classification, a crucial task in remote sensing and agriculture, involves identifying and categorizing different types of agricultural land based on the crops being cultivated. Accurate cropland classification is essential for various purposes, including land management, resource allocation, yield prediction, and environmental monitoring. In recent years, the integration of machine learning and deep learning techniques has revolutionized the field of cropland classification, enabling more efficient and accurate analysis of agricultural landscapes.

Cropland classification with machine learning [11, 12] and deep learning [13–18] has demonstrated remarkable accuracy improvements compared to traditional methods. Yet, as the scale and complexity of data continue to grow, the limitations of classical deep learning become increasingly apparent: large models, extensive parameters, and substantial training resources.

Recently, Quantum machine learning(QML) [6, 7] has been developed and introduced into several fields. QML merges principles from quantum mechanics and machine learning to enable the development of more powerful and efficient computational models. This new paradigm offers a promising solution to tackle complex computational challenges by harnessing the unique properties of quantum systems. At its core, quantum machine learning leverages the principles of superposition and entanglement, which are fundamental to quantum mechanics. These principles enable quantum computers to process and manipulate information in ways that are fundamentally different from classical computers. Quantum bits, or qubits, can exist in a superposition of states, allowing them to represent and process multiple possibilities simultaneously. Additionally, qubits can become entangled, leading to correlations between qubits that enable novel methods of information processing.

In the realm of machine learning, quantum computing holds the potential to accelerate tasks such as optimization, matrix inversion, and sampling, which are often time-consuming for classical computers. Quantum algorithms, such as the Quantum Support Vector Machine (QSVM) [19], Quantum Neural Network (QNN) [20], and Quantum Principal Component Analysis (QPCA) [21], are being explored to enhance the efficiency and capabilities of various machine learning tasks.

Inspired by the advantages of quantum computing and quantum machine learning, this section presents the Cropland Quantum Learning (CQL), a hybrid quantum-classical model based on the variational shadow quantum learning (VSQL) method [22], to locate cultivation regions of the given target crop. The model consists of a quantum module and a classical module. The former is used for feature encoding and extraction, while the latter is a fully connected layer followed by a softmax function to output the classification results. Note that quantum computing and quantum machine information application [23] is still in its infant stages, and building practical and scalable quantum computers remains a significant challenge. As a result, the proposed CQL model is implemented using the Paddle Quantum [24], a simulating platform on classical computers developed by Baiduquantum machine learning is primarily simulated on classical computers. Platforms such as PennyLane [25], IBM Qiskit [26], and Baidu Paddle Quantum [24] are among the leading tools for implementing hybrid quantum-classical [27] models and quantum machine learning at this stage.

3.1.2 Main Contributions

The main contributions of this chapter are summarized as follows:

- A quantum-classical hybrid Learning framework (Variable Shadow Learning, VSL) is proposed for the remote sensing cropland classification task. This method combines the Quantum Shadow theory with the variable component sub-algorithm for the first time to construct a quantum-classical hybrid neural network architecture suitable for remote sensing images, effectively enhancing the feature expression ability for complex spatiotemporal data.

- The quantum data coding strategy for cropland classification is designed and verified. Based on the characteristics of remote sensing image data, this study proposed a quantum Encoding scheme combining Amplitude Encoding and shadow measurement, which gave full play to the high-dimensional expression advantage of quantum states and significantly enhanced the data characterization ability.

- Extensive experiments are conducted to verify the validity and robustness of the proposed model. In this study, real remote sensing datasets are used for testing. The performance of the proposed CQL method in terms of classification accuracy, generalization ability, and stability is evaluated in detail, and a strict comparison is made with multiple classical benchmark methods (including CNN and LSTM). The results show that the proposed method is superior to the classical methods in key performance indicators.
- The advantages and potential practical application value of quantum methods are deeply discussed. This chapter not only theoretically expounds why the proposed model has advantages in the classification task of high-dimensional remote sensing data, but also analyzes the feasibility, potential advantages and limitations of this method in actual agricultural management application scenarios, providing valuable theoretical and practical basis for the future application of quantum machine learning in the field of agricultural remote sensing.

3.2 Related Work

The study of cropland classification has witnessed significant advancements through the application of machine learning and deep learning techniques. Numerous researchers have explored the integration of these methodologies to accurately discern and classify cropland regions. On the contrary, quantum machine learning and quantum neural networks represent a highly emerging research direction that serves as a crucial bridge between quantum computation and real-world classical problems. They also constitute a significant endeavor to apply quantum algorithms to classical issues. In this chapter, we delve into the existing deep learning-based approaches for cropland research and explore some of the achievements in quantum neural networks.

The application of deep learning techniques in remote sensing-based cropland classification has gained significant traction due to its ability to automatically extract complex features from high-dimensional and heterogeneous remote sensing data. Deep learning methods, particularly Convolutional Neural Networks (CNNs) [28] and Recurrent Neural Networks (RNN) [29], have demonstrated remarkable success in various land cover and land use classification tasks. Researchers have increasingly explored the potential of these techniques to accurately classify cropland areas using multispectral imagery, including indices such as the Normalized Difference Vegetation Index (NDVI) [30] and Enhanced Vegetation Index (EVI) [31].

CNNs have emerged as a powerful tool for feature extraction from remote sensing images. They excel at capturing spatial patterns and hierarchies of features that are essential for accurate classification. In cropland classification, CNNs have been employed to analyze multispectral imagery, effectively distinguishing different land cover types, including cropland. For instance, Ao et al. [32] utilized CNNs to classify agricultural land use from sentinel-2 images.

Another approach gaining momentum in cropland classification is the utilization of Recurrent Neural Networks to incorporate temporal information. RNNs are well-suited for sequences of data, making them suitable for analyzing the time-series nature of remote sensing data. To illustrate, an LSTM network is employed to learn and classify long-term land cover in China spanning from 1982 to 2015 [33].

Despite the successes of pure deep learning models, challenges persist, including the need for substantial amounts of labeled data and their susceptibility to overfitting. To address these limitations, researchers have explored transfer learning and data augmentation strategies [34]. Transfer learning [35], where pre-trained models are fine-tuned on specific tasks, has been shown to improve the classification accuracy for cropland areas.

While deep learning models have shown significant promise in cropland classification, there is growing interest in leveraging quantum computing to further enhance classification accuracy. Our proposed quantum-classical hybrid neural network builds upon the foundations laid by deep learning methods while using the quantum computing paradigm to accelerate feature extraction and enhance classification performance. By integrating quantum and classical resources, we aim to create a synergistic approach that overcomes the limitations of pure classical methods and advances the state-of-the-art in cropland classification accuracy.

3.3 Methodology

In this section, we provide a comprehensive statement of the methodology employed in this chapter. Our elucidation encompasses various facets, including data collection, data encoding, network architecture, and algorithmic specifics. Each of these elements is thoroughly expounded upon to ensure a comprehensive understanding of our research approach. we embark on a case analysis centered on Jackson County, located in Minnesota, within the United States, serving as our designated study area. This is followed by a detailed explanation of data and its designed coded representation. Furthermore, we propose a quantum-classical hybrid model to handle these data effectively, addressing the binary classification task of determining whether corn cultivation is present in the planting regions.

3.3.1 Data Collection

In this context, the bands corresponding to corn crops within the CDL dataset were selected for 2020, 2021, and 2022 within the specified county. Within each unit area, data relating to both EVI and NDVI during the growth stages of corn were extracted. Furthermore, the areas where corn was cultivated were labeled as "1", while regions without corn cultivation were designated as "0". A cumulative total of 9228, 9353, and 9119 data instances were acquired within this region.

NDVI: The Normalized Difference Vegetation Index (NDVI) dataset is a widely used remote sensing measurement that provides valuable information about the health and vigor of vegetation cover. NDVI is derived from satellite imagery and is commonly used to monitor and assess vegetation growth, land cover changes, and environmental conditions. It is instrumental in agriculture, ecology, and environmental studies. The NDVI is calculated using the following formula:

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{3.1}$$

Where

- NIR is the Near-Infrared band reflectance from the satellite image.
- Red is the Red band reflectance from the same image.

EVI: The Enhanced Vegetation Index dataset is another remote sensing measurement used to assess the health and density of vegetation cover, similar to the NDVI. The EVI was developed to address some of the limitations of NDVI, particularly in areas with dense vegetation or high levels of soil background reflectance. EVI takes into account atmospheric influences and enhances sensitivity to variations in vegetation cover. The EVI is calculated using the following formula:

$$EVI = 2.5 \times \frac{NIR - Red}{NIR + 6 \times Red - 7.5 \times Blue + 1} \tag{3.2}$$

Where

- NIR is the Near-Infrared band reflectance from the satellite image.

- Red is the Red band reflectance from the same image.
- Blue is the Blue band reflectance from the same image.

Labels: The labels are extracted from the Cropland Data Layer (CDL) [36]. The CDL constitutes a yearly generated crop-specific land cover dataset encompassing the entire continental United States. This compilation is crafted utilizing moderate-resolution satellite imagery in conjunction with comprehensive agricultural ground truth information. This dataset furnishes the ground truth essential for our binary cropland classification task, where "0" signifies cultivated areas, while "1" corresponds to non-cultivated regions.

Throughout the growth cycle of corn, a total of nine data points, along with a corresponding label, can be extracted for both NDVI and EVI. Summing up, the ultimate raw dataset is composed of individual one-dimensional vectors, each representing a $\{[x_i^j], y_i \in \{0, 1\}, \{j \mid j \in \mathbb{Z}, 0 \leq j \leq N\}\}$.

3.3.2 Data Encoding

Both the original EVI and NDVI data are regarded as foundational datasets, necessitating their conversion into quantum-encoded representations. Quantum encoding methods encompass a spectrum of techniques, encompassing basis encoding, amplitude encoding, and angle encoding, among others. Given the nature of the task, intrinsic data attributes, and the optimization of computational inference, the present study adopts an amplitude encoding approach.

Amplitude encoding encodes a vector $\mathbf{x}$ of length N into amplitudes of an n-qubit quantum state with $n = [\log_2(N)]$:

$$|\mathbf{x}\rangle = \sum_{j}^{N} x^j |j\rangle \tag{3.3}$$

Where $\{|j\rangle\}$ is the computational basis for the Hibert Space. As the amplitudes of a quantum state are constructed from classical information, it is imperative for the input to adhere to the requirement of normalization: $|\mathbf{X}|^2 = 1$.

In this manner, our feature points have been encoded into quantum states and subsequently subjected to padding. The original NDVI and EVI signals can now be fed into the quantum network component for inference.

3.3.3 Model Structure

In this section, we present the underlying architecture of the network, which is composed of three main components. We design our hybrid neural network based on variational shadow quantum learning (VSQL) [22] which utilizes a low-parameter variational quantum circuit U for feature extraction. The first component involves a quantum neural network section that takes pre-encoded quantum states as inputs. This enables the extraction of all features with minimized parameter degrees. These features are then transmitted as inputs to the second component of the network, which is a concatenation of a fully connected network and a softmax layer. This configuration ultimately generates a probability distribution for classification outcomes. The entire architecture is optimized through a backpropagation process, aiming to minimize binary entropy. The optimization pertains to the parameters θ in the quantum part and parameters ω in the classical part. The holistic framework of the network, illustrated in Figure 3.1, packages these components in a structure. The feature engineering module $U(\theta)$ of the network consists of a series of variational quantum circuits. Its purpose is to conduct feature extraction and dimensionality reduction on the pre-encoded raw data D to derive the feature set V. The detailed architecture of $U(\theta)$ is depicted in Figure 2.

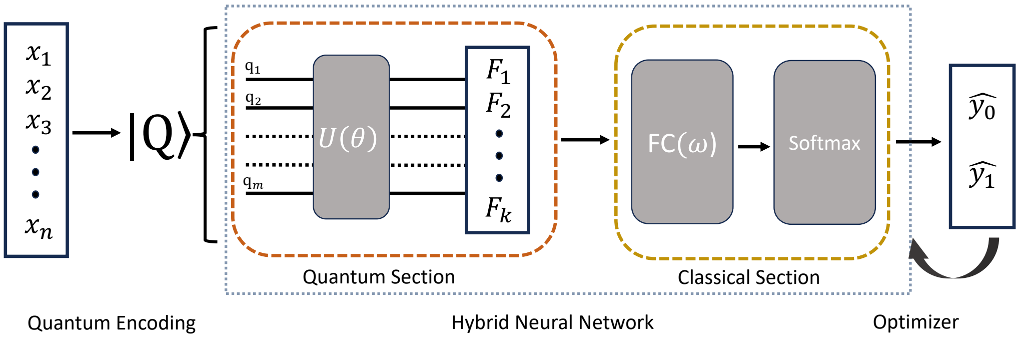


Figure 3.1: The overall architecture of the network.

The approach employed here involves a $U^{'}(\theta)$ sub-network that operates within the subspace formed by two single qubits. Subsequently, a similar sliding mechanism is applied to the following two signal qubits subspace, and this process iterates progressively until the ith to $i+1$th quantum state's subspace is reached the last two single qubits. Within each subspace, the $U^{'}(\theta)$ quantum circuit is executed, followed by a Pauli-X measurement to observe and compute the corresponding feature v.

Functioning as a sub-quantum circuit, $U^{'}(\theta)$ constitutes an integral component of the core $U(\theta)$. Its specific structure is illustrated in Figure 3.3. It encompasses a

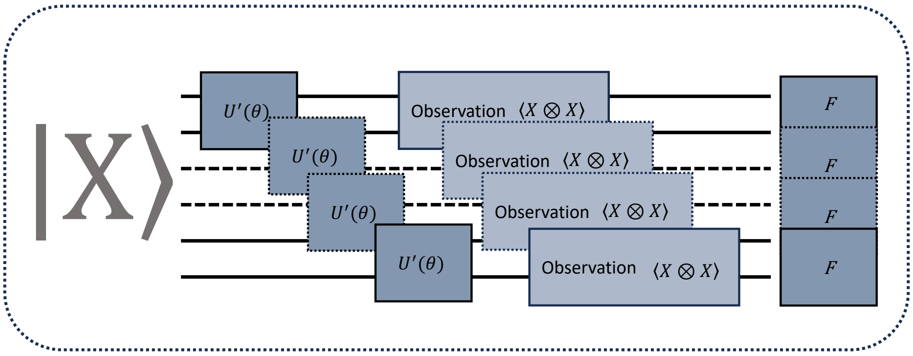


Figure 3.2: Details presentation of $U(\theta)$.

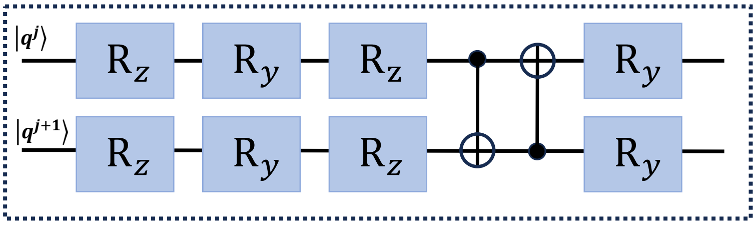


Figure 3.3: Details presentation of $U^{'}(\theta)$.

sequence of $Rz - Ry - Rz$ rotations applied successively to the two individual qubit states within the subspace, followed by a controlled-NOT $CNOT$ operation between the two qubit states. In the final step of the $U^{'}(\theta)$ quantum circuit, there is a single qubit gate rotation, specifically a Ry rotation gate. The matrices for the rotations R and the $CNOT$ operation are provided as follows:

$$CNOT = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad R_y(\theta) = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \quad R_z(\theta) = \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \tag{3.4}$$

R_y and R_x are fundamental single-qubit rotation gates in quantum computing used to introduce specific rotation operations on quantum bits(qubits). Meanwhile, $CNOT$ is a type of gate operation in quantum computing that enables controlled flipping between quantum bits. In a $CNOT$ gate operation, one quantum bit serves as the control bit, while another quantum bit serves as the target bit. When the control bit is 1, the target bit undergoes a flip (0 becomes 1 and 1 becomes 0); when the control bit is 0, the target bit remains unchanged.

Algorithm 1 Cropland Quantum Learning: Encoding and training process

```
Input: Data set $D := \{[x_i^j], y_i \in \{0,1\}, \{j \mid j \in \mathbb{Z}, 0 \leq j \leq N\}, Epoch$
Output: $Loss$, Quantum Circuit Parameters $\theta$, Classical Parameters $\omega$, $b$
Initialize the parameters $\theta$ of quantum circuit
Initialize the parameters $\omega$ and $b$ of classical block
Quantum Encoding on $[x_i]$, $|X_i\rangle = AE([x_i])$, $AE$ is an Amplitude Encoder.
for $epoch$ = (1, $Epoch$) do
    Feed Encoded Data $|X_i\rangle$ to the varitional quantum circuit.
    Utilize $\langle X \otimes X\rangle$ operation measuring the quantum part and receive the quantum part features $v_i$
    Feed $v_i$ into the classical network and obtain $\hat{y}_i$
    Caculate the the Binary Crossentropy $(\hat{y}_i - y_i)^2$ and update the parameters $\theta$, $\omega$ and $b$
end for
```

By utilizing the aforementioned structured quantum circuit block, we can extract the feature sets v during the forward propagation of the network. Subsequently, this feature vector is fed into the subsequent classical network module, which comprises a fully connected network interlinked with a $softmax$ layer. The fully connected layer can capture the complex relationship between the features obtained before, combine these features linearly, and input them into the activation function to generate the probability distribution required by the binary classification problem, to realize the cropland classification task.

The overall structure of the proposed network is a hybrid model that employs a Quantum encoder to encode the raw data and use it as input. It incorporates a parameterized quantum circuit as the feature engineering module, followed by a classical neural network's fully connected layer and softmax activation function as the prediction module.

3.3.4 Algorithm

The training and convergence methods of the entire network are similar to traditional deep learning training methods, as illustrated in Algorithm 1.

Firstly, initialize the parameters θ for the quantum block and ω for the classical block separately. Given the dataset $D := \{[x_i^j], y_i \in \{0,1\}, \{j \mid j \in \mathbb{Z}, 0 \leq j \leq N\}$, where i and j represent the indices of the sample and the feature in the dataset, respectively. N refers to the feature length of each sample. Apply amplitude quantum encoding AE

to the data, the feature dimensions are reduced from N to $[\log_2(N)]$.

$$|X_i\rangle = AE([x_i]) \tag{3.5}$$

Then feed the encoded data into the quantum for feature extraction to obtain the feature set v_i. Continuing, the obtained features are input to the classical network for classification. The corresponding loss function is calculated, and the parameters of both modules are updated accordingly.

$$\mathcal{L}_{\text{BCE}}(y, \hat{y}) = -(y \cdot \log(\hat{y}) + (1 - y) \cdot \log(1 - \hat{y})) \tag{3.6}$$

where $\hat{y}$ is the predictive classification of the entire network on the training data.

$$\hat{y} = \sigma[F_c(F_q(|X\rangle, \theta), \omega)] \tag{3.7}$$

For more details, F_c serves as an abstract representation of the classical network component, which essentially comprises a fully connected layer.

$$F_c = v \cdot \omega + b \tag{3.8}$$

where v is extracted by F_q which is the denotation of the quantum network section.

With this approach, the overall logical objective of the algorithm is to iteratively update the parameters θ of the quantum circuit and the parameters ω of the classical network. By utilizing the Adam optimizer [37], the algorithm aims to minimize the value of the loss function, thereby achieving the training of the hybrid network for classification.

3.4 Experiments

3.4.1 Experimental Setup

Datasets

The overview of the datasets split in our experiments is given in Table **??**I. The data includes pre-processed NDVI and EVI concatenated collection from 2020 to 2022 in County Jackson, Minnesota, USA. The split processing randomly selects 80% of the datasets of each year as the training dataset, and the rest 20% is reserved for testing the trained models. Within the training set, a further 20% is set aside as a validation set. From the Table 3.1, the 2022 dataset contains a significantly lower number of positive samples (e.g., around 22%). Therefore, after splitting the data for 2022, the Synthetic Minority Oversampling Technique (SMOTE) [38] is applied to oversample the 2022 training set to ensure a balanced class distribution. The resultant dataset is denoted as 2022^a in the rest of this chapter. On the other hand, both the validation and test sets retain their original, unchanged distribution.

Baselines

To compare with traditional deep learning methods, we employed both LSTM and CNN models. For CNN models, we conducted both 1-dimensional CNN (1D CNN) and 2-dimensional CNN (2D CNN).

Implementation

For the proposed Cropland Quantum Learning hybrid network, we devised two distinct network architectures and corresponding experiments. The first approach treats NDVI and EVI data within a single sample and keeps them as a unified entity, termed as the One-Head Cropland Quantum Learning (1Head CQL). The alternative, the Two-Head Cropland Quantum Learning (2Head CQL), treats NDVI and EVI as separate features, encoding them individually and subsequently feeding them into two distinct quantum circuits. To realize the quantum-related feature engineering, our experimental environment primarily uses Paddle-Quantum (2.4.0). Additionally, the baselines utilize the PaddlePaddle (2.3.0) [39]. Both of them are open-source packages from Baidu.

Hyper-Parameters

To facilitate a more effective analysis of the results, we fix certain hyperparameters across different networks. These include a batch size of 32 and a learning rate of 0.001. For each model on different datasets, we conducted training for 10 epochs. Furthermore, we use Adam as the optimizer. Consequently, the results obtained are more conducive to a controlled variable analysis.

3.4.2 Performance and Evaluation

Throughout the experiment, we aim to compare the performance of different networks across various datasets. By analyzing the experiment results, we gain insights into the strengths and weaknesses of each network for the given task.

Table 3.1: Dataset Split Detail (The number of samples in the train, test, and validation sets for each dataset, with the percentage of positive samples in each set given in parentheses.)

Data	Train Set	Test Set	Validation Set
2020	5905(47.1%)	1846(46.9%)	1477(45.6%)
2021	5986(43.5%)	1871(45.0%)	1497(45.3%)
2022	5836(22.3%)	1824(22.9%)	1459(22.7%)
2022^a	9038(50%)	1824(22.4%)	1459(22.3%)

Table 3.2: Comparison of Model Complexity

Methods	Parameters	Layers	FCNN
1D CNN	154	1× CNN	✓
2D CNN	170	1 ×CNN	✓
LSTM	138	1× LSTM	✓
1Head CQL	13	1 × CQL	✓
2Head CQL	23	1× CQL	✓

Model Complexity

Before delving into the analysis of the results, we first consider the complexity of the network. To assess model complexity, we utilize the model's training parameters and network depth, with specific network configurations detailed in Table 3.2.

The Table 3.2 illustrates that all models consist of a single layer, excluding pooling layers, activations, and other non-parametric components. Moreover, before getting the final outputs, each network terminates with a fully connected layer. Despite their shallow architecture, the Cropland Quantum Learning approach possesses fewer training parameters. what's more, when dealing with more intricate data and transitioning to deeper, more complex networks, the parameter count for quantum networks could exponentially decrease due to their unique encoding mechanism.

Metrics

In our study, a variety of metrics are employed to evaluate and compare the performance of hybrid networks with that of traditional neural networks. These metrics encompass the Area Under the Curve (AUC), the F1 score, and accuracy.

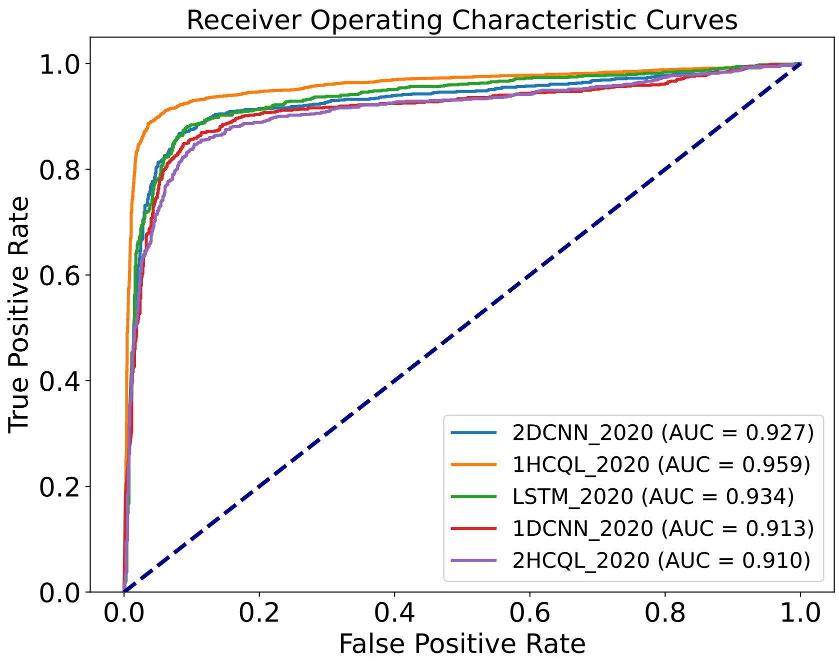


Figure 3.4: ROC curves of models on dataset 2020

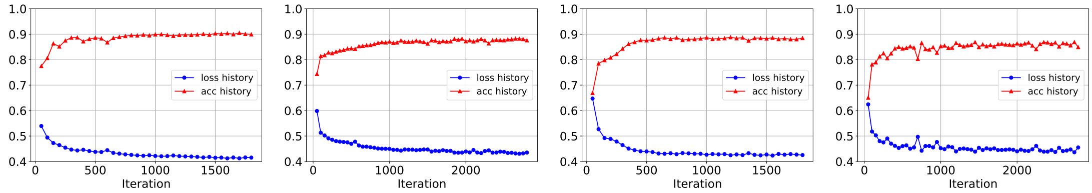


Figure 3.5: Validation accuracy and loss history of CQL Hybrid Model on dataset 2022 and 2022^a (From left to right, the performances of 1Head CQL on dataset 2022 and 2022^a, then 2Head CQL on these two datasets, respectively)

Among them, AUC represents the area under the ROC curve. It primarily quantifies the classifier's ability to rank a randomly chosen positive sample higher than a randomly chosen negative one. The AUC value ranges between 0.5 and 1.0, with values closer to 1.0 indicating superior performance. The F1 score is a crucial performance metric for finding a balance between precision and recall. For a balanced dataset, accuracy serves as an effective performance metric.

Evaluation

Our experimental results are summarized in Table 3.3. Within the original 2020 dataset, the 1Head CQL model outperforms others in terms of F1 score, AUC, and accuracy, surpassing both CNN and LSTM by two percentage points or more in each metric. We have specifically illustrated the AUC performance, considering its significance in binary classification tasks. As shown in Figure 3.4, it is evident that the 1Head CQL model exhibited the best performance on the 2020 dataset. Although the CQL model performance on the 2021 dataset does not achieve the best results, it also shows a result comparable to traditional CNN and LSTM models.

As the table shows, traditional networks fail to yield results on the original 2022 dataset. In fact, under this experimental setup, traditional models tend to overfit the relatively imbalanced 2022 data, failing to produce accurate test results. However, the CQL model demonstrates good convergence during the training process, as shown in Figure 3.5. Furthermore, the 1Head CQL model outperforms the 2Head version. For the oversampled 2022^a data, the 1Head CQL model also achieve the best performance in terms of F1 score and accuracy.

In summary, the hybrid models based on quantum machine learning, when applied to binary classification of agricultural land use, are able to match and often surpass traditional CNN and LSTM networks in most cases. A more significant finding is that, in this task, the CQL model performs better on imbalanced datasets compared to traditional networks. This indicates that quantum-based feature engineering can be more effective than conventional methods in certain specific datasets.

3.5 Discussion

Utilizing a hybrid neural network for cropland classification proves to be a promising and effective approach. The quantum-classical hybrid neural approach allows us to extract complex features from remote sensing data, capturing patterns and relationships between different features like traditional neural networks. Simultaneously, we conduct training and testing on multiple datasets, and the proposed method yields consistent performance. On certain datasets, it even exhibits slightly superior performance compared to results obtained from traditional deep learning feature mappings. However, the feature embedding approach using quantum-inspired techniques capitalizes on the distinctive properties of quantum bits, achieving comparable effectiveness within a network configuration that utilizes fewer parameters. This quantum-inspired approach opens new avenues and solutions for future applications involving larger-scale remote sensing data and more complex tasks. Importantly, while the current implementation involves quantum simulation on conventional hardware, the prospect of utilizing authentic quantum computers offers the potential to leverage quantum algorithms for accelerating classical

Table 3.3: Corpland Classification on 3 different years(10 epochs training)

Methods	Test Year	F1 Score	AUC	Acc(%)
1D CNN	2020	0.869	0.913	0.878
	2021	**0.921**	0.969	**0.931**
	2022	–	–	–
	2022[a]	0.762	**0.938**	0.878
2D CNN	2020	0.882	0.927	0.889
	2021	0.920	0.965	0.930
	2022	–	–	–
	2022[a]	0.756	0.925	0.872
LSTM	2020	0.886	0.934	0.894
	2021	0.917	**0.971**	0.928
	2022	–	–	–
	2022[a]	0.740	0.917	0.870
1Head CQL	2020	**0.911**	**0.959**	**0.920**
	2021	0.900	0.948	0.919
	2022	**0.780**	**0.920**	**0.903**
	2022[a]	**0.880**	0.935	**0.885**
2Head CQL	2020	0.890	0.910	0.885
	2021	0.915	0.966	0.924
	2022	0.748	0.917	0.890
	2022[a]	0.739	0.912	0.868

algorithms when processing traditional data. As we move forward, the incorporation of real quantum computation could lead to substantial advancements, ultimately bridging the gap between quantum-inspired methodologies and the untapped power of quantum computing. This promises to provide novel solutions for tackling intricate challenges posed by expansive remote sensing datasets and intricate tasks, transcending the capabilities of conventional deep learning approaches.

Chapter 4

A Quantum-Classical Hybrid Feature Mapping for Yield Prediction

Contents

4.1 Overview

Accurate yield prediction models are essential in modern agricultural technology for optimizing crop management and enhancing production efficiency. Meanwhile, quantum machine learning successfully enhances feature extraction in multiple disciplines. Considering the success of deep learning in yield prediction and quantum machine learning in various fields, this chapter introduces a hybrid feature extraction approach. This method aims to boost deep learning model performance in yield prediction by integrating quantum-extracted features. This method embeds quantum feature mapping modules into classical neural networks, enabling higher-dimensional feature extraction. Focusing on the United States, this study conducts quantum and classical feature extraction at the county level, considering weather, soil, and other static factors, to predict annual crop yields. Our method achieves prediction results that match or surpass the current deep learning methods.

4.1.1 Problem Statement

Yield prediction is an important process used across various industries to estimate future outputs based on historical data and influencing factors. This predictive ability is particularly significant in agriculture, manufacturing, and financial markets, where precise forecasting enhances decision-making and efficiency. For yield prediction, the methodology of yield prediction integrates sophisticated statistical methods and machine learning techniques. The data of yield prediction can vary widely depending on the specific domains including weather, soil, remote sensing data from satellites, market data, interest rates, and so on. Each of these sectors relies on different datasets that are critical for developing accurate predictive models. The evolution of yield prediction methodologies has seen significant advancements due to the integration of various machine learning techniques. Traditional methods primarily rely on statistical models like linear regression, which are simple yet powerful tools for understanding relationships between variables. However, the complexity of factors affecting yields is immense. Current research increasingly adopts deep learning in yield prediction because it can handle high-dimensional spatial and temporal data.

Quantum machine Learning integrates quantum computing with machine learning. This emerging file capitalizes on the unique properties of quantum mechanics, such as superposition and entanglement, to operate in high-dimensional spaces efficiently, leveraging these dimensions far beyond what classical computing can manage. Quantum machine learning demonstrates its capabilities across multiple domains, including healthcare, computer vision, and finance [40–44]. However, the limitation of quantum machine learning is primarily confined to labeled datasets and standardized tasks.

Benefiting from the advantages of Quantum Neural Networks (QNN) and Deep Neural Networks (DNN), we propose a hybrid quantum-classical feature mapping as a component of our feature engineering strategy aimed at achieving precise crop yield predictions. Our approach addresses the challenge of engineering features related to various crop yields. It utilizes both neural network (NN) methods and quantum neural network (QNN) approaches to perform feature engineering, integrating these features through channel concatenation before classical post-processing. This integration facilitates the subsequent regression tasks required for accurate yield forecasting. Our method underscores the potential of combining quantum and classical computational paradigms to enhance the accuracy and efficiency of predictive models in agricultural applications.

4.1.2 Main Contributions

This work intends to make the contributions as follows.

- We collect a variety of rich data from multiple sources, including weather data, surface feature data, soil data, and other static data that do not change over time. This diverse dataset allows us to comprehensively understand and analyze environmental and agronomic factors that influence crop yields, providing a robust foundation for our predictive models.
- We propose a hierarchical quantum-classical hybrid model to perform feature mapping on these diverse datasets. This innovative model processes and integrates various environmental and agronomic features, leveraging the unique computational advantages of both quantum and classical approaches.

4.1.3 The Outline of The Chapter

The remaining parts of the chapter are organized as follows.
Section 4.2 presents related works. Section 4.3 provides the overview of the proposed approach and detailed designs of each component. Section 4.4 comprehensively evaluates the proposed approach and compares the performance with the state-of-the-art baselines. Finally, section 4.5 concludes the Chapter.

4.2 Related Work

4.2.1 Yield Prediction

In yield prediction, many methods emerge, generally encompassing both traditional machine learning techniques and advanced deep learning methods. Traditional approaches often utilize algorithms such as lasso [45, 46], linear regression [47], decision trees [48], and support vector machines [49], which prove effective in handling structured data and clear-cut prediction tasks. With the increase in data dimensionality and diversity, deep neural network approaches are gradually becoming the preferred method. Convolutional Neural Networks (CNNs) [50] are employed to focus on structured features, while Long Short-term Memory Networks (LSTMs) [51] can be utilized to handle sequential features. Furthermore, a combined approach [52] leveraging the strengths of both can be adopted for enhanced performance. Some scholars have achieved notable success in yield prediction by mapping geographic information into topological data and harnessing Graph Neural Networks (GNNs) [53] to capture high-dimensional spatial and geographic information. Building upon graph embedding, they further enhance predictions by integrating diverse data and employing varied network architectures, resulting in significant advancements.

4.2.2 Quantum Machine Learning

Quantum Machine Learning (QML) [6, 7] emerges as a rapidly developing field that integrates the cutting-edge concepts of quantum computing and machine learning. Its core aim is to leverage the unique advantages of quantum computing to enhance and optimize the performance of machine learning algorithms. Methods such as Variational Quantum Circuits (VQC) [54] and Variational Quantum Eigensolvers (VQE) [55] play pivotal roles within this realm. Beyond VQC and VQE, QML extends into numerous other research areas, such as Quantum Support Vector Machines (QSVM) [56], Quantum Principal Component Analysis (QPCA) [21], and Quantum Autoencoders (QAE) [57], among others. These algorithms exploit quantum computing's characteristics to various extents to enhance traditional machine learning performance and solve problems that conventional methods struggle with. Among the research in QML, Quantum Neural Networks (QNNs) attract significant attention. By constructing parameterizable and trainable quantum circuits, QNNs introduce a new type of machine learning model. Compared to traditional neural networks, QNNs exhibit higher efficiency and better performance in certain tasks, such as complex tasks of image recognition and natural language processing, showcasing their unique advantages. Utilizing properties like superposition and entanglement of quantum states, QNNs more effectively handle high-dimensional data and nonlinear issues, thus improving the accuracy of classification and recognition.

Although a considerable amount of work based on QNN [22, 58–60] and hybrid methods [27, 61, 62] has been conducted in various directions, these efforts are mostly based on standard datasets or synthetic data sets. Applying quantum neural network-related methods to real-world tasks remains a current challenge.

4.3 Methodology

Our overall architecture of the hybrid mapping process is illustrated in Figure 4.1. The diagram illustrates a quantum-classical architecture where convolutional neural networks first process various data including temporal, soil, and extra inputs, which are then transformed through parallel classical and quantum feature engineering methods, and finally, the features are merged through channel concatenation to form a rich, unified dataset for subsequent analysis. Before we delve into a detailed description of our methods, we first abstract and formally define the problem.

4.3.1 Problem Formulation

As our goal is to predict annual yields at the county level, which means we need to transform the high-frequency daily updated data into a form that matches annual yield prediction for specific counties. To address this issue, we represent these time-dependent features as $X^T_{c,t} \in \mathbb{R}^{n_T \times Interval}$, where c represent the county level,t represents the year and T means these features are time-dependent. Soil-related features do not vary over time but are associated with soil depth. These features are expressed as $X^S_c \in \mathbb{R}^{n_S \times Depth}$. Similarly, c represents the county, but its dimensions are defined by zones and layers of the soil. In addition to the temporal features and soil depth features, there are also features related to crop yield, which we denote as X^E_C. These features are solely related to geographical location. For the groundtruth, county-level yields are represented as $y_{c,t}$, c, and t respectively representing the county and year. Thus, our objective becomes to use various features as inputs, with the output being the annual yield prediction for each county. Therefore, our question is reformulated as follows:

$$\hat{y}_{c,t} = F(X^T_{c,t}, X^S_c, X^E_c, y_{c,t})$$

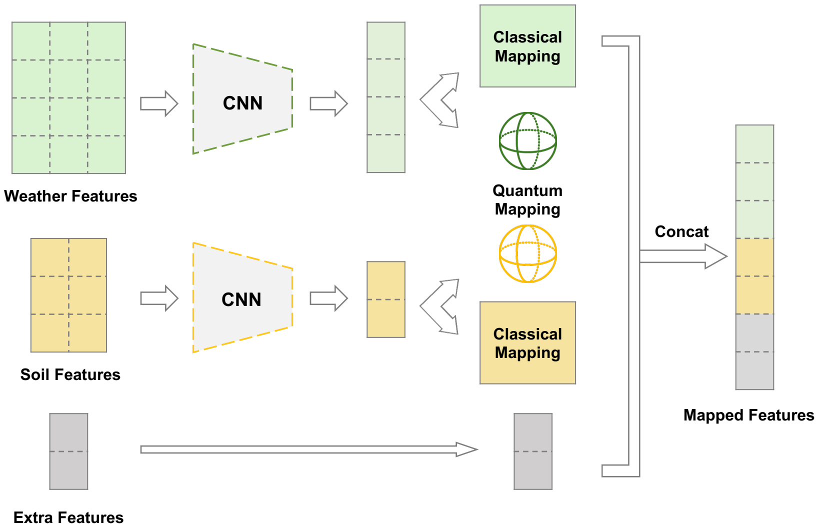


Figure 4.1: The overall Architecture of the hybrid Mapping Process.

In this context, F represents our entire nonlinear prediction methodology, functioning as a hybrid model designed for feature mapping and feature engineering. This model is then applied to accomplish the ultimate task of yield prediction.

4.3.2 First Level Classical Mapping

After obtaining raw data from various sources, we conduct preprocessing operations such as clarification, completion, and normalization. This preprocessed data then serves as the input for our model. For the first level classical feature mapping, we employ a classical network structure CNN. For time-series features, we restructure them based on the number of features and their respective time intervals. Similarly, attributes related to soil depth are reshaped according to the feature matrix and specific soil depths. The output of the first level classical feature mapping is $C_{c,t}^{T} = f_T(X_{c,t}^{T})$ and $C_c^S = f_S(X_c^S)$. Other static features remain unchanged at this stage. The function f_t is a CNN network designed to process temporally related weather, while f_s is another CNN employed to handle depth-related soil features.

At this level, CNN mapping not only maps the original features to a new space but also serves a major role in data dimensionality reduction. Even though the dimensionality of data undergoes an exponential decrease through quantum encoding in quantum neural networks, the original data involves more than ten quantum bits. Considering that we are currently simulating on classical devices, we initially utilize the characteristics of CNNs to reduce the dimensions accordingly.

4.3.3 Second Level Feature Mapping

From the previous level of mapping, we obtain the Ct and Cs features. The feature engineering at this level will follow a hybrid quantum-classical mode, where the classical secondary feature engineering is implemented through a simple neural network. Our focus, however, is on the quantum feature mapping segment. In this segment, to minimize training parameters and prevent gradient vanishing, our architecture is based on the VSQL model [22], comprising three main modules: the quantum encoding process, a parametrized quantum circuit for feature mapping, and observation to obtain the final quantum features.

Quantum Encoding: A widely used approach for encoding classical data into the quantum state involves associating normalized input data with the probability amplitudes of the quantum state. The encoding scheme is referred to as amplitude encoding(AE).

$$AE(x) : x \in \mathbb{R}^N \to |X\rangle = \frac{1}{\|x\|}\sum_{i=1}^{N} x_i|i\rangle$$

In this study, we focus on encoding the output features of the first level feature mapping using two encoding functions AE_S and AE_T, both of which utilize amplitude encoding mentioned above. If the number of original features is not a power of two, we typically encounter the need for padding during the amplitude encoding process. However, in our research, the input for encoding originates from the output of a Convolutional Neural Network, which facilitates the adjustment of the input to fit the powers of two. Therefore, we can perform feature engineering very efficiently, effectively avoiding the potential information loss or increased processing complexity that padding might introduce. So after the encoding, the features are represented at Bloch space, which is compact and low dimensionality. $|\phi_T\rangle = AE_T(C^T_{c,t})$ and $|\phi_S\rangle = AE_S(C^S_c)$ are the encoded feature which is the input of the parameterized quantum circuit.

Parameterized Quantum Circuit: For the quantum circuit in our study, we utilize the variational shadow quantum learning framework. The core idea behind the quantum circuit configuration is the use of a shadow circuit which is illustrated in Figure 4.2. In the diagram, Rz and Ry represent single-qubit gates that rotate around the z-

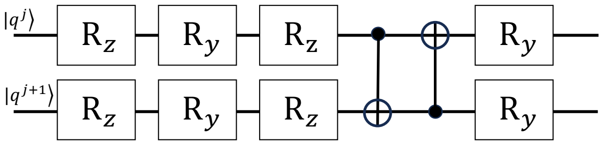


Figure 4.2: The shadow circuit of VSQL framework

axis and y-axis of the Bloch sphere, respectively, where the rotation angles serve as the parameters of these quantum gates. The $CNOT$ gate, on the other hand, is a two-qubit gate used primarily to entangle qubits. The Ry, Rz, and $CNOT$ are represented below.

$$R_y(\theta) := \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) & -\sin\left(\frac{\theta}{2}\right) \\ \sin\left(\frac{\theta}{2}\right) & \cos\left(\frac{\theta}{2}\right) \end{bmatrix} \; R_z(\theta) := \begin{bmatrix} e^{-i\frac{\theta}{2}} & 0 \\ 0 & e^{i\frac{\theta}{2}} \end{bmatrix} \; \text{CNOT} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{4.1}$$

Instead of processing the entire Hibert space, the VSQL uses the quantum circuit as a subspace feature extraction. In our implementation, we employ a shadow circuit approach, applying it between every two adjacent qubits, specifically between the ith and the $i+1$th qubits. By utilizing a sliding window technique, we progressively capture the corresponding shadow features, extending this process up to the final pair of qubits. The overall quantum encoding, quantum circuits, and observation architecture are shown in Figure **??**.

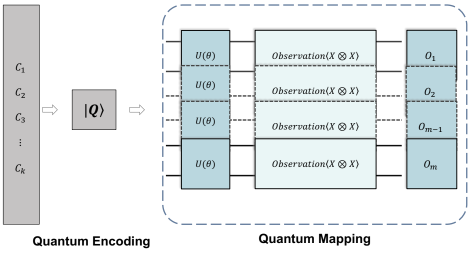

C_1
C_2
C_3
⋮
C_k
$|Q\rangle$
$U(\theta)$
$U(\theta)$
$U(\theta)$
$U(\theta)$
$Observation\langle X \otimes X\rangle$
$Observation\langle X \otimes X\rangle$
$Observation\langle X \otimes X\rangle$
$Observation\langle X \otimes X\rangle$
O_1
O_2
O_{m-1}
O_m
Quantum Encoding
Quantum Mapping

Figure 4.3: The process of quantum encoding and quantum mapping

In summary, quantum feature mapping achieves a process whereby classical inputs are encoded into a quantum state and subsequently processed via a series of quantum operations and measurements to yield outcomes $O^T = Q_T(|\phi_T\rangle, \theta)$ and $O^S = Q_S(|\phi_S\rangle, \theta)$ where Q_T and Q_S represent the quantum engineering parts for temporal features and soil features, respectively.

Channel Concatenation We have integrated the temporal information and the classical features of soil data obtained through hierarchical feature engineering with quantum features, by concatenating them across channels.

$$h_{integrated} = Concat(O^T, O^S, C^{T'}, C^{S'}, X^E)$$

$C^{T'}$ and $C^{S'}$ represent the classical features derived from second level feature engineering, while the addition feature X^E remains invariant.

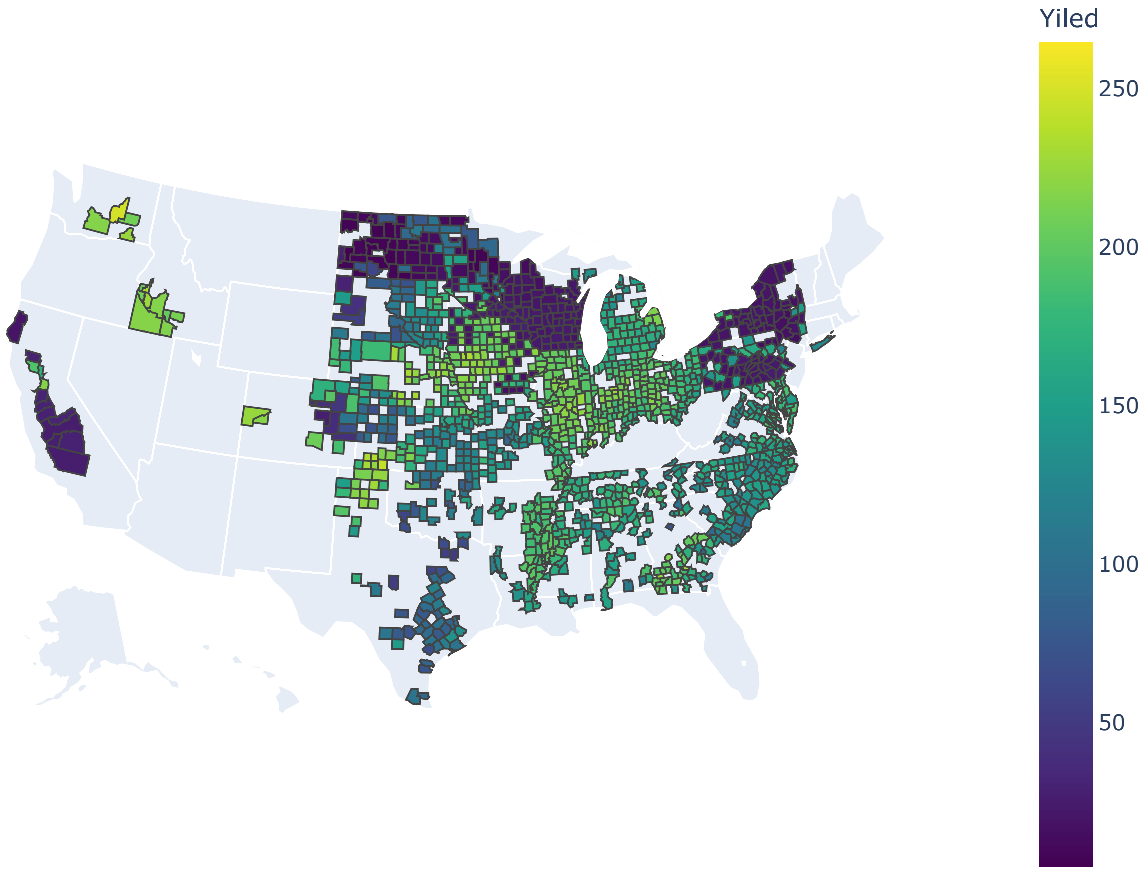


Figure 4.4: The USA county-level corn yield in 2021

After the hybrid quantum-classical feature engineering process, the high-dimensional feature vector $h_{integrated}$ extracted will be effectively integrated and transferred to a classical fully connected neural network. This network is designed to further process and refine the features, capturing and learning complex nonlinear relationships among them through densely connected layers of neurons, thereby enhancing the model's predictive capabilities. Additionally, the transfer of hybrid quantum-classical features to the fully connected neural network represents the final step in the integration of quantum and classical features. This fusion not only enhances the multi-dimensional expression of data but also improves the model's ability to recognize and process complex data patterns.

4.4 Experiments

As illustrated in Figure 4.4, the 2021 county-level data on corn yield per unit across the United States reveals that corn cultivation is predominantly concentrated in the central

Table 4.1: Performance Comparison of Different Methods on Various Datasets. RMSE: Lower values are better. R^2 and Corr: Higher values are better.

Method	Training Data	Test Year	Crop	RMSE	R^2	Corr
lasso	Saeed et al.	2018	soybean	0.66	0.59	0.76
	Saeed et al.	2018	corn	0.79	0.36	0.72
	Ours	2023	soybean	0.57	0.61	0.74
	Ours	2023	corn	0.68	0.38	0.67
Gradient-Boosting	Saeed et al.	2018	Soybean	0.62	0.55	0.77
	Saeed et al.	2018	corn	0.76	0.44	0.74
	Ours	2023	Soybean	0.59	0.55	0.76
	Ours	2023	Corn	0.73	0.21	0.68
CNN	Saeed et al.	2018	soybean	0.61	0.57	0.79
	Saeed et al.	2018	corn	0.66	0.52	0.74
	Ours	2023	Soybean	0.54	0.64	0.78
	Ours	2023	Corn	0.59	0.54	0.72
CNN w/ QM	Saeed et al.	2018	soybean	0.59	0.58	0.79
	Saeed et al.	2018	corn	0.63	0.54	0.72
	Ours	2023	Soybean	0.50	0.71	0.81
	Ours	2023	Corn	0.55	0.66	0.73

and eastern regions, with particularly high yields in southeastern counties. In contrast, there are very few areas with yield statistics in the western part of the country. This phenomenon is closely related to the wide geographic dimensions, climate diversity, and rich soil properties across the United States. Thus, the yield of crops is closely related to factors such as geographical location, climate diversity, and soil characteristics. So we obtain various data from different channels and sources. These data include weather data, land surface data, soil data, annual yields of different crops, and some other crop-related data, spanning from 2000 to 2023. The following is a detailed description of these dataset sources.

Climate Data: For some weather and surface data, we obtain them from the open-source dataset ERA5-Land Daily Aggregated - ECMWF Climate Reanalysis [63], available on Google Earth Engine. This dataset, provided by the European Centre for Medium-Range Weather Forecasts (ECMWF), is a comprehensive collection of daily aggregated land surface climate data. This dataset encompasses a wide range of variables such as temperature, precipitation, humidity, wind speed, and radiation, among others. It offers high-resolution, global coverage spanning multiple decades, making it valuable for various environmental and climate-related studies. The data is derived from the ERA5-Land reanalysis system, which integrates observations from multiple sources using advanced data assimilation techniques. This dataset is widely used by researchers, policymakers, and practitioners for climate monitoring, modeling, and impact assessment purposes.

Soil Data: In our study, we source soil information from two primary data repositories. The first is the SoilGrids250m dataset [64] from Google Earth Engine, which offers 11 different soil properties, each comprising six data layers. However, for the property of organic carbon density, only a single data layer was available, leading us to treat this as static data. The second source is the Gridded Soil Survey Geographic Database (gSSURGO) (Soil Survey Staff, 2023) [65], which contains soil features that do not change over time and are available as a raster map layer in a 10-meter resolution. We collected several attributes relevant to crop yield, some of which are depth-dependent and thus have values for 6 depth levels, such as soil organic carbon and available water storage, while others are not depth-dependent, such as the national commodity crop productivity index. Each variable is aggregated to a county level by computing the weighted average over all plant-covered grid cells in a single county. The plant-covered grid cells are defined as the total area of herbaceous, hay/pasture, and cultivated crops, computed using the National Land Cover Database [66].

Yield Data: We retrieve the crop yield data from the USDA dataset, provided by the United States Department of Agriculture, which encompasses a wide range of agricultural and environmental data about the United States. This dataset includes information on crop yields, land use, soil properties, climate conditions, agricultural practices, and more. It serves as a valuable resource for researchers, policymakers, farmers, and other stakeholders involved in agriculture and environmental management. The USDA dataset is widely used for various purposes such as crop yield prediction, land management, agricultural policy development, and environmental monitoring. It provides comprehensive insights into the agricultural landscape of the United States and supports informed decision-making in the agricultural sector.

Other Data: In addition to data that varies with time and soil depth, we also have some static data. These static data include soil-related information as well as crop-related National Commodity Crop Productivity Index (NCCPI) data.

According to Saeed's [52] work, we adjust our time-series features by converting daily observations to weekly averages, thereby reducing data redundancy. This dataset includes 24 weather features such as temperature, humidity, soil temperature, and wind speed. Therefore, with data aggregated over 52 weeks, the final dimensionality of these features is 24 multiplied by 52. Additionally, the soil characteristics incorporate 10 properties from SoilGrids250m and three properties from the gSSURGO dataset, each with six bands. Hence, the total dimensionality of the soil features is 13 properties each with six bands. Additionally, our dataset includes extra features related to specific crops, such as the National Commodity Crop Productivity Index (NCCPI) for individual crops and the total NCCPI, along with drought indices.

4.4.1 Metrics

As our task is regression-related work. We assess all methods using three widely-recognized regression metrics: root mean square error (RMSE), the coefficient of determination R^2, and Pearson correlation coefficient (Corr) where inspired by Fan's [53] work. RMSE and R^2 indicate the regression model's predictive accuracy in absolute and percentage terms, respectively. Corr fundamentally represents a normalized measure of covariance between two data sets and quantifies the strength of linear correlation between actual and predicted values.

4.4.2 Baselines

We employ a variety of methods to conduct in-depth analyses on data collected between 2000 and 2023, and we perform validation and testing on the datasets on 2023. The techniques we adopt include Convolutional Neural Networks (CNNs), In addition, we explore machine learning methods such as Lasso regression and Gradient Boosting. Furthermore, we experiment with combining quantum features with CNN and conduct deep analyses of feature engineering using hybrid models. In addition, we conduct experiments on the publicly available dataset by Saeed et al. [52], referred to hereafter as the "Saeed et al." Based on their soybean data available up to 2018, we further integrated corn yield data from the same year and conduct tests on yield 2018.

4.4.3 Experiment Set-up

All our work is conducted on uniform equipment and environment, specifically utilizing hardware equipped with a single GPU, the INTEL UHD Graphics 630. The deep learning experiments are carried out using PaddlePaddle [39], while the construction of quantum circuits is performed in the Paddle Quantum [24] environment, both developed by Baidu.

4.4.4 Hyper Parameters

Regarding hyperparameter settings, we applied each method to various datasets, with each set of experiments running for 50 epochs. The Adam optimizer [37] is utilized, with a learning rate set at 0.001 and a weight decay parameter of 1e-5. This configuration is aimed at balancing training speed and model performance, ensuring stable and reproducible results across multiple training iterations.

4.4.5 Resutls and Disscussion

Table 4.1 presents the results of all experiments conducted. We select two machine learning models—Lasso and Gradient Boosting—along with a CNN model, to serve as the baseline for our quantum-classical hybrid models. The results indicate that, from the perspective of crop type, all models perform better on soybean datasets than on corn datasets. From the model perspective, the CNN and the models incorporating quantum mapping outperformed traditional machine learning models. Notably, our hybrid model featuring a novel feature mapping approach via CNN achieved the best performance across numerous datasets and evaluation metrics. Different crops may have significantly different feature distributions, data densities, and noise levels related to the prediction targets. Soybean datasets in these experiments might exhibit more regular or consistent feature patterns, which makes it easier for models to capture and learn relevant patterns. In summary, in exploring the integration of quantum and classical features in machine learning models, we propose a novel approach that leverages the strengths of both types of features to achieve richer information.

4.5 Conclusion

In our research, we collect historical data related to crop yield across various dimensions and aspects and engage in feature engineering that combines classical and quantum techniques. This unique approach enables us to connect features from different data spaces, optimizing data representation and enhancing the performance of predictive models. Our methodology is innovative in theory and has proven effective in practical application. Our results show comparable or superior performance by comparing our hybrid feature engineering strategy with traditional machine learning methods and single Convolutional Neural Network (CNN) approaches. This highlights the potential of quantum computing to enhance classical data processing techniques and demonstrates its prospects in addressing real-world complex problems, such as crop yield prediction.

Chapter 5

Quantum Random Walk on Graph

Contents

5.1 Overview

In recent years, Graph Neural Networks (GNNs) have made significant strides in various applications, demonstrating their potential in handling complex networked data. Simultaneously, quantum machine learning has emerged as a rapidly advancing and promising field, leveraging quantum computing principles to enhance machine learning models. Benefiting from the advancements in both GNNs and quantum machine learning, we propose a novel hybrid model called CTQW-GraphSAGE. This model aims to combine the strengths of classical and quantum approaches to improve performance on graph-related tasks. The model is built on the GraphSAGE framework, enhanced with quantum feature mapping and Continuous-Time Quantum Walk (CTQW). These enhancements are used to calculate aggregation weights for neighboring nodes relative to the target node, thereby integrating quantum properties into the classical model. We evaluate the proposed model on various benchmark datasets and compare our results with several baseline graph neural network methods. CTQW-GraphSAGE achieves comparable results to the classical models on most of the selected datasets on node classification tasks.

5.1.1 Problem Statement

Graph-structured data analysis has become an indispensable tool in modern data science, especially Graph Neural Networks (GNNs) have achieved great success across various domains, including social networks, biological networks, knowledge graphs, and more [67–69]. GNNs are designed to perform computation on graphs, enabling the model to learn and generalize from topological structures. The fundamental operation in GNNs is the message passing or neighborhood aggregation mechanism. In this framework, nodes aggregate feature information from their neighbors, allowing the model to capture both local structures and global graph properties. This iterative process involves updating the representation of each node by aggregating representations of its neighbors and potentially its features. Then, undertake different downstream tasks fit to different domains and scenarios.

However, the traditional neighborhood aggregation strategy only considers the neighboring nodes of the target node, ignoring the more complex topological information. Moreover, the aggregation methods based on pooling and mean represent a fixed-parameter approach, involving no trainable parameters during the aggregation.

To preserve the complex topological information based on target nodes during graph analysis, this chapter introduces a k-hop sampling method. This approach extends beyond merely considering a node's immediate neighbors to include structural information with k-hops in the graph. A weighted aggregation that leverages both the topological information and the node features of the graph is then employed to enhance the adaptability and flexibility of the aggregation strategy. The weights for the aggregation are derived using Continuous-Time Quantum Walk (CTQW) within the graph. Furthermore, a variational quantum circuit is utilized to generate parametric embedding, thereby transforming it into a trainable Continuous-Time Quantum Walk.

This chapter introduces a quantum-classical hybrid model with an aggregation method based on Continuous-Time Quantum Walk, the CTQW-GraphSAGE. This method aims to aggregate more complex topological information between nodes, not just relying on direct neighbors, and utilizes a parametrically embedded CTQW for weighted information aggregation.

5.1.2 Main Contributions

Our main contributions are as follows:

- We utilize a novel approach to graph sampling that emphasizes the hop-based selection of nodes, diverging from traditional methods that primarily consider neighbors based on their layer-wise proximity. This strategy is designed to capture and preserve the intricate local topological structures surrounding a target node, which are often crucial for understanding the node's role and characteristics within the broader network.

- We employ Continuous-Time Quantum Walk (CTQW) on the subgraph to obtain corresponding aggregation probabilities. Instead, we incorporate the similarity of features between different nodes into the Hamiltonian. This inclusion significantly enhances the representation of node feature aggregation.

- A variational Quantum Circuit (VQC) is utilized to create parametric embedding of node features. This strategic implementation allows for the weights, which are generated through the CTQW process within the graph, to be optimizable. It makes the weights not only dynamic but also learnable to enhance the overall efficacy of the information aggregation process in the graph.

5.1.3 The Outline of This Chapter

The remaining parts of this chapter is organized as follows. Section 5.2 presents related works. Section 5.3 provides the overview of the proposed approach and detailed designs of each component. Section 5.4 comprehensively evaluates the proposed approach and compares the performance with the state-of-the-art baselines. Finally, section 5.5 concludes the chapter.

5.2 Related Work

5.2.1 Graph Neural Network

In recent years, graph neural networks (GNNs) [9] have emerged as a powerful tool for processing and learning from graph-structured data. Previous approaches using MLP-like [70, 71] networks focused primarily on feature information, ignoring the characteristics of the graph structure itself. A significant leap was the development of Graph Convolutional Networks (GCN) [72], which is simplified graph convolutional using efficient layer-wise propagation rules based on adjacency and degree matrices. This was followed by Graph Attention Networks (GAT and GATII) [73, 74] that introduce an attention mechanism, assigning dynamic importance to neighbors. The difference between spectral and spatial approaches in GNNs further broadened the scope of the field. Spectral methods [75, 76] leverage the graph Laplacian's eigenvalues for convolution, offering a mathematical approach. On the other hand, GraphSAGE [77] directly aggregates neighbor features, addressing scalability and enabling inductive learning on large graphs. The challenge of heterogeneous and dynamic graphs led to the development of specialized models like Heterogeneous GNNs (HetGNNs) [78] and Dynamic GNNs (DGCNNs) [79], catering to diverse node types and evolving graph structures. The concept of Message Passing Neural Networks (MPNNs) [80] unified various GNN architectures under a single framework, streamlining the understanding of how GNNs update node features. Besides these, DeepWalk [81] utilizes random walks on the graphs and treats each node visit in a walk as a word. For reducing the computational complexity, SGC [82] removes the non-linearities between layers and replaces multiple layers with a single line classifier.

The applicability of GNNs extends to various domains. For instance, in social network analysis, GNNs are used for community detection and recommendation systems. In bioinformatics, they assist in protein-protein interaction predictions, and in computer vision, GNNs are applied in scene graph generation and object detection.

5.2.2 Quantum Machine Learning

Simultaneously, Quantum Machine Learning (QML) as an emerging interdisciplinary field at the intersection of quantum computing and machine learning has developed rapidly over the years. Numerous machine learning algorithms have been realized into quantum versions [20, 83, 84], offering acceleration for classical problems. Moreover, many scholars and researchers have harnessed the advantages of quantum computing and integrated them with traditional neural networks, constructing a variety of hybrid models for different domains.

The field of Quantum Machine Learning (QML) [7] has witnessed significant growth and innovation over the past decade. Variational Quantum Eigensolvers (VQEs) [85] have emerged as the fundamental quantum algorithm. Quantum Support Vector Machines (QSVM) [19] aims to provide quantum speedup in classification tasks as the analogs of classical support vector machines. Quantum algorithms for feature selection, such as Quantum Principle Component Analysis (QPCA) [21], aim to reduce dimensionality and improve the efficiency of classical machine learning algorithms.

The fusion of quantum computing and machine learning techniques has extended to graph data, presenting novel solutions for graph-related problems. Quantum Graph Neural Networks represent a significant development, combining the power of Quantum Computing and Graph Neural Networks (GNNs). QGNNs aim to perform graph-based machine learning tasks, such as node classification, link prediction, and graph classification, with quantum-enhanced capabilities. Notably, Verdon et al. [86] pioneered the development of the Quantum Graph Neural Network (QGNN), a quantum computing-based model for graph classification. QGNN leverages a quadratic Hamiltonian to encode graph structures and employs quantum circuits to extract pertinent structural details. Though Peter et al. introduced the Equivalent Quantum Graph Circuit (EQGC) [87]. EQGC excels at capturing permutation-invariant topologies of input graphs. Its scalability is constrained, as the number of required qubits scales linearly with the number of nodes, restricting its applicability to small-scale synthetic datasets. A notable breakthrough in quantum graph algorithms is the Graph Quantum Neural Tangent Kernel (GraphQNTK) [88], which stands as the sole quantum algorithm capable of handling realistically sized graph data. For utilizing the advancement of quantum computing, ego-graph based Quantum Graph Neural Network (egoQGNN) [89] greatly reduced the trainable parameters compared to the traditional GNNs.

In summary, within the domain of combining Graph Neural Networks (GNNs) and Quantum Machine Learning (QML), numerous networks have achieved success. How-

Table 5.1: Notations and Functions

Symbols	Definition or Meanings
k	The maximum sampling hop
m	The number of k-hop samples for the target node
$N_k^m(v)$	A sampled nodes set of target node v
X_v	Features sets of $N_k^m(v)$
ρ_v	The feature mapping after parametric quantum embedding.
P_v	Probablities of N_v after quantum walk
$\odot$	Hadamard Product (element-wise product)

ever, many of these networks have not effectively harnessed the power of Quantum Machine Learning to simultaneously leverage both feature information and graph structure. This article introduces a novel approach that utilizes Quantum Machine Learning for feature mapping and employs Quantum Random Walks for probability computation. This innovative approach has demonstrated excellent results in node classification tasks, showcasing the potential of quantum-enhanced techniques in graph-based machine learning.

5.3 Methodology

5.3.1 Preliminaries

A graph G can be described using two matrices: an adjacency matrix A, where $A \in \{0,1\}^{n\times n}$, and a node feature matrix X, where $X \in \mathbb{R}^{n\times d}$. In these matrices, n represents the total number of nodes, and d signifies the dimensionality of the node features. Within the adjacency matrix A, an entry $A[i,j] = 1$ indicates the presence of an edge between nodes v_i and v_j; conversely, $A[i,j] = 0$ implies the absence of an edge. Based on the definition of a graph, some additional notations and explanations that are utilized in subsequent processes are presented in Table 5.1.

5.3.2 Framework

Before the theoretical foundation of the method, this subsection provides an overall introduction to the entire network structure of the method we propose. The architecture of the system is illustrated in Figure 5.1.

The whole process depicted begins with node sampling to extract a subgraph, from

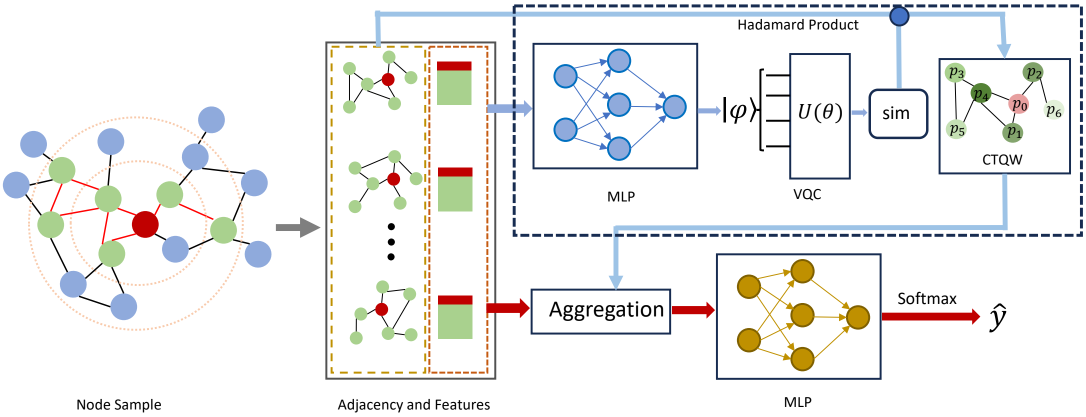


Figure 5.1: The overall architecture of CTQW-GraphSAGE.

which the adjacency and feature matrices are obtained. The features undergo a hybrid quantum-classical transformation. The classical component consists of a multilayer perceptron (MLP), which functions to reduce dimensionality. In parallel, the quantum component comprises a variational quantum circuit (VQC), which encodes classical data into a quantum state and performs a quantum feature mapping through the circuit.

The mapped features are then used to calculate similarity measures, which are combined with the sampled adjacency matrix through a Hadamard product. This product result serves as a Hamiltonian operator to govern the time evolution in a continuous-time quantum walk (CTQW), resulting in a probability distribution. The probabilities represent the expectation of the walker landing on each node within the subgraph.

Leveraging this probability distribution, feature aggregation is performed for the target node. The process concludes with a classical MLP followed by a softmax classifier, which collectively serves to classify the nodes based on the aggregated features. The methodology shows a blend of classical machine learning techniques and quantum computing principles to enhance the analysis of graph-structured data.

5.3.3 Detail Description

In this subsection, we focus on the critical aspects of the network, including node sampling, the implementation of quantum mapping embedding with VQC, the Continuous-Time Quantum Walk operating on the graph, and feature aggregation.

Sampling: Unlike the layer-wise random sampling in GraphSAGE, where the number of neighbors for each layer and each target node is fixed to ensure consistency and controllability at each layer, the proposed sampling method involves sampling within different hops. The sampling process also retains neighbor information among nodes

Algorithm 2 k-hop Nodes Sampling

Input: Graph: $\{G = (V, E), |V| = N\}$; Node Features $X = \{x_v | \forall v \in V\}$, sample nodes: m; sample hops: k

Output: Adjacency Matrix A_v and Features Matrix X_v of subgraphs.

```
Initialize node set $N_k^m(v)$
for $i$ in range $(1, k)$ do
    Find the $i$-hop neighbours of node $v$ in $G$, add these nodes to the temporary set $T(v)$
end for
Select $m$ nodes from $T(v)$ and set them as $N_k^m(v)$
Generate adjacency matrix $A_v$ and features matrix $X_v$ for $N_k^m(v)$
return $N_k^m(v)$, $A_v$, $X_v$
```

other than the target node, trying to preserve local structures and complexities. The corresponding adjacency matrix of these nodes and their features is obtained through this sampling, as shown in Algorithm 2.

The objective of sampling is to obtain a set of nodes from the Graph $G(V, E)$ which consists of vertices V and edges E for each target node v in the vertex set V, the k-hop neighborhood is defined as follows:

$$N_k^m(v) = \{u \mid d(v, u) \leq k\} \tag{5.1}$$

where $d(v, u)$ is the shortest path distance between nodes v and u. From this k-hop neighborhood, a subset of $N_k^m(v)$ m nodes is randomly selected.

The proposed sample algorithm implies that $N_k^m(v)$ includes the target node v, its immediate neighbors, and all other nodes that can be reached by traversing up to k edges. The k-hop sampling strategy is valuable for capturing a comprehensive local context around each node, as it not only considers the direct connections (1-hop neighbors) but also includes nodes that are indirectly connected within k steps. After the sampling, we treat the corresponding adjacency matrices A_v and feature set X_v as the input of Trainable Continuous-Time Quantum Walk ,which includes a variable quantum circuit (VQC) and an evolution operator.

Trainable Continuous-Time Quantum Walk: The Trainable Continuous-Time Quantum Walk (T-CTQW) is composed of two parts. One is a hybrid quantum-classical part, which consists of an MLP and a variational quantum circuit. The other one is a Continuous-Time Quantum Walk (CTQW). The overall illustration of a Trainable Continuous-Time Quantum Walk is presented in Algorithm 3.

For each node v in the graph node set V, the quantum state $|\psi_0\rangle$ is prepared in a way

Algorithm 3 Trainable Continuous-Time Quantum Walk

Input: Adjacency matrices A_v, Features matrices X_v and evolution time t
Parameter: Parameters of Variable Quantum Circuit and MLP θ, ω_q.
Output: Probabilities

```
Initialize $\theta$, $\omega_p$ and set $t$
for each node $v$ in $V$ do
    Initialize the quantum state $\psi_0$ by setting the target node $v$ state as $|1\rangle$ and other nodes $u$ states as $|0\rangle$
    Features dimensions reduction by MLP: $X_v \mapsto X_v^{'}$
    Encode features to quantum state $X_v^{'} \mapsto |X_v^{'}\rangle$
    Feed the re-encoded features to VQC $|X_v^{'}\rangle \mapsto |\rho_v\rangle$
    Measure and get the output $|\rho_v\rangle \mapsto \rho_v$
    Calculate the similarity matrix $S_v = sim(\rho_v)$
    Continuous-Time Quantum Walk evolution: $|\psi_t\rangle = e^{-iS_v \odot A_v t} |\psi_0\rangle$
    Calculate the probabilities vector $P_v$
end for
return all $P_v$
```

that the target node v is set $|1\rangle$, and all other nodes u are set to $|0\rangle$. Feature reduction is then performed on the feature matrices X_v using an MLP, effectively compressing the data before it is encoded into a quantum state. The reduced features $X_v^{'}$ undergo amplitude encoding, transforming them into a quantum state $|X_v^{'}\rangle$. These quantum-encoded features are fed into the VQC, which processes them and generates an output quantum state mapping ρ_v, and it is illustrated in Figure 5.2.

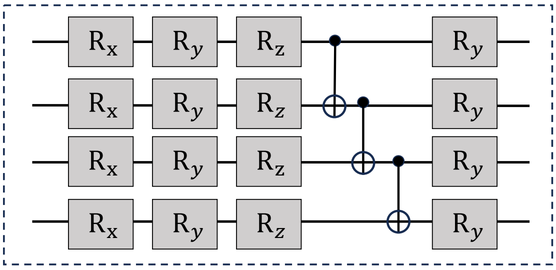


Figure 5.2: The VQC architecture.

For the specialized VQC, which is constructed using a combination of specific quantum gates. These gates include Rx(Pauli-X rotation gate), Ry (Pauli-Y rotation gate), and Rz (Pauli-Z rotation gate), which are each responsible for rotating a qubit around their respective axes on the Bloch sphere. Additionally, the circuit incorporates CNOT (Controlled NOT) gates, which are utilized for creating entanglements between qubits, a fundamental characteristic of quantum computing.

Each qubit initially passes through a sequence of parameterized rotation gates and then proceeds through an entanglements section composed of several Controlled-Not gate (CNOT), followed by an additional layer of rotation gates to enhance expressivity. The VQC is represented as $U(\theta)$. Then the mapping process can be described as $|\rho_v\rangle = U(\theta)\,|\rho_v\rangle$. After the measurement $\rho_v = \langle\rho_v|\,M\,|\rho_v\rangle$ where M represent the observation computation.

In a traditional CTQW, the walker's evolution is governed by the system's Hamiltonian $\mathcal{H}$, which is typically related to the adjacency matrix or Laplacian matrix of the graph or lattice on which the walk is taking place. The state of the walker $|\psi\rangle_t$ at time t is given by the solution to the Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}|\psi(t)\rangle = \mathcal{H}|\psi(t)\rangle \tag{5.2}$$

Where the Hamiltonian is the adjacent matrix. Solving the Schrödinger equation with $\hbar = 1$, we get the state of the quantum walker at time t:

$$|\psi_t\rangle = e^{-i\mathcal{H}t}\,|\psi_0\rangle \tag{5.3}$$

This work considers more than just the topological structure of the graph. We aim to provide a richer, higher-dimensional representation that encompasses both the graph's topological information and the connections based on node features. Therefore, the Hamiltonian operator is revised to be the Hadamard product of the adjacency matrix and the feature similarity computed by the output of the VQC.

$$\mathcal{H} = sim(\rho_v) \odot A_v. \tag{5.4}$$

Equation 4 shows that the Hamiltonian matrix is constructed by first calculating the similarities between specific features mapped by VQC. Then, we mask these similarity matrices using the adjacency matrices to eliminate expressions of similarity between nodes that are not connected.

To determine the probability of finding the quantum walker at a particular node u, which is one of the nodes in the subgraph based on the target node v at time t, we project the time-evolved state onto the basis state $|u\rangle$ corresponding to the node. The probability $P_u(t)$ is calculated as the absolute square of the amplitude:

$$P_u(t) = |\langle u\,|\psi(t)\rangle\,|^2 \tag{5.5}$$

This process is repeated for each node in the subgraph to obtain the complete probability distribution across all nodes P_v.

In summary, the process of CTQW translates the classical information of a graph, including the connectivities between nodes and their features, into weights used for the aggregation of graph features. This transformation effectively captures the structure and attributes of the graph within a quantum framework, facilitating computations that leverage quantum properties such as superposition and entanglement. The specific mathematical expression of this process can be detailed as follows:

In practical terms, the sum of the probabilities for all nodes should be equal to 1,

$$\sum_u P_u(t) = 1, u \in N_k^m(v), \tag{5.6}$$

reflecting the normalization condition and ensuring that the quantum walk describes a complete and physically possible quantum system.

This rule is not only a fundamental principle of quantum mechanics but also a key condition that must be adhered to when designing and interpreting quantum algorithms in quantum computing. In practical quantum computing applications, ensuring the normalization of quantum states is crucial, as it guarantees the correctness and reliability of quantum computational results.

Feature Aggregation: In the feature aggregation part, we leverage the probabilities derived from continuous quantum random walks to present a novel approach to synthesizing enriched node representations in graphs. This aggregation method takes into account not only the direct connections of nodes or the topological structure of the graph but also fully integrates the interconnections among features

$$\begin{gathered} h_v^l = \sigma(W(\cdot\{h_v^{l-1}\} \cup \{WeightedMean(h_u^{l-1}, P_u)\})) \\ (\forall u \in N_k^m(u), u \neq v) \end{gathered} \tag{5.7}$$

The equation describes the process of updating the hidden state of node v at layer l in the neural network. The hidden state h_v^l is computed as a function of the node's previous layer state h_v^{l-1} and the weighted features of its neighboring nodes. The activation function σ introduces nonlinearity. W represents the weight matrix for the linear transformation of node features. The sum is taken over all neighboring nodes u of v, where each neighbor's contribution to the new feature representation is weighted by the probability P_u, derived from processes like quantum random walks. This formulation captures both the direct connections of the nodes and the topological structure of the

Table 5.2: Datasets Details

Dataset	# Nodes	# Edges	# Features	# Labels
Cora	2708	5429	1433	7
Pubmed	19717	44388	500	3
Citeseer	3327	4732	3703	6
Wisconsin	251	499	1703	5
Texas	183	309	1703	5
Cornell	183	295	1703	5

graph, integrating the relationships among features for a richer representation. In addressing the node classification task, after the described feature aggregation process, we implement a conventional post-processing methodology. This involves the use of a multilayer perceptron integrated with a softmax function to categorize the nodes.

5.4 Experiments

This section presents the evaluation results of the proposed CTQW-GraphSAGE on various datasets and compares them against the state-of-the-art baselines.

5.4.1 Datasets

Six datasets, Cora, PubMed, Citeseer, Wisconsin, Texas, and Cornell, are chosen to evaluate the proposed approach. Table 5.2 contains detailed statistical information about these datasets, and a brief introduction about each dataset is provided below.

Cora: This is a citation network dataset where nodes represent scientific papers, and edges represent citations between these papers. The papers are classified into different classes based on their topics, such as Machine Learning, Neural Networks, etc. Each paper (node) is described by a binary word vector indicating the absence/presence of the corresponding word from the dictionary.

PubMed: Similar to Cora, the PubMed dataset is also a citation network but focuses on diabetes-related publications from the PubMed database. Each paper in the dataset is described by a TF/IDF weighted word vector from a dictionary. The papers are categorized into three classes based on the type of diabetes they discuss.

Citeseer: This is another citation network where nodes represent publications and edges represent citations. Like Cora, each document is associated with a word vector indicating the presence of words from a dictionary. The papers are categorized into different classes based on their research topics.

Wisconsin, Texas, and Cornell: These three datasets are part of the WebKB dataset collection and consist of web pages from computer science departments of various universities, categorized into classes like faculty, student, project, etc. Each webpage is described by a word vector similar to the Cora and Citeseer datasets. The datasets represent a network where edges are hyperlinks between these web pages.

Table 5.3: Performance for the node classification task on six datasets (mean accuracy(%) and standard deviation over 10 trials), In each dataset, the best performance is highlighted in **bold**, while the second-best is underlined.

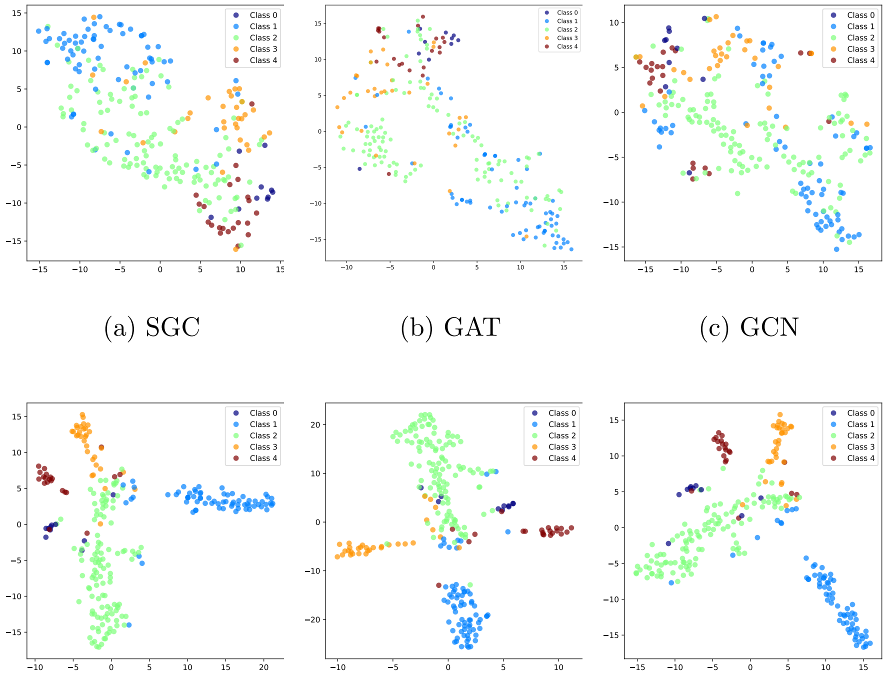
Methods	Cora	Citeseer	Pumbed	Cornell	Wisconsin	Texas
DeepWalk	82.32$_{\pm 0.72}$	60.78$_{\pm 1.11}$	61.25$_{\pm 1.30}$	50.81$_{\pm 5.37}$	50.39$_{\pm 5.48}$	47.57$_{\pm 3.01}$
Graph-based Method (2-layers)						
GCN	87.39$_{\pm 0.27}$	76.5$_{\pm 0.42}$	87.24$_{\pm 0.25}$	58.65$_{\pm 2.1}$	44.22$_{\pm 1.2}$	44.59$_{\pm 1.36}$
GAT	87.08$_{\pm 0.48}$	77.54$_{\pm 0.40}$	87.28$_{\pm 0.28}$	59.73$_{\pm 1.89}$	46.86$_{\pm 3.33}$	50.27$_{\pm 2.16}$
GraphSAGE-mean	87.50$_{\pm 0.31}$	77.21$_{\pm 0.42}$	87.45$_{\pm 0.41}$	75.96$_{\pm 3.5}$	72.94$_{\pm 1.92}$	**71.81$_{\pm 2.36}$**
GraphSAGE-pool	86.71$_{\pm 0.55}$	75.60$_{\pm 0.54}$	86.68$_{\pm 0.38}$	71.89$_{\pm 2.76}$	69.22$_{\pm 3.40}$	68.91$_{\pm 2.18}$
CTQW-GraphSAGE	**87.89$_{\pm 0.95}$**	**78.04$_{\pm 0.28}$**	**88.60$_{\pm 2.58}$**	73.64$_{\pm 3.08}$	**74.35$_{\pm 2.40}$**	70.84$_{\pm 3.85}$
Graph-based Method (1-layer)						
SGC	84.20$_{\pm 0.27}$	77.04$\pm$0.17	85.20$_{\pm 0.45}$	60.00$_{\pm 1.08}$	45.29$_{\pm 0.59}$	51.89$_{\pm 1.08}$
GCN	85.45$_{\pm 0.52}$	77.32$_{\pm 0.28}$	86.64$_{\pm 0.40}$	61.08$_{\pm 1.79}$	45.29$_{\pm 0.59}$	49.46$_{\pm 1.73}$
GAT	84.23$_{\pm 0.72}$	77.66$_{\pm 0.31}$	85.18$_{\pm 0.49}$	63.51$_{\pm 3.25}$	47.84$_{\pm 3.19}$	53.24$_{\pm 2.72}$
GraphSAGE-mean	83.93$_{\pm 0.26}$	77.30$_{\pm 0.18}$	84.89$_{\pm 0.31}$	**77.29$_{\pm 2.16}$**	72.75$_{\pm 1.84}$	70.54$_{\pm 0.81}$
GraphSAGE-pool	83.02$_{\pm 0.40}$	77.74$_{\pm 0.26}$	84.15$_{\pm 0.49}$	73.24$_{\pm 2.25}$	65.50$_{\pm 1.80}$	67.57$_{\pm 2.09}$
CTQW-GraphSAGE	85.47$_{\pm 0.35}$	76.78$_{\pm 0.90}$	86.93$_{\pm 0.86}$	73.47$_{\pm 3.40}$	69.68$_{\pm 2.81}$	69.78$_{\pm 2.12}$

5.4.2 Baselines

To evaluate the performance of our model on different datasets, we implement several state-of-the-art models, including Simplifying Graph Convolutional Networks (SGC), DeepWalk, Graph Convolutional Networks (GCN), Graph Attention Networks (GAT), and GraphSAGE with pooling and mean aggregation, and compare their performance on the above datasets with our proposed approach.

5.4.3 Experiments set-up

We assess and compare our model based on the accuracy achieved in node classification tasks. To ensure fairness and credibility, we conduct ten trials for each model on every dataset and the accuracy report is based on the average accuracy. For each trial, we run the model 100 epochs. For the neural networks method, we use Adam optimizer. The setting of the hyperparameters: dropout at 0.6, learning rate at 0.001, weight decay =



(a) SGC (b) GAT (c) GCN

(d) GraphSAGE-Mean (e) GraphSAGE-Pool (f) CTQW

Figure 5.3: t-SNE of various networks on the Wisconsin dataset, with subfigures from (a) to (f) representing SGC, GAT, GCN, GraphSAGE with mean aggregation, GraphSAGE with pool aggregation, and our CTQW-GraphSAGE, respectively (2-layer versions for graph-based models).

$5e^{-4}$, the hidden dimension within {32, 64, 128}. Especially for the DeepWalk method, the number of walks is 10, the window size is within 5, the embedding size is 128, and the walk length is 80. For the proposed method, the number of node samples in the first layer is set to {20, 15}, and in the second layer, it is {10, 5}. The implementation environment includes python (3.8), torch (2.0.1), torch-geometric (2.4) [90], and Paddle-Quantum (2.4.0) [24]. In our study, all experiments are conducted on a Unix device equipped with a Radeon Pro 5500M GPU. We conducted comparative experiments for different neural network architectures by implementing both 1-layer and 2-layer variants. This approach allows us to evaluate the impact of network depth on the model's performance.

5.4.4 Performance

Table 5.3 comprehensively demonstrates the performance of various networks across multiple datasets, where we implement experiments with both 1-layer and 2-layer configurations for graph-based networks. Notably, our method achieved the best results in most of the selected datasets, such as Cora, Citeseer, Pubmed, and Wisconsin, and the second-best in the Texas dataset, while performing at an average level on the Cornell dataset compared to other graph-based methods. Both 1-layer and 2-layer networks effectively handled node classification tasks. In terms of stability, compared to DeepWalk, GCN, GAT, and SGC methods, our approach and GraphSAGE adapted well to different datasets for node classification. For instance, as shown in Figure 5.3, the t-SNE visualization of the final embeddings of trained networks on the Wisconsin dataset, SGC, GAT, and GCN does not classify the dataset effectively, whereas CTQW and GraphSAGE distinctly differentiated between different nodes.

5.5 Conclusion

In this work, we introduce a novel hybrid network called CTQW-GraphSAG. This network utilizes a parameterized variational quantum circuit for node feature mapping, followed by computing the similarity between these mappings. We then construct a Hamiltonian using the Hadamard product of the similarity matrix and the adjacency matrix to facilitate the evolution of continuous quantum walks. The resulting probabilities of the walker residing on the nodes are used for weighted aggregation. This approach not only leverages the structural information of the graph but also considers the node feature information in the quantum space. Additionally, in our node sampling process, we take into account the k-hop neighbors of the target node, focusing on preserving the complex information about the target node. This method has also achieved comparable performance to current mainstream graph neural networks in node classification tasks. However, as the quantum part of our method is simulated on classical computers using quantum platforms, it incurs a relatively high time cost for training. In the future, we hope to utilize real quantum devices and more efficient computational methods for acceleration.

Chapter 6

Quantum State Hyperdimensional Computing

Contents

6.1 Overview

Hyperdimensional Computing (HDC) is a brain-inspired computational paradigm that utilizes high-dimensional vector spaces to represent, bind, and process information efficiently. Its robustness, scalability, and adaptability make HDC particularly effective for machine learning tasks such as classification and pattern recognition. In this work, we introduce Quantum-StateHD (QSHD), a novel approach that leverages random quantum states as encoding mechanisms for high-dimensional spaces within the HDC framework. Quantum states, with their inherent properties of high-dimensionality, near-orthogonality, and probabilistic distributions, provide a natural substrate for constructing robust representations. Quantum-StateHD encodes classical data into the coefficients of random quantum states and seamlessly integrates these representations into the HDC pipeline. Experimental results on several benchmark classification datasets demonstrate that Quantum-StateHD achieves competitive or superior performance compared to traditional state-of-the-art HDC methods, showcasing its potential for robust and efficient high-dimensional information processing. This study bridges the gap between quantum information science and HDC, offering a promising direction for hybrid quantum-classical machine learning architectures.

6.1.1 Problem Statement

Hyperdimensional Computing (HDC) is an emerging computation paradigm inspired by the human brain processing information using distributed and hyperdimensional representations. Its inherent properties, such as small model size, low computational complexity, and support for one-shot and few-shot learning, make HDC well-suited for resource-constrained environments. HDC is foreseen as a valuable tool in applications where efficiency is prioritized over absolute accuracy. It shows performance in many applications, including signal processing [91], robotics [92], natural language processing [93], and data security [94].

HDC can be conceptualized as a computational approach that employs high-dimensional vectors to represent information. These vectors, typically with thousands of elements, enable efficient processing through vector operations. Various encoding approaches are proposed to map objects in the input space to the hyperspace, and the choice of encoding functions depends on the characteristics of the input data. For example, binary encoding represents data as binary strings, which are then mapped to high-dimensional vectors.

N-grams decompose sequential data, such as text or time series, into overlapping subsequences of length n. Each n-gram is then encoded as a vector. These encoding strategies share a common characteristic: they begin by encoding elementary data units, such as characters, pixel intensities, or signal amplitudes, which are subsequently aggregated to represent composite objects (e.g., images, sentences, and signals).

Although encoding is crucial for representing data within the HDC framework, existing methods face inherent limitations, particularly with respect to computational cost and potential information loss [95]. Generation and manipulation of high-dimensional vectors, especially for complex data types, can require substantial computational resources, posing a scalability challenge for large datasets [96]. Furthermore, the encoding process itself may lead to a reduction in the fidelity of the data representation. This loss of information can be due to dimensionality reduction techniques or inherent constraints of the chosen encoding scheme, which can affect the accuracy of subsequent HDC operations [97]. Therefore, balancing representational capacity with computational efficiency and minimizing information loss remain key considerations in the development of effective HDC encoding strategies.

To address the above issues, this work presents Quantum-StateHD (QSHD), a novel framework designed to enhance encoding efficiency within the HDC paradigm by exploiting the principles of quantum mechanics. Quantum-StateHD leverages quantum encoding, a technique used in quantum computing and communication where information is represented through quantum states. A pure state is the simplest kind of quantum state, which can be described by a normalized state vector, $\psi = \alpha |0\rangle + \beta |1\rangle$, where α and β are complex numbers that satisfy $|\alpha|^2 + |\beta|^2 = 1$. This representation describes the state of a qubit, the quantum analog of a classical bit. A key property of qubits is their ability to exist in a superposition of states, which means that they can simultaneously embody a combination of $|0\rangle$ and $|1\rangle$. This characteristic allows n qubits to span a Hilbert space of dimension 2^n. Consequently, the information capacity of a quantum system scales exponentially with the number of qubits, offering a substantial advantage in information storage compared to classical systems where capacity increases linearly. The proposed QSHD utilizes this exponential scaling to achieve more efficient encoding within the HDC framework.

Integrating quantum encoding into the HDC paradigm is non-trivial. The primary challenge is how to utilize quantum mechanisms for data encoding. Common quantum encoding methods include binary encoding and amplitude encoding. To address this challenge, we adopt a combined strategy that integrates the real and imaginary parts along with the amplitude encoding. The second challenge involves an approximate computation of orthogonality and similarity. In high-dimensional spaces, approximate orthogonality is a crucial property of hyperdimensional computing (HDC), which ensures robustness in noisy environments. Therefore, proving the approximate orthogonality and similarity computation of the encoded random quantum states in high-dimensional Hilbert spaces presents a significant challenge. The final challenge is computational complexity. On classical computing platforms, HDC implementations typically have low complexity, which is one of the main reasons why HDC is widely chosen for real-time tasks such as edge computing and embedded AI. Thus, validating the efficiency of our proposed method on benchmark datasets is a practical challenge.

6.1.2 Main Contributions

To address the aforementioned challenges, the main contributions of our proposed QSHD method are as follows:

- **Proposing a novel encoding method using quantum states:** We present a method that utilizes random quantum states as the foundation for encoding data into hyperdimensional spaces. Quantum states, with their inherent high-dimensionality and probabilistic distributions, provide a natural substrate for representing complex data. This encoding mechanism replaces traditional deterministic or classical random hypervector schemes, offering improved representational capacity by exploiting the unique properties of quantum states, such as near-orthogonality and stochasticity. The method efficiently maps classical data into quantum-state-based representations, preserving the underlying structure of the data while enhancing robustness.

- **Theoretical Analysis of Quantum-State Properties in HDC:** We provide a rigorous theoretical analysis that demonstrates how quantum states align with the requirements of HDC. This includes proving that the near-orthogonality of random quantum states enhances the separability of encoded representations, ensuring minimal interference between distinct data points. This theoretical foundation establishes the feasibility and effectiveness of quantum-state-based encoding within HDC.

- **Extensive Evaluation on Benchmark Datasets:** The performance of Quantum-StateHD was evaluated on multiple benchmark datasets commonly used in classification tasks. To validate the practical implementation, we implemented our method using standard Python libraries and also leveraged IBM's Qiskit framework for quantum computing simulations. The experimental results demonstrate that Quantum-StateHD achieves competitive or superior performance compared to traditional HDC methods in terms of accuracy, robustness, and scalability. These findings highlight the potential benefits of integrating quantum-inspired encoding strategies into classical machine learning pipelines.

6.1.3 The Outline of the Chapter

The remaining parts of this chapter are organized as follows. Section 6.2 presents related works. Section 6.3 introduces the traditional HD methods. Section 6.4 provides the overview of the proposed approach and detailed designs of each component. Section 6.5 and 6.6 comprehensively evaluate the proposed approach and compare the performance with the state-of-the-art baselines. Finally, section 6.7 concludes Chapter 6.

6.2 Related Work

Inspired by the neural activity patterns of the human brain, hyperdimensional computing (HDC) has emerged as a promising computing framework. The basic HDC model [98] establishes the fundamental principle of representing data with high-dimensional random vectors. Significant research efforts focus on improving the efficiency of the HDC model, encoding strategies, and hardware implementation. Wang proposes DistHD [99], a learner-aware dynamic encoding method that achieves a significant reduction in dimension while improving inference speed. Similarly, Imani et al. develop QuantHD [100], an efficient HDC framework that reduces reliance on floating-point numbers, thus greatly improving energy efficiency. Duan et al. propose LeHDC [101], which enhances learning-based HDC classifiers through binary neural network training and improves the accuracy of inference. Training efficiency is also a key area of focus. AdaptHD [102] is an adaptive training method to optimize HDC learning, achieve faster training, and achieve significant energy savings. OnlineHD [103] supports single-shot online learning and enhances the robustness to hardware failures. Several works address resource efficiency and hardware adaptability. Morris et al. introduce CompHD [104], focusing on model compression to reduce storage costs while improving execution speed. SparseHD [105], which uses sparse-aware computing for hardware acceleration, resulting in lower resource consumption. To support real-time edge computing, Zou et al. propose NeuralHD [106], a neural adaptation-inspired dynamic coding system that can scale HDC across IoT systems with faster training performance. These advances in HDC illustrate the collaborative efforts to develop scalable, efficient, and accurate models. Through innovations in coding schemes, adaptive learning, hardware-aware design, and edge computing, HDC has solidified its role as a powerful framework for future computing paradigms, especially in energy-constrained environments.

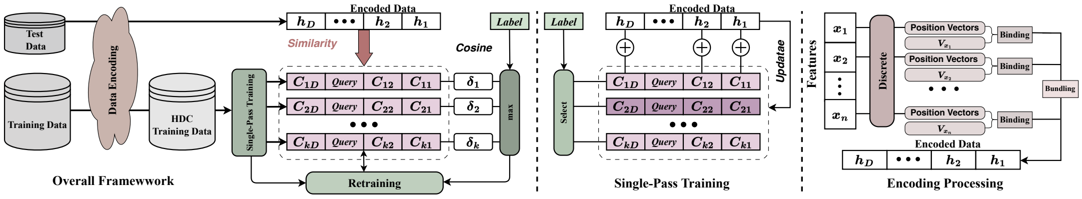


Figure 6.1: Traditional HDC for Classification framework.

Quantum encoding is a critical part of quantum computing, translating classical data into quantum states through encoding techniques such as amplitude encoding, basis encoding, and hybrid methods. These encoding schemes facilitate quantum computation, enhance data security, and enable efficient quantum communication. Ranga et al. [107] highlighted the transformative role of data encoding in quantum machine learning, focusing on improving model accuracy through optimized quantum state preparation. Similarly, Rath and Date [108] explored classical-to-quantum mapping, demonstrating its impact on the performance of quantum algorithms. Weigold et al. [109] expanded quantum data encoding patterns for gate-based quantum computers, while Bravo-Prieto [110] developed enhanced quantum autoencoders for data compression. In the field of quantum cryptography, Yang et al. [111] proposed a novel encryption scheme using quantum Fourier transforms for secure image communication. Image encoding applications were advanced by Yetiş and Karaköse [112], who designed cost-effective quantum circuits. Hybrid quantum encoding techniques, explored by Smanchat et al. [113], demonstrate improved quantum data storage and processing. Furthermore, Majji et al. [114] introduce a quantum-based approach for image data compression, enhancing efficiency in quantum sensing. Nguyen et al. (2024) revisit the encoding in quantum neural networks, optimizing visual feature extraction for computer vision tasks. Recently, Govorov et al. [115] explore time-encoded photonic quantum states for applications in secure quantum communication protocols like quantum key distribution (QKD). These advancements showcase quantum encoding's versatility in enabling next-generation computing, data security, and communication technologies.

6.3 Background

6.3.1 Hyperdimensional Computing

Hyperdimensional computing can be applied to a variety of learning problems. This chapter focuses on classification tasks and common problems in supervised learning. As shown in Figure 6.1, applying hyperdimensional computing for classification tasks involves encoding, single-pass training, and similarity checking.

Encoding

Encoding is a key step in HDC to map input data into a high-dimensional vector space. The features of the input data and its structural relationships are embedded into a single high-dimension vector while retaining its information. Let $\mathbf{x} = [x_1, x_2, ..., x_n]$ denote preprocessed input data. Encoding $\mathbf{x}$ into a high-dimensional vector space usually takes the following steps.

Value Encoding: Value encoding maps each feature in the input data to a high-dimensional space. For discrete features, each possible value v_j is assigned a unique hypervector. If the features are continuous, such that x_i is within a range$[x^{low}, x^{high}]$, the value vector is computed as $\mathbf{V}_{x_i} = \alpha \cdot \mathbf{V}_{x^{low}} + (1 - \alpha) \cdot \mathbf{V}_{x^{high}}$, where $(\alpha = x_i - x^{low})/(x^{high} - x^{low})$. This ensures that continuous values are smoothly encoded into high-dimensional space. Alternatively, if the value is discretized into binary or hot vectors, such as [1, 1, 1, 0, 0, 0], thermometer encoding can be applied to represent the feature's relative position with its range.

Position Encoding: Position encoding is a technique used to represent the position of a feature in a sequence. This is done by adding a vector, $\mathbf{P}_i \in \mathbb{R}^D$, where D is the dimension of the high-dimensional space, to the feature's representation that encodes its position. There are many different ways to do position encoding, but one common approach is to use a set of sinusoidal functions with different frequencies. Position encoding can be used to improve the performance of HDC models on tasks that involve sequential data, such as natural language processing.

Based on the above encoding method, a unique property of hyperdimensional space is a large number of nearly orthogonal hypervectors, enabling highly parallel operations. We can use these operations to combine the positional information $\mathbf{P}_i$ with the value information $\mathbf{V}_{x_i}$ for each feature, creating a bound vector $\mathbf{B}_i$. This is achieved by mathematical operations **Binding**. Binding operation can be performed using various methods, such as dot product, $\mathbf{B}_i = \mathbf{P}_i \cdot \mathbf{V}_{x_i}$, element-wise multiplication, $\mathbf{B}_i = \mathbf{P}_i \odot \mathbf{V}_{xi}$, or XOR for binary vectors, $\mathbf{B}_i = \mathbf{P}_i \oplus \mathbf{V}_{x_i}$. The result of this operation ensures that both position and value information are encoded into a single vector for each feature. Once all features have been encoded as bound vectors, bundling is performed to combine them into a hypervector $\mathbf{H}$ that represent the entire input data. The **bundling** operation aims to superimpose the bund vectors, typically by summing them: $\mathbf{H} = \sum_{i=1}^{n} \mathbf{B}_i$, where n is the total number of features. Optionally, weights ω_i can be assigned to the features during binding, resulting in a weighted sum: $\mathbf{H} = \sum_{i=1}^{n} \omega_i \cdot \mathbf{B}_i$. This superposition ensures that all feature information is integrated into a single vector, capturing both individual feature contributions and their relationships. Finally, the hypervector $\mathbf{H}$ is normalized to ensure numerical stability for further processing. Normally, L2 normalization is applied.

Training

In the HDC classification task, most existing HDC algorithms perform the training in a single iteration. It can be divided into several steps. At the beginning of training, each class $\mathbf{C}_k$ is associated with a class hypervector $\mathbf{C}_k \in \mathbb{R}^D$, where D is the dimensionality of the hypervector space. These class hypervectors are initialized as $\mathbf{C}_k = 0, \forall k \in 1, 2, ..., K$, where K is the number of classes. This initialization allows the hypervectors to accumulate information from the samples during training. Then, for every training sample $(\mathbf{x}, y)$, where $\mathbf{x}$ represents the features and y is the class label, the encoded normalized hypervector of the sample, $\mathbf{H}$ is added to the corresponding class hypervector: $\mathbf{C}_k = \mathbf{C}$. After all samples have been processed, the class hypervectors C_k finish updating. Normally, each class hypervector should be normalized to ensure consistency during inference. The class hypervectors represent the comprehensive feature information of their respective class and serve as reference vectors for classification during inference.

Inference

To classify the input sample, the similarity between the encoded sample hypervector $\mathbf{H_x}$ and each class hypervector $\mathbf{C}_k$ is calculated. There are many ways to calculate similarity. The most popular one is the cosine similarity that measures the cosine of the angle between two hypervectors: $\delta(\mathbf{H_x}, \mathbf{C}_k)$ where both $\mathbf{H_x}$ and $\mathbf{C}_k$ are normalized. If the hypervectors are binarized, the similarity can be computed as the complement of the Hamming distance: $d(\mathbf{H_x}, \mathbf{C}_k) = \sum_{i=1}^{D}(\mathbf{H_x}[i] \oplus \mathbf{C}_k[i])$, where $\oplus$ is also the XOR operation, and the smaller the distance, the higher the similarity. For the real-valued hypervectors, a simple dot product can also serve as a similarity metric. The predicted label is determined by selecting the class k with the highest similarity score:

$$\hat{y} = \arg\max_{k} \delta(\mathbf{H}_{\mathbf{x}}, \mathbf{C}_k)$$

6.3.2 Quantum state and quantum encoding

Qubits and quantum states

A **qubit** is the fundamental unit of quantum information, analogous to a classical bit but with significant differences. Unlike a classical bit, which can only be in one of two states (0 or 1), a qubit can exist in a superposition of states 0 and 1. The state of a qubit is represented as: $\psi = \alpha |0\rangle + \beta |1\rangle$, where α and β are complex numbers that satisfy $|\alpha|^2 + |\beta|^2 = 1$. In addition, Assuming an n-qubit space exists, the state of a quantum system is represented as a vector in a 2^n-dimensional Hibert space. It captures all possible combinations of the n qubits in superposition. A general **quantum state** for n-qubits can be written as

$$|\psi\rangle = \sum_{i=0}^{2^n-1} c_i |i\rangle$$

where $|\cdot\rangle$ represents Dirac notation, $|i\rangle$ represent the basis vector like $|0...000\rangle$,$|0...001\rangle$, or $|0...010\rangle$, $c_i \in \mathbb{C}$ are the coefficients corresponding to each basis state, satisfying the normalization condition:

$$\sum_{i=0}^{2^n-1} {c_i}^2 = 1$$

which also matches the definition of a pure state.

Quantum encoding

Quantum encoding is the process of mapping classical information into quantum states for storing, processing, or transmission. One of the efficient, simple, and pure-state methods is amplitude coding. Consider a given classical data vector $\mathbf{x} = [x_0, x_1, ..., x_{N-1}]$, where $N = 2^n$ and n is the number of qubits, amplitude encoding transforms this vector into a quantum state $|\psi\rangle$ as follows:

$$|\psi\rangle = \frac{1}{||\mathbf{x}||} \sum_{i=0}^{N-1} x_i |i\rangle$$

Based on such an encoding method and HD encoding process, we can prepare and characterize the random quantum pure states of the original data.

6.4 Proposed Quantum-StateHD

The overall structure of our proposed Quantum-StateHD is shown in Figure 6.2, which integrates quantum encoding and representation into the conventional HDC framework. The details of quantum encoding block, as well as the training and inference processes are detailed in the following sections.

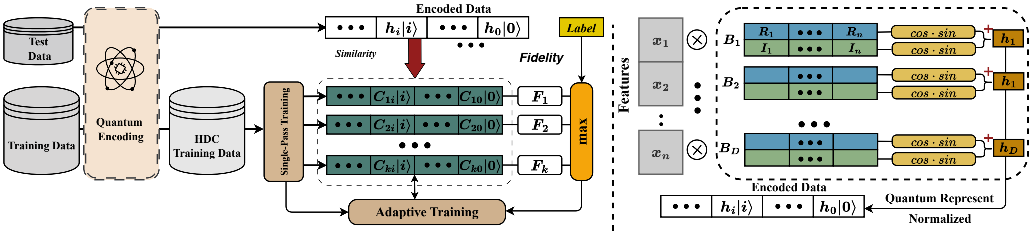


Figure 6.2: Quantum-StateHD for Classification framework.

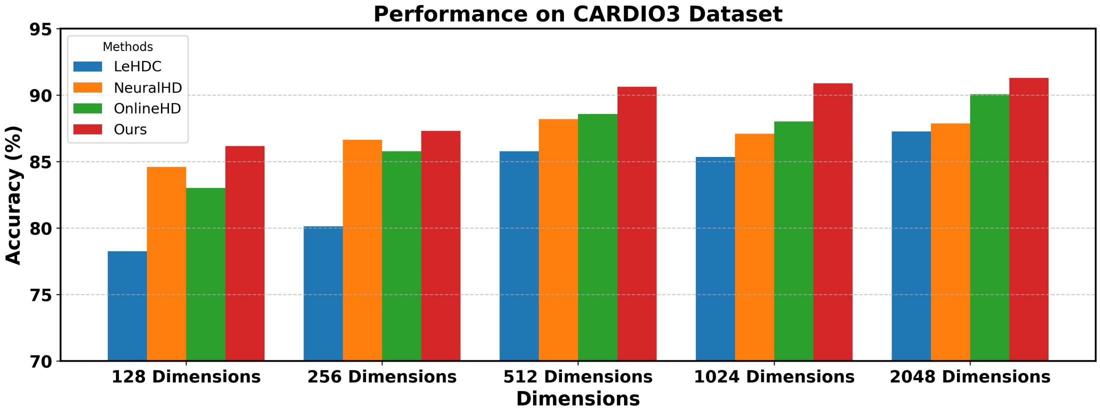


Figure 6.3: Classification accuracy comparison between different methods

6.4.1 Quantum-StateHD Encoding

This section presents the proposed encoding method inspired from the quantum mechanism and signal processing. By mapping the original feature to a high-dimensional quantum space, the nonlinear strengthening and numerical stability of the feature can be obtained simultaneously. The specific methods and instructions are as follows. Suppose $\mathbf{x} = [x_1, x_2, ..., x_n]$ is the input data with n features, which is required to be mapped to a hypervector $\mathbf{H} = \{h_1, h_2..., h_D\}$ with dimensions D, where $D = 2^N$ and $h_i \in \mathbb{C}$, and $h_i = Re(h_i) + iIm(h_i)$ is a complex number with $Re(h_i)$ and $Im(h_i)$ represent the real and imaginary parts, respectively. Firstly, two vectors $\mathbf{B}^{Re} = \{B_1^{Re}, B_2^{Re}, ...B_D^{Re}\}$ $\mathbf{B}^{Im} = \{B_1^{Im}, B_2^{Im}, ...B_D^{Im}\}$ are randomly generated from a Gaussian distribution with the mean $\mu = 0$ and the standard deviation $\sigma = 1$. We encode each element h_i of $\mathbf{H}$ by equation

$$Re(h_i) = cos(\mathbf{B}_i^{Re} \cdot \mathbf{x} + b_i^{Re}) \times sin(\mathbf{B}_i^{Re} \cdot \mathbf{x}) \tag{6.1}$$

$$Im(h_i) = cos(\mathbf{B}_i^{Im} \cdot \mathbf{x} + b_i^{Im}) \times sin(\mathbf{B}_i^{Im} \cdot \mathbf{x}) \tag{6.2}$$

where b_i^{Re} and b_i^{Im} are random shift value generated uniformly from $[0, 2\pi]$. This encoding method, based on random projection, trigonometric transformation, and random phase shifts, offers several key advantages.

Firstly, the proposed encoding method produces a complex hypervector $\mathbf{H}$ which is assumed to be normalized. This complex hypervector can directly serve as the coefficients of a pure quantum state composed of N-qubits. Each component h_i represents the complex amplitude of the corresponding quantum basis state $|i\rangle$ in the computational basis.

$$\mathbf{H} := |\psi\rangle = \sum_{i=0}^{2^N-1} h_i \, |i\rangle \, , h_i \in \mathbb{C}$$

This encoding process effectively transforms the original classical feature vector into a D-dimensional hypertvector, which can equivalently be viewed as a pure quantum state in an N-qubit Hibert space where $D = 2^N$.

Secondly, our method can also ensures to map different input vectors to nearly orthogonal quantum states even for large datasets. This kind of orthogonality is the reflection of the concentration of measure in classical high-dimensional space and also can extended to quantum space. Consider there are two random pure states:

$$|\psi\rangle = \sum_{i=0}^{2^N-1} c_i \, |i\rangle \, , \quad |\phi\rangle = \sum_{i=0}^{2^N-1} d_i \, |i\rangle \, ,$$

where $c_i, d_i \in \mathbb{C}$ are the normalized coefficients. In the quantum space, the orthogonality of two quantum states can be measured by **fidelity**. The fidelity of two pure state is defined as $F(|\psi\rangle \, , |\phi\rangle) = |\, \langle\psi| \, |\phi\rangle \, |^2$. The value approaching zero indicates that the orthogonality between two quantum states becomes stronger (if $F(\cdot) = 1$, the two states are identical, and if $F(\cdot) = 0$, the two states are orthogonal). The inner product between the states is:

$$\langle\psi|\phi\rangle = \sum_{i=0}^{2^N-1} c_i^* d_i$$

Since c_i and d_i are independent and symmetrically distributed, we have expectation value $\mathbf{E}[\langle\psi|\phi\rangle] = 0$. The variance should be the expectation of the squared magnitude of the inner product:

$$\mathbf{E}[|\langle\psi|\phi\rangle|^2] = \sum_{i=0}^{2^N-1} \mathbf{E}[|c_i|^2] \cdot \mathbf{E}[|d_i|^2]$$

Since $\mathbf{E}[|c_i|^2] = \mathbf{E}[|d_i|^2] = 1/2^N$, it follows that:

$$\mathbf{E}[|\langle\psi|\phi\rangle|^2] = \frac{1}{2^N}$$

Thus, fidelity $F(|\psi\rangle, |\phi\rangle) = |\langle\psi|\phi\rangle|^2$ decreases as 2^N increases, and at the limit $2^N \to \infty$, the fidelity tends to zero. The fidelity calculation is also used as a measure of similarity in subsequent classification training and inference. This method makes different feature vectors encoded into high-dimensional quantum states distinguishable. Therefore, this encoding approach is well suited for application in hyperdimensional computing.

6.4.2 Quantum-StateHD Training

Quantum-StateHD computing supports efficient one-pass training. The encoder maps all training data to the quantum state $|\psi\rangle$. Training $|\psi\rangle$ in the same class $\{|\psi_l^1\rangle, |\psi_l^2\rangle, ..., |\psi_l^m\rangle\}$ can be aggregated to generate a single class $|C_l\rangle$, where m means the number of samples in this class and l indicates the class tag. To ensure that the class superposition state is a pure state, we cannot add directly as in classical space.

$$|C_l\rangle = |\psi_l^1\rangle + |\psi_l^2\rangle, ..., |\psi_l^m\rangle \tag{6.3}$$

Assuming that each sample contributes equally to this class, the average superposition of quantum states should be used to obtain a pure state to express the class superposition state. The method is as follows:

$$|C_l\rangle = \frac{\sum_{i=1}^m |\psi_l^i\rangle}{\|\sum_{i=1}^m |\psi_l^i\rangle\|} = \frac{\sum_{i=1}^m |\psi_l^i\rangle}{\sqrt{m + \sum_{i\neq j} \langle\psi_l^i|\psi_l^j\rangle}} \tag{6.4}$$

In the quantum encoding section, it is explained that $|\psi_l^i\rangle$ and $|\psi_l^j\rangle$ $(i \neq j)$ are approximately orthogonal, so

$$\sum_{i\neq j} \langle\psi_l^i|\psi_l^j\rangle \approx 0$$

To implement this on a classical computer, the above mathematical operations need to be performed. However, on quantum devices or quantum platforms, multi-qubits unitary operations must be used to achieve the desired effect. For different classes, we use this method to obtain the superposition state for each class, allowing the classification of test data. Moreover, the iterative learning process is conducted as described in Algorithm 1. The QSHD incremental learning algorithm efficiently updates the model by building on previously learned representations instead of starting from scratch. In each training iteration, the algorithm processes each data point by calculating its similarity to the current model and predicting its category. If the prediction is correct, the model remains unchanged. However, if a misclassification occurs, the model is updated by adding the data point to the correct class representation and subtracting it from the incorrectly predicted class. This selective update mechanism allows the model to improve its performance while retaining prior knowledge, achieving fast convergence and continuous learning through multiple iterations.

Algorithm 4 Quantum-StateHD Incremental Learning

```
Input: Dataset $D = \{(x_i, y_i)\}$, Current Model $M$, Number of Iterations $T$
Output: Updated Model $M$
for $t = 1$ to $T$ do
    for each $(x_i, y_i) \in D$ do
        Compute similarity $S$ of $x_i$ with the current model $M$
        $\hat{y}_i \leftarrow \arg\max S$                    ▷ Predicted class for $x_i$
        if $\hat{y}_i == y_i$ then
            continue                                         ▷ No update if correctly classified
        else
            $M[y_i] \leftarrow M[y_i] + x_i$                 ▷ Add to correct class
            $M[\hat{y}_i] \leftarrow M[\hat{y}_i] - x_i$     ▷ Subtract from incorrect class
        end if
    end for
end for
return $M$
```

6.4.3 Quantum-StateHD Inference

The inference process is similar to the classical HD classification method. A sample of test data $\mathbf{x_t}$ should be encoded into a quantum state $|\psi_t\rangle$. The similarity metric fidelity is used to measure the strength of a match between $|\psi_t\rangle$ and each class $|C_l\rangle$. After the fidelity is computed, the class corresponding to the highest fidelity value is selected as the output.

6.5 Experimental Setup

We evaluate the proposed method by conducting comparative experiments on various datasets and dimensions, comparing it against state-of-the-art baseline methods to ob-

Table 6.1: Classification accuracy of Quantum-StateHD and other HDC algorithms on 5 different datasets. The results show the average of the ten experiments and the standard deviation is in parentheses, the highest accuracy is **bolded**, and the runner-up is underlined.

	CARDIO10			CARDIO3			ISOLET			MNIST			UCIHAR		
	128	512	2048	128	512	2048	128	512	2048	128	512	2048	128	512	2048
AdaptHD	66.04 (3.10)	78.09 (3.52)	79.66 (2.93)	81.30 (4.54)	87.09 (3.91)	90.22 (2.78)	40.65 (2.07)	69.36 (4.01)	86.68 (0.62)	49.73 (8.19)	75.28 (2.07)	83.06 (7.21)	65.32 (0.78)	79.05 (4.78)	86.70 (2.70)
CompHD	44.05 (2.74)	56.73 (0.22)	57.51 (0.69)	65.18 (1.72)	67.53 (2.93)	72.93 (1.06)	36.84 (0.43)	70.09 (0.79)	83.39 (0.69)	49.68 (0.67)	70.06 (1.23)	80.00 (0.21)	61.25 (2.58)	75.36 (1.26)	81.81 (1.39)
DistHD	68.15 (2.15)	70.81 (3.60)	72.54 (2.21)	69.01 (5.31)	59.62 (6.93)	67.06 (6.82)	89.10 (0.59)	92.47 (0.61)	91.40 (1.39)	85.02 (0.59)	87.40 (0.33)	88.59 (0.44)	**92.83** **(0.36)**	89.93 (4.89)	92.22 (2.59)
LeHDC	35.92 (7.47)	54.77 (6.46)	60.09 (4.89)	78.25 (4.50)	85.76 (0.44)	87.25 (1.25)	18.02 (2.28)	30.38 (4.24)	46.87 (1.66)	28.34 (6.07)	51.75 (3.66)	69.73 (2.04)	41.51 (2.19)	44.41 (2.89)	64.23 (1.04)
NeuralHD	69.56 (1.73)	77.93 (1.45)	78.33 (0.11)	84.59 (1.86)	88.18 (0.86)	87.87 (0.48)	87.21 (0.34)	91.36 (0.56)	92.28 (0.40)	82.07 (0.64)	87.35 (0.12)	88.21 (0.04)	87.26 (0.28)	89.20 (0.34)	89.80 (0.12)
OnlineHD	71.05 (1.23)	**78.33** **(1.44)**	**81.22** **(1.33)**	83.02 (2.21)	88.58 (1.28)	90.06 (2.31)	88.28 (0.70)	**92.50** **(0.52)**	93.93 (0.26)	81.25 (0.52)	92.47 (0.41)	**95.51** **(0.99)**	88.42 (1.94)	92.09 (0.98)	92.87 (0.39)
QuantHD	43.27 (0.77)	55.48 (3.81)	56.42 (0.44)	62.44 (3.27)	68.70 (2.40)	69.33 (0.80)	33.93 (4.11)	59.53 (0.71)	79.13 (0.59)	41.35 (3.32)	63.04 (0.72)	76.72 (0.79)	55.51 (1.60)	69.01 (1.86)	78.74 (1.56)
SparseHD	42.96 (5.85)	63.30 (3.62)	73.32 (1.09)	78.09 (3.28)	85.76 (0.62)	84.82 (2.46)	26.38 (1.31)	54.27 (1.19)	79.45 (0.76)	30.29 (5.92)	56.77 (3.42)	78.03 (1.23)	55.53 (2.56)	72.20 (3.53)	85.65 (1.03)
Vanilla	50.31 (2.55)	57.51 (3.37)	58.53 (0.95)	69.01 (4.49)	71.36 (0.84)	72.22 (1.46)	56.02 (1.39)	78.19 (0.23)	86.40 (0.43)	58.77 (3.06)	75.95 (0.43)	81.36 (0.52)	65.42 (0.92)	79.18 (0.37)	83.11 (0.87)
	7	9	11	7	9	11	7	9	11	7	9	11	7	9	11
Ours	**72.30** **(2.50)**	76.62 (1.59)	80.28 (0.48)	**86.15** **(1.39)**	**90.61** **(0.98)**	**90.28** **(0.65)**	**89.36** **(1.74)**	92.15 (0.49)	**94.17** **(0.37)**	**85.69** **(0.47)**	**92.95** **(0.14)**	93.08 (0.15)	88.12 (0.26)	**92.58** **(1.25)**	**93.48** **(0.87)**

Table 6.2: Quantitative information of the datasets

Dataset	# features	# classes	# training size	# test size	Description
CARDIO3	21	3	1700	426	Cardiac Signals Classification
CARDIO10	21	10	1700	426	Cardiac Signals Classification
MNIST	784	10	60000	10000	Handwritten Digit Recognition
ISOLET	617	26	6238	1559	Speech Recognition
UCIHAR	561	6	7352	2947	Human Activity Recognition

tain detailed experimental results and parameter information. Specific details are provided below.

6.5.1 Datasets

We test our methods and state-of-the-art methods on 5 benchmark classification datasets.
CARDIO3 and CARDIO10: CARDIO [116] is a dataset designed for in-depth research on cardiac signals. It provides tools to load and process ECG signals and includes predefined workflows for training models and performing inference. This dataset is often utilized for detecting PQ, QT, QRS segments, and calculating heart rate.
MNIST: The MNIST dataset [117] is a reference for handwritten digit recognition tasks. It contains grayscale images of digits ranging from 0 to 9 and is widely used in machine learning and deep learning research to evaluate image classification algorithms.
ISOLET: The ISOLET dataset [118] is designed for speech recognition tasks, specifically to classify spoken letters of the English alphabet. It provides acoustic data collected from multiple speakers and is commonly used to develop and test audio classification models.
UCI HAR: The UCI Human Activity Recognition (UCI HAR) [119] dataset focuses on identifying human activities using data collected from wearable sensors. It has been widely used in machine learning to study time series data and activity recognition applications.
The quantitative information of these data is shown in Table 6.2.

6.5.2 Baselines

A variety of HD baselines in multiple dimensions are used as the baselines in our experiments. Specifically, since our method operates in the Hilbert space, all high-dimensional settings are chosen as powers of 2 for a fair comparison. The baselines used include Vanilla HD, AdaptHD, DistHD, CompHD, NeuralHD, SparseHD, QuantHD, LeHDC, and OnlineHD, with dimensions {128, 256, 512, 1024, 2048, 4096, 8192}. These correspond to our method operating with {7, 8, 9, 10, 11, 12, 13} qubits, respectively.

6.5.3 Platforms

The proposed Quantum-StateHD is implemented in both classical and quantum-dependent environments. Specifically, our approach is implemented in a classical environment using the PyTorch framework, while the related baseline approach is done with the help of the TorchHD [120] library. In a quantum environment, our approach is implemented through IBM's Qiskit platform [121], leveraging its quantum computing resources and simulation capabilities.

In order to ensure the comprehensiveness and reliability of the experimental results, we conduct several tests and comparative analyses on the performance of all methods in a classical environment, focusing on evaluating the accuracy, stability, and computational efficiency of the methods. In addition, we compare the performance of our methods in classical and quantum environments and analyze the influence of different environments on the performance of our methods. The dual-environment test validates the adaptability and robustness of our approach while highlighting the potential advantages and limitations of quantum computing for specific tasks.

6.6 Results

6.6.1 Convergence

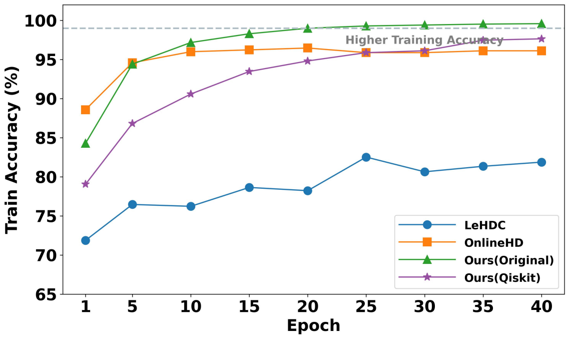


Figure 6.4: Comparison of training processes between different models and implementations (case on CARDIO3 with dimension 128 or qubit 7)

Figure 6.4 illustrates the relationship between training accuracy and epochs, comparing the performance of four methods: LeHDC, OnlineHD, Ours (Original) implemented in a traditional Python environment, and Ours (Qiskit) implemented. Among these, Ours (original) achieves the best performance, with training accuracy rapidly increasing to nearly 100% within the first 5 epochs and maintaining this level throughout. Ours (Qiskit) also performs exceptionally well, showing a steady improvement in accuracy and finally surpassing OnlineHD, reaching greater than 95%. While OnlineHD achieves high accuracy early on, it slightly underperforms compared to Ours (Qiskit) in the later stages. In contrast, LeHDC shows the weakest performance, with the accuracy gradually increasing and only reaching around 80% at the end. Overall, both Ours (Original) and Ours (Qiskit) demonstrate outstanding performance, with Ours (Original) achieving the highest accuracy and Ours (Qiskit) ultimately outperforming OnlineHD with its stable and reliable improvement. In general, our method demonstrates remarkable stability and rapid convergence during multiple training iterations, consistently outperforming other approaches in terms of both accuracy and training efficiency.

6.6.2 Accuracy

Table 6.1 shows the classification performance of various algorithms in five datasets (CARDIO10, CARDIO3, ISOLET, MNIST, UCIHAR) and different feature dimensions (128, 512, 2048). In general, as the dimension increases from 128 to 2048, the classification accuracy of all methods generally improves significantly, indicating that high-dimensional feature representation plays an important role in improving the classification effect. In comparison of different methods, the method proposed by our model performs best in the combination of various datasets and dimensions, showing a strong classification ability. For example, on the MNIST dataset, our method achieves 95.51% accuracy at 2048 dimensions, which is significantly higher than other methods. In the UCIHAR dataset, our method achieves an accuracy of 93.48% at 2048 dimensions, also much ahead of other methods. In the ISOLET dataset, our method also leads with an accuracy rate of 92.15%, but OnlineHD performs well on this dataset, reaching 92.47%. In contrast, traditional methods such as AdaptHD and DistHD perform well on some datasets (such as ISOLET and CARDIO10), but are still slightly inferior overall, while CompHD and QuantHD have mediocre classification performance and significantly lower accuracy on multiple datasets. Overall, our method shows strong generalization and stability, especially in the case of high-dimensional features, reflecting a breakthrough in the processing power of complex datasets.

In addition to implementing our algorithm with PyTorch, given that our algorithm involves quantum states and quantum coding, we also deployed it on IBM's Qiskit platform. The main difference between the two is the way the quantum states are implemented and fidelity calculations. Specifically, the implementation in the Qiskit platform stores and calculates quantum states through the state form in Qiskit. There are some differences in the results obtained by the two implementation methods, as shown in Table 6.3 .

The table presents the performance on the CARDIO3 dataset, with the seven dimensions indicating the number of quantum bits used in Qiskit, as shown in parentheses. Although both methods show good performance across different dimensions, the overall results are quite comparable. Therefore, our method demonstrates strong performance in both environments.

Table 6.3: Accuracy comparison between different platforms (case on CARDIO3)

	128(7)	256(8)	512(9)	1024(10)	2048(11)	4096(12)	8192(13)
Qiskit	86.38	88.26	91.07	90.37	89.90	90.84	90.37
Classical	86.15	87.09	90.61	90.31	90.28	90.28	90.84

6.6.3 Time Consumption

Table 6.4: Time consumption (in seconds) per epoch of different models at various dimensions. (case on CARDIO3 Ours-C means calculate the similarity traditionally)

	512(9)	1024(10)	2048(11)	4096(12)	8192(13)
Vanilla	0.0520	0.0626	0.102	0.2988	0.543
OnlineHD	0.0595	0.0832	0.0965	0.1001	0.120
LeHDC	0.0951	0.108	0.194	0.349	0.727
CompHD	0.0522	0.0606	0.107	0.316	0.626
Ours-C	0.130	0.195	0.221	0.256	0.457

To compare the complexity of our proposed method with existing approaches, we intuitively observe time consumption as a key metric. Table 6.4 presents the time consumption per epoch (in seconds) for different models in various dimensions, using the CARDIO3 dataset as a case study. Our proposed method shows a higher time consumption compared to some existing approaches. However, it is important to emphasize that the increase in time consumption is not exponential. This validates that the computational complexity of our method on classical platforms remains comparable to current approaches and meets the low-complexity requirements of HDC. Furthermore, it is worth noting that if our method is implemented on actual quantum hardware or platforms, the time consumption is expected to decrease exponentially. This is due to the inherent advantages of quantum computing, such as parallelism and superposition, further demonstrating the potential efficiency improvements that can be achieved in future quantum implementations.

6.7 Conclusion

In this chapter, a hyperdimensional computation algorithm based on quantum state encoding is proposed. With this algorithm, hyperdimensional information of features can be extracted more efficiently. We both verify it theoretically and implement it in two ways: one in a classic Python environment, and the other experimentally in a Qiskit environment. The experimental results of the two implementation methods on different datasets are comparable to or even better than current mainstream state-of-the-art hyperdimensional computing algorithms. However, the experimental effect of the algorithm relying on the Qiskit environment is slightly different from that of the classical method, which may be due to the difference in simulation accuracy and the randomness of quantum simulation itself, so it is necessary to obtain a more accurate average value through multiple statistics. In addition, the storage and computing resources required by the method in quantum space are much lower than those of traditional hyperdimensional computing, and the reduction advantages are exponential. However, in both original Python and Qiskit environments, our experiments are based on simulations. In the future, we will aim to implement and test this approach in a real quantum environment.

Chapter 7

Quantum-enhanced Hyperdimensional Computing

Contents

7.1 Overview

In Chapter 6, we have gained a certain understanding of hyperdimensional computing. Hyperdimensional Computing (HDC) is a robust computing framework inspired by human cognition, characterized by performing simple and efficient operations in high-dimensional vector Spaces. We have also proved the feasibility of conducting high-dimensional computations in Hilbert space in Chapter 6. However, only theoretical calculations and simulations on traditional devices were carried out, but an end-to-end quantum-level high-dimensional calculation was not achieved. This chapter proposes quantum-enhanced Hyperdimensional Computing (QeHDC), which extends the classical HDC by leveraging the properties of quantum mechanics to enhance computational efficiency. In this chapter, we propose a novel quantum HDC framework, which adopts a one-time training method and effectively projects classical data onto quantum amplitude states by using sine and quantum coding. This part is also explained in Chapter 6, but we made relevant improvements. Our framework introduces an innovative quantum binding operation based on reference states, which is implemented through quantum circuits. Furthermore, we propose a superclass generation strategy based on the density matrix, which effectively extracts key quantum state features by using eigenvalue decomposition, thereby achieving a more accurate and robust class representation. Experimental evaluations conducted on standard benchmark datasets indicate that, compared with traditional classical and existing quantum enhancement methods, our method has superior performance, robustness to noise, and computational feasibility. The results highlight the practical advantages and potential of quantum HDC in quantum enhanced classification tasks and pave the way for the future development of quantum-inspired computing paradigms.

Among them, in the related work and basic knowledge sections, although they are similar to many parts of Chapter 6. In order to be able to read this chapter independently better, we still re-describe the relevant content.

7.1.1 Problem Statement

Hyperdimensional Computing [98], also known as Vector Symbolic Architectures [122], is a computational paradigm that leverages high-dimensional (typically larger than

10,000 dimensions) vectors for information representation, processing, and learning. Inspired by cognitive science and neuroscience principles, HDC aims to mimic how the human brain encodes, stores, and retrieves information by operating in high-dimensional space. Instead of using traditional low-dimensional numerical representations, HDC uses high-dimensional vectors. Each vector represents a concept in a distributed way.

Information is processed through algebraic operations. These include bundling (element-wise addition with normalization), binding (component-wise multiplication or XOR), and permutation (reordering of dimensions). These operations allow HDC to efficiently perform classification, pattern recognition, associative memory, and temporal sequence modeling tasks. One of the key advantages of HDC is its robustness to noise, as high-dimensional representations inherently provide redundancy and fault tolerance, making it well suited for real-world applications with noisy data. In addition, HDC is computationally efficient because it relies primarily on vector arithmetic rather than complex matrix operations, making it well suited for low-power, real-time, and edge computing scenarios such as embedded systems, IoT devices, and neuromorphic computing platforms. Unlike traditional deep learning methods that require extensive training using backpropagation and gradient descent, HDC can learn through a fast, one-time encoding and updating mechanism, which makes it particularly effective for applications with limited training data.

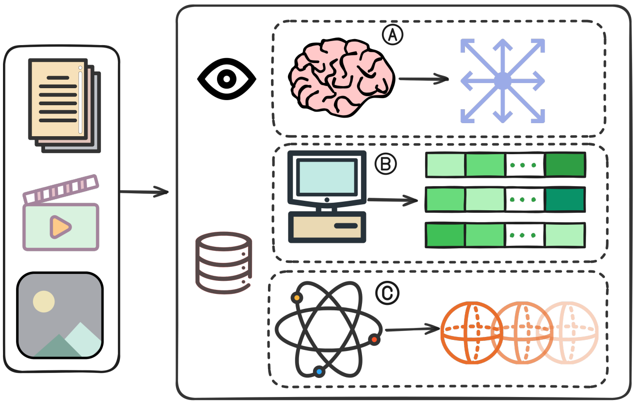


Figure 7.1: Conceptual representations of high-dimensional computing (HDC) across three paradigms. (A) Brain-inspired representation and associative reasoning;(B) Classical computing using binary or hypervector structures;(C) Quantum-enhanced HDC utilizing quantum states and Bloch sphere representations for encoding and binding.

Although HDC has demonstrated substantial advantages, it still faces serious computing and storage bottlenecks with the increase in data complexity and scale. Therefore, using quantum enhancement to break through the traditional architecture and mechanism of HDC is the motivation for this study.

QeHDC breaks through traditional architectural limitations via three core mechanisms. **Dimensional Compression:** Quantum states utilize the geometric properties of the Bloch sphere to represent traditional vector space. By using only $log_2 N$ qubits, they achieve the same representation capability as classical high-dimension spaces while enabling exponential dimensional compression. **Computational Architecture:** Quantum Circuit replaces traditional $O(N^2)$ complexity matrix operations with quantum gate operations based on unitary evolution. Through superposition-based parallel processing and entanglement gates, these circuits facilitate efficient feature interactions. **Representation transformation:** Quantum system overcomes the constraints of classical linear algebra by constructing nonlinear entangled states using quantum circuits. By leveraging quantum interference effects, they achieve superlinear coupling between features, enhancing the interaction dynamics beyond classical approaches.

7.1.2 Main Contribution

Based on the above challenges and motivations, our proposed Quantum-enhanced hyperdimensional computing mainly achieves the following contributions.

- First, we introduce a **two-step hybrid encoder**. It initially maps the original data features into a high-dimensional space using random projection and trigonometric transformations. Then, it expresses the high-dimensional vectors in Bloch space using only a few qubits. This implicitly reduces the dimension of data representation. The proposed encoding method lowers the computational resources overhead required for data processing while making data encoding more suitable for quantum computing implementation.

- Second, we develop an innovative **Quantum Binding** operation called **Referencec-State-Based Quanutm Binding**. Traditional HDC binding operations often rely on simple XOR, circular convolution, or Hadamard product. In contrast, our quantum binding and bundling approaches leverage a predefined quantum reference estate as a binding anchor. By designing specialized quantum circuits, such as the controlled rotation gate (CRX) and the entanglement gate (CX), our method achieves a more efficient data association and representation. This reference-state-based binding not only enhances the expressiveness of high-dimensional vectors but also provides a natural operation mode that is well-suited for quantum hardware implementation.

- Third, we propose a Density-Matrix-Based template state construction method, known as **Density-Matrix-Based Superclass Generation**. This method decomposes the density matrix of quantum states and selects key feature vectors to construct a representative template state (SuperClass) for each class. By capturing core features from the data more accurately, this approach improves the model's robustness and classification accuracy.
- Additionally, we built a complete classification experimental verification system based on the IBM quantum computing framework Qiskit. First, we complete the classification experimental modeling in high-fidelity quantum noise environment in Qiskit Aer Simulator. We then carry out further confirmatory experiments on a small sample of physical qubits (QPUs) on IBM processors. It is worth noting that despite the essential differences in the calculation principle and noise environment, we still get the corresponding results. This fully proves that this scheme has a unique cross-platform adaptability, which can not only validate and optimize the algorithm in the existing classical computing system but also seamlessly connect with the quantum computing unit in the NISQ era to realize prototype verification.

Through these innovations, our QeHDC approach overcomes the efficiency bottleneck of traditional HDC when processing large and complex data. It provides a new idea and a feasible solution for the practical application of quantum computing in machine learning.

7.1.3 The Outline of the Chapter

The remaining parts of the Chapter is organized as follows. Section 7.2 presents the background, including HDC-related work and quantum computing basics. Section 7.3 introduces the overview framework of our proposed method and detailed designs of each component. Section 7.4 shows the experiments set-up and data in detail. Section 7.5 provides the results on classification tasks and the comprehensive evaluation of our approach. The final section 7.6 concludes the Chapter 7 and envisions the direction of future work.

7.2 Backgroud

7.2.1 Hyperdimensional Computing

HDC is originally conceived in the 1990s as cognitive models that depend on very high dimensionality and randomness [98]. It is inspired by the way the human brain represents and processes information. HDC has the advantages of robustness, fault tolerance, and efficient parallel computing by using high dimensional random vectors to represent and process information [123]. In recent years, with the rapid development of artificial intelligence and computing hardware, HDC has regained widespread attention from researchers. In 2017, Rahimi et. al. proposed HDC for machine learning tasks, demonstrating its high efficiency and low complexity in simple text classification [124]. Subsequently, Hersche et al. evaluated a variety of HDC coding strategies in the motor imagination classification of EEG, demonstrating high energy efficiency on embedded devices [125].

In recent years, many innovative HDC methods have emerged. AdaptHD [102] can dynamically adapt to data changes and effectively deal with non-stationary data environments. The real-time incremental learning framework built by OnlineHD [103] can support continuous updating of streaming data tasks. In terms of representation optimization, CompHD [104] adopts a composite coding strategy to enhance feature interaction, LeHDC [101] optimizes high-dimensional space mapping through the learnable embedding mechanism, and NeuraldHD [106] deeply integrates a neural network architecture to achieve joint representation optimization. In the field of computational efficiency optimization, the dynamic coding technology designed by DistHD [99] not only improves accuracy but also strengthens noise immunity. QuantHD [100] adopts low bandwidth representation to significantly reduce the storage overhead on the premise of ensuring performance, while SparseHD [105] reduces the computational complexity based on sparse coding theory. For specific application scenarios, BioHD [126] developed a biometric coding protocol to enhance the sensitivity of genome analysis. The graph structure coding operator constructed by GraphHD [127] highly improved the accuracy of node classification and SecureHD [128] ensured the privacy of medical data processing through the homomorphic encryption protocol. In terms of application, HDC has been widely used in a variety of complex tasks in recent years, including biological signal analysis [129, 130], natural language processing [131], robot perception and navigation [132, 133], and remote sensing image processing [134].

In summary, HDC is gradually becoming an important research direction in the field of artificial intelligence due to its computational simplicity, robustness, and compatibility with emerging computing architectures, especially in the context of quantum computing convergence, which has a broader prospect.

7.2.2 Quantum Computing Basics

Quantum computing is a computational model based on the principles of quantum mechanics, which uses qubits as the basic computing unit. Unlike classical bits, which exist strictly in either 0 or 1, qubits can both be in state $|0\rangle$ and $|1\rangle$. Qubits can exist in a **superposition** [124] of both states, represented mathematically as:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle, \text{where } \alpha, \beta \in \mathbb{C}, |\alpha|^2 + |\beta|^2 = 1 \tag{7.1}$$

α and β are the amplitudes of the complete complex probabilities, indicating the probabilities of obtaining $|0\rangle$ and $|1\rangle$, respectively, when measuring. Due to the existence of superposition states, quantum computing can simultaneously process multiple computation paths in the most exponential level of state space, thus increasing computing power.

Non-local correlations can be formed between multiple qubits through entanglement, where the measurement of one qubit instantaneously affects the state of the other one, even if they are far apart. For example, the **Bell State** [135] of two qubits is one of the most basic entangled states:

$$|\Phi^+\rangle = \frac{|00\rangle + |11\rangle}{\sqrt{2}} \tag{7.2}$$

Entanglement is particularly important in high-dimensional computing because it allows data to form more complex relationships between multi-qubit systems, improving the ability to represent information. Quantum computing manipulates qubits through a series of quantum gates, which are unitary transformations that guarantee the reversibility of quantum information. Typical quantum gates include the Hadamard gate (H), Pauli-X (bit-flip) gate, rotation gates (RX, RY, RZ), and controlled gates such as controlled-NOT (CNOT or CX gate), which flip the state of a target qubit conditioned on the state of a control qubit. For example, the widely used H gate generates superposition states from computational basis states:

$$H|0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}, \quad H|1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}} \tag{7.3}$$

Moreover, practical quantum systems typically cannot be described completely by pure states due to noise and environmental interaction. Thus, introduce the concept of **density matrices**. A density matrix [136] is a generalized mathematical tool used to represent mixed states (statistical ensembles) of a quantum system:

$$\rho = \sum_i p_i |\psi_i\rangle \langle\psi_i|, \quad \sum_i p_i = 1 \tag{7.4}$$

where p_i represents the probability that the quantum system exists in the state ψ_i. Density matrices are crucial for quantum computing tasks involving statistical modeling and noisy environments, providing a powerful way to encode complex statistical information beyond pure-state representation. In this chapter, the density matrix method is used to construct the Superclass vector, and the core information of the class is extracted by feature decomposition to improve the classification model's generalization ability. Finally, information encoded in quantum states is obtained through the operation of **measurement**. Measurement collapses the quantum superposition into one of the classical basis states with probabilities determined by the amplitude squared of each basis state. Specifically, measuring a qubit $|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ yields outcomes with probabilities:

$$P(|0\rangle) = |\alpha|^2, \quad P(|1\rangle) = |\beta|^2 \tag{7.5}$$

In our study, measurement is used in the classification decision process of quantum high-dimensional computation. The data features of hyperdimensional representation are extracted by measurement, and the final pattern recognition task is performed using the classical computational framework. In summary, quantum computing provides the most new information representation and computation framework; through the superposition of quantum states, entanglement, statistical representation of density matrices, operation of quantum circuits, and measurement mechanisms, it can enhance the feature representation of hyper-dimensional computing. Based on these quantum properties, a novel Quantum-enhanced Hyperdimensional Computing (QeHDC) method is proposed to encode, bind, and classify high-dimensional data more efficiently.

7.3 Quantum-enhanced Hyperdimensional Computing

In this chapter, we propose a Quantum-enhanced Hyperdimensional Computing (QeHDC) system that leverages quantum computational properties to enhance the efficiency and expressiveness of classical HDC. The system comprises four core modules. First, a 2-step classical-quantum hybrid encoder transforms classical data into quantum states. Then, a Quantum HD Operator performs HDC operations like binding and construction using quantum circuits. Superclass construction builds a density matrix over bound states and extracts features via eigen-decomposition. Finally, classification is achieved

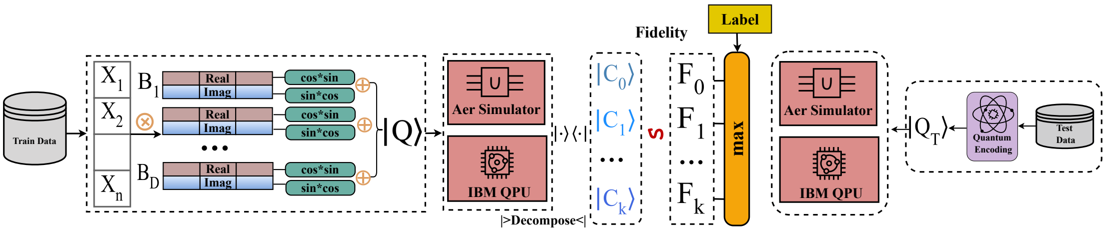


Figure 7.2: The overall structure of QeHDC Framework

by comparing input states with learned superclasses using quantum state fidelity. We will introduce each of these parts in detail below. Before that, we define a system input sample as:

$$D = (X_i, y_i)_{i=1}^{N} \tag{7.6}$$

where $X_i \in \mathbb{R}^d$ is the feature vector of dimension d and y_i is the class label, typically encoded as a categorical variable.

7.3.1 2-step Hybrid Encoder

The core of the coding process is to efficiently map classical eigenvectors to quantum states while preserving the structural information of the input data. The first step is a classical transformation. Given a classical input vector $X \in \mathbb{R}^d$, we perform a random projection using a randomly initialized weight matrix $W \in \mathbb{R}^{n*d}$, where $n = 2^{N_{\textbf{qubits}}}$ represents the target quantum dimension. The transformation is computed as:

$$Z = WX \tag{7.7}$$

To introduce nonlinearity and phase variation in the coding process, we introduce a bias term $B \in \mathbb{R}^n$ whose values are uniformly sampled from the interval $[0, 2\pi]$ forming a transformed projection:

$$Z' = Z + B \tag{7.8}$$

To convert the projected values into **quantum-compatible** complex-valued presentations, we Then, the real part and imaginary part are mapped separately using the sinusoidal transformations:

$$\Phi_{\text{re}} = cos(Z')sin(Z), \quad \Phi_{\text{im}} = sin(Z')cos(Z) \tag{7.9}$$

This formulation which we called **Cross-Multiplicative** achieves the same uniform phase angle distribution while using only single weight matrix, making it the most efficient approach (See Appendix A for explanation) The final quantum state representation is given by:

$$\Psi = \frac{\Phi_{\text{re}} + i\Phi_{\text{im}}}{||\Phi_{\text{re}} + i\Phi_{\text{im}}||} \tag{7.10}$$

Compared to other encoding schemes, this approach ensures a well-distributed representation in the complex space, enhancing separability and stability in quantum computation. Further theoretical analysis, including its impact on data distribution, is provided in Appendix 9.1.

7.3.2 QeHDC Operation

Quantum operations form the core of our method by enabling transformations and interactions between encoded quantum states. Once classical data is converted into quantum representations via Cross-Multiplicative Encoding, a set of quantum operations is applied to enhance, bind, and extract relevant information. The goal of these operations is to exploit quantum properties such as superposition and entanglement to improve computational efficiency and feature representation. We focus on the **binding operation**, which is critical in representing relationships between encoded quantum states. Unlike classical models that rely on matrix multiplications or convolutional operations, our approach leverages quantum state interactions to achieve a compact and efficient representation of information.

In the Quantum hyperdimensional system, direct implementation of classical HDC binding operation causes an exponential increase in the required qubits (See Appendix 9.2). To address this, we propose a specifically compressed binding method, employing a quantum circuit for efficient information binding and compression. Consider the encoded quantum state for a single input sample as :

$$|\Phi_{\text{in}}\rangle = \sum_{i=0}^{2^N-1} \alpha_i |i\rangle, \quad \sum_i |\alpha_i|^2 = 1 \tag{7.11}$$

Also define a predefined reference quantum state as:

$$|\Phi_{\text{ref}}\rangle = \sum_{j=0}^{2^N-1} \beta_j |j\rangle, \quad \sum_j |\beta_j|^2 = 1 \tag{7.12}$$

We initialize these two states into two separate quantum registers, each containing

n qubits, resulting in the initial state:

$$|\Psi_{\text{init}}\rangle = |\Phi_{\text{in}}\rangle \otimes |\Phi_{\text{ref}}\rangle \tag{7.13}$$

Next, we apply Controlled-RX (CRX) gates between each pair of corresponding qubits (q_i, q_{i+n}) with rotation angle $\pi/4$:

$$CRX(\pi/4) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes e^{-i\frac{\pi}{8}X} \tag{7.14}$$

At this point, the state evolves into:

$$|\Psi'\rangle = \prod_{i=0}^{n-1} CRX(\pi/4)\, |\Psi_{\text{init}}\rangle \tag{7.15}$$

Subsequently, apply single-qubit RY rotations on each qubit in the reference register for additional nonlinear compression:

$$RY(\pi/4) = e^{-i\frac{\pi}{8}Y} \tag{7.16}$$

The state then becomes:

$$|\Psi''\rangle = \prod_{i=0}^{n-1} RY_i(\pi/4)\, |\Psi'\rangle \tag{7.17}$$

To further enhance the binding and compression effect, we add chain entanglement operations (CX gates) within the reference register. The final highly entangled and compressed quantum representation is:

$$|\Psi_{\text{comp}}\rangle = \prod_{\{q_a, q_b\} \in Q} CX_{q_a, q_b}\, |\Psi''\rangle \tag{7.18}$$

Where $Q = \{n, n+1, ..., 2n-1\}$. For all samples within the same class c (assuming a total of m samples), we perform the above binding and compression process individually, obtaining:

$$\{|\Psi^{(1)}_{\text{comp}}\rangle, |\Psi^{(2)}_{\text{comp}}\rangle, ..., |\Psi^{(m)}_{\text{comp}}\rangle\} \tag{7.19}$$

The binding circuit described above is a fully entangled structure, meaning any two qubits in the latter half are directly entangled. The circuit for the example with 4 qubits is shown in Fig. 3.

We also employed some structures with lower complexity. Please refer to Appendix 9.3. Through these detailed steps, we achieve quantum state binding and compression

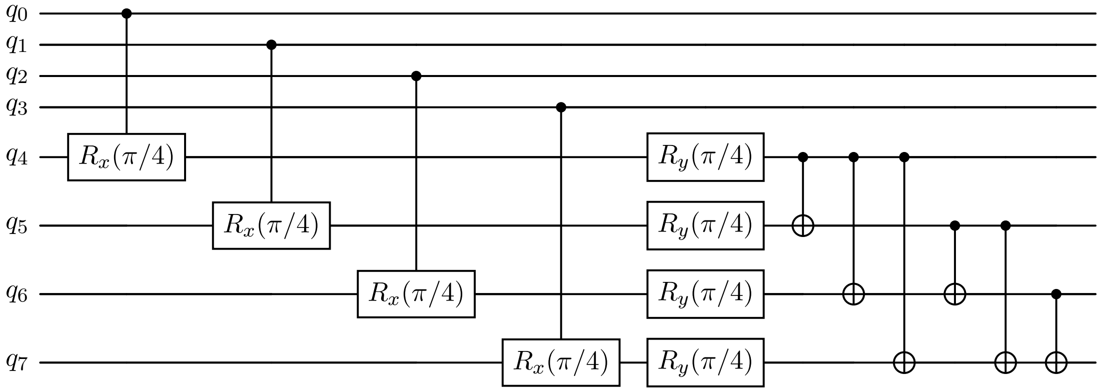


Figure 7.3: Quantum Bind Circuit

at both sample and class level, significantly avoiding the exponential qubit resource increase caused by the traditional classical HDC method. This approach holds substantial potential and expansion possibilities in future practical applications. In our proposed compressed binding method, we implement quantum state binding compression using specifically designed quantum circuits. In our practical study, we employ two approaches for realizing and verifying our quantum bound states:

- Ideal Quantum State simulation via the Qiskit Aer Simulator (no measurement required)
- Physical experiments on IBM's Quantum Processing Unit (QPU), requiring measurement and quantum state tomography for reconstruction.

It should be noted that using the ideal state vector from the Aer simulator applies only to theoretical analysis and algorithm validation. On real quantum devices (e.g., IBM QPU), measurement and quantum state tomography reconstruction are essential steps. Our research encompasses both ideal simulations and real-device experiments to comprehensively demonstrate both the theoretical performance and the practical feasibility of our quantum binding algorithm.

7.3.3 SuperClass Template Generate

In our method, using the previously introduced compressed binding method, each class contains numerous samples represented by bound quantum states. Therefore, from multiple sample states with the same class, we must extract a representative and stable quantum state, termed the **SuperClass Template State**. Suppose a class c has m data samples, after obtaining a holistic representation for the class, we need to comprehensively capture the statistical characteristics of these sample states. The quantum density matrix provides the ideal mathematical tool for this purpose:

$$\rho_c = \frac{1}{m} \sum_{i=1}^{m} |\Psi^i_{\text{comp}}\rangle \langle \Psi^i_{\text{comp}}| \tag{7.20}$$

The density matrix ρ_c is essentially a statistical representation of all bound states in class c. It reflects information from all states and reveals intrinsic correlations among them. The density matrix has dimensions $2^{2n} \times 2^{2n}$. To address the computational challenge due to dimensionality, we employ two strategies based on matrix dimension. Directly construct the density matrix using standard methods and perform eigen-decomposition:

$$\rho_c |\phi_j\rangle = \lambda |\phi_j\rangle \tag{7.21}$$

We choose the eigenvector corresponding to the largest eigenvalue $\lambda_{\max}$ as the representative state for the class:

$$|\Phi_c^{\text{template}}\rangle = |\phi_{\max}\rangle , \quad \rho_c |\phi_{\max}\rangle = \lambda |\phi_{\max}\rangle \tag{7.22}$$

Through the described SuperClass template state generation approach, we effectively extract representative and stable class-level quantum representations from multiple bound states.

7.3.4 Single-Pass Training

In our method, we use one-pass training. Thanks to the characteristics of the high-dimensional computing framework, the model can be constructed with only one data traversal, avoiding the computational cost and parameter tuning complexity caused by multiple iterative optimizations in traditional machine learning methods. This approach not only significantly reduces the occupation of quantum resources, but is more suitable for the current environment with limited quantum hardware resources, and effectively improves the actual deployment capability and generalization performance of the algorithm.

7.3.5 Classification

In the classification phase, for each unseen classical data sample X_{test}, we first encode it into a quantum state $|\Phi_{X_{\text{test}}}\rangle$ using the Cross-Multiplicative Encoding approach described previously. Subsequently, the encoded state is bound to the same predefined reference quantum state $|\Phi_{\text{ref}}\rangle$ via the compressed Binding process, obtaining a quantum bound state:

$$|\Psi_{\text{test}}\rangle = \mathcal{U}_{\text{bind}}\left(|\Phi_{x_{\text{test}}}\rangle, |\Phi_{\text{ref}}\rangle\right).$$

Classification is then performed by calculating the quantum state fidelity between the bound test state $|\Psi_{\text{test}}\rangle$ and each pre-computed SuperClass template state $|\Psi_{c_j}^{(\text{template})}\rangle$:

$$F\left(|\Psi_{\text{test}}\rangle, |\Psi_{c_j}^{(\text{template})}\rangle\right) = \left|\langle\Psi_{\text{test}}|\Psi_{c_j}^{(\text{template})}\rangle\right|^2.$$

The class prediction for the test sample is obtained by selecting the SuperClass template that yields the highest fidelity value:

$$c^* = \arg\max_{c_j} F\left(|\Psi_{\text{test}}\rangle, |\Psi_{c_j}^{(\text{template})}\rangle\right).$$

On ideal quantum simulators (e.g., Qiskit Aer), the fidelity is directly computed using state vectors, whereas on real quantum hardware (e.g., IBM QPUs), quantum state tomography is first performed to reconstruct the density matrix ρ_{test}. Subsequently, fidelity is calculated as:

$$F(\rho_{\text{test}}, |\Psi_{C_j}^{(\text{template})}\rangle) = \langle\Psi_{C_j}^{(\text{template})}|\rho_{\text{test}}|\Psi_{C_j}^{(\text{template})}\rangle.$$

This quantum fidelity-based classification approach naturally leverages quantum coherence and quantum state similarity, providing robust, noise-resistant classification performance that is particularly suitable for resource-constrained quantum computing environments.

7.4 Experiments

In this section, we evaluate the performance and effectiveness of our proposed QeHDC framework. We provide details on the datasets, quantum simulation, and hardware environments, as well as the parameter settings shared across all experiments.

7.4.1 Dataset

We evaluate our method on three widely used datasets: **ISOLET** [118], **MNIST** [117], and **UCI HAR** [119]. The quantitative details of the datasets are shown in Table 7.1.

The ISOLET dataset consists of spoken letter recordings, where each sample corresponds to one of the 26 English alphabet letters spoken by different speakers. Each

Table 7.1: Quantitative details of the datasets

Dataset	# Features	# Classes	# Traing Samples	# Test Samples
MNIST	784	10	60000	10000
ISOLET	617	26	6238	1559
UCI HAR	561	6	7352	2947

instance is represented by 617 acoustic features. Due to its high-dimensional input and large class space, ISOLET presents a challenging scenario for scalable classification. The MNIST dataset is a benchmark image classification task composed of handwritten digits from 0 to 9. Each image is 28×28 pixels, flattened into a 784-dimensional vector. MNIST is widely used for evaluating pattern recognition and machine learning algorithms.

Table 7.2: Binary Classification Results (2 Classes) for Three Datasets by Method and Dimension

Method	ISOLET-2Classes			MNIST-2Classes			UCI HAR-2Classes		
	16D	32D	64D	16D	32D	64D	16D	32D	64D
Vanilla	72.78±8.12	76.11±5.11	76.39±4.37	78.50±8.90	90.12±2.04	94.47±0.92	60.15±1.64	67.63±2.12	70.70±2.79
AdaptHD	67.50±13.96	68.33±6.49	79.72±8.20	77.37±8.20	90.10±2.82	96.71±0.83	57.15±1.96	62.50±8.14	70.29±1.45
OnlineHD	**88.89±4.53**	95.28±0.39	98.89±1.04	**95.19±1.76**	**99.16±0.34**	99.67±0.12	**87.80±4.17**	88.38±4.15	90.90±1.03
NeuralHD	86.39±2.58	95.83±1.80	96.39±0.79	94.44±1.03	98.49±0.38	99.01±0.32	83.52±2.98	87.56±2.56	91.97±1.35
CompHD	55.83±5.14	68.89±1.57	75.83±4.76	72.37±12.39	87.42±7.31	90.76±2.61	57.05±5.21	68.05±2.05	64.84±4.27
SparseHD	66.94±1.42	61.94±4.43	70.83±4.25	68.31±17.31	72.58±2.37	66.49±9.15	57.33±2.83	62.84±6.95	68.39±4.71
QuantHD	55.00±3.60	75.56±6.85	74.17±1.36	70.50±19.59	83.04±0.60	82.81±6.94	55.33±5.70	61.67±12.14	67.56±6.21
LeHDC	59.44±5.15	56.39±1.04	65.00±5.31	51.06±5.21	61.61±1.58	67.30±11.97	52.36±2.04	55.05±0.83	53.43±2.39
	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits
Ours-Aer	86.35±1.11	**96.05±0.57**	**98.97±0.49**	89.99±0.62	95.61±1.34	**99.97±0.83**	87.59±1.36	**89.72±1.43**	**92.04±0.26**

The UCI HAR (Human Activity Recognition) dataset contains 561-dimensional feature vectors extracted from smartphone sensor signals. The data represents six human activities (e.g., walking, standing, lying down) collected from multiple subjects. It is a compact but structured dataset that tests the model's ability to distinguish between temporal movement patterns. These datasets vary in feature dimension, number of categories, and sample complexity, providing different assessment environments. To investigate the extensibility of our model, we performed a classification task ranging from 2 classes to 10 classes, rather than using a complete set of classes (for example, all 26 letters in ISOLET or all 10 numbers in MNIST). We observed that the full class classification task led to a significant decline in performance, so we adopted a progressive evaluation strategy that better reflects the actual capabilities of current quantum systems.

7.4.2 Low-Dimensional HDC and Quantum Encoding

While our approach is inspired by traditional HDC, we intentionally work in a significantly lower dimension([16,32,64]) to reflect the limitations of current quantum hardware. In the classical HDC literature, extremely high sizes (up to 10,000) are often used to ensure robustness and orthogonality. In contrast, our QeHDC performs well in low dimensions using the expressiveness of quantum states. For fair comparison, all HDC baselines are also constrained to the same vector dimension. This highlights a key difference in the design. While traditional techniques rely on the force dimension, our approach seeks to preserve a quantum-efficient representation of core HDC principles.

7.4.3 Implementation Details

Experiments are conducted in the following environments:

- **Qiskit Aer Simulator:** [121] An ideal, noiseless quantum simulation environment used for baseline accuracy and fidelity evaluation. The majority of our experiments are conducted using the Aer simulator, as it allows scalable, noise-free validation of our method across different datasets and class counts.

- **IBM QPU:** [137]A small scale of experiments is deployed on real IBM QPU backends to assess the practical feasibility of our framework under hardware constraints. Quantum state tomography is used to reconstruct output states due to the non-availability of state vectors on real hardware. Basic error mitigation techniques are applied where applicable. For comparison, the same scale and data samples used on IBM devices are also evaluated using the Aer simulator to establish a controlled benchmark.

- **Classical Baselines**: For traditional HDC and classical parts of our methods, we use TorchHD [138] implementations. All classical experiments are run on a workstation equipped with a NVIDIA RTX A6000, GPU acceleration is used for efficient batching in TorchHD. Dimensionality (16, 32, 64) is matched with the quantum model, and all baselines are evaluated on the same training/testing splits as the quantum system.

All quantum operations, including encoding, binding, and fidelity computation, are implemented using Qiskit. State preparation and circuit compilation are optimized for each backend. In real-device executions, we use 8192 shots and transpile circuits to the native gate set of the selected backend. Fidelity measurements are obtained by reconstructing the output density matrices via tomography. Hyperparameters, such as the number of qubits ({4,5,6}), repeats (5), and choice of reference state, are fixed across experiments unless specified in dedicated analysis sections.

7.5 Results and Analysis

In this section, we present the key findings from our experiments. We focus on evaluating classification accuracy and the contribution of each core component in the QeHDC framework. All results are organized by task complexity and dataset, followed by comparative studies to highlight the advantages and limitations of our approach.

7.5.1 Overall Performance

Table 7.2 provides a comprehensive comparison of our QeHDC framework against various classical hyperdimensional computing (HDC) baselines, such as VanillaHD, AdaptHD, OnlineHD, NeuralHD, CompHD, SparseHD, QuantHD, and LeHDC—across three benchmark datasets (ISOLET, MNIST, and UCI HAR) under different dimensional constraints (16D, 32D, and 64D). The dimensions in the classical methods correspond respectively to quantum states encoded by 4, 5, and 6 qubits in our approach.

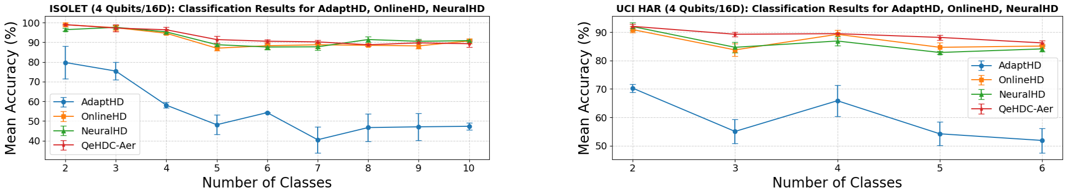
From these results, our QeHDC consistently achieves robust and superior classification accuracy. Specifically, on the ISOLET binary classification task, our model demonstrates accuracy that increases significantly with the dimension, reaching 98.97% at 6 qubits (64D), clearly outperforming all classical baselines except OnlineHD, with which it achieves comparable performance. In the MNIST binary classification scenario, our approach further excels, reaching an accuracy of 99.97% at 6 qubits—surpassing or matching the best classical results. Likewise, for the UCI HAR dataset, which presents complex temporal feature patterns, our QeHDC method attains an accuracy of 92.04% at 6 qubits, which notably exceeds most classical baselines.

Importantly, even at lower-dimensional settings , our method maintains competitive performance. This highlights the effective expressive capacity of our quantum-inspired encoding and compression-based binding mechanisms. Such robust performance under dimensional constraints is particularly relevant for near-term quantum computing applications, where resource availability (e.g., number of qubits) is significantly limited.

Beyond binary classification tasks, our method also demonstrates promising results in more complex multi-class classification settings, systematically evaluated from 3-class up to 10-class problems. These additional experiments confirm that the QeHDC framework consistently retains strong classification capability across increasing classification complexity and various datasets. Detailed multi-class classification results and further analysis can be found in the Appendix 9.4.

Overall, these extensive experimental evaluations indicate that our quantum-based HDC approach not only competes favorably against classical HDC methods but also shows distinct advantages regarding efficiency, scalability, and robustness across different classification scenarios.

7.5.2 Limitation Analysis



(a) Performance on ISOLET Dataset

(b) Performance on UCI HAR

Figure 7.4: Mean classification accuracy of AdaptHD, OnlineHD, NeuralHD, and QeHDC-Aer on (a) ISOLET and (b) UCI HAR datasets with varying numbers of classes.

From the figures Fig. 7.4a and Fig. 7.4b, it is evident that our proposed Quantum Hyperdimensional Computing framework (QeHDC-Aer) consistently demonstrates stable and superior classification accuracy, around 90%, across varying classification complexities (from 2-class up to 10-class scenarios) on both ISOLET and UCI HAR datasets. In contrast, classical methods such as AdaptHD experience significant accuracy degradation and instability as the number of classes increases, highlighting the robustness of our approach under more complex classification tasks. Moreover, our method achieves performance comparable to, or better than, top-performing classical methods like OnlineHD and NeuralHD, especially notable on the UCI HAR dataset. Other classical baselines (e.g., VanillaHD, CompHD, SparseHD) fail to maintain satisfactory performance at such low-dimensional settings, indicating difficulty in handling complex tasks under limited dimensional constraints. These results clearly demonstrate the strong generalization capability and scalability of our QeHDC framework, emphasizing its advantage over classical HDC approaches in low-dimensional, resource-constrained quantum environments.

7.5.3 Aer Simulation and IBM QPU

To evaluate both the ideal performance and practical feasibility of our proposed QeHDC framework, we conduct experiments using the Qiskit Aer simulator and IBM Quantum

Table 7.3: Small-batch Classification on UCI HAR (4 Qubits)

Method	# Qubits	# Train	# Test	Accuracy	Fidelity (to Aer Ideal)
QeHDC-Aer	4	50	20	0.80	0.98
QeHDC-QPU	4	50	20	0.75	0.87

Processing Units (QPUs). The Aer simulator provides a noise-free quantum environment with access to full quantum statevectors, allowing us to assess the expressiveness and fidelity of our model under ideal conditions, free from decoherence and gate imperfections. In contrast, IBM QPUs introduce realistic limitations such as noise, qubit connectivity constraints, and readout errors, offering a platform to validate the framework under physical quantum computing constraints.

On the Aer simulator, we observed high fidelity and classification accuracy for the quantum states generated after binding and template construction. These results validate the theoretical soundness of our encoding, binding, and template generation procedures. Specifically, the fidelity between the ideal post-binding statevector and the density matrix reconstructed via simulated tomography consistently exceeded 0.98, confirming the accuracy and stability of our tomography pipeline in a noise-free setting.

Quantum State Tomography (QST) plays a critical role in real-device testing where full access to quantum states is unavailable. By preparing and measuring the same quantum state across multiple bases and statistically reconstructing its density matrix, QST enables us to recover the output state from QPU measurements and perform fidelity-based classification. In our workflow, the test state is first encoded and the binding circuit is executed on the QPU. The resulting measurement data is then used to reconstruct a density matrix via QST, from which we extract a quantum representation that is compared with the stored SuperClass template to perform final classification.

Due to the significant cost and scaling complexity of QST and QPU, where the number of measurement circuits increases exponentially with the number of qubits. We adopted a small-batch testing strategy. Combined with device runtime constraints, this makes full-scale experiments infeasible on current hardware. As a representative case, we selected a small batch from the UCI HAR dataset (4-qubit setting), training on 50 samples and testing on 20. We redo this experiment for 5 times and get the average results in Table 7.3.

These results indicate that even under real noise, the quantum states reconstructed from the QPU retain sufficient structure to support meaningful classification, with fidelity and accuracy close to ideal conditions. Notably, the difference in fidelity between the QPU and simulator also quantifies the impact of hardware noise on the final inference performance.

The examples of our binding circuits, representative tomography circuits, and visual comparisons of ideal and reconstructed states are included in the Appendix 9.5.

Overall, this experiment validates the expressiveness of our method under ideal conditions and its deployability on current meso-scale noisy quantum devices (NISQs), providing a practical basis and experimental experience for further expanding the QeHDC model on more powerful quantum hardware platforms in the future.

7.5.4 Ablation Study

In order to evaluate the effect of the coding mechanism on the overall performance of QeHDC systems, we conduct an ablation on the Quantum encoder. The objective of this study is to analyze the effects of different coding strategies on the quality of class templates and the final classification accuracy.

7.6 Conclusion and Future Work

In this chapter, a new framework Quantum-enhanced hyperdimensional computing (QeHDC) is proposed, which combines the basic idea of traditional hyperdimensional computing (HDC) with the ability to express quantum states. We design a classic-quantum two-step hybrid coding mechanism and introduce a reference state-based compression binding method to efficiently generate a compact high-dimensional quantum representation that can be used for classification tasks. At the same time, we construct a SuperClass template by means of density matrix decomposition, so that class abstraction can be realized in quantum representation space.

Unlike traditional HDC models, which typically rely on tens of thousands of dimensional vectors, our quantum scheme is able to operate in significantly low-dimensional (e.g. 64-dimensional) spaces, and achieves competitive and even superior results on multiple classification tasks. Experiments on ISOLET, MNIST, and UCI HAR datasets demonstrate the effectiveness and robustness of the proposed method in simulated environments (Aer) and real devices (IBM QPU). Especially in the small batch test of QPU, the model can still obtain high quantum state fidelity and classification accuracy even if there is hardware noise, which indicates that the method has good usability and deployment potential.

In the future, we will promote our work from the following directions: On the one hand, we will expand this method to more complex task scenarios (such as sequence modeling, online learning), and explore the combination of quantum binding and class incremental update mechanisms. On the other hand, to further optimize the computational cost of quantum state chromatography, or to design a more suitable reasoning mechanism for the hardware structure to adapt to larger-scale quantum systems. In addition, the introduction of learnable quantum encoders and differentiable template generation processes is also expected to further enhance the integration of QeHDC with deep architectures such as neural networks. With the continuous development of quantum hardware, we believe that the framework proposed in this work will show its broad application prospects in larger-scale scenarios and provide a solid foundation for building the next generation of quantum cognitive computing systems.

Chapter 8

Summary and Future Directions

Contents

8.1 Summary of Achievements

This dissertation systematically studies the integration mechanism of quantum computing and machine learning, and applies it to key tasks such as remote sensing data classification and high-dimensional computing, achieving the following main results:

- The theoretical basis of quantum machine learning has been established, covering quantum feature mapping, variable component sub-circuit design, and a hybrid modeling framework combined with classical tasks.

- The VSQL is utilized, which solved the problems of small samples and category imbalance in remote sensing cropland classification and yield prediction. This model combines the quantum shadow mechanism with the variational optimization method and is significantly superior to the classical model in terms of generalization ability.

- A hyperdimensional computing framework based on quantum mechanisms is constructed. For the first time, the mapping of core operations such as binding, superposition, and superclass representation in classical HDC on the quantum state space was achieved, and experimental verification was completed on a real quantum platform.

- The end-to-end experiments were completed on the IBM simulator and the real QPU platform, and the system performance was evaluated on the real farmland remote sensing dataset, verifying the advantages of the proposed model in terms of accuracy, robustness, and data utilization efficiency.

8.2 Research Limitations

Research limitations Although this research has achieved positive results, the following deficiencies still exist:

- The scalability of quantum hardware is limited: Current real quantum devices still have limitations in terms of bit count, coherence time, and noise control, making it difficult to support large-scale model deployment.
- The problem of circuit depth and vanishing gradients: The variable component sub-circuits used may encounter the ”plain phenomenon” in deep structures, resulting in low training efficiency.
- Theoretical understanding is still insufficient: Although the empirical results are good, the strict theoretical proof of achieving quantum advantage in all learning scenarios still needs further research.

8.3 Future Directions

Based on the achievements of this research, the subsequent work can be carried out around the following directions:

- Enhance the scalability and noise resistance of quantum circuits to adapt to more complex real quantum hardware in the future.
- Explore automated Quantum structure search methods, such as AutoQML or Quantum NAS, for the optimal quantum circuit design of different tasks.
- Construct the generalization theoretical system of the quantum model, with a focus on studying its performance mechanism under noisy environments and small sample conditions.
- Expand to multimodal and cross-disciplinary remote sensing tasks (such as crop prediction and pest and disease monitoring), as well as other fields like medical imaging, bioinformatics and scientific computing.

Conduct in-depth research on the interpretability and fusion intelligence mechanism of the quantum-classical model to provide theoretical and technical support for constructing trusted and interpretable hybrid intelligent systems.

Chapter 9

Appendix

9.1 Parameter Usage Comparison

In QeHDC, the choice of encoding strategy significantly affects the **phase angle distribution** in the complex plane. It directly impacts the separability and expressiveness of quantum states. This section compares and analyzes three sinusoidal transformation strategies: Single-Weight Cross-Multiplicative (our method), Dual-Weight Same-Product, and Single-Weight Same-Product. We compare these methods with respect to parameter efficiency, phase angle distribution, and normalization properties.

9.1.1 Phase Angle Concentration in Single-Weight Same-Product

In **Single-Weight Same-Product Encoding**, both the real and imaginary components share the **same weight matrix**, defined as:

$$\mathrm{Re}(\Psi') = \cos(WX + B) \cdot \sin(WX), \tag{9.1}$$

$$\mathrm{Im}(\Psi') = \cos(WX + B) \cdot \sin(WX) \tag{9.2}$$

Since the transformation for both components is identical, their ratio remains constant:

$$\frac{\mathrm{Im}(\Psi')}{\mathrm{Re}(\Psi')} = 1 \tag{9.3}$$

This results in all encoded data points having the same phase angle. Consequently, the data distribution in the complex plane becomes **highly concentrated**, reducing the effectiveness of encoding.

9.1.2 Parameter Efficiency in Cross-Multiplicative Transformation

Single-Weight Cross-Multiplicative Encoding solves this issue by using a single weight matrix while applying different transformations to the real and imaginary components:

$$\mathrm{Re}(\Phi) = \cos(WX + B) \cdot \sin(WX), \tag{9.4}$$

$$\mathrm{Im}(\Phi) = \sin(WX + B) \cdot \cos(WX) \tag{9.5}$$

This ensures a uniform phase angle distribution while minimizing parameter usage.

In contrast, Dual-Weight Single-Product Encoding requires two independent weight matrices to achieve the same effect:

$$\mathrm{Re}(\Psi'') = \cos(W_{\mathrm{re}}X + B) \cdot \sin(W_{\mathrm{re}}X), \tag{9.6}$$

$$\mathrm{Im}(\Psi'') = \cos(W_{\mathrm{im}}X + B) \cdot \sin(W_{\mathrm{im}}X) \tag{9.7}$$

While this method also maintains a uniform phase angle distribution, it doubles the parameter count, increasing computational overhead.

9.2 Compressed Binding

This appendix provides a detailed analysis demonstrating that traditional binding methods used in classical HDC, such as element-wise multiplication or circular convolution, lead to an exponential explosion in the number of required qubits when directly applied to quantum systems. Although theoretically feasible to use quantum state amplitude as classical high-dimensional vectors for binding, such an approach is practically infeasible in quantum hardware. Additionally, this approach has been explored in previous studies, but predominantly through classical binding in classical HDC, which is usually defined via element-wise multiplication or XOR operation simulations rather than quantum implementation. Consider 2 D-dimensional vectors:

$$A = [a_1, a_2, .., a_D], B = [b_1, b_2, ..., b_D] \tag{9.8}$$

Their binding result (element-wise multiplication) is expressed as:

$$C = A \circ B = [a_1 b_1, a_2 b_2, ..., a_D b_D] \tag{9.9}$$

Consider 2 D-dimensional vectors:

$$A = [a_1, 1_2, .., a_D], B = [b_1, b_2, ..., b_D] \tag{9.10}$$

Their binding result (element-wise multiplication) is expressed as:

$$C = A \circ B = [a_1b_1, a_2b_2, ..., a_Db_D] \tag{9.11}$$

This type of binding operation is efficient and easy to perform in a classical system, but if it is implemented directly in a quantum system, it faces huge resource challenges.

In quantum system, a quantum state of n qubits represents a 2^n-dimensional complex vector:

$$|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha |i\rangle , \quad \sum |\alpha|^2 = 1 \tag{9.12}$$

If we want to bind two quantum states using the traditional HDC binding method, we first need to simultaneously represent two full $D = 2^n$-dimensional quantum states in the quantum computer. Each stare requires n qubits and the joint representation of two states thus requires $2n$ qubits. If we bind k quantum states, the toatal number of qubits required scales as:

$$Q_{total} = k \cdot n$$

This implies that if we directly adopt traditional binding methods, adding each additional state linearly increases the required qubits. Given that each quantum state's dimensionality grows exponentially, total computational resources effectively explode exponentially. Theoretically, we could directly use quantum state amplitudes as classical high-dimensional vectors for binding. However, this approach does not genuinely leverage quantum computing advantages, as it simply treats quantum amplitudes as classical data. Hence, it deviates from our quantum algorithmic context, failing to harness quantum parallelism or the computational benefits of quantum superposition. It is worth noting that previous studies have explored this "amplitude-as-vector" binding approach. Nevertheless, these studies typically relied on classical simulations rather than actual implementations on real quantum hardware.

9.3 Bind Circuits Alternatives

Below, we describe two alternative entanglement structures of bind Circuits with lower complexity than the fully-connected entanglement structure.

9.3.1 Linear (Chain) Entanglement

The circuit for the example with four qubits is shown in the figure.

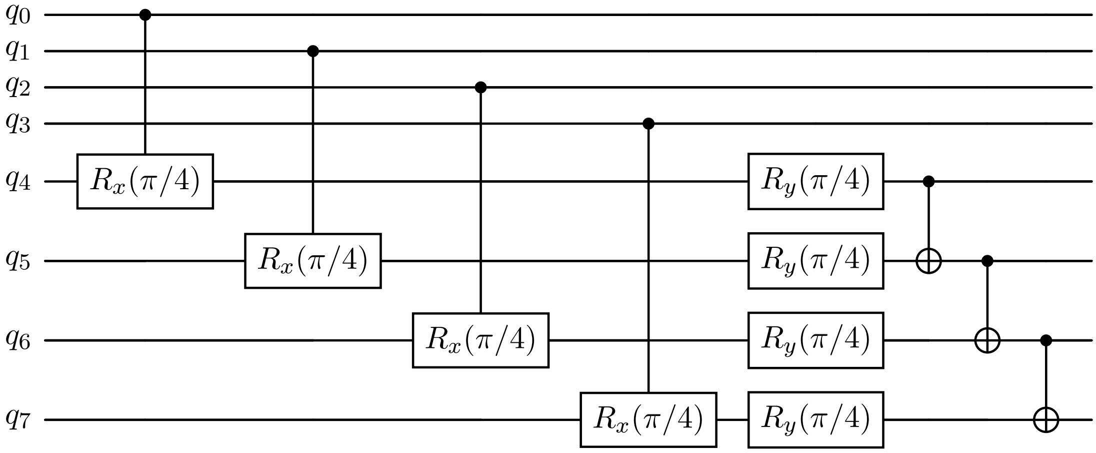


Figure 9.1: Linear Entanglement Binding Circuit with 4 qubits.

In this structure which, qubits in the second half (indexed from n to $2n-1$) are entangled sequentially in a linear (chain-like) configuration. Each qubit is entangled directly with its immediate neighbor, resulting in a simpler, lower-complexity structure.

The **unitary operator** representing this structure is expressed as follows:

$$U_{\text{linear}} = U_{\text{chain}} \cdot U_{\text{param}} \tag{9.13}$$

where

$$U_{\text{chain}} = \prod_{i=n}^{2n-2} CX_{i,\,i+1}, \tag{9.14}$$

and

$$U_{\text{param}} = \prod_{i=0}^{n-1} \left[RY\left(\frac{\pi}{4}\right)_{n+i} \cdot CRX(\theta_i)_{i,\,n+i} \right]. \tag{9.15}$$

9.3.2 Ring (Circular) Entanglement

The circuit for the example with four qubits is shown in the figure.

In this entanglement structure, the qubits in the latter half form a closed loop. Each qubit is entangled directly with its neighbor, including an additional connection between the first and last qubits to complete the ring. This structure is slightly more complex than linear entanglement but simpler than the fully connected entanglement.

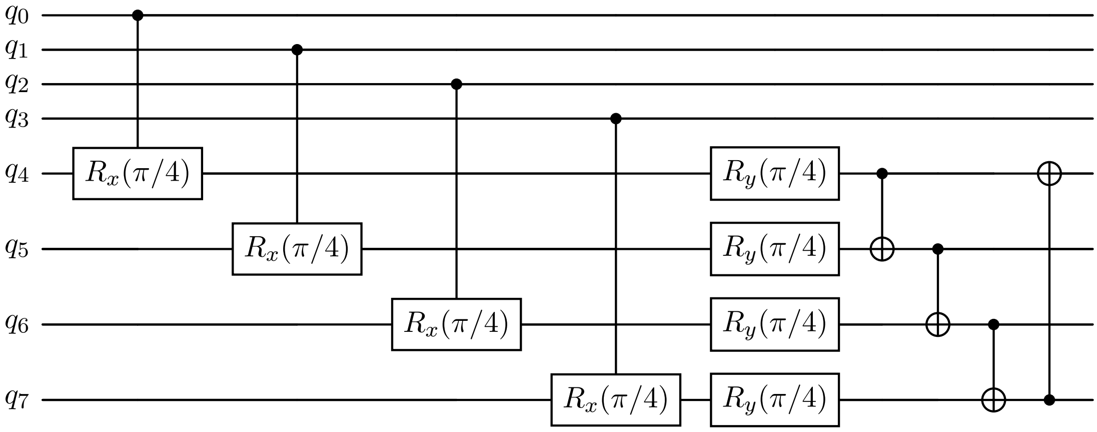


Figure 9.2: Ring Entanglement Binding Circuit with 4 qubits.

The **unitary operator** for ring entanglement is expressed as:

$$U_{\text{ring}} = U_{\text{circle}} \cdot U_{\text{param}} \tag{9.16}$$

where

$$U_{\text{circle}} = \prod_{i=n}^{2n-1} CX_{i,\,((i-n+1) \mod n)+n}, \tag{9.17}$$

and U_{param} is as defined above.

Here, qubits indexed n to $2n-1$ form a circular chain, ensuring each qubit is entangled with exactly two neighbors.

9.4 Complete Results for Classification

In addition to the two-class evaluations presented in the main text, we extend our experiments to multi-class classification tasks with increasing numbers of classes. Specifically, we conduct classification from 3 to 10 classes on the ISOLET and MNIST datasets, and from 3 to 6 classes on the UCI HAR dataset.

These experiments follow the same QeHDC framework, using one pass training and fidelity-based inference under fixed qubit and dimension configurations. Class-balanced subsets are selected for each setting.

The goal is to evaluate the scalability and stability of our method under increasing classification complexity. The results, summarized in the Table 9.1 to 9.8, show that although accuracy naturally decreases as the number of classes increases, our method maintains strong relative performance compared to classical HDC baselines, especially under low-dimensional constraints.

9.5 Tomography Circuits and Resconstructed State Comparison

This appendix illustrates the tomography process used in our quantum classification framework, including representative tomography circuits and a comparison between the ideal post-binding state and the reconstructed density matrix.

Figure 9.3, Figure 9.4, and Figure 9.5 show three example circuits generated by the tomography module. These circuits correspond to different measurement bases required for full quantum state reconstruction. To emphasize the core structure, all circuits are presented in a cleaned form with the initialization components removed.

After executing all tomography circuits, we reconstruct the output density matrix and compare it with the ideal quantum statevector obtained from the Aer simulator. Figure 9.6 and 9.7 show the 3D cityscape visualizations of the complex amplitudes. The real and imaginary parts of both the ideal and reconstructed states are shown side by side for comparison.

As seen in the figures, the overall structure and amplitude distribution between the ideal and reconstructed states are highly consistent, validating the effectiveness of our QST pipeline even under limited measurement noise. These results demonstrate that the reconstructed quantum states retain sufficient fidelity for classification, as further supported by the numerical fidelity scores in the main text.

Table 9.1: 3-Class Results (ISOLET, MNIST, UCI HAR)

Method	ISOLET-3Classes			MNIST-3Classes			UCI HAR-3Classes		
	16D	32D	64D	16D	32D	64D	16D	32D	64D
Vanilla	56.85±6.04	65.37±2.73	71.67±5.90	54.90±7.66	76.18±4.68	83.34±2.26	47.85±2.16	54.29±1.29	54.63±1.98
AdaptHD	54.81±10.82	63.89±7.26	75.37±4.48	55.36±11.27	71.80±6.06	83.53±1.92	45.93±5.30	46.84±4.49	55.06±4.18
OnlineHD	86.85±5.45	94.81±1.89	97.41±1.39	74.96±8.72	83.15±2.88	94.53±0.41	65.66±7.46	77.34±5.70	83.75±2.21
NeuralHD	86.48±2.28	93.89±0.79	**97.59**±0.94	79.84±3.33	88.20±2.15	93.37±1.14	68.42±3.57	77.72±0.51	84.67±1.83
CompHD	52.78±7.74	63.33±2.36	68.15±4.12	49.66±3.36	59.93±1.56	75.29±1.62	45.06±1.18	46.07±1.85	53.06±3.37
SparseHD	51.48±5.00	61.48±7.61	73.33±7.86	44.46±12.26	43.00±4.77	64.65±17.43	41.14±3.05	46.43±3.78	51.43±3.60
QuantHD	52.96±8.95	58.89±3.63	71.30±2.14	64.07±5.01	67.10±5.01	67.92±1.87	38.02±8.45	44.97±3.41	50.54±5.82
LeHDC	46.67±7.07	48.15±2.73	53.89±9.50	44.55±5.35	53.21±5.02	57.91±11.07	37.66±0.65	38.62±2.94	42.78±2.03
	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits
Ours-Aer	**86.95**±1.43	**95.01**±0.84	97.22±0.67	75.06±2.98	**87.27**±1.41	**95.23**±2.35	**81.34**±0.63	**84.43**±0.35	**89.26**±1.32

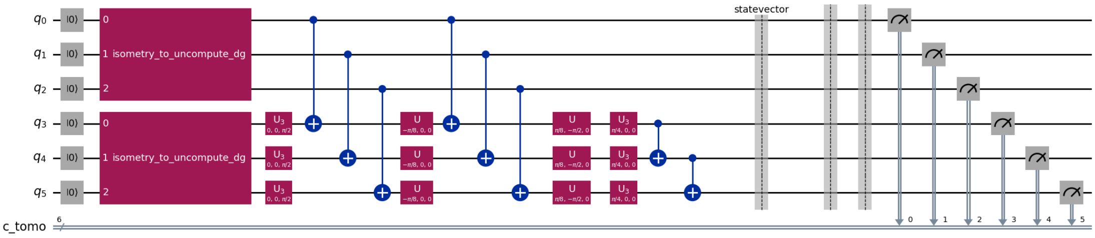


Figure 9.3: Quantum State Tomography Circuit (Z bias)

Table 9.2: 4-Class Results (ISOLET, MNIST, UCI HAR)

Method	ISOLET-4Classes			MNIST-4Classes			UCI HAR-4Classes		
	16D	32D	64D	16D	32D	64D	16D	32D	64D
Vanilla	43.75±6.49	59.03±6.33	65.83±0.59	47.17±2.89	62.33±3.42	75.81±2.35	55.75±4.70	62.74±2.83	69.36±1.56
AdaptHD	33.61±6.13	45.14±6.58	58.06±1.37	34.86±7.01	62.61±6.05	71.77±5.90	46.96±9.54	51.24±8.91	65.87±5.50
OnlineHD	79.44±2.60	84.17±1.89	94.58±0.68	61.32±3.67	78.00±2.44	87.53±2.04	72.10±2.94	81.82±2.17	89.24±1.55
NeuralHD	**80.69**±1.87	89.17±2.91	95.28±1.04	**73.07**±4.01	80.76±2.49	87.82±0.62	77.53±3.43	83.28±1.20	86.88±1.54
CompHD	39.44±0.71	46.53±2.60	59.72±5.40	37.90±2.92	52.08±1.22	64.29±1.58	42.55±11.07	56.59±1.59	61.87±5.04
SparseHD	42.08±3.25	41.94±4.82	61.67±5.31	35.54±4.85	37.40±4.10	38.35±10.15	39.94±3.58	50.66±2.50	56.37±4.04
QuantHD	34.17±5.62	52.50±3.25	55.00±2.91	43.78±0.69	43.47±1.50	50.85±7.99	32.14±6.62	52.48±2.35	60.21±6.60
LeHDC	32.64±3.99	40.83±2.57	38.33±7.27	32.79±5.02	38.92±8.12	50.01±5.80	43.91±5.97	40.47±3.11	47.46±1.29
	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits
Ours-Aer	80.14±1.28	**90.98**±1.32	**96.34**±1.82	68.87±1.69	**81.48**±1.82	**88.87**±0.64	**78.32**±1.61	**84.91**±1.71	**89.47**±0.73

Table 9.3: 5-Class Results (ISOLET, MNIST, UCI HAR)

Method	ISOLET-5Classes			MNIST-5Classes			UCI HAR-5Classes		
	16D	32D	64D	16D	32D	64D	16D	32D	64D
Vanilla	34.33±2.13	54.89±4.12	56.11±5.65	44.54±4.14	58.66±3.18	71.17±3.59	47.25±3.31	51.81±2.70	59.97±3.75
AdaptHD	32.56±4.73	42.56±10.02	48.11±4.94	34.43±5.58	46.27±5.24	62.39±11.57	32.45±4.27	49.07±2.78	54.23±4.15
OnlineHD	62.22±3.45	78.44±0.79	87.00±1.25	53.30±10.33	76.50±2.72	86.98±0.93	62.14±4.12	75.19±3.11	84.67±1.73
NeuralHD	70.78±4.36	80.33±1.96	88.78±0.87	**69.59**±0.80	**79.67**±3.36	86.95±1.38	71.44±3.02	79.52±1.56	82.85±0.54
CompHD	31.22±4.01	36.89±4.31	46.67±0.72	38.79±6.67	48.76±3.28	57.24±0.80	41.96±4.42	45.39±0.50	54.54±2.33
SparseHD	23.78±3.50	38.78±0.57	44.78±2.78	28.71±2.32	30.85±3.36	35.12±3.32	33.87±9.18	38.87±4.20	47.40±0.76
QuantHD	24.22±5.97	34.67±4.23	50.11±5.12	29.78±8.10	36.56±6.67	53.29±1.56	34.34±1.32	43.62±5.63	44.48±3.34
LeHDC	25.67±2.42	31.33±2.33	32.56±4.24	32.54±3.07	30.31±3.60	30.73±5.82	27.62±5.46	33.89±4.39	38.38±1.14
	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits
Ours-Aer	**72.39**±2.73	**85.49**±1.28	**91.37**±0.64	65.73±2.34	75.26±2.49	**87.04**±1.30	**74.94**±0.87	**81.92**±0.85	**88.15**±0.29

Table 9.4: 6-Class Results (ISOLET, MNIST, UCI HAR)

Method	ISOLET-6Classes			MNIST-6Classes			UCI HAR-6Classes		
	16D	32D	64D	16D	32D	64D	16D	32D	64D
Vanilla	38.70±3.46	53.06±3.15	55.46±2.93	44.58±3.79	47.49±5.53	57.42±2.79	36.91±4.12	46.08±5.89	57.26±0.48
AdaptHD	30.56±1.64	39.17±7.20	54.26±0.13	27.00±9.06	32.54±8.22	55.82±6.17	38.91±10.41	42.11±2.05	51.89±4.34
OnlineHD	63.52±3.10	80.93±3.67	88.24±1.82	51.21±3.14	67.49±1.42	80.39±1.85	68.47±2.94	76.96±3.53	85.11±0.67
NeuralHD	75.09±0.92	82.69±2.16	87.59±1.33	**63.05**±1.87	71.76±0.66	82.70±0.98	73.56±1.65	80.24±0.98	84.11±0.99
CompHD	29.44±4.47	41.67±3.20	52.50±2.40	30.46±1.68	40.80±5.94	51.21±4.03	32.47±4.98	41.42±7.17	51.95±3.16
SparseHD	23.98±4.81	36.76±2.92	50.74±1.98	23.76±4.26	24.99±5.00	27.75±5.64	30.84±1.15	35.69±3.96	41.78±4.06
QuantHD	29.81±3.19	40.19±2.87	46.39±4.74	31.76±10.03	36.68±3.68	39.92±6.45	28.06±2.35	40.48±3.19	38.89±0.99
LeHDC	23.61±3.60	28.15±4.93	34.35±4.67	25.86±4.28	30.90±2.35	32.76±4.65	21.67±2.43	31.21±3.91	34.02±6.37
	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits
Ours-Aer	**76.43**±3.59	**84.94**±2.46	**90.53**±1.35	62.45±1.84	**73.18**±2.47	**84.41**±1.28	**73.85**±0.66	**81.05**±0.60	**86.22**±1.75

Table 9.5: 7-Class Results (ISOLET and MNIST)

Method	ISOLET-7Classes			MNIST-7Classes		
	16D	32D	64D	16D	32D	64D
Vanilla	36.75±7.88	47.94±2.59	54.21±3.77	34.69±7.28	43.94±1.91	53.18±1.64
AdaptHD	26.43±3.03	30.32±2.35	40.48±6.62	32.28±6.47	30.57±4.43	54.22±2.76
OnlineHD	67.14±1.47	80.71±1.85	88.73±1.63	50.60±3.36	63.90±1.21	80.04±1.12
NeuralHD	71.51±0.59	81.75±3.34	87.70±1.81	52.95±2.54	71.00±1.94	80.43±1.30
CompHD	28.49±4.89	30.32±0.74	45.95±2.21	26.21±2.02	36.59±0.51	45.29±2.88
SparseHD	24.92±2.65	31.35±3.57	41.59±2.63	22.16±2.90	19.84±2.22	25.70±4.52
QuantHD	30.16±5.63	37.54±5.35	44.05±4.60	25.84±4.11	31.91±6.93	39.29±6.00
LeHDC	20.48±3.41	22.14±2.44	29.44±1.29	22.17±4.49	24.20±2.38	34.02±6.17
	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits
Ours-Aer	**72.57**±1.43	**87.45**±1.49	**90.19**±1.49	**57.62**±2.39	**72.88**±1.98	**82.07**±2.03

Table 9.6: 8-Class Results (ISOLET and MNIST)

Method	ISOLET-8Classes			MNIST-8Classes		
	16D	32D	64D	16D	32D	64D
Vanilla	33.33±3.96	40.49±3.45	57.85±3.47	30.77±1.49	43.19±3.33	58.51±1.72
AdaptHD	27.50±1.53	37.71±6.78	46.67±7.04	24.16±1.67	33.92±5.43	47.44±7.57
OnlineHD	65.42±5.35	80.90±1.37	88.47±0.39	42.01±3.21	60.49±1.76	77.44±1.54
NeuralHD	70.90±1.87	85.76±2.06	**91.39**±1.43	**56.90**±2.44	69.39±2.67	80.10±0.66
CompHD	25.35±0.55	32.85±0.94	42.57±4.53	24.54±3.71	32.65±2.69	42.71±0.97
SparseHD	17.50±4.36	30.00±4.34	35.14±3.09	22.64±5.33	19.76±4.34	26.02±1.65
QuantHD	18.61±4.03	30.62±2.86	39.79±1.84	16.57±6.14	37.69±0.89	39.40±2.17
LeHDC	16.32±1.90	19.93±1.82	28.19±2.77	16.70±4.37	23.32±1.14	29.05±3.52
	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits
Ours-Aer	**71.62**±1.38	**86.42**±1.36	88.72±3.13	56.31±2.83	**71.94**±2.29	**81.46**±0.34

Table 9.7: 9-Class Results (ISOLET and MNIST)

Method	ISOLET-9Classes			MNIST-9Classes		
	16D	32D	64D	16D	32D	64D
Vanilla	29.44±4.92	41.98±1.83	57.16±2.60	27.87±0.53	39.08±4.12	52.22±2.58
AdaptHD	25.43±2.57	24.01±6.12	47.04±6.91	21.74±3.17	23.52±7.59	47.90±3.54
OnlineHD	62.78±4.74	84.51±0.76	88.15±1.46	37.52±4.24	54.04±2.83	71.56±1.28
NeuralHD	**70.62**±3.33	83.21±2.06	**90.49**±1.01	53.65±1.53	68.81±1.62	77.13±0.85
CompHD	26.91±1.44	30.43±1.57	41.54±5.18	23.36±1.98	29.93±2.70	38.34±1.48
SparseHD	17.35±4.61	24.26±0.69	37.41±4.45	15.79±2.40	18.57±2.11	21.57±3.18
QuantHD	20.37±2.72	26.42±3.29	38.02±3.97	20.80±2.57	29.95±0.28	36.52±4.86
LeHDC	20.06±4.52	22.65±4.83	31.60±1.29	19.33±0.34	18.30±0.39	25.10±4.19
	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits
Ours-Aer	69.32±1.74	**85.88**±2.67	89.71±2.49	**54.87**±2.82	**69.04**±1.73	**79.24**±2.49

Table 9.8: 10-Class Results (ISOLET and MNIST)

Method	ISOLET-10Classes			MNIST-10Classes		
	16D	32D	64D	16D	32D	64D
Vanilla	27.72±2.73	39.78±3.68	54.11±2.70	23.49±3.11	30.96±4.59	46.61±4.41
AdaptHD	16.28±1.95	25.11±2.91	47.33±1.78	16.42±4.03	21.82±4.04	36.27±4.52
OnlineHD	62.89±4.83	81.89±1.47	90.67±1.30	34.03±1.42	53.24±4.90	69.79±0.30
NeuralHD	68.67±3.42	85.61±1.80	90.83±0.49	47.04±1.52	63.92±1.89	75.73±0.36
DistHD	**75.67**±3.31	86.00±2.12	**92.94**±0.63	50.84±2.48	**67.90**±2.13	78.97±1.54
CompHD	21.39±4.47	32.44±3.85	35.22±1.61	20.25±0.67	26.88±1.51	36.88±2.37
SparseHD	13.94±1.59	22.06±2.79	36.11±3.94	14.07±1.66	14.53±0.92	23.50±3.23
QuantHD	21.61±2.09	28.33±4.90	39.39±6.28	21.17±2.65	25.27±1.64	30.93±2.74
LeHDC	14.78±5.18	17.17±4.25	22.33±1.91	15.43±3.16	17.43±4.05	26.50±2.60
	4 Qubits	5 Qubits	6 Qubits	4 Qubits	5 Qubits	6 Qubits
Ours-Aer	70.73±1.16	**86.83**±2.02	89.25±0.93	**53.27**±2.34	66.79±2.51	**80.27**±1.63

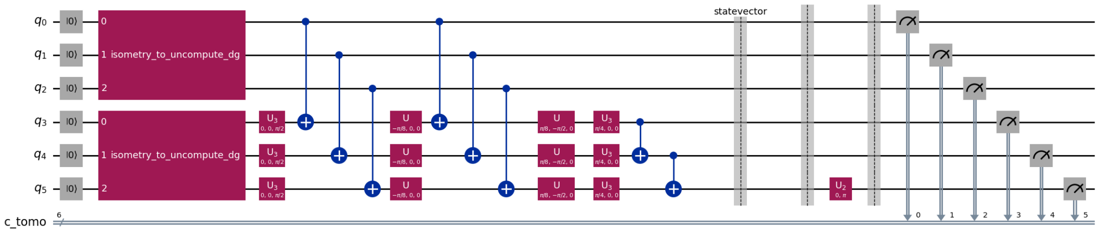


Figure 9.4: Quantum State Tomography Circuit (X bias)

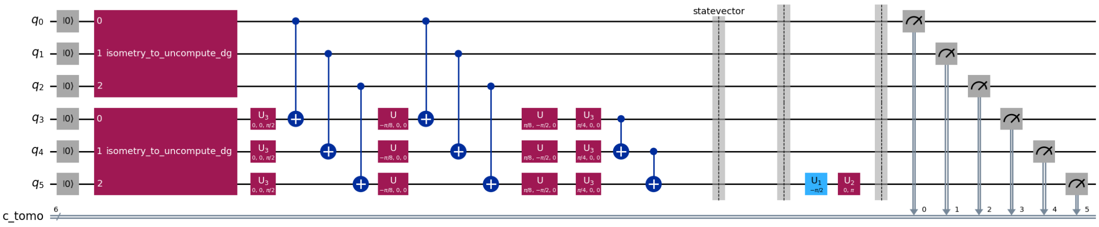


Figure 9.5: Quantum State Tomography Circuit (Y bias)

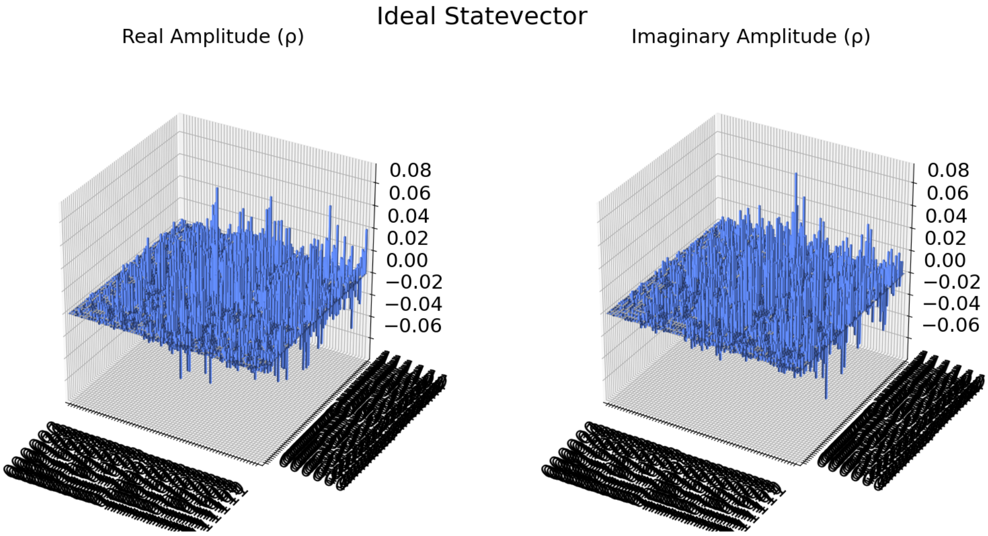


Figure 9.6: Ideal Statevector (Real and Imaginary Components)

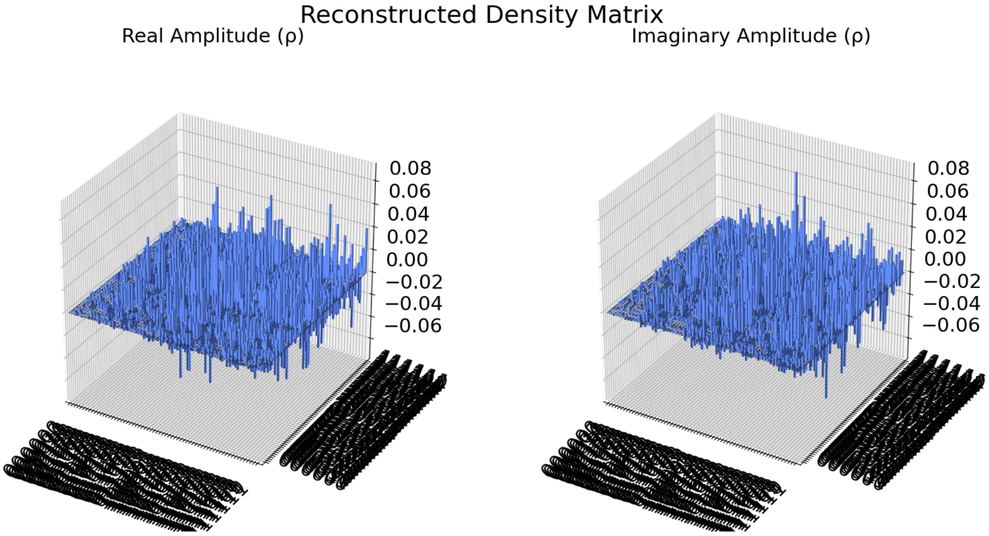


Figure 9.7: Reconstructed Density Matrix (Real and Imaginary Components)

Bibliography

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach.* pearson, 2016.

[3] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, pp. 78–87, 2012.

[4] M. I. Jordan and T. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, pp. 255 – 260, 2015.

[5] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[6] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.

[7] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.

[8] N. Young, *An introduction to Hilbert space.* Cambridge university press, 1988.

[9] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI open*, vol. 1, pp. 57–81, 2020.

[10] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 4–24, 2020.

[11] J. Kang, H. Zhang, H. Yang, and L. Zhang, “Support vector machine classification of crop lands using sentinel-2 imagery,” in *2018 7th International Conference on Agro-geoinformatics (Agro-geoinformatics).* IEEE, 2018, pp. 1–6.

[12] A. Tariq, J. Yan, A. S. Gagnon, M. Riaz Khan, and F. Mumtaz, “Mapping of cropland, cropping patterns and crop types by combining optical remote sensing images with decision tree classifier and random forest,” *Geo-Spatial Information Science*, vol. 26, no. 3, pp. 302–320, 2023.

[13] Z. Sun, L. Di, and H. Fang, “Using long short-term memory recurrent neural network in land cover classification on landsat and cropland data layer time series,” *International journal of remote sensing*, vol. 40, no. 2, pp. 593–614, 2019.

[14] L. Zhong, L. Hu, and H. Zhou, “Deep learning based multi-temporal crop classification,” *Remote sensing of environment*, vol. 221, pp. 430–443, 2019.

[15] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, “Deep learning classification of land cover and crop types using remote sensing data,” *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, 2017.

[16] Z. Du, J. Yang, W. Huang, and C. Ou, “Training segnet for cropland classification of high resolution remote sensing images,” in *AGILE Conference*, 2018.

[17] H. Li, C. Zhang, S. Zhang, X. Ding, and P. M. Atkinson, “Iterative deep learning (idl) for agricultural landscape classification using fine spatial resolution remotely sensed imagery,” *International Journal of Applied Earth Observation and Geoinformation*, vol. 102, p. 102437, 2021.

[18] P. Hao, L. Di, C. Zhang, and L. Guo, “Transfer learning for crop classification with cropland data layer data (cdl) as training samples,” *Science of The Total Environment*, vol. 733, p. 138869, 2020.

[19] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big data classification,” *Physical review letters*, vol. 113, no. 13, p. 130503, 2014.

[20] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, “Quanvolutional neural networks: powering image recognition with quantum circuits,” *Quantum Machine Intelligence*, vol. 2, no. 1, p. 2, 2020.

[21] S. Lloyd, M. Mohseni, and P. Rebentrost, “Quantum principal component analysis,” *Nature Physics*, vol. 10, no. 9, pp. 631–633, 2014.

[22] G. Li, Z. Song, and X. Wang, "Vsql: Variational shadow quantum learning for classification," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 9, 2021, pp. 8357–8365.

[23] M. A. Nielsen and I. L. Chuang, *Quantum computation and quantum information.* Cambridge university press, 2010.

[24] "Paddle Quantum," 2020. [Online]. Available: https://github.com/PaddlePaddle/Quantum

[25] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi *et al.*, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," *arXiv preprint arXiv:1811.04968*, 2018.

[26] Qiskit contributors, "Qiskit: An open-source framework for quantum computing," 2023.

[27] J. Liu, K. H. Lim, K. L. Wood, W. Huang, C. Guo, and H.-L. Huang, "Hybrid quantum-classical convolutional neural networks," *Science China Physics, Mechanics & Astronomy*, vol. 64, no. 9, p. 290311, 2021.

[28] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 international conference on engineering and technology (ICET).* Ieee, 2017, pp. 1–6.

[29] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, no. 64-67, p. 2, 2001.

[30] N. Pettorelli, *The normalized difference vegetation index.* Oxford University Press, USA, 2013.

[31] A. Huete, K. Didan, T. Miura, E. P. Rodriguez, X. Gao, and L. G. Ferreira, "Overview of the radiometric and biophysical performance of the modis vegetation indices," *Remote sensing of environment*, vol. 83, no. 1-2, pp. 195–213, 2002.

[32] A.-M. Simón Sánchez, J. González-Piqueras, L. de la Ossa, and A. Calera, "Convolutional neural networks for agricultural land use classification from sentinel-2 image time series," *Remote Sensing*, vol. 14, no. 21, p. 5373, 2022.

[33] H. Wang, X. Zhao, X. Zhang, D. Wu, and X. Du, "Long time series land cover classification in china from 1982 to 2015 based on bi-lstm deep learning," *Remote Sensing*, vol. 11, no. 14, p. 1639, 2019.

[34] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[35] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques.* IGI global, 2010, pp. 242–264.

[36] D. M. Johnson, R. Mueller *et al.*, "The 2009 cropland data layer," *Photogramm. Eng. Remote Sens*, vol. 76, no. 11, pp. 1201–1205, 2010.

[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[38] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, "Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary," *Journal of artificial intelligence research*, vol. 61, pp. 863–905, 2018.

[39] Y. Ma, D. Yu, T. Wu, and H. Wang, "Paddlepaddle: An open-source deep learning platform from industrial practice," *Frontiers of Data and Domputing*, vol. 1, no. 1, pp. 105–115, 2019.

[40] S. Rani, P. K. Pareek, J. Kaur, M. Chauhan, and P. Bhambri, "Quantum machine learning in healthcare: Developments and challenges," *2023 IEEE International Conference on Integrated Circuits and Communication Systems (ICICACS)*, pp. 1–7, 2023.

[41] D. Maheshwari, B. Garcia-Zapirain, and D. Sierra-Sosa, "Quantum machine learning applications in the biomedical domain: A systematic review," *IEEE Access*, vol. 10, pp. 80 463–80 484, 2022.

[42] S. Panyaram, "Utilizing quantum computing to enhance artificial intelligence in healthcare for predictive analytics and personalized medicine," *FMDB Transactions on Sustainable Computing Systems*, 2024.

[43] D. Guijo, V. Onofre, G. Bimbo, S. Mugel, D. Estepa, X. D. Carlos, A. Adell, A. Lojo, J. Bilbao, and R. Orús, "Quantum artificial vision for defect detection in manufacturing," *ArXiv*, vol. abs/2208.04988, 2022.

[44] M. Pistoia, S. F. Ahmad, A. Ajagekar, A. Buts, S. Chakrabarti, D. Herman, S. Hu, A. Jena, P. Minssen, P. Niroula, A. G. Rattew, Y. Sun, and R. Yalovetzky, "Quantum machine learning for finance," *ArXiv*, vol. abs/2109.04298, 2021.

[45] S. KUMAR, S. ATTRI, and K. Singh, "Comparison of lasso and stepwise regression technique for wheat yield prediction," *Journal of Agrometeorology*, 2021.

[46] K. Singh, K. Singh, S. Kumar, S. Panwar, B. Gurung *et al.*, "Forecasting crop yield through weather indices through lasso," *Indian J. Agric. Sci*, vol. 89, no. 3, pp. 540–544, 2019.

[47] V. Sellam and E. Poovammal, "Prediction of crop yield using regression analysis," *Indian journal of science and technology*, vol. 9, 2016.

[48] S. Raj, S. Patle, and S. Rajendran, "Predicting crop yield using decision tree regressor," *2022 International Conference on Knowledge Engineering and Communication Systems (ICKES)*, pp. 1–5, 2022.

[49] M. Jun and S. Chao, "The study on soybean yield prediction based on support vector machine," *Mathematics in Practice and Theory*, 2011.

[50] P. Nevavuori, N. G. Narra, and T. Lipping, "Crop yield prediction with deep convolutional neural networks," *Comput. Electron. Agric.*, vol. 163, 2019.

[51] U. Bhimavarapu, G. Battineni, and N. Chintalapudi, "Improved optimization algorithm in lstm to predict crop yield," *Comput.*, vol. 12, p. 10, 2023.

[52] S. Khaki, L. Wang, and S. V. Archontoulis, "A cnn-rnn framework for crop yield prediction," *Frontiers in Plant Science*, vol. 10, p. 492736, 2020.

[53] J. Fan, J. Bai, Z. Li, A. Ortiz-Bobea, and C. P. Gomes, "A gnn-rnn approach for harnessing geospatial and temporal information: application to crop yield prediction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 11, 2022, pp. 11 873–11 881.

[54] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio *et al.*, "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.

[55] M. Cerezo, K. Sharma, A. Arrasmith, and P. J. Coles, "Variational quantum state eigensolver," *npj Quantum Information*, vol. 8, pp. 1–11, 2020.

[56] P. Rebentrost, M. Mohseni, and S. Lloyd, “Quantum support vector machine for big feature and big data classification,” *Physical review letters*, vol. 113 13, p. 130503, 2013.

[57] J. Romero, J. Olson, and A. Aspuru-Guzik, “Quantum autoencoders for efficient compression of quantum data,” *Quantum Science and Technology*, vol. 2, 2016.

[58] J. Landman, N. Mathur, Y. Li, M. Strahm, S. Kazdaghli, A. Prakash, and I. Kerenidis, “Quantum methods for neural networks and application to medical image classification,” *Quantum*, vol. 6, p. 881, 2022.

[59] T. Sakuma, “Application of deep quantum neural networks to finance,” *arXiv: Computational Finance*, 2020.

[60] M. Ostaszewski, P. Sadowski, and P. Gawron, “Quantum image classification using principal component analysis,” *ArXiv*, vol. abs/1504.00580, 2015.

[61] D. Zhu, N. Linke, M. Benedetti, K. Landsman, N. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque, L. Egan, O. Perdomo, and C. Monroe, “Training of quantum circuits on a hybrid quantum computer,” *Science Advances*, vol. 5, 2018.

[62] G. Hellstem, “Hybrid quantum network for classification of finance and mnist data,” *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*, pp. 1–4, 2021.

[63] J. Muñoz-Sabater, E. Dutra, A. Agustí-Panareda, C. Albergel, G. Arduini, G. Balsamo, S. Boussetta, M. Choulga, S. Harrigan, H. Hersbach *et al.*, “Era5-land: A state-of-the-art global reanalysis dataset for land applications,” *Earth system science data*, vol. 13, no. 9, pp. 4349–4383, 2021.

[64] T. Hengl, J. Mendes de Jesus, G. B. Heuvelink, M. Ruiperez Gonzalez, M. Kilibarda, A. Blagotić, W. Shangguan, M. N. Wright, X. Geng, B. Bauer-Marschallinger *et al.*, “Soilgrids250m: Global gridded soil information based on machine learning,” *PLoS one*, vol. 12, no. 2, p. e0169748, 2017.

[65] USDA Natural Resources Conservation Service, “Gridded soil survey geographic (gssurgo) database,” 2023, accessed: date-of-access. [Online]. Available: https://websoilsurvey.sc.egov.usda.gov/

[66] C. G. Homer, J. A. Fry, and C. A. Barnes, “The national land cover database,” US Geological Survey, Tech. Rep., 2012.

[67] Y. Li, Y. Ji, S. Li, S. He, Y. Cao, Y. Liu, H. Liu, X. Li, J. Shi, and Y. Yang, "Relevance-aware anomalous users detection in social network via graph neural network," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[68] H. Zhang, Z. Feng, and C. Wu, "A non-local graph neural network for identification of essential proteins," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–8.

[69] C. Ren, L. Zhang, L. Fang, T. Xu, Z. Wang, S. Yuan, and E. Chen, "Ontological concept structure aware knowledge transfer for inductive knowledge graph embedding," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[70] H. Taud and J. Mas, "Multilayer perceptron (mlp)," *Geomatic approaches for modeling land change scenarios*, pp. 451–455, 2018.

[71] Y. Hu, H. You, Z. Wang, Z. Wang, E. Zhou, and Y. Gao, "Graph-mlp: Node classification without message passing in graph," *arXiv preprint arXiv:2106.04051*, 2021.

[72] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[73] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[74] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?" *arXiv preprint arXiv:2105.14491*, 2021.

[75] Y. Ma, J. Hao, Y. Yang, H. Li, J. Jin, and G. Chen, "Spectral-based graph convolutional network for directed graphs," *arXiv preprint arXiv:1907.08990*, 2019.

[76] A. Qin, Z. Shang, J. Tian, Y. Wang, T. Zhang, and Y. Y. Tang, "Spectral–spatial graph convolutional networks for semisupervised hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 2, pp. 241–245, 2018.

[77] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[78] C. Zhang, D. Song, C. Huang, A. Swami, and N. V. Chawla, “Heterogeneous graph neural network,” in *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, 2019, pp. 793–803.

[79] F. Manessi, A. Rozza, and M. Manzo, “Dynamic graph convolutional networks,” *Pattern Recognition*, vol. 97, p. 107000, 2020.

[80] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Message passing neural networks,” *Machine learning meets quantum physics*, pp. 199–214, 2020.

[81] B. Perozzi, R. Al-Rfou, and S. Skiena, “Deepwalk: Online learning of social representations,” in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 701–710.

[82] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, “Simplifying graph convolutional networks,” in *International conference on machine learning.* PMLR, 2019, pp. 6861–6871.

[83] D. Dong, C. Chen, H. Li, and T.-J. Tarn, “Quantum reinforcement learning,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 5, pp. 1207–1220, 2008.

[84] P.-L. Dallaire-Demers and N. Killoran, “Quantum generative adversarial networks,” *Physical Review A*, vol. 98, no. 1, p. 012324, 2018.

[85] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth *et al.*, “The variational quantum eigensolver: a review of methods and best practices,” *Physics Reports*, vol. 986, pp. 1–128, 2022.

[86] G. Verdon, T. McCourt, E. Luzhnica, V. Singh, S. Leichenauer, and J. Hidary, “Quantum graph neural networks,” *arXiv preprint arXiv:1909.12264*, 2019.

[87] P. Mernyei, K. Meichanetzidis, and I. I. Ceylan, “Equivariant quantum graph circuits,” in *International Conference on Machine Learning.* PMLR, 2022, pp. 15 401–15 420.

[88] Y. Tang and J. Yan, “Graphqntk: Quantum neural tangent kernel for graph data,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 6104–6118, 2022.

[89] X. Ai, Z. Zhang, L. Sun, J. Yan, and E. Hancock, “Decompositional quantum graph neural network,” *arXiv preprint arXiv:2201.05158*, 2022.

[90] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[91] G. Karunaratne, M. L. Gallo, M. Hersche, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, "Energy efficient in-memory hyperdimensional encoding for spatio-temporal signal processing," *IEEE Transactions on Circuits and Systems II: Express Briefs*, pp. 1725–1729, 2021.

[92] P. Neubert, S. Schubert, and P. Protzel, "An introduction to hyperdimensional computing for robotics," *KI - Künstliche Intelligenz*, pp. 319 – 330, 2019.

[93] R. Thapa, B. Lamichhane, D. Ma, and X. Jiao, "Spamhd: Memory-efficient text spam detection using brain-inspired hyperdimensional computing," *2021 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 84–89, 2021.

[94] S. Zhang, Z. Wang, and X. Jiao, "Adversarial attack on hyperdimensional computing-based nlp applications," *2023 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pp. 1–6, 2023.

[95] A. Hernández-Cano, Y. Ni, Z. Zou, A. Zakeri, and M. Imani, "Hyperdimensional computing with holographic and adaptive encoder," *Frontiers in Artificial Intelligence*, vol. 7, p. 1371988, 2024.

[96] A. Thomas, B. Khaleghi, G. K. Jha, S. Dasgupta, N. Himayat, R. Iyer, N. Jain, and T. Rosing, "Streaming encoding algorithms for scalable hyperdimensional computing," *ArXiv*, vol. abs/2209.09868, 2022.

[97] B. Khaleghi, J. Kang, H. Xu, J. Morris, and T. Simunic, "Generic: highly efficient learning engine on edge using hyperdimensional computing," *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022.

[98] P. Kanerva, "Hyperdimensional computing: An introduction to computing in distributed representation with high-dimensional random vectors," *Cognitive Computation*, vol. 1, pp. 139–159, 2009.

[99] J. Wang, S. Huang, and M. Imani, "DistHD: A learner-aware dynamic encoding method for hyperdimensional classification," *2023 60th ACM/IEEE Design Automation Conference (DAC)*, pp. 1–6, 2023.

[100] M. Imani, S. Bosch, S. Datta, S. Ramakrishna, S. Salamat, J. M. Rabaey, and T. Rosing, "QuantHD: A quantization framework for hyperdimensional computing," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 2268–2278, 2019.

[101] S. Duan, Y. Liu, S. Ren, and X. Xu, "LeHDC: Learning-based hyperdimensional computing classifier," in *Proceedings of the 59th ACM/IEEE Design Automation Conference*, 2022, pp. 1111–1116.

[102] M. Imani, J. Morris, S. Bosch, H. Shu, G. De Micheli, and T. Rosing, "Adapthd: Adaptive efficient training for brain-inspired hyperdimensional computing," in *Proceedings of 2019 IEEE Biomedical Circuits and Systems Conference (BioCAS)*, 2019, pp. 1–4.

[103] A. Hernández-Cano, N. Matsumoto, E. Ping, and M. Imani, "Onlinehd: Robust, efficient, and single-pass online learning using hyperdimensional system," in *Proceedings of 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2021, pp. 56–61.

[104] J. Morris, M. Imani, S. Bosch, A. Thomas, H. Shu, and T. Rosing, "CompHD: Efficient hyperdimensional computing using model compression," in *Proceedings of 2019 IEEE/ACM International Symposium on Low Power Electronics and Design (ISLPED)*, 2019, pp. 1–6.

[105] M. Imani, S. Salamat, B. Khaleghi, M. Samragh, F. Koushanfar, and T. Rosing, "SparseHD: Algorithm-hardware co-optimization for efficient high-dimensional computing," in *Proceedings of 2019 IEEE 27th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, 2019, pp. 190–198.

[106] Z. Zou, Y. Kim, F. Imani, H. Alimohamadi, R. Cammarota, and M. Imani, "Scalable edge-based hyperdimensional learning system with brain-like neural adaptation," in *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, 2021, pp. 1–15.

[107] D. Ranga, A. Rana, S. Prajapat, P. Kumar, K. Kumar, and A. V. Vasilakos, "Quantum machine learning: Exploring the role of data encoding techniques, challenges, and future directions," *Mathematics*, p. 3318, 2024.

[108] M. Rath and H. Date, "Quantum data encoding: a comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy," *EPJ Quantum Technology*, p. 72, 2024.

[109] M. Weigold, J. Barzen, F. Leymann, and M. Salm, "Expanding data encoding patterns for quantum algorithms," in *Proceedings of 2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*, 2021, pp. 95–101.

[110] C. Bravo-Prieto, "Quantum autoencoders with enhanced data encoding," *Machine Learning: Science and Technology*, p. 035028, 2021.

[111] Y.-G. Yang, X. Jia, S.-J. Sun, and Q.-X. Pan, "Quantum cryptographic algorithm for color images using quantum fourier transform and double random-phase encoding," *Information Sciences*, pp. 445–457, 2014.

[112] H. Yetiş and M. Karaköse, "An improved and cost reduced quantum circuit generator approach for image encoding applications," *Quantum Information Processing*, p. 203, 2022.

[113] B. Bhabhatsatam and S. Smanchat, "Hybrid quantum encoding: Combining amplitude and basis encoding for enhanced data storage and processing in quantum computing," in *Proceedings of 2023 20th International Joint Conference on Computer Science and Software Engineering (JCSSE)*, 2023, pp. 512–516.

[114] S. R. Majji, A. Chalumuri, and B. S. Manoj, "Quantum approach to image data encoding and compression," *IEEE Sensors Letters*, pp. 1–4, 2023.

[115] X.-B. Nguyen, H.-Q. Nguyen, H. Churchill, S. U. Khan, and K. Luu, "Quantum visual feature encoding revisited," *Quantum Machine Intelligence*, p. 61, 2024.

[116] D. Campos and J. Bernardes, "Cardiotocography," UCI Machine Learning Repository, 2000, DOI: https://doi.org/10.24432/C51S4N.

[117] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, pp. 2278–2324, 1998.

[118] R. Cole and M. Fanty, "ISOLET," UCI Machine Learning Repository, 1991, DOI: https://doi.org/10.24432/C51G69.

[119] E. Bulbul, A. Cetin, and I. A. Dogru, "Human activity recognition using smartphones," in *Proceedings of 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2018, pp. 1–6.

[120] M. Heddes, I. Nunes, P. Vergés, D. Kleyko, D. Abraham, T. Givargis, A. Nicolau, and A. Veidenbaum, "Torchhd: An open source python library to support research

on hyperdimensional computing and vector symbolic architectures," *Journal of Machine Learning Research*, pp. 1–10, 2023.

[121] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, "Quantum computing with Qiskit," 2024.

[122] R. W. Gayler, "Vector symbolic architectures answer jackendoff's challenges for cognitive neuroscience," *arXiv preprint cs/0412059*, 2004.

[123] C.-Y. Chang, Y.-C. Chuang, C.-T. Huang, and A.-Y. Wu, "Recent progress and development of hyperdimensional computing (hdc) for edge intelligence," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 13, no. 1, pp. 119–136, 2023.

[124] F. R. Najafabadi, A. Rahimi, P. Kanerva, and J. M. Rabaey, "Hyperdimensional computing for text classification," in *Design, automation test in Europe conference exhibition (DATE), University Booth*, 2016, pp. 1–1.

[125] K. A. Schindler and A. Rahimi, "A primer on hyperdimensional computing for ieeg seizure detection," *Frontiers in neurology*, vol. 12, p. 701791, 2021.

[126] Z. Zou, H. Chen, P. P. Poduval, Y. Kim, M. Imani, E. Sadredini, R. Cammarota, and M. Imani, "Biohd: an efficient genome sequence search platform using hyperdimensional memorization," *Proceedings of the 49th Annual International Symposium on Computer Architecture*, 2022.

[127] I. Nunes, M. Heddes, T. Givargis, A. Nicolau, and A. Veidenbaum, "Graphhd: Efficient graph classification using hyperdimensional computing," in *2022 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2022, pp. 1485–1490.

[128] R. Chaurasia and A. Sengupta, "Securehd: Designing low-cost reliable and security aware hardware accelerators during high-level synthesis for computationally intensive application frameworks," in *2024 IEEE International Symposium on Smart Electronic Systems (iSES)*, 2024, pp. 380–383.

[129] M. Eggimann, A. Rahimi, and L. Benini, "A 5 μw standard cell memory-based configurable hyperdimensional computing accelerator for always-on smart sensing," *IEEE Transactions on Circuits and Systems I: Regular Papers*, pp. 4116–4128, 2021.

[130] U. Pale, T. Teijeiro, and D. A. Alonso, “Systematic assessment of hyperdimensional computing for epileptic seizure detection,” *2021 43rd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, pp. 6361–6367, 2021.

[131] G. Karunaratne, M. L. Gallo, G. Cherubini, L. Benini, A. Rahimi, and A. Sebastian, “In-memory hyperdimensional computing,” *Nature Electronics*, vol. 3, pp. 327 – 337, 2019.

[132] A. Menon, A. Natarajan, L. I. G. Olascoaga, Y. Kim, B. C. Benedict, and J. Rabaey, “On the role of hyperdimensional computing for behavioral prioritization in reactive robot navigation tasks,” *2022 International Conference on Robotics and Automation (ICRA)*, pp. 7335–7341, 2022.

[133] H. Kwon, K. Kim, J. Lee, H.-S. Lee, J. Kim, J. Kim, T. Kim, Y. Kim, Y. Ni, M. Imani, I. Suh, and Y. Kim, “Brain-inspired hyperdimensional computing in the wild: Lightweight symbolic learning for sensorimotor controls of wheeled robots,” *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5176–5182, 2024.

[134] A. Mitrokhin, P. Sutor, C. Fermüller, and Y. Aloimonos, “Learning sensorimotor control with neuromorphic sensors: Toward hyperdimensional active perception,” *Science Robotics*, vol. 4, 2019.

[135] J.-M. A. Allen, “Quantum superpositions cannot be epistemic,” *Quantum Studies: Mathematics and Foundations*, vol. 3, pp. 161–177, 2015.

[136] G. Thekkadath, L. Giner, Y. Chalich, M. Horton, J. Banker, and J. S. Lundeen, “Direct measurement of the density matrix of a quantum system.” *Physical review letters*, vol. 117 12, p. 120401, 2016.

[137] A. Sturm, “Theory and implementation of the quantum approximate optimization algorithm: A comprehensible introduction and case study using qiskit and ibm quantum computers,” 2023.

[138] M. Heddes, I. Nunes, P. Vergés, D. Kleyko, D. Abraham, T. Givargis, A. Nicolau, and A. Veidenbaum, “Torchhd: An open source python library to support research on hyperdimensional computing and vector symbolic architectures,” *Journal of Machine Learning Research*, vol. 24, no. 255, pp. 1–10, 2023.