Machine Learning Applications of Parameterized Quantum Circuits

by Guangxi Li

Thesis submitted in fulfilment of the requirements for
the degree of

Doctor of Philosophy

under the supervision of Profs. Sanjiang Li and Yuan Feng

University of Technology Sydney
Faculty of Engineering and Information Technology

Feb 2023

Certificate of Original Authorship

I, Guangxi Li, declare that this thesis is submitted in fulfillment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note:
Signature removed prior to publication.

Date: 10/02/2023

To my family.

Acknowledgements

During my Ph.D. study, I enjoyed a lot of life in Sydney, learned a lot of new knowledge, and made myself grow in all aspects. However, my Ph.D. study process was not smooth, especially under the attack of Covid-19, which led to a significant reduction in overall learning efficiency. Therefore, I cannot complete my studies successfully without the support and dedication of many people.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Sanjiang Li, for providing me with this opportunity to study at the Centre for Quantum Software and Information (CQSI), University of Technology Sydney. His rigorous academic style and patient character have helped me a lot and inspired me. Since I had not yet enrolled in the university, he has patiently guided me in matters related to my enrollment through multiple emails. After entering the university, he also helped me immensely in my research. He not only gave me total freedom in research contents and research directions but also often reminded me at the group meeting that I should not be impetuous in my research and should calm down to do fundamental research. It had a significant impact on my later research style. He also gave me a lot of help in my life, such as asking me if I was still used to eating and living and if I had gone to the surrounding places for sightseeing.

I am also grateful to my co-supervisor, Prof. Yuan Feng, for his solid academic skills and patient communication with me. He really knows a lot and often gave me a lot of discussions and helped me with the proof details I wrote, which also let me know that mathematical proof is very rigorous. Moreover, I am grateful to Prof. Mingsheng Ying, Prof. Zhengfeng Ji, Min-Hsiu Hsieh, Nengkun Yu, Christopher Ferrie and Yulei Sui for their helpful research guidance.

I would really like to thank Prof. Runyao Duan for supporting me with opportunities to visit Baidu Quantum Research Institute. At that time, due to Covid-19, I could not return to Sydney, which led to slow progress in my scientific research. Fortunately, Prof. Duan gave me the opportunity to visit Baidu, which not only enabled me to continue my research but also made me feel the strong research atmosphere in the company. I would especially like to thank my mentor, Dr. Xin Wang, for his academic guidance. His

academic style and his style of not fearing difficulties attracted me profoundly and also inspired me to keep learning. His efficient behavior style also deeply affected me. He is not only my mentor but also my friend and often talks with me about interesting things in the quantum field.

I would like to extend my sincere thanks to my co-authors or co-workers: Youle Wang, Xin Hong, Yu Luo, Xiangzhen Zhou, Zhixin Song, Xuanqiang Zhao, Ruilin Ye, Zhiwei Liu, Gang Tang, Afrad Basheer, Benchi Zhao, Zhan Yu, Hongshun Yao, Daokun Zhang, Adi Lin, Shufeng Kong, Fujin Zhu, Dian Ouyang and all those not mentioned but helped me a lot. I am very happy to work with these wonderful people. I would also like to acknowledge the scholarships that support my study and living stipend: China Scholarship Council, Joint CSC-UTS International Research Scholarship, Living Stipend CSC Top Up, Sydney Quantum Academy Top Up and Extension Scholarship, Baidu-UTS AI Meets Quantum project and Australian Research Council project.

Thanks should also go to all the members of CQSI for creating an excellent learning and working environment. This environment lets me have the opportunity to experience the immersive learning atmosphere and the open and relaxed living environment abroad. In particular, I would like to mention Lily Qian and Robyn Barden for their help in my admission guidance, scholarship application and travel reimbursement. Of course, I would also like to thank all the members of Baidu Quantum for providing a vibrant working atmosphere. I would be remiss in not mentioning my friends: Rong Hu, Yuxuan Qiu, Wenhao Ma and Youle, Yu Luo, and Xiangzhen, whom I mentioned above, for the joyful days we spent together in Sydney.

Lastly, I'd like to mention my family for their unconditional support, funding and trust. Especially thanks to my girlfriend for her understanding, support and remote accompany almost every day. Sincerely thank you all.

Abstract

With the development of near-term quantum devices, hybrid quantum-classical computing has been acknowledged as a promising framework to realize near-term quantum advantages on important tasks, including chemistry, optimization and machine learning. The performance of such frameworks significantly relies on the power of parameterized quantum circuits (PQCs). However, it is challenging to design more suitable PQC architectures showing quantum superiorities for practical quantum machine learning tasks. In this thesis, we make progress in studying the power of PQCs in quantum classification and quantum natural language processing, and exploring the limitations of PQCs in quantum data encoding.

Specifically, we first propose variational shadow quantum learning for quantum classification, which in particular utilizes the local PQCs inspired by classical shadows to extract features of quantum data in a convolution way. We show this method could avoid the notorious barren plateaus issue and has superiorities with respect to accuracy and parameter numbers compared with baselines. Secondly, we propose a quantum self-attention neural network, where we introduce the self-attention mechanism into PQCs and then utilize a Gaussian projected quantum self-attention serving as a sensible quantum version of self-attention. We show this approach outperforms 1) the best existing QNLP model based on syntactic analysis, and 2) a simple classical self-attention neural network in text classification tasks on public data sets. Lastly, we prove that, for the PQC-based data encoding strategies, the average encoded state will concentrate on the maximally mixed state at an exponential speed on circuit depth.

In conclusion, we propose two new quantum neural network (QNN) models for handling practical machine learning tasks, demonstrating QNN's ability to extract features and the potential of quantum machine learning in real-world applications. In addition, we also reveal the concentration of data encoding, which seriously limits the performance of downstream quantum supervised learning tasks. Such concentration might also guide the practical data encoding design. All these progress would benefit practical quantum machine learning.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

After Feynman put forward the concept of quantum computing in the 1980s [1], quantum computing has become a promising paradigm [2] for fast computations that can provide substantial advantages in solving valuable problems [3–7]. With major academic and industry efforts on developing quantum algorithms and quantum hardware, it has led to an increasing number of applications in areas including cryptography [8], chemistry [9, 10], optimization [11], and machine learning [7, 12–14] and so on. The reason is that quantum computing has the inherent nature of parallel computing, which makes it able to show significant computational advantages for some specific problems [3].

At the same time, machine learning [15, 16], as the core of artificial intelligence, has also changed almost all aspects of our lives in recent years with applications ranging from face recognition, speech recognition, product recommendation, autonomous driving and so on.

Quantum machine learning is the combination of quantum computing and machine learning [7, 17–19]. In this field, on the one hand, one hopes to enhance some traditional machine learning algorithms with the help of quantum computers, so as to realize the advantages in inference accuracy or running speed. On the other hand, one also hopes that machine learning methods can help solve some problems in the quantum field. Quantum machine learning is also expected to have a wide range of applications like classical machine learning.

At present, some quantum machine learning algorithms, such as quantum data fitting algorithm [20], can be expected to achieve exponential acceleration under certain conditions. Most of these algorithms are based on Shor's algorithm [21] or HHL algorithm [22]. They generally need fault-tolerant quantum computers to run, and they also require quantum random access memory (QRAM) [23] technology similar to classical random access memory. However, neither fault-tolerant quantum computing nor QRAM is available in the near future.

Quantum devices available currently, also known as the noisy intermediate-scale quantum (NISQ) devices [24], have up to a few hundred physical qubits. They are affected by coherent and incoherent noises, making the practical implementation of many advantageous quantum algorithms less feasible. But such devices with 50-100 qubits already allow one to achieve quantum advantage against the most powerful classical supercomputers on certain carefully designed tasks [25, 26]. A natural question is how to design powerful quantum machine learning algorithms by employing these near-term quantum devices for practical applications in the NISQ era.

A currently feasible and popular way is to adopt the hybrid quantum-classical framework, that is, allocate some relatively difficult tasks, which is generally considered to be classically intractable, to the quantum computer and leave those relatively simple tasks to the classical computer to run. Some of the earliest representative works are variational quantum eigensolver (VQE) [27, 28] and quantum approximate optimization algorithm (QAOA) [29], which are used to solve molecular ground state preparation problems and combinatorial optimization problems, respectively. They encode the process of generating quantum states with exponential dimensions into parameterized quantum circuits and run them on quantum computers, and leave the update process of these parameters to classical computers, so as to make full use of near-term quantum devices.

The key to this hybrid framework lies in the design of parameterized quantum circuits (PQCs), i.e., PQC design. A PQC mainly includes some quantum gates that are easy to implement on NISQ devices and have low noise, such as single-qubit rotation gates, two-qubit control-not (CNOT) gate, etc. The classical parameters that are required to be updated are usually encoded as the rotational angles of these single qubit gates, e.g., the Pauli-Y rotation gate with parameter θ, $R_y(\theta)$. The two or multi-qubit gates have no adjustable parameters. Our purpose is to explore how to place these quantum gates, such as sequential or parallel placement, to express appropriate transformations. Theoretically, any unitary transformation can be constructed from these simple gates with parameters, and then the functions we need can be constructed. But the cost is that the circuit depth it needs is exponentially related to the number of qubits. Unfortunately, a too-deep circuit depth will make the error of the final quantum state of the circuit significantly large on the NISQ devices. Therefore, it is urgent to study how to use relatively shallow parameterized quantum circuits to express the complex functional relationship, and then demonstrate the actual quantum advantages for machine learning.

Since PQCs are usually regarded as quantum neural networks [30], we can also understand them from the perspective of neural networks. In the classic case, different neural network architectures have been proposed to match different problems, such as convolutional neural network [31] for visual problems, recurrent neural network [32] for natural language processing. Similarly, in the quantum case, what kind of PQCs should we devise to adapt to different application tasks?

PQCs can be used not only as models. They can also be used in quantum data encoding, that is, taking the input as a parameter in a PQC in the data preprocessing stage. This is because, for classical data, the input is generally classical, so they need to be encoded into quantum states first, and PQC is an effective encoding method in the NISQ era. Therefore, what kind of PQCs could provide a better encoding circuit is also worth studying.

This thesis mainly focuses on the capabilities and limitations of PQCs. Specifically, we mainly explore the capabilities and potential quantum advantages of PQCs in machine learning applications, as well as some limitations of PQCs in data encoding. These results may shed light on the future research of near-term quantum machine learning.

1.2 Research Problems

As an emerging and promising interdisciplinary research direction in the fields of quantum computing and artificial intelligence, the area of quantum machine learning abounds with scientific research problems worth exploring. Theoretically speaking, quantum computing can be applied to almost all machine learning tasks, such as image classification, recommendation systems, natural language processing, etc. However, to realize quantum advantages, it usually requires fault-tolerant quantum computers and quantum random access memory. Unfortunately, in the current NISQ era, these cannot be realized. On the contrary, on recent quantum devices, we can only realize some shallow-depth quantum circuits to ensure the accuracy of the final quantum state. Therefore, our goal is to explore, by employing shallow PQCs, whether we can still accomplish the above machine learning tasks with (potential) quantum advantages. Next, we list some relevant concrete research problems that this thesis tries to solve.

- How to design suitable parameterized quantum circuit architectures for different application tasks?

From the above background introduction, we know that the role of PQCs in processing quantum machine learning tasks is similar to that of neural networks in classical machine learning. In other words, whether the PQC architecture is appropriate or not will directly affect the performance of variational quantum algorithms for specific tasks. In the classical case, neural networks have various architectures in deep learning to be applied to different tasks, such as convolutional neural networks [31] suitable for vision tasks, recurrent neural networks [32] suitable for natural language tasks, and even attention-based neural networks [33] with excellent performance for both tasks. In the quantum case, a few PQC architectures such as quantum convolutional neural networks [34] and quantum long short-term memory [35] have been proposed for some physical classification problems. However, it is still uncertain whether they are suitable for classical machine learning tasks and whether they have potential quantum advantages. Therefore, it is urgent to study how to design more suitable PQC architectures to realize potential quantum advantages for different quantum machine learning tasks.

- How to choose various PQC-based data encoding strategies for classical data?

In quantum machine learning, the inputs are generally classical, so these inputs need to be encoded into corresponding quantum states in a form similar to data preprocessing before they can be placed on quantum computers for execution. Encoding classical information into quantum one is nontrivial [7], especially more difficult on NISQ devices. One of the feasible schemes is to use PQC for encoding. However, most of the existing PQC-based encoding schemes such as angle encoding, and IQP encoding are designed by experience and lack theoretical support. Therefore, how to systematically understand the encoding strategies based on PQCs and how to select these strategies need to be further studied.

This thesis makes progress to the above research problems by proposing two new PQC architectures, i.e., shadow quantum learning for general classification tasks and quantum self-attention neural networks (QSANN) for natural language processing tasks. Furthermore, as a response to the last research problem, we point out the concentration issue of PQC-based data encoding strategies, which could significantly influence the eventual performance of quantum machine learning tasks. Next, we give the motivation and contributions for these results.

1.3 Motivation

1.3.1 Shadow Quantum Learning

The main idea of a hybrid quantum-classical algorithm is employing parameterized quantum circuits (as a unitary neural network architecture) to search the parameter space and combining classical optimization methods like gradient descent (GD) to find the best parameters [36–39]. These hybrid algorithms have been applied to many topics such as quantum eigensolver [36], quantum simulation [40], quantum state distance estimation [41, 42] and quantum matrix decomposition [43]. So far, most proposals for variational quantum classification process information in the global sense such that the quantum circuit always acts on the whole Hilbert space fulfilling high-dimensional transformation. And the classical feature/information extracted from the quantum system is achieved through measurement. This formulation faces two potential challenges. One is the quantum resource such as the number of quantum gates required may be exponential/polynomial in the number of qubits. However, more efficient architectures could exist by limiting the operating scope to a few selected qubits to achieve the same task performance but significantly reduce the quantum resource required, e.g., constant dependence on the number of qubits. The other challenge is the notorious Barren Plateau problem [44]. As the problem size increases, it will exhibit exponentially vanishing gradients, making the optimization landscape flat and hence untrainable using gradient-based optimization methods.

To overcome the above challenges, we explore a significantly different hybrid architecture inspired by classical shadows. Classical shadows [45], devised from shadow tomography [46], represent a series of succinct classical descriptions of quantum states. These descriptions are generally obtained by employing simple or even local observables to measure on a computational basis. Furthermore, some important quantum properties such as quantum fidelities and entanglement entropies can be predicted using classical shadows rather than possessing full information of quantum states. This provides us the intuition that the idea of obtaining classical shadows may also be helpful in quantum classification. Concretely speaking, our method extracts only "local" features from the subspace of quantum states, which we call *shadow features*, by using only local parameterized quantum circuits acting on a few selected qubits. Then these shadow features are fed into a fully-connected neural network to complete the quantum classification tasks.

1.3.2 Quantum Self-Attention Neural Networks

In recent years, the self-attention mechanism [47], due to its capability of capturing long-term information, has become a dominant neural network framework in machine learning, especially in natural language processing. *Natural language processing* (NLP) is a key subfield of AI that aims to give machines the ability to understand human language. Self-attention neural networks have excellent performance on various NLP tasks such as language modeling [48], machine translation [33], question answering [49], and text classification [50]. This motivates us that it is desired to introduce a self-attention mechanism into quantum neural networks to enhance the performance of the latter. Detailed mathematical description of the self-attention mechanism is deferred to Subsec. 2.5.2.

Due to human language's high complexity and flexibility, NLP tasks are generally challenging to implement. Thus, it is natural to think about whether and how quantum computing can enhance machines' performance on NLP. Some works focus on quantum-inspired language models [47, 51–53] with ideas borrowed from quantum mechanics. Another approach, known as *quantum natural language processing* (QNLP), seeks to develop quantum-native NLP models that can be implemented on quantum devices [54–57]. Most of these QNLP proposals, though at the frontier, lack scalability as they are based on syntactic analysis, which is a preprocessing task requiring significant effort, especially for large data sets. Furthermore, these syntax-based methods employ different PQCs for sentences with different syntactical structures and thus are not flexible enough to process the innumerable complex expressions possible in human language.

To overcome these drawbacks in current QNLP models, we propose the *quantum self-attention neural network* (QSANN), where the self-attention mechanism is introduced into quantum neural networks. Because of the word-embedding technique [58], our method can avoid the problems of the model based on the syntactic analysis mentioned above and also makes use of the excellent ability of the self-attention mechanism. We also note that a recently proposed method [59] for quantum state tomography, an important task in quantum computing, adopts the self-attention mechanism and achieves decent results.

1.3.3 Concentration of Quantum Data Encoding

For a typical quantum machine learning task, the quantum circuit used in the variational quantum algorithms consists of two parts: a data encoding circuit and a QNN. Hence, the design of variational quantum algorithms could be further decomposed into the design of the data encoding circuit and the design of QNN architecture. On the one hand, developing various QNN architectures is the most popular way to improve these algorithms' ability to deal with practical tasks. Numerous architectures such as strongly entangling circuit architectures [60], quantum convolutional neural networks [34], tree-tensor networks [61], and even automatically searched architectures [62–65] have been proposed. On the other hand, one has to carefully design the encoding circuit, which could significantly influence the generalization performance of these algorithms [66, 67]. Consider an extreme case. If all classical inputs are encoded into the same quantum state, these algorithms will fail to do any machine learning tasks. In addition, the kernel's perspective [68–71] also suggests that data encoding strategy plays a vital or even leading role in quantum machine learning algorithms [14, 72, 73]. However, there is much less literature on data encoding strategies, which urgently requires to be studied.

Encoding classical information into quantum data is nontrivial [7], and it is even more difficult on near-term quantum devices. One of the most feasible and popular encoding strategies on NISQ devices is based on *parameterized quantum circuits* (PQCs) [74], such as the empirically designed angle encoding [14, 75], IQP encoding [76], etc. It is natural to ask how to choose these encoding strategies and whether there are theoretical guarantees of using them. More specifically, it is necessary to systematically understand the impact of such PQC-based encoding strategies on the performance of QNNs in quantum machine learning tasks.

1.4 Contributions

The main contributions of this thesis are summarized as follows:

- As the first attempt to extract local features inspired by classical shadows, we propose a variational shadow quantum learning (VSQL) framework that could be adapted to many near-term quantum applications. In particular, we apply this framework to develop quantum classifiers for near-term quantum devices. Firstly, we employ the parameterized shadow quantum circuits $U(\boldsymbol{\theta})$ (denoted as *shadow circuits*) acting on selected local qubit subspace rather than the whole qubit Hilbert space, which considers the operating scope efficiency and the connectivity limit on quantum hardware. Secondly, the shadow features of the input data (encoded as quantum states $\rho^{(m)}$ with labels $y^{(m)}$) will be computed via measuring the Pauli $X \otimes X \cdots \otimes X$ observables on the quantum devices. The final step is to utilize a classical Fully-Connected Neural Network (FCNN) to post-process these shadow features, and we could then decide the label prediction $\hat{y}^{(m)}$ through an activation function. The advantages of this work are multi-fold. First, VSQL can be easily implemented on quantum devices with topological connectivity limitations, since it mainly considers locally-operated quantum circuits. Second, we show that VSQL involves significantly fewer parameters (independent of the problem size) than existing variational quantum classifiers [60, 77]. Notably, we prove that VSQL could naturally avoid the Barren Plateau issue [44] (gradients vanishing issue in QML) by limiting the operating scope. Finally, we demonstrate real-world applications of VSQL to do quantum state classification and handwritten digit recognition. We in particular show that VSQL outperforms existing variational quantum classifiers in the test accuracy while requiring much fewer parameters. (Chapter 3)

- We propose a quantum self-attention neural network (QSANN), where the self-attention mechanism is introduced into quantum neural networks. In each quantum self-attention layer of QSANN, we first encode the inputs into high-dimensional quantum states, then apply PQCs on them according to the layout of the self-attention neural networks, and finally adopt a *Gaussian projected quantum self-attention* (GPQSA) to obtain the output effectively. To evaluate the performance of our model, we conduct numerical experiments of text classification with different data sets. The results show that QSANN outperforms the currently best-known QNLP model as well as a simple classical self-attention neural network on test accuracy, implying the potential quantum advantages of our method. The advantages of this work are multi-fold. First, our proposal is the first QNLP algorithm with a detailed circuit implementation scheme based on the self-attention mechanism. This method can be implemented on NISQ devices and is more practicable on large data sets compared with previously known QNLP methods based on syntactic analysis. Second, in QSANN, we introduce the Gaussian projected quantum self-attention, which can efficiently dig out the correlations between words in high-dimensional quantum feature space. Furthermore, visualization of self-attention coefficients on text classification tasks confirms its ability to focus on the most relevant words. Last, we experimentally demonstrate that QSANN outperforms existing QNLP methods based on syntactic analysis [78] and simple classical self-attention neural networks on several public data sets for text classification. Numerical results also imply that QSANN is resilient to quantum noise. (Chapter 4)

- We show that for the usual PQC-based data encoding strategies with a fixed width, the average encoded state is close to the maximally mixed state at an exponential speed in depth. In particular, we establish the following. Firstly, we theoretically give the upper bound of the quantum divergence between the average encoded state and the maximally mixed state, which depends explicitly on the hyper-parameters (e.g., qubit number and encoding depth) of PQCs. From this bound, we find that for a fixed qubit number, the average encoded state concentrates on the maximally mixed state exponentially on the encoding depth. Secondly, we show that the quantum states encoded by deep PQCs will seriously limit the trainability of a quantum classifier and further limit its classification ability. Thirdly, we show that the quantum states encoded by deep PQCs are indistinguishable from a quantum information perspective. Finally, we support the above findings by numerical experiments on both synthetic and public data sets. (Chapter 5)

1.5 Publications

1.5.1 Related to the Thesis:

1. **Guangxi Li**, Zhixin Song, and Xin Wang. "VSQL: variational shadow quantum learning for classification." *Proceedings of the AAAI Conference on Artificial Intelligence.* Vol. 35. No. 9. 2021. (Chapter 3)

2. **Guangxi Li**, Xuanqiang Zhao, and Xin Wang. "Quantum Self-Attention Neural Networks for Text Classification." *arXiv preprint arXiv:2205.05625.* 2022. (Chapter 4)

3. **Guangxi Li**, Ruilin Ye, Xuanqiang Zhao, and Xin Wang. "Concentration of Data Encoding in Parameterized Quantum Circuits." *arXiv preprint arXiv:2206.08273.* 2022. To appear in NeurIPS 2022. Spotlight. (Chapter 5)

1.5.2 Others (*equal contribution):

4. Youle Wang*, **Guangxi Li***, and Xin Wang. "Variational quantum Gibbs state preparation with a truncated Taylor series." *Physical Review Applied* 16.5 (2021): 054035.

5. Youle Wang, **Guangxi Li**, and Xin Wang. "A Hybrid Quantum-Classical Hamiltonian Learning Algorithm." *arXiv preprint arXiv:2103.01061.* 2021.

6. **Guangxi Li**, Youle Wang, Yu Luo, and Yuan Feng. "Quantum data fitting algorithm for non-sparse matrices." *arXiv preprint arXiv:1907.06949.* 2019.

1.6 Thesis Outline

The outline of this thesis is organized as follows:

- *Chapter 2:* This chapter introduces some necessary quantum basics, which are helpful for machine learning researchers: quantum state, density matrix, quantum gate, quantum circuit and so on. We then give an overview of the hybrid quantum-classical computing framework, because the algorithms involved in this thesis basically follow this framework. Next, we introduce some quantum machine learning tasks, like quantum classification, and quantum natural language processing. Finally, we present some quantum data encoding strategies.

- *Chapter 3:* Classification is one of the most important tasks in quantum machine learning. This chapter proposes a new variational shadow quantum learning framework to deal with this problem. First, we introduce the model, loss function, analytical gradient, parameter quantity, and the analysis of the computational complexity and theoretical classification ability of the framework in the case of binary classification. At the same time, we show how it can avoid the barren plateau issue. Then we show the similarities and differences of the framework in the above components in the case of multi-label classification. Finally, we demonstrate the advantages of this framework compared with some existing variational quantum algorithms through numerical experiments.

- *Chapter 4:* Quantum natural language processing is a promising research direction. This chapter focuses on text classification and proposes a quantum self-attention neural network framework to solve the problem. First, we introduce the main components of the framework: the quantum self-attention layer, the selection of parameterized quantum circuits, the construction of loss function, the analysis of analytical gradients, and the analysis of overall complexity. Then we carried out numerical experiments on some toy-scale and medium-scale datasets, and the results show that this framework has advantages in accuracy and parameter quantity compared with some existing methods.

- *Chapter 5:* Quantum data encoding is a very important direction that is seldom studied at present. This chapter focuses on the data encoding strategies based on PQCs and points out the important concentration issue. Specifically, we show that for PQCs with a fixed number of qubits, the average encoded state will close to the maximum mixed state at an exponential speed with the increase of depth. This result shows that adopting this encoding strategy will severely impact downstream tasks. We also verified this by numerical experiments in the quantum supervised learning task.

- *Chapter 6:* We summarize the main contents and contributions of this thesis and discuss possible future research directions.

Chapter 2

Preliminaries

In this chapter, we briefly introduce some preliminaries that are necessary for this thesis, like quantum basics, parameterized quantum circuits, hybrid quantum-classical computing framework, quantum classification, quantum natural language processing, and quantum data encoding.

2.1 Quantum Basics

Here, we briefly introduce the basic concepts of quantum computation that are necessary for this thesis. Interested readers are recommended to the celebrated textbook by Nielsen and Chuang [79].

2.1.1 Quantum States

Information in the quantum computing field is represented by n-qubit quantum states over Hilbert space $\mathbb{C}^{2^n \times 2^n}$, which could be mathematically described by positive semi-definite matrices $\rho \succeq 0$ with property $\mathrm{Tr}(\rho) = 1$. Following this density matrix formulation, a quantum state is pure if $\mathrm{Rank}(\rho) = 1$; otherwise, it is mixed. For a pure state ρ, it can be represented by a unit vector in the form that $\rho = |\psi\rangle\langle\psi|$, where the *ket* notation $|\psi\rangle \in \mathbb{C}^d$ denotes a column vector and *bra* notation $\langle\psi| = |\psi\rangle^\dagger$ with † denoting conjugate transpose.

In general, we would also use $|\psi\rangle$ to denote a pure state for simplicity. A mixed state could be represented as $\rho = \sum_i q_i |\psi_i\rangle\langle\psi_i|$, where the coefficients $q_i > 0$ records the probability for a quantum system to be in each corresponding pure state $|\psi_i\rangle\langle\psi_i|$ and hence $\sum_i q_i = 1$. Specifically, a mixed state whose density matrix is proportional to the identity matrix is called the maximally mixed state $\mathbb{1} \equiv \frac{I}{2^n}$.

2.1.2 Quantum Gates

Typical single-qubit gates include Pauli gates,

$$X \equiv \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad Z \equiv \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \tag{2.1}$$

and their corresponding rotation gates $R_P(\theta) \equiv \mathrm{e}^{-i\theta P/2}$ with a parameter $\theta \in [0, 2\pi)$ and $P \in \{X, Y, Z\}$. Another commonly used gate $U3$ that appeared in this thesis is defined as $U3(\theta_1, \theta_2, \theta_3) \equiv R_z(\theta_3)R_y(\theta_2)R_z(\theta_1)$, which can implement an arbitrary single-qubit unitary transformation with appropriate parameters. In this thesis, R_z, R_y are equivalent to R_Z, R_Y without specified. A multi-qubit gate can be either an individual gate (e.g., CNOT) or a tensor product of single-qubit gates, e.g., $Z \otimes Z$, $Z \otimes I$, $Z^{\otimes n}$ and so on.

2.1.3 Quantum Evolution

The evolution of a pure quantum state $|\psi\rangle$ is mathematically described by applying a quantum circuit (or a quantum gate), i.e., $|\psi'\rangle = U|\psi\rangle$, where U is the unitary operator (matrix) representing the quantum circuit and $|\psi'\rangle$ is the quantum state after evolution. Similarly, due to the linearity, the evolution of a mixed quantum state $\rho = \sum_i q_i |\psi_i\rangle\langle\psi_i|$ could also be mathematically described by employing a quantum circuit $\rho' = U\rho U^\dagger = \sum_i q_i U |\psi_i\rangle\langle\psi_i| U^\dagger$, where the coefficients $q_i \geq 0$ and $\sum_i q_i = 1$.

Since quantum evolution is essentially a matrix operation, we can use a series of quantum gates with adjustable parameters to evolve so that the initial state can evolve to any desired target quantum state. The work of adjusting the parameters in the quantum gate is very similar to the parameter updating process of the neural network. We will further explain this later.

2.1.4 Quantum Measurement

Quantum measurement is usually introduced at the end of algorithms to extract classical information from the final quantum state. For instance, given a pure quantum state $|\psi\rangle$ and an observable O, one could design quantum measurements to obtain the information $\langle\psi| O |\psi\rangle$. Similarly, if a quantum state ρ is in density matrix form, the quantum measurement is in the form $\langle O\rangle = \mathrm{Tr}(O\rho)$, which we call the expectation of the observable O. Here, O is Hermitian.

Now we describe these two kinds of quantum measurements through concrete examples. For the pure state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, if we measure it on a computational basis, then we will get the state $|0\rangle$ with probability

$$p_0 = \langle\psi| M_0 |\psi\rangle = \langle\psi| \cdot |0\rangle\langle 0| \cdot |\psi\rangle = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = |\alpha|^2,$$

and similarly, get the state $|1\rangle$ with probability $p_1 = |\beta|^2$. Also for the pure state $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$, the expectation of Pauli Z operator is defined as $\langle Z\rangle = \mathrm{Tr}(Z |\psi\rangle\langle\psi|) = \langle\psi| Z |\psi\rangle = \langle\psi| M_0 |\psi\rangle - \langle\psi| M_1 |\psi\rangle = p_0 - p_1$. Here, $M_0 \equiv |0\rangle\langle 0|$ and $M_1 \equiv |1\rangle\langle 1|$ denote the quantum observables. Obviously, $\langle Z\rangle$ varies in the range $[-1, 1]$ and we cannot estimate it through just one-time measurement. Hence, we need to run and measure the entire circuit multiple times (or multiple shots) to get multiple measurement results. Suppose we repeat it S times and obtain S_0 times 0 and $S_1 = S - S_0$ times 1, then $\langle Z\rangle \approx \frac{S_0 - S_1}{S}$. From the Chernoff bound, the number of repetitions S scales of $O(\frac{1}{\epsilon^2} \log \frac{1}{\eta})$ such that $|\langle Z\rangle - \frac{S_0 - S_1}{S}| < \epsilon$ with probability at least $1 - \eta$. Empirically, 2048 or 4096 shots are sufficient to satisfy most requirements.

Within this thesis, we focus on the hardware-efficient Pauli measurements, i.e., setting O as Pauli operators or their tensor products. For instance, we could choose $Z_1 \equiv Z \otimes I^{\otimes(n-1)}$, $X_2 \equiv I \otimes X \otimes I^{\otimes(n-2)}$, $Z_1 Z_2 \equiv Z \otimes Z \otimes I^{\otimes(n-2)}$, etc., with n qubits in total.

2.2 Parameterized Quantum Circuits

The parameterized quantum circuit is the most studied formalism in NISQ algorithms and also is the core of this thesis. Therefore, in this section, we briefly introduce how PQC is defined and designed.

Generally speaking, a PQC refers to a unitary operation with a series of adjustable parameters. By applying it to some common initial states, such as the state $|0\rangle$, one can obtain the corresponding variational quantum state. Its purpose is to adjust these parameters so that the final variational quantum state approximates the desired state. Similar to the universal approximation theorem in neural networks [80], the universal PQC always exists. Actually, it is known from [81] that the collection of all one-qubit gates together with any set of imprimitive[1] two-qubit gates are universal. However, the difficulty is that such PQCs often require an exponential level of depth, so it is unlikely to be achieved at the current stage. Fortunately, the authors of [82] pointed out that the data generated by real physical systems usually have symmetry and locality, which means that we may only employ some simple PQCs, that is, they are not so deep, but we can still approach the quantum states we want. Therefore, how to design and select appropriate PQCs are very significant for solving practical machine learning problems.

Different PQC architectures will significantly affect the performance of NISQ algorithms, and one usually designs a PQC from two perspectives. On the one hand, from the perspective of the problem heuristic, PQCs will affect the convergence speed and the approximation degree between the final variational quantum state and the target state that can solve the problem optimally. For example, the unitary coupled cluster scheme [83] is one prominent case of adopting this perspective. On the other hand, from the perspective of quantum hardware, deeper PQCs will bring errors that cannot be ignored, and some complex PQCs are difficult to construct through native and simple quantum gates. Therefore, one usually needs to consider which one of these two perspectives is preferred according to the actual problems and applications.

[1] A two-qubit gate is called *imprimitive* if it can map a two-qubit product state into a non-product state. A typical example of imprimitive two-qubit gates is the frequently used CNOT or CZ gate.

This thesis mainly focuses on hardware-efficient PQCs [74], mainly because we regard PQCs as alternatives to neural networks. We all know that neural networks have excellent performance in machine learning, which also inspires us whether we can use these hardware-efficient PQCs to obtain excellent performance in some practical machine learning tasks. We think this might be a fascinating yet challenging research direction.

Next, we will specifically introduce some hardware-efficient PQCs. An early work [28] proposed a kind of hardware-efficient PQCs suitable for hardware constraints. These PQCs not only use a limited set of quantum gates but also need to obey the topological connections between qubits in hardware devices. These quantum gates usually include single-qubit Pauli rotation gates and two-qubit entangling gates. These single-qubit gates act on some or all of the qubits in parallel and form a *block* together with the entangling gates, also known as a *layer*. The hardware-efficient PQCs usually contain multiple such layers.

In general, a hardware-efficient parameterized quantum circuit with L layers [74] has the form

$$U(\boldsymbol{\theta}) = \prod_{j}^{L} U_j(\theta_j)V_j, \tag{2.2}$$

where $U_j(\theta_j) = \exp(-i\theta_j P_j/2)$ denotes a unitary derived from a Hermitian operator P_j and V_j denotes some fixed operators such as Identity, CNOT and so on. Typically, P_j are chosen as Pauli string operators. i.e., tensor products of Pauli operators and V_j are selected according to the architectures of the actual quantum hardware, for example, CNOT or CZ gates for superconducting computers [84] or XX gates for trapped ion computers [85], see Fig. 2.1 for an illustration [74].

2.3 Hybrid Quantum-Classical Algorithms

After introducing PQCs, we now introduce the most popular algorithm framework in the current NISQ era, the hybrid quantum-classical computing framework. According

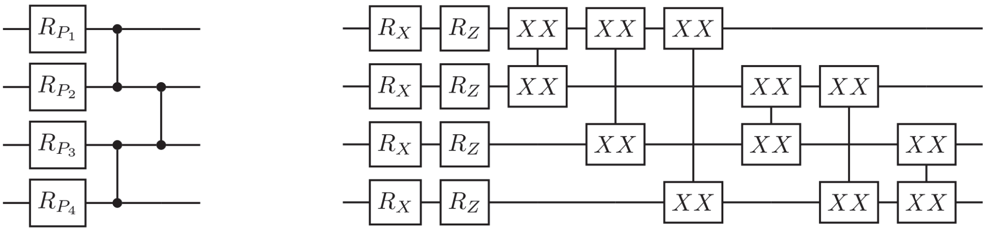


FIGURE 2.1: Examples of hardware-efficient parameterized quantum circuits with CZ gates (left) for superconducting computers and XX gates (right) for trapped ion computers.

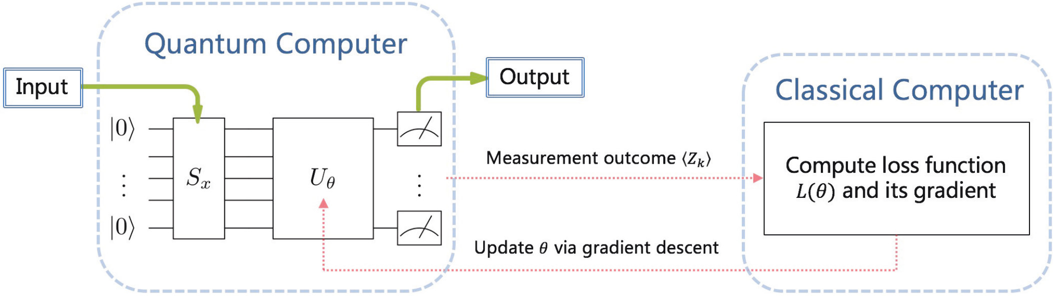

Quantum Computer
Input
Output
$|0\rangle$
$|0\rangle$
S_x
U_θ
Measurement outcome $\langle Z_k \rangle$
Update θ via gradient descent
Classical Computer
Compute loss function $L(\theta)$ and its gradient

FIGURE 2.2: The architecture of hybrid quantum-classical algorithms, where the quantum computer is considered as the main body and the classical computer is an assistant. In the figure, S_x denotes the encoder circuit and U_θ is the parameterized quantum circuit. The classical computer uses the estimation of the measurement outcome, e.g., expectations of Pauli-Z_k $\langle Z_k \rangle$, to construct a loss function and compute its gradient, and after that makes use of some optimization methods to update the parameters θ.

to this framework, we can define various machine learning models to adapt to different applications.

The architecture of hybrid quantum-classical algorithms is depicted in Fig. 2.2. It mainly contains the encoder circuit S_x and the parameterized quantum circuit U_θ, which are implemented on quantum computers, and the whole learning procedure is done with the assistance of classical computers that are used to construct loss function and compute gradients of parameters. In the following subsections, we will introduce each of these components.

2.3.1 Encoder Circuit

The encoder circuit S_x refers specifically to the data pre-processing operation which aims to encode the classical data vector x into a quantum Hilbert space that usually has a higher dimensionality. On the one hand, the classical input x could be easily fed into a quantum computer by being transformed to a quantum state through the encoder circuit; on the other hand, the encoding process generally involves some non-linear feature maps, which makes the classical data sets becoming easier to extract useful features for classification or other tasks.

There are various encoding methods have been proposed. One of the most direct is amplitude encoding, which encodes the classical input vector into the amplitudes of quantum states [86]. The most significant advantage of this encoding is that only n-qubit quantum states can represent the classical vectors with 2^n dimension; that is, the required memory can be compressed exponentially. Therefore, it is an essential source for some quantum machine learning algorithms to accelerate exponentially [20]. Unfortunately, this encoding requires a depth equivalent to the exponential number of qubits and requires quantum random access memory, which is currently unavailable. Even if we can perform such encoding, the time complexity required for data loading and readout is at least linear or polynomial [87]; hence it might still be impossible to achieve the above exponential acceleration.

Another major category is to use PQC-based encoding [74], i.e., to encode each element of the input vector as a parameter of PQCs. This scheme is the easiest to realize at present and perhaps the most capable of realizing potential quantum advantages. The most common example is angle encoding [88], i.e., each classical element corresponds to a Pauli rotation gate on a qubit, and the number of qubits required is equal to the dimension of the classical vector. The advantage of this encoding is that it can bring some nonlinear feature mapping and better solve downstream tasks; The disadvantage is that it consumes too many qubit resources. Some other encoding strategies may also provide potential quantum advantages, such as IQP [76], a quantum version of the random kitchen sink [89] and so on. More data encoding strategies are concluded in Sec. 2.6 with a detailed mathematical description.

2.3.2 Loss Function

With the above preparation of the encoder circuit and parameterized quantum circuit, one can obtain the final parameterized quantum state $|\psi(\theta)\rangle$. Next, we can design the corresponding loss function with the help of classical computers and according to different target tasks. Generally speaking, there are two kinds of learning tasks. One is to make the parameterized quantum state approach the target quantum state, such as in quantum chemistry. The objective is to find a distance measure between the target and the generated states. Another one is to extract useful information through this parameterized quantum state, such as in the supervised learning classification tasks. Similarly, this also aims to design a distance measure between the useful information and the target label information. In this subsection, we will introduce the most commonly used loss functions for these two types of tasks: fidelity and expectation of Pauli operators.

2.3.2.1 Fidelity

Quantum fidelity, which represents the overlap between two quantum states, is a typical distance measure of quantum states in the quantum area. It can be defined as

$$F(|\psi_{targ}\rangle, |\psi(\theta)\rangle) \equiv |\langle\psi_{targ}|\psi(\theta)\rangle|^2, \tag{2.3}$$

where $|\psi_{targ}\rangle$ denotes the target quantum state and $|\psi(\theta)\rangle$ is the parameterized quantum state generated by the parameterized quantum circuit. The fidelity-based loss function could be designed as one minus fidelity or just the negative fidelity, i.e.,

$$L(\theta) = 1 - F(|\psi_{targg}\rangle, |\psi(\theta)\rangle); \tag{2.4}$$

$$L(\theta) = -F(|\psi_{targ}\rangle, |\psi(\theta)\rangle). \tag{2.5}$$

This kind of loss function is employed in various state preparation algorithms, and for specific examples, this target state is often designed as a computational basis state $|e_i\rangle$, e.g.,

$$F(|e_i\rangle, |\psi(\theta)\rangle) \equiv |\langle e_i|\psi(\theta)\rangle|^2. \tag{2.6}$$

Typical instances are included in quantum optics [90–92], excited state preparation [93, 94] and quantum machine learning [73, 74, 95, 96].

2.3.2.2 Expectation of Pauli Operators

As mentioned in Subsec. 2.1.4, the quantum measurement is a common method to extract classical information from quantum states. Therefore, if we want to construct the loss function according to this classical information, we must design different measurement methods. In the NISQ era, a simple and effective way is to adopt the expectation of Pauli operators. Concretely speaking, given the final quantum state $|\psi(\theta)\rangle$, the expectation $\langle H\rangle$ of a Pauli operator H is defined as

$$\langle H\rangle_\theta \equiv \mathrm{Tr}(H\,|\psi(\theta)\rangle\langle\psi(\theta)|) = \langle\psi(\theta)|\,H\,|\psi(\theta)\rangle\,. \tag{2.7}$$

Here, H is a linear combination of Pauli operators or their primitive tensor product, i.e.,

$$H = \sum_i \alpha_i P_i, \tag{2.8}$$

where each simple Pauli operator P_i is summed with the corresponding coefficient α_i.

The expectation-based loss function could be directly defined as

$$L(\theta) = \langle H\rangle_\theta \tag{2.9}$$

for the variational quantum eigensolver problem, which aims to find the minimum energy or the minimum eigenvalue of the Hermitian operator H.

For the supervised learning task, here we take the binary classification as an example, the expectation-based loss function is defined as

$$L(\theta; x, y) \equiv \left|\langle H\rangle_{\theta,x} - y\right|^2 = \left|\langle\psi(\theta; x)| H |\psi(\theta; x)\rangle - y\right|^2, \tag{2.10}$$

where x denotes the input and $y \in \{-1, 1\}$ is the corresponding label.

2.3.3 Optimizer

After defining the loss function, we need to use some optimization methods to optimize these loss functions. In this subsection, we mainly introduce some gradient-based optimization methods, such as finite difference, parameter shift rule and quantum natural gradient and so on. We also briefly mention some gradient-free methods.

2.3.3.1 Gradient-based Optimization

The gradient-based optimization method is the most commonly used approach to optimize a smoothing loss function. This is because according to the optimization theory, the negative gradient represents the direction in which the loss function drops most rapidly. Given the loss function $L(\theta)$, we generally have the following update rule

$$\theta_i^{(t+1)} \longleftarrow \theta_i^{(t)} - \eta \frac{\partial L(\theta)}{\partial \theta_i}, \tag{2.11}$$

where θ_i denotes the i-th element of the parameter vector θ, $t = 0, 1, 2, \ldots$ denotes the t-th iteration and η is the learning rate which could influence the convergence speed. Here, $\theta_i^{(0)}$ means the initial guess which usually obeys some distributions such as Gaussian or uniform, for the parameters.

The above update rules can be applied to almost all situations, but sometimes in order to improve the convergence speed during training, some improved rules are proposed such as stochastic gradient descent (SGD) and gradient update with momentum, i.e., Adam. The former takes into account the problem of large number of data samples, that is, the sample size is too large to be loaded into memory all at once for computing derivatives. Instead, only one or some data samples are randomly selected to calculate the derivative, and it is proved that the convergence accuracy of SGD is theoretically the same as that of gradient descent. Adam considers the “zigzag” problem of gradient, that is, the real negative gradient only represents the direction of the fastest descent at the current moment, but not the direction of the fastest descent overall. Concretely, Adam not only considers the current gradient, but also considers the previous gradient, and adds them linearly with different weights to ensure the fastest overall descent. What’s more, the experience in engineering practice also tells us that Adam has an excellent performance in (quantum) machine learning, thus it is also the update rule mainly adopted by this thesis.

Various methods for estimating gradients on quantum computers have been proposed [97], here we will briefly introduce some of them.

Finite Difference The gradient estimation process by using a finite difference scheme could be described as follows

$$\frac{\partial L(\theta)}{\partial \theta_i} \approx \frac{L(\theta + \varepsilon \mathrm{e}_i) - L(\theta - \varepsilon \mathrm{e}_i)}{2\varepsilon}, \tag{2.12}$$

where e_i denotes the unit vector in the i-th direction and ε is some smaller value. Note that we need to evaluate two times of the loss function to obtain the gradient $\frac{\partial L(\theta)}{\partial \theta_i}$ and more sample times are required to get a better estimation of the gradient due to the limited accuracy obtained from the quantum devices.

Parameter Shift Rule From the chain rule, the partial derivative $\frac{\partial L(\theta)}{\partial \theta_i}$ could be written as a function of the partial derivatives of the Pauli expectation values $\frac{\partial \langle P \rangle_\theta}{\partial \theta_i}$, where P denotes some Pauli operator. Then the gradient estimation process by using parameter shift rule [98] is described as

$$\frac{\partial \langle P \rangle_\theta}{\partial \theta_i} = \frac{\langle P \rangle_{\theta + \frac{\pi}{2}\mathrm{e}_i} - \langle P \rangle_{\theta - \frac{\pi}{2}\mathrm{e}_i}}{2}, \tag{2.13}$$

where e_i is the unit vector along the i-th direction of θ.

Here we note that this method also requires estimating the Pauli expectation values twice. But different from the finite difference method, the gradient estimated here is theoretically exact. Therefore, this becomes a popular quantum method for gradient estimation.

Quantum Natural Gradient The quantum natural gradient is an extension of the classical natural gradient. In the classical case, the efficiency of standard gradient descent is affected by the flat Euclidean parameter space, hence the (classical) natural gradient of non-Euclidean parameter space is proposed to improve it [99]. The updating rule of quantum natural gradient [100] is as follows

$$\theta_i^{(t+1)} \longleftarrow \theta_i^{(t)} - \eta \mathcal{F}^{-1}(\theta) \frac{\partial L(\theta)}{\partial \theta_i}, \tag{2.14}$$

where $\mathcal{F}(\theta)$ denotes the Fubini-Study metric tensor or quantum Fisher information metric.

Compared with other gradient descent methods, the quantum natural gradient has the advantage that it can effectively avoid falling into local minima [101], and thus has better performance [100]. However, the disadvantage is that under the current quantum hardware conditions, the Fubini-Study metric tensor is very difficult to estimate.

2.3.3.2 Gradient-free Optimization

In addition to the above gradient-based methods, some gradient-free methods have also been proposed. For example, [102] proposes a natural evolution algorithm, which uses the estimation of a natural gradient to update parameters instead of directly calculating the gradient. Some authors [103, 104] use reinforcement learning algorithms to learn an optimal strategy to optimize the parameters of quantum approximate optimization algorithms. In addition, [105] uses the sequential minimization optimization method, which is proven to be effective in classical support vector machines. Its core idea is to decompose the optimization process into smaller components that are easy to solve.

2.4 Quantum Classification

Classification is one of the most important tasks in machine learning. Its purpose is to classify the input data of a given category into the correct category, which can be one category or multiple categories. A typical classical example is the spam filtering system, where we can divide an email into spam and non-spam. In addition, there are also other tasks or challenges: for example, recognizing handwritten numbers and classifying them into specific number symbols; Identifying whether the user's reviews of a restaurant are positive or negative, and so on.

Generally speaking, for different tasks, we should use different models to optimally adapt them. However, no matter what kind of models we employ, a training data set containing a large number of inputs and outputs (or labels) is required so that the model can be trained from it. Furthermore, the training data set should cover all possible scenarios as much as possible, and provide enough data samples for each category, so that the model can be trained correctly.

Quantum classification, as its name implies, is to add quantum elements to the classical classification. There are usually three forms: First, only the training data set is quantum, i.e., the inputs are some quantum states with known labels. This situation is also known as using classical classification models to classify quantum data [106–108]; Second, only the classification model is quantum, that is, the inputs are still classical but they need to be encoded into quantum inputs (i.e., quantum states) first, and then a quantum classifier is adopted to classify [60, 74, 76]; Third, both the training data sets and the models are quantum. This thesis mainly focuses on the latter two forms, namely, using quantum models to classify classical data or quantum data.

2.4.1 Classification Task

The classification task could be described as follows: given a labeled training data set $\mathcal{D}^{(train)} = \{(\rho^{(m)}, y^{(m)})\}$, our purpose is to learn a complex mapping f between each set element (input) $\rho^{(m)}$ and its corresponding label $y^{(m)}$ such that

$$f(\rho^{(m)}) \approx y^{(m)}. \tag{2.15}$$

However, our expectation is that this well-trained mapping f is not only suitable for the training data set, but also for the test data set (i.e., recognize an unseen dog as a dog)

$$f(\rho^{(unseen)}) \approx y^{(unseen)}. \tag{2.16}$$

This property is called generalization ability, which is extremely essential for all classification problems and is also an important indicator to benchmark the performance of a classifier. Here, we need to note that the input ρ denotes the quantum data. However, if the input is classical, then ρ denotes the encoded quantum state of the classical input.

2.4.2 Quantum Classifiers

Quantum classifiers are widely studied and proposed with the purpose of using quantum-enhanced features to achieve quantum advantages against the classical ones [76, 109]. Especially in the NISQ era, due to the unavailability of fault-tolerant quantum computers, a large number of variational quantum classifiers are developed as temporary schemes [34, 60, 61, 77, 98, 110, 111]. One could also refer to the relatively comprehensive review [112] for more quantum classifiers.

As the main focus of this thesis, the variational quantum classifiers are actually the hybrid quantum-classical algorithms that are mentioned above. Therefore, although our ultimate goal is that quantum classifiers can achieve quantum advantages, the current variational quantum classifiers obviously cannot achieve this goal. The dilemma mainly comes from two aspects. On the one hand, the advantages of the expression ability of the variational quantum circuits compared with that of the classical neural networks are still unclear; On the other hand, the variational quantum classifiers are faced with a serious gradient vanishing problem, i.e., the notorious barren plateaus [44]. Fortunately, some researchers have analyzed the possibility of quantum advantages from the kernel's perspective [68], but further exploration is still needed.

2.5 Quantum Natural Language Processing

Natural language processing (NLP) is an important branch of artificial intelligence, which aims to let machines learn to understand human language. The purpose of quantum natural language processing is to explore whether and how to use quantum computing to enhance the performance of machines in the NLP field. The currently proposed quantum natural language processing methods are generally divided into two categories: One is quantum-inspired language models [47, 51–53] with borrowed ideas from quantum mechanics; Another approach, just called *quantum natural language processing* (QNLP), seeks to develop quantum-native NLP models that can be implemented on quantum devices [54–57]. The focus of this thesis is the latter, where potential quantum advantages are expected via building complex quantum circuits that are classically intractable. In this section, we introduce some background knowledge in the QNLP field involved in this thesis.

2.5.1 Text Classification

As one of the central and basic tasks in the NLP field, text classification is to assign a given text sequence to one of the predefined categories. Examples of text classification tasks considered in this thesis include topic classification and sentiment analysis. A commonly adopted approach in machine learning is to train a model with a set of pre-labeled sequences. When fed a new sequence, the trained model will be able to predict its category based on the experience learned from the training data set.

2.5.2 Self-Attention Mechanism

The self-attention mechanism leads to an excellent leap in classical machine learning because it can connect two words at any distance, alleviating the problem of weak long-range relationships in long short-term memory. This impact is particularly evident in natural language processing. This is also the primary reason why we introduce self-attention into quantum neural networks.

In a self-attention neural network layer [33], the input data $\{x_s \in \mathbb{R}^d\}_{s=1}^S$ are linearly mapped, via three weight matrices, i.e., query $W_q \in \mathbb{R}^{d\times d}$, key $W_k \in \mathbb{R}^{d\times d}$ and value $W_v \in \mathbb{R}^{d\times d}$, to three parts $W_q x_s$, $W_k x_s$, $W_v x_s$, respectively, and by applying the inner product on the query and key parts, the output is computed as

$$y_s = \sum_{j=1}^{S} a_{s,j} \cdot W_v x_j \tag{2.17}$$

with

$$a_{s,j} = \frac{\mathrm{e}^{x_s^\top W_q^\top W_k x_j}}{\sum_{l=1}^{S} \mathrm{e}^{x_s^\top W_q^\top W_k x_l}}, \tag{2.18}$$

where $a_{s,j}$ denote the self-attention coefficients.

2.5.3 DisCoCat Model

DisCoCat model [113] denote by tensors the meaning of words, where the order of each tensor is specified by the grammatical types of words. A word type p has a left (p^l) and a right adjoint (p^r), with two contraction rules:

$$p \cdot p^r \to 1 \qquad p^l \cdot p \to 1. \tag{2.19}$$

And a transitive verb type will return an s. For example, a transitive sentence such as “Alice likes apples” has the following derivation:

$$n \cdot (n^r \cdot s \cdot n^l) \cdot n \to (n \cdot n^r) \cdot s \cdot (n^l \cdot n) \to 1 \cdot s \cdot 1 \to s. \tag{2.20}$$

The overall pipeline has the following steps: first, do syntactic analysis on sentences; Step 2: deduce according to the syntax tree of DisCoCat; Step 3: rewrite according to the diagram of DisCoCat; The fourth step is to design the corresponding quantum circuit; Fifth, use the quantum compiler to compile the quantum circuit into the executable language of the quantum computer; The sixth step is to run it on a quantum computer. In the last step, post-processing such as measurement is carried out, and the results are obtained.

2.6 Quantum Data Encoding

Quantum data encoding [86] is an essential step for quantum algorithms to accept classical input. In the classical case, the information is extracted from the input to do the downstream machine learning tasks; Similarly, in the quantum case, the useful information needs to be reserved as much as possible via quantum data encoding strategies to do quantum machine learning tasks. Hence, a good encoding strategy is necessary, especially in the current NISQ era. In this section, we outline some common encodings.

2.6.1 Amplitude Encoding

Amplitude encoding is the most direct encoding strategy. It encodes an d-dimensional vector $\boldsymbol{x}$ into an $\log_2 d$-qubit quantum state whose amplitudes in the computational basis are the elements of the vector,

$$\boldsymbol{x} \rightarrow |x\rangle \equiv \sum_{i=0}^{d-1} x_i |i\rangle . \tag{2.21}$$

Here we assume $d = 2^n$ with some integer n, and if that is not the case, we pad the vector to $2^{\lceil \log_2 d \rceil}$ dimensions with zero. Since it can express information of $O(d)$ by only using quantum memory of $O(\log d)$, amplitude encoding is a primary reason why many quantum algorithms can achieve exponential speedup. Unfortunately, this encoding cannot be implemented accurately on current quantum devices, which leads to exponential acceleration only at the theoretical level. Therefore, it is urgent to find more practical encoding strategies to realize quantum advantages.

2.6.2 Repeated Amplitude Encoding

Repeated amplitude encoding means we repeat amplitude encoding k times in a tensor product form,

$$\boldsymbol{x} \rightarrow |x\rangle \otimes |x\rangle \otimes \cdots \otimes |x\rangle . \tag{2.22}$$

This can enhance the power of amplitude encoding by employing $O(k \log_2 d)$ qubits. In general, the richer the feature types of input data contained in an encoding, the more beneficial the following quantum model circuit is, i.e., the stronger its encoding power. From this point of view, compared with the original amplitude encoding, the repeated amplitude encoding contains more polynomial features, e.g., $O(x^2), O(x^3), \ldots, O(x^k)$, hence it can be viewed as an enhanced version.

2.6.3 Basis Encoding

Basis encoding is a ubiquitous encoding strategy in qubit-based quantum computing. It encodes the binary representation of an input into a computational basis state,

$$x \rightarrow |x_{n-1}\rangle \cdots |x_1\rangle |x_0\rangle , \tag{2.23}$$

where $x = \sum_{k=0}^{n-1} 2^k x_k$. For example, 5 is encoded into $5 = 101 \rightarrow |1\rangle |0\rangle |1\rangle$. It requires $O(n)$ qubits to encode a scalar and $O(nd)$ qubits a vector with d dimensions.

2.6.4 Angle Encoding

Angle encoding is possibly the most common encoding strategy in the NISQ era. It encodes n-dimensional vector $\boldsymbol{x}$ into n-qubit quantum product state,

$$\boldsymbol{x} \rightarrow R_y(x_0) |0\rangle \otimes R_y(x_1) |0\rangle \otimes \cdots \otimes R_y(x_{n-1}) |0\rangle , \tag{2.24}$$

where x_k's are generally normalized to $[0, 2\pi]$. Here, we note that the Pauli-Y rotation gate could be changed into other (Pauli) gates, e.g., $R_x(x_k)$.

2.6.5 PQC-based Encoding

PQC-based encoding strategy means a d-dimensional vector $\boldsymbol{x}$ is embedded into a parameterized quantum circuit with d parameters, and the parameters are usually the rotation angles of Pauli gates. The above angle encoding can be regarded as the most straightforward PQC-based encoding strategy. Theoretically speaking, a kind of PQC corresponds to a concrete encoding strategy. Therefore, among these numerous PQC encoding strategies, it is promising to explore which ones can better serve quantum machine learning.

Chapter 3

Variational Shadow Quantum Learning

3.1 Introduction

Quantum computers are expected to have significant applications in solving challenging problems in information processing. Inspired by the powerful capacity of classical supervised learning and its growing community [15, 114], it is natural to develop their quantum counterparts and explore the emerging field of quantum machine learning (QML) [7, 17–19]. Among many topics in this area, classification is one of the most important tasks, e.g., distinguishing quantum states [106–108] or recognizing classical data [60, 74, 76]. Classification is usually described as a decision-making process with discrete variables where the processing unit is provided with a labeled training set $\mathcal{D}^{(train)} = \{(\rho^{(m)}, y^{(m)})\}$ in order to find the convoluted mapping $\mathcal{F}$ between each set element $\rho^{(m)}$ and its corresponding label $y^{(m)}$. Once the training process is complete, we would expect the classifier $\mathcal{F}$ not only learns the map $\mathcal{F}(\rho^{(m)}) = y^{(m)}$ precisely, but also generalizes its capacity of discrimination to discover some hidden features shared with similar test data $\mathcal{F}(\rho^{(new)}) = y^{(new)}$ (i.e. recognize an unseen cat as a cat). This ability of generalization is valuable to all classification tasks, and hence it is frequently used to benchmark the performance of a classifier.

In classical machine learning, various approaches have been proposed to implement classification tasks, including perceptron-based algorithms, support vector machines, and the most prevalent neural network (NN) framework [15]. With the quantum computing community growing in the NISQ era [24], similar ideas have been developed respectively, including the quantum perceptron model [115], kernel-based method [70], and the quantum neural network (QNN) framework [60, 61, 76, 77, 98, 109, 116, 117]. This chapter focuses on QNN-based algorithms, also referred to as Variational Quantum Algorithms (VQA) or hybrid quantum-classical algorithms.

The main content of this chapter was published in [111], and the remainder is organized as follows: in Sec. 3.2, we introduce the variational shadow quantum learning framework for binary classification, which includes model sketch, loss function, analytical gradients, model complexity, theoretical classification ability and escape of barren plateau. The similar contents of variational shadow quantum learning framework for multi-label classification are discussed in Sec. 3.3. In Sec. 3.4, numerical experiments are conducted to verify the accuracy and efficiency of VSQL methods, including the classification of quantum states, MNIST classification and distinguishing noisy quantum states. Lastly in Sec. 3.5, some discussions are concluded to inspire future research.

3.2 Variational Shadow Quantum Learning for Binary Classification

3.2.1 Sketch of Method

We now present the sketch of VSQL for binary classification. Our goal is to find the optimal parameters $\boldsymbol{\theta}^*$ in the local parameterized quantum circuits $U(\boldsymbol{\theta})$, which we call *shadow circuit*, and the best weights $\{\boldsymbol{w}^*, b^*\}$ in the fully-connected neural network such that the algorithm could correctly predict the label of an unknown input quantum state. The original meaning of the word ‘shadow’ is that we focus on the information of things in a particular aspect rather than the whole one. This aspect of information usually comes from projecting things in some directions. Here in this chapter, we regard the features that

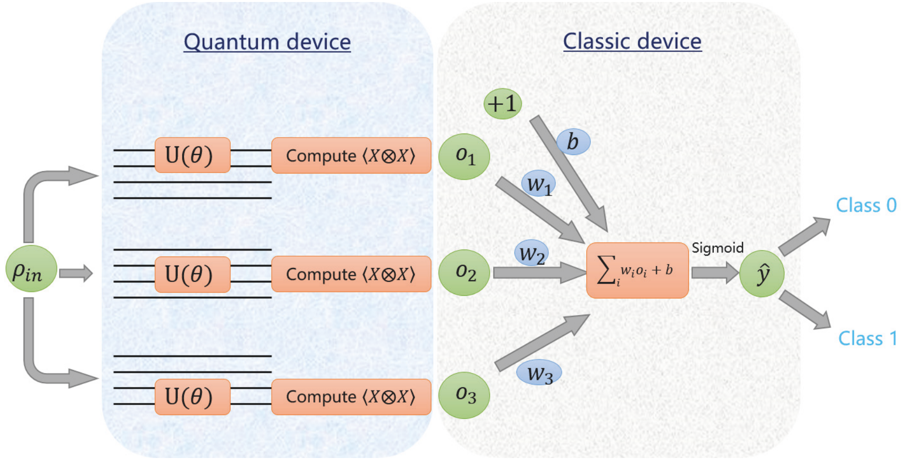


FIGURE 3.1: Sketch of variational shadow quantum learning (VSQL) for binary classification with $n = 4$ and $n_{qsc} = 2$. In the quantum device, the shadow circuit is implemented on the subspace of input state ρ_{in}. Sliding through the whole system to collect the Pauli-$(X \otimes X)$ expectations, i.e., shadow features. In the classic device, the resulting shadow features o_i's are fed into a fully-connected neural network. Here, the output $\hat{y}$ is a value between 0 and 1 for the binary case. We should denote that all the shadow circuits $U(\theta)$'s sliding through the n-qubit Hilbert space are identical.

the local PQCs extract as the projection information of the input quantum state on some qubit spaces. Therefore, we call the local PQCs the shadow circuits and the projection information the shadow feature.

Like most classifiers, VSQL consists of two separate processes, viz. training and inference. During the training process (illustrated in Fig. 3.1), we are given the training data set encoded in n-qubit quantum state $\mathcal{D}^{(train)} \equiv \{(\rho_{in}^{(m)}, y^{(m)})\}_{m=1}^{N_{train}}$, where $y^{(m)} \in \{0, 1\}$ denotes the binary label for the m-th input density matrix $\rho_{in}^{(m)}$. Then, the n_{qsc}-local shadow circuit acts on the first n_{qsc} qubits and the corresponding Pauli-$(X \otimes \cdots \otimes X)$ expectation value is estimated, recorded as *shadow feature* o_1. Here $(X \otimes \cdots \otimes X)$ has n_{qsc} Pauli-X gates. Next, the same shadow circuit is implemented on the subspace spanned from the 2^{nd} up to the $(2 + n_{qsc} - 1)^{th}$ qubit to extract the second shadow feature o_2. As the shadow circuit slides down, we obtain $n - n_{qsc} + 1$ shadow features in total. This convolution-like way of sliding through the qubit positions can be adjusted according to the hardware connectivity. Pauli-X gates employed to do measurements could be replaced by other Pauli gates, which are also easy to be measured, e.g., Pauli-Z gates. Actually, due to the universality of $U(\theta)$, Pauli-X and Pauli-Z or other Pauli gates would have the same effects. We also note that there is only one shadow circuit here. However,

Algorithm 1 Variational shadow quantum learning (VSQL) for binary classification: the training process

```
Input: The training data set $\mathcal{D}^{(train)} \equiv \{(\rho_{in}^{(m)}, y^{(m)} \in \{0,1\})\}_{m=1}^{N_{train}}$, $EPOCH$, optimization procedure
Output: The final parameters $\boldsymbol{\theta}^*$, $\boldsymbol{w}^*$ and $b^*$, and the list of losses
Initialize the parameters $\boldsymbol{\theta}$ of the 2-local (for example) shadow circuit $U(\boldsymbol{\theta})$ from uniform distribution Uni$[0, 2\pi]$ and $\boldsymbol{w}, b$ from Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbb{I})$
for $ep = 1, \ldots, EPOCH$ do
    for $m = 1, \ldots, N_{train}$ do
        Apply multi-times the shadow circuit $U(\boldsymbol{\theta})$ to the input density matrix $\rho_{in}^{(m)}$
        Measure the subsystem and estimate a series of expectations $\langle X \otimes X \rangle$, recorded as $o_i$'s
        Feed the shadow features $o_i$'s into the classical neural network and obtain the output $\hat{y}^{(m)}$
        Compute the accumulated loss $\left(\hat{y}^{(m)} - y^{(m)}\right)^2$ and update accordingly the parameters $\boldsymbol{\theta}$, $\boldsymbol{w}$ and $b$ via gradient-based optimization procedure
    end for
    if the stopping criterion is satisfied then
        Break
    end if
end for
```

the **n**umber of **s**hadow circuits (n_s) could be increased appropriately to accommodate the difficulty of classification tasks, with $n_s(n - n_{qsc} + 1)$ shadow features. Sequentially, we feed these local features $\{o_i\}$ into a classical FCNN, which means they are summed with weights $\boldsymbol{w} \in \mathbb{R}^{n-n_{qsc}+1}$, bias $b \in \mathbb{R}$ and mapped into the range $\hat{y}^{(m)} \in [0,1]$ via the sigmoid activation function $\sigma(z) = \left(1 + \mathrm{e}^{-z}\right)^{-1}$. Repeat the same procedure for each input data and compute the accumulated loss $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{w}, b; \mathcal{D}^{(train)})$ between the predicted value $\hat{y}^{(m)}$ and its true label $y^{(m)}$. Finally, VSQL utilizes a gradient-based optimizer to update the shadow circuit parameters $\boldsymbol{\theta}$ and the neural network parameters $\boldsymbol{w}, b$, thus gradually minimizing the loss function. Repeat these steps until the loss is converged with tolerance $\Delta\mathcal{L} \leq \varepsilon$ or other stopping criteria are satisfied. Check Algorithm 1 for details.

During the inference process, the unseen test data set $\mathcal{D}^{(test)} \equiv \{(\rho_{in}^{(m)}, y^{(m)} \in \{0,1\})\}_{m=1}^{N_{test}}$ is provided to the classifier $\mathcal{F}$. We feed each sample in the test set to the trained hybrid framework (combination of shadow circuit and FCNN) to predict its label. Then, the test accuracy could be calculated by comparing the predicted labels and the true labels. The details are provided in Algorithm 2. Furthermore, VSQL can be naturally generalized

Algorithm 2 Variational shadow quantum learning (VSQL) for binary classification: the inference process

```
Input: The test data set $\mathcal{D}^{(test)} \equiv \{(\rho_{in}^{(m)}, y^{(m)} \in \{0,1\})\}_{m=1}^{N_{test}}$, the parameters $\boldsymbol{\theta}$, $\boldsymbol{w}$ and $b$ from the training process
Output: The list of predicted labels and the test accuracy
Set the counter $n\_c = 0$, denoting the number of correctly predicted labels
for $m = 1, \dots, N_{test}$ do
    Apply multi-times the shadow circuit $U(\boldsymbol{\theta})$ to the input density matrix $\rho_{in}^{(m)}$
    Measure and estimate a series of expectations $\langle X \otimes X \rangle$, recorded as $o_i$'s
    Feed these shadow features $o_i$'s into the classical neural network and obtain the output $\hat{y}^{(m)} \in [0,1]$
    if $\hat{y}^{(m)} \leq 0.5$ then
        Set the predicted label as '0'
    else
        Set the predicted label as '1'
    end if
    if the predicted label $==$ $y^{(m)}$ then
        $n\_c = n\_c + 1$
    end if
end for
Compute the test accuracy as $n\_c / N_{test}$
```

to multi-label classification by replacing the sigmoid activation function with a softmax function.

3.2.2 Loss Function

Given the data set $\mathcal{D} \equiv \{(\rho_{in}^{(m)}, y^{(m)})\}_{m=1}^{N}$ and n_{qsc}-local shadow circuits, the loss function of VSQL for binary classification is designed to be the mean square error[1] [118]:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{w}, b; \mathcal{D}) \equiv \frac{1}{2N} \sum_{m=1}^{N} \left[\hat{y}^{(m)} \left(\rho_{in}^{(m)}; \boldsymbol{\theta}, \boldsymbol{w}, b \right) - y^{(m)} \right]^2 . \tag{3.1}$$

Here, the predicted label $\hat{y}^{(m)}$ is defined as follows:

$$\hat{y}^{(m)} \left(\rho_{in}^{(m)}; \boldsymbol{\theta}, \boldsymbol{w}, b \right) \equiv \sigma \left(\sum_i w_i o_i^{(m)} \left(\rho_{in}^{(m)}; \boldsymbol{\theta} \right) + b \right), \tag{3.2}$$

[1] The cross-entropy loss is considered in the multi-label case.

where $\sigma(z)$ denotes the sigmoid activation function and the shadow features o_i are calculated through

$$o_i^{(m)}\left(\rho_{in}^{(m)};\boldsymbol{\theta}\right)=\operatorname{Tr}\left(\rho_{in}^{(m)}(\mathbb{I}\otimes\cdots\otimes U^{\dagger}(\boldsymbol{\theta})OU(\boldsymbol{\theta})\otimes\cdots\otimes\mathbb{I})\right). \tag{3.3}$$

Note that the shadow circuit $U(\boldsymbol{\theta})$ and the physical observable $O = X\otimes\cdots\otimes X$ are applied on the same local qubits. Additionally, $U(\boldsymbol{\theta})$ is usually decomposed as a chain of unitary operators:

$$U(\boldsymbol{\theta}) = \prod_{l=L}^{1} U_l(\theta_l)V_l, \tag{3.4}$$

where $U_l(\theta_l) = \exp(-i\theta_l P_l/2)$ and V_l denotes a fixed operator such as Identity, CNOT and so on.

3.2.3 Analytical Gradients

With the above preparation, we can easily derive the analytical gradients, with which VSQL could naturally update its parameters $\boldsymbol{\theta}$ and $\{\boldsymbol{w}, b\}$ via gradient-based optimization method, e.g., SGD [119]. For each input $\rho_{in}^{(m)}$,

$$\frac{\partial\mathcal{L}}{\partial w_i} = \left(\hat{y}^{(m)} - y^{(m)}\right)\cdot\hat{y}^{(m)}\left(1-\hat{y}^{(m)}\right)\cdot o_i^{(m)}, \tag{3.5}$$

$$\frac{\partial\mathcal{L}}{\partial b} = \left(\hat{y}^{(m)} - y^{(m)}\right)\cdot\hat{y}^{(m)}\left(1-\hat{y}^{(m)}\right), \tag{3.6}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\theta_l} &= \frac{\partial\mathcal{L}}{\partial\hat{y}^{(m)}}\cdot\sum_i\frac{\partial\hat{y}^{(m)}}{\partial o_i^{(m)}}\cdot\frac{\partial o_i^{(m)}}{\partial\theta_l}\\ &= \left(\hat{y}^{(m)} - y^{(m)}\right)\cdot\sum_i\hat{y}^{(m)}\left(1-\hat{y}^{(m)}\right)w_i\cdot\frac{\partial o_i^{(m)}}{\partial\theta_l},\end{aligned} \tag{3.7}$$

The partial derivatives w.r.t w_i and b are written in Eqs. (3.5) and (3.6) could be directly computed in the classical device and used to update w_i, b through the backpropagation algorithm [114]. And the partial derivative w.r.t θ_l in Eq. (3.7) can be regarded as a weighted sum of several partial gradients $\partial o_i^{(m)}/\partial\theta_l$,

$$\frac{\partial o_i^{(m)}\left(\boldsymbol{\theta};\rho_{in}^{(m)}\right)}{\partial\theta_l} = -\frac{i}{2}\operatorname{Tr}\left(U_{>l}^\dagger O U_{>l}\left[P_l, U_{\leq l}\rho_i U_{\leq l}^\dagger\right]\right). \tag{3.8}$$

where $\rho_i = \operatorname{Tr}_{-i}(\rho_{in}^{(m)})$ denotes the partial trace of $\rho_{in}^{(m)}$ corresponding to the index i, $U_{\leq l} = \prod_{j=l}^{1} U_j(\theta_j)V_j$ and $U_{>l} = \prod_{j=L}^{l+1} U_j(\theta_j)V_j$ and $[\rho,\sigma] = \rho\sigma - \sigma\rho$ denotes the commutator between ρ and σ. This gradient can be calculated exactly on the quantum device with the $\pi/2$ parameter shift rule proposed by Mitarai et al.. Compared with the finite difference scheme, this method leads to a faster convergence [120] and is more suitable to the existing quantum devices.

3.2.4 Number of Parameters of VSQL

In the hybrid quantum-classical framework, the number of parameters in the quantum circuit is an important quantity to measure its complexity and efficiency. The main reason is that updating each parameter is costly in terms of quantum resources as it requires rerunning the entire circuit multiple times. Therefore, algorithms with a smaller number of parameters are preferable in the NISQ era. Here, we exhibit this advantage for VSQL.

There are two kinds of parameters in VSQL, i.e., the parameters $\boldsymbol{\theta}$ in the shadow circuits and the parameters $\boldsymbol{w}, b$ in the classical NN. Assume the action mode of the shadow circuits is "shadow sliding" (illustrated in Fig. 3.1), which is also employed throughout this chapter. The number of parameters of VSQL for binary classification is summarized as follows.

Proposition 3.1. *For an n-qubit quantum system, if we use n_s shadow circuits, then the number of parameters of VSQL for binary classification is*

$$\begin{aligned}\#\ \mathit{Params} &= \#\ \mathit{Params}\Big|_{\mathit{in\ shadow\ circuits}} + \#\ \mathit{Params}\Big|_{\mathit{in\ NN}} \\ &= n_s n_{qsc} D + \left[n_s\left(n - n_{qsc} + 1\right) + 1\right],\end{aligned} \tag{3.9}$$

where we denote by n_{qsc} the ***n****umber of* ***q****ubits of the* ***s****hadow* ***c****ircuits and assume each shadow circuit consists of D layers with n_{qsc} parameters in each layer.*

Thus, VSQL has a parameter quantity that is linearly related to n and D separately, rather than nD that commonly appears in most of the ansatzes employed in the existing literature [60, 77, 98]. For a 50-qubit quantum system, if we use just one 2-local shadow circuit with 20 layers, i.e., $n_s = 1, n_{qsc} = 2, D = 20$, then the number of parameters of VSQL is $40 + (50 - 2 + 1) + 1 = 90$, which is much smaller than $nD = 1000$.

3.2.5 Number of repetitions for computing each shadow feature

As we need to repeat the shadow circuits multiple times to estimate the shadow features, here we give the number of repetitions required in VSQL.

Proposition 3.2. *Given a precision ϵ, the number of repetitions of the shadow circuit for computing each shadow feature at error ϵ, with probability at least $1 - \eta$, scales as $O\left(\log(1/\eta)/\epsilon^2\right)$.*

This proposition is directly derived from the Chernoff–Hoeffding theorem [121]. Furthermore, by utilizing these estimated shadow features, VSQL outputs the prediction value $\hat{y}$ and gives a label according to the following prediction rule

$$\text{predicted label} = \begin{cases} 0, & \hat{y} < 0.5 \\ 1, & \hat{y} \geq 0.5. \end{cases} \tag{3.10}$$

Therefore, in the inference process of VSQL, for an input state with the label $y \in \{0, 1\}$, if the predicted label is correct and the gap between the prediction value and 0.5 is τ under an infinite number of repetitions of the shadow circuits, then the actual number of repetitions, required to ensure that the input state is not misclassified, will be related to the gap τ.

Proposition 3.3. *For an n-qubit quantum system, if we use n_s shadow circuits and assume the final weights w_i of the neural networks in VSQL are bounded as $|w_i| \leq C_w$, and the prediction gap is $\tau \in (0, 0.5)$, then the actual number of repetitions for computing each shadow feature, with probability at least $1 - \eta$, scales as $O\left(n_s^2 n^2 C_w^2 \log(1/\eta)/\tau^2\right)$.*

Proof. If the estimated error of each shadow feature o_i is δ, then from Proposition 3.2, the number of repetitions, with probability at least $1-\eta$, is $O\left(\log(1/\eta)/\delta^2\right)$. What's more, due to

$$\frac{\partial \hat{y}}{\partial o_i} \equiv \frac{\partial \sigma\left(\sum_i w_i o_i + b\right)}{\partial o_i} = \hat{y}\,(1-\hat{y}) \cdot w_i \leq \frac{|w_i|}{4} \leq \frac{C_w}{4}, \tag{3.11}$$

the error of $\hat{y}$ could be bounded as $\frac{1}{4}n_s n C_w \delta$, where the first inequality follows from $0 < \hat{y} < 1$ and the term $n_s n$ means there are at most $n_s n$ shadow features. If we let $\frac{1}{4}n_s n C_w \delta \leq \tau$, the number of repetitions for computing each shadow feature is obtained. □

From Proposition 3.3, we know, in the inference process of VSQL, if there is a large prediction gap, then the output of VSQL will be allowed to have significant errors. This means VSQL will require much fewer repetitions for computing each shadow feature yet still ensure obtaining a correct predicted label.

3.2.6 Theoretical Classification Ability

In this subsection, we explore the theoretical classification ability of VSQL and give the corresponding necessary and sufficient conditions.

Theorem 3.4. *Given two types of input density matrices $\rho_{in}^{(0)}$ and $\rho_{in}^{(1)}$ with labels 0 and 1, respectively, VSQL can distinguish them if, and only if, there exists a group of $\boldsymbol{\theta}$ that makes at least one pair of shadow features $o_i^{(0)}$ and $o_i^{(1)}$ different, i.e., $|o_i^{(0)} - o_i^{(1)}| > 0$.*

Proof. Sufficiency: Without loss of generality, we assume $i = 1$ and $o_1^{(0)} < o_1^{(1)}$. By simply setting $w_1 = 1$ and other w_i's as 0, and setting $b = -(o_1^{(0)} + o_1^{(1)})/2$, we could obtain

$$\hat{y}^{(0)} \equiv \sigma(\sum_i w_i o_i^{(0)} + b) = \sigma[(o_1^{(0)} - o_1^{(1)})/2] < \sigma\,(0) = 0.5;$$
$$\hat{y}^{(1)} \equiv \sigma(\sum_i w_i o_i^{(1)} + b) = \sigma[(o_1^{(1)} - o_1^{(0)})/2] > \sigma\,(0) = 0.5.$$

By taking 0.5 as the decision boundary, we know VSQL could distinguish these two types of input density matrices theoretically.

Necessity: Assuming all pairs of shadow features $o_i^{(0)}$ and $o_i^{(1)}$ are identical for any group of $\boldsymbol{\theta}$, then $\hat{y}^{(0)}$ and $\hat{y}^{(1)}$ will always be same. Hence, VSQL fails to theoretically distinguish these two input density matrices, which is in contradiction with the condition. □

Corollary 3.5. *Given two types of n-qubit input density matrices. If each pair of their corresponding m-local partial traces are identical ($m < n$), then VSQL is theoretically incapable of distinguishing them via m-local shadow circuits, and vice versa.*

The proof of Corollary 3.5 could be immediately derived from Theorem 3.4, because getting identical partial traces is equivalent to having same shadow features (cf. Eq. (3.3)).

After exploring the necessary and sufficient conditions for the theoretical classification ability of VSQL, we now discuss this ability under different local shadow circuits. Intuitively, larger shadow circuits will give VSQL stronger classification ability. The following Theorem will give a detailed statement.

Theorem 3.6. *Given two types of n-qubit input density matrices $\rho_{in}^{(0)}$ and $\rho_{in}^{(1)}$. If VSQL can not theoretically distinguish them via m-local shadow circuits, then neither can via m'-local shadow circuits, where $m' < m < n$.*

Proof. From Corollary 3.5, we know every pair of the corresponding m-local partial traces of these two states are identical, i.e.,

$$\left(\rho_{in}^{(0)}\right)_{m\text{-local}} = \left(\rho_{in}^{(1)}\right)_{m\text{-local}}, \tag{3.12}$$

where $(\rho)_{m\text{-local}} \equiv \mathrm{Tr}_{n/m\text{-local}}\,(\rho)$ denotes the partial trace of ρ on all n other than m-local qubit system and the subscripts "m-local" on both sides mean they are in the same m local qubit system. If we similarly define the following

$$\left(\rho_{in}^{(0)}\right)_{m'\text{-local}} \equiv \mathrm{Tr}_{m/m'\text{-local}}\left(\left(\rho_{in}^{(0)}\right)_{m\text{-local}}\right) \tag{3.13}$$

$$\left(\rho_{in}^{(1)}\right)_{m'\text{-local}} \equiv \mathrm{Tr}_{m/m'\text{-local}}\left(\left(\rho_{in}^{(1)}\right)_{m\text{-local}}\right), \tag{3.14}$$

then we have

$$\left(\rho_{in}^{(0)}\right)_{m'\text{-local}} = \left(\rho_{in}^{(1)}\right)_{m'\text{-local}}. \tag{3.15}$$

Due to the arbitrariness of m-local, we obtain m'-local can also be arbitrary, which means each pair of the corresponding m'-local partial traces of these two states are identical. From Corollary 3.5 again, we could finish the proof. □

The following example shows that the other direction does not hold. Assume

$$\rho_{in}^{(0)} = \frac{1}{\sqrt{2}}\left(|000\rangle + |110\rangle\right) \cdot \frac{1}{\sqrt{2}}\left(\langle 000| + \langle 110|\right) \tag{3.16}$$

$$\rho_{in}^{(1)} = \frac{1}{\sqrt{2}}\left(|000\rangle - |110\rangle\right) \cdot \frac{1}{\sqrt{2}}\left(\langle 000| - \langle 110|\right) \tag{3.17}$$

and let $m' = 1$ and $m = 2$. In the following, we use $(\rho_{in})_i$ and $(\rho_{in})_{i,j}$ to denote the 1-local and 2-local partial traces, respectively, where $i, j = 1, 2, 3, i < j$. Now we verify their 1-local and 2-local partial traces:

$$\left(\rho_{in}^{(0)}\right)_1 = \left(\rho_{in}^{(0)}\right)_2 = \frac{1}{2}\left(|0\rangle\langle 0| + |1\rangle\langle 1|\right), \quad \left(\rho_{in}^{(0)}\right)_3 = |0\rangle\langle 0| \tag{3.18}$$

$$\left(\rho_{in}^{(1)}\right)_1 = \left(\rho_{in}^{(1)}\right)_2 = \frac{1}{2}\left(|0\rangle\langle 0| + |1\rangle\langle 1|\right), \quad \left(\rho_{in}^{(1)}\right)_3 = |0\rangle\langle 0|; \tag{3.19}$$

$$\left(\rho_{in}^{(0)}\right)_{1,2} = \frac{1}{2}\left(|00\rangle + |11\rangle\right) \cdot \left(\langle 00| + \langle 11|\right) \tag{3.20}$$

$$\left(\rho_{in}^{(1)}\right)_{1,2} = \frac{1}{2}\left(|00\rangle - |11\rangle\right) \cdot \left(\langle 00| - \langle 11|\right). \tag{3.21}$$

We see for these two states there exists different 2-local partial traces, even though each pair of their corresponding 1-local partial traces are identical. This indicates, from Corollary 3.5, VSQL could theoretically distinguish them via 2-local shadow circuits, but could not via 1-local ones. Hence, the two states in Eqs. (3.16) and (3.17) could be a successful counterexample.

From Theorem 3.6, we confirm the intuition that the larger the number of qubits n_{qsc} of the shadow circuits is, the stronger the theoretical expressive ability of VSQL is. However, if this number is too large, it will lead to other problems, such as the Barren Plateau issue described in the next subsection. Therefore, in this chapter, we set it as a hyper-parameter whose value is chosen according to engineering experience.

3.2.7 Escape of Barren Plateau

In the last subsection, we have shown that VSQL has a strong theoretical classification ability for a wide range of quantum states, especially by using large local shadow circuits. However, the sizeable operating scope of the shadow circuits will increase network parameters and the cost of compiling given limited hardware connections and leads to the Barren Plateau issue. The barren plateau issue [44, 122] refers to the vanishing gradient problem during the training process of QNN. That is, for a wide range of variational quantum circuits, the partial gradients of the objective function have a zero mean and an exponentially vanishing variance, which makes it difficult for the optimizer to find the correct direction to decrease the objective function. Therefore, it is important to discuss whether the BP problem exists when proposing a new variational quantum algorithm.

Next, we evaluate the mean and variance of the analytical gradients in VSQL. There is no barren plateau issue for the partial gradients (see Eqs. (3.5) and (3.6)) with respect to the parameters w_i and b of the classical NN. And for the partial gradient (see Eq. (3.7)) with respect to θ_l of QNN, the barren plateau issue is mainly reflected on the last term, i.e., the partial derivatives (see Eq. (3.8)) of the shadow features o_i with respect to θ_l. Hence, it is sufficient to evaluate the mean and the variance of the partial gradient in Eq. (3.8) to explore the barren plateau problem in VSQL. The results are summarized in the following proposition.

Definition 3.7. A unitary t-design [123] is defined as a finite set of unitaries $\{U_k\}_{k=1}^K$ on a d-dimensional Hilbert space such that

$$\frac{1}{K} \cdot \sum_k P_{(t,t)}(U_k) = \int_{\mathcal{U}(d)} d\mu_{Haar}(U) P_{(t,t)}(U), \tag{3.22}$$

where $P_{(t,t)}(U)$ denotes a polynomial of degree at most t on the elements of U and at most t on the elements of $U^\dagger$.

Proposition 3.8. *If $U_{>l}$ or $U_{\leq l}$ forms at least an n_{qsc}-local unitary 2-design, the mean and the variance of the analytical gradients with respect to θ_l in VSQL (see Eq.* (3.8)*) are evaluated as*

$$\mathbb{E}\left[\frac{\partial o_i}{\partial \theta_l}\right] = 0; \quad Var\left[\frac{\partial o_i}{\partial \theta_l}\right] = -\frac{1}{4} \cdot \frac{C\left(\rho_i\right)}{2^{2n_{qsc}} - 1}, \tag{3.23}$$

where $C\left(\rho_i\right) \in \left(-4 \times 2^{n_{qsc}}, 0\right)$ denotes a constant and n_{qsc} is the number of qubits of the shadow circuits.

Proof. Before start, we need the following two lemmas [122–124]:

Lemma 3.9. *Let $\{U_k\}_{k=1}^K \in \mathcal{U}(d)$ form a unitary t-design [123] with $t \geq 1$, and let A, B be arbitrary linear operators. Then*

$$\frac{1}{K} \cdot \sum_k \operatorname{Tr}\left(U_k A U_k^\dagger B\right) = \int_{\mathcal{U}(d)} d\mu_{Haar}(U) \cdot \operatorname{Tr}\left(U A U^\dagger B\right) = \frac{\operatorname{Tr}(A)\operatorname{Tr}(B)}{d}. \tag{3.24}$$

Lemma 3.10. *Let $\{U_k\}_{k=1}^K \in \mathcal{U}(d)$ form a unitary t-design [123] with $t \geq 2$, and let A, B, C, D be arbitrary linear operators. Then*

$$\begin{aligned}\frac{1}{K} \cdot \sum_k \operatorname{Tr}\left(U_k A U_k^\dagger B U_k C U_k^\dagger D\right) &= \int_{\mathcal{U}(d)} d\mu_{Haar}(U) \cdot \operatorname{Tr}\left(U A U^\dagger B U C U^\dagger D\right) \\ &= \frac{\operatorname{Tr}(A)\operatorname{Tr}(C)\operatorname{Tr}(BD) + \operatorname{Tr}(AC)\operatorname{Tr}(B)\operatorname{Tr}(D)}{d^2 - 1} \\ &\quad - \frac{\operatorname{Tr}(AC)\operatorname{Tr}(BD) + \operatorname{Tr}(A)\operatorname{Tr}(B)\operatorname{Tr}(C)\operatorname{Tr}(D)}{d(d^2 - 1)};\end{aligned} \tag{3.25}$$

$$\begin{aligned}\frac{1}{K} \cdot \sum_k \operatorname{Tr}\left(U_k A U_k^\dagger B\right)\operatorname{Tr}\left(U_k C U_k^\dagger D\right) &= \int_{\mathcal{U}(d)} d\mu_{Haar}(U) \cdot \operatorname{Tr}\left(U A U^\dagger B\right)\operatorname{Tr}\left(U C U^\dagger D\right) \\ &= \frac{\operatorname{Tr}(AC)\operatorname{Tr}(BD) + \operatorname{Tr}(A)\operatorname{Tr}(B)\operatorname{Tr}(C)\operatorname{Tr}(D)}{d^2 - 1} \\ &\quad - \frac{\operatorname{Tr}(A)\operatorname{Tr}(C)\operatorname{Tr}(BD) + \operatorname{Tr}(AC)\operatorname{Tr}(B)\operatorname{Tr}(D)}{d(d^2 - 1)}.\end{aligned} \tag{3.26}$$

According to Eq. (3.8), i.e.,

$$\begin{aligned}\frac{\partial o_i^{(m)}}{\partial \theta_l} &= -\frac{i}{2}\operatorname{Tr}\left(U_{>l}^\dagger O U_{>l}\left[P_l, U_{\leq l}\rho_i U_{\leq l}^\dagger\right]\right) &(3.27)\\ &= \frac{i}{2}\operatorname{Tr}\left(U_{\leq l}\rho_i U_{\leq l}^\dagger\left[P_l, U_{>l}^\dagger O U_{>l}\right]\right), &(3.28)\end{aligned}$$

(i) if $U_{>l}$ forms at least a n_{qsc}-local unitary 2-design, from Eqs. (3.27), (3.24) and (3.26), we have

$$\begin{aligned}\mathbb{E}\left[\frac{\partial o_i^{(m)}}{\partial \theta_l}\right] &= -\frac{i}{2}\cdot\frac{\operatorname{Tr}(O)\,\mathbb{E}\left[\operatorname{Tr}\left([P_l, U_{\leq l}\rho_i U_{\leq l}^\dagger]\right)\right]}{2^{n_{qsc}}} = 0, &(3.29)\\ \operatorname{Var}\left[\frac{\partial o_i^{(m)}}{\partial \theta_l}\right] &= -\frac{1}{4}\cdot\frac{\operatorname{Tr}\left(O^2\right)\mathbb{E}\left[\operatorname{Tr}\left([P_l, U_{\leq l}\rho_i U_{\leq l}^\dagger]^2\right)\right]}{2^{2n_{qsc}}-1}; &(3.30)\end{aligned}$$

(ii) if $U_{\leq l}$ forms at least a n_{qsc}-local unitary 2-design, from Eqs. (3.28), (3.24) and (3.26), we have

$$\mathbb{E}\left[\frac{\partial o_i^{(m)}}{\partial \theta_l}\right] = \frac{i}{2}\cdot\frac{\operatorname{Tr}(\rho_i)\,\mathbb{E}\left[\operatorname{Tr}\left([P_l, U_{>l}^\dagger O U_{>l}]\right)\right]}{2^{n_{qsc}}} = 0, \tag{3.31}$$

$$\operatorname{Var}\left[\frac{\partial o_i^{(m)}}{\partial \theta_l}\right] = -\frac{1}{4}\cdot\left(\frac{\operatorname{Tr}\left(\rho_i^2\right)\mathbb{E}\left[\operatorname{Tr}\left([P_l, U_{>l}^\dagger O U_{>l}]^2\right)\right]}{2^{2n_{qsc}}-1} - \frac{\operatorname{Tr}^2(\rho_i)\,\mathbb{E}\left[\operatorname{Tr}\left([P_l, U_{>l}^\dagger O U_{>l}]^2\right)\right]}{2^{n_{qsc}}(2^{2n_{qsc}}-1)}\right). \tag{3.32}$$

Now let's consider the term $\mathbb{E}\left[\operatorname{Tr}\left([P, U^\dagger A U]^2\right)\right]$, where P is a Pauli product operator, U denote a series of unitary matrices that the expectation acts on and $A = \sum_j \lambda_j \left|\lambda_j\right\rangle\!\left\langle\lambda_j\right|$

denotes a Hermitian operator, where we ssume $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{2^{n_{qsc}}}$. Then we have

$$\begin{aligned}
\mathbb{E}\left[\operatorname{Tr}\left([P, U^\dagger A U]^2\right)\right] &= \mathbb{E}\left[\operatorname{Tr}(PU^\dagger AU - U^\dagger AUP)^2\right] && (3.33)\\
&= 2\mathbb{E}\left[\operatorname{Tr}(PU^\dagger AU)^2\right] - 2\mathbb{E}\left[\operatorname{Tr}(PU^\dagger AUU^\dagger AUP)\right] && (3.34)\\
&= 2\mathbb{E}\left[\sum_{i,j} \lambda_i\lambda_j \operatorname{Tr}(\underbrace{\langle\lambda_j| UPU^\dagger |\lambda_i\rangle\langle\lambda_i| UPU^\dagger |\lambda_j\rangle}_{p_{ij}})\right] - 2\operatorname{Tr}(A^2) && (3.35)\\
&= 2\mathbb{E}\left[(\vec{\lambda})^\dagger P_\Lambda \vec{\lambda}\right] - 2\operatorname{Tr}(A^2). && (3.36)
\end{aligned}$$

Here, $\vec{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_{2^{n_{qsc}}}]^\top$ and we define a matrix $P_\Lambda = [p_{ij}]$, where each element is defined as

$$p_{ij} = \langle\lambda_i| UPU^\dagger |\lambda_j\rangle\langle\lambda_j| UPU^\dagger |\lambda_i\rangle. \tag{3.37}$$

From the fact that $p_{ij} \geq 0$ and $\sum_i p_{ij} = \sum_j p_{ij} = 1$, we know P_Λ is a doubly stochastic matrix. Now in order to bound the term $(\vec{\lambda})^\dagger P_\Lambda \vec{\lambda}$, we can repeatedly perform the following procedure followed from the *Rearrangement inequality*, i.e., for any $i \leq k$ and $j \leq l$, we have

$$\begin{array}{ccccccc}
\lambda_i\lambda_j & + & \lambda_k\lambda_l & \geq & \lambda_i\lambda_l & + & \lambda_k\lambda_j. \\
p_{ij} & & p_{kl} & & p_{il} & & p_{kj} \\
\uparrow^{\Delta_1}\ (or \downarrow_{\Delta_2}) & & \uparrow^{\Delta_1}\ (or \downarrow_{\Delta_2}) & & \downarrow_{\Delta_1}\ (or \uparrow^{\Delta_2}) & & \downarrow_{\Delta_1}\ (or \uparrow^{\Delta_2})
\end{array} \tag{3.38}$$

That is, for the four elements in the four corners of any rectangle (e.g., indexed by rows i, k and columns j, l) in P_Λ, we could increase p_{ij}, p_{kl} and decrease p_{il}, p_{kj} by Δ_1 simultaneously to get close to its upper bound; Or conversely by Δ_2 to get close to its lower bound (see also Eq. (3.38)). Here, we can set $\Delta_1 = \min\{p_{il}, p_{kj}\}$ and $\Delta_2 = \min\{p_{ij}, p_{kl}\}$ to satisfy the nonnegativity. An intuitive example for one step of this procedure is referred to below:

$$
\begin{aligned}
(\vec{\lambda})^\dagger \left[\begin{array}{cccccc}
 & & j & & l & \\
 & & \vdots & & \vdots & \\
i & \cdots & 0.3 \xrightarrow{-0.2} 0.1 & \cdots & 0.4 \xrightarrow{+0.2} 0.6 & \cdots \\
 & & \vdots & & \vdots & \\
k & \cdots & 0.5 \xrightarrow{+0.2} 0.7 & \cdots & 0.2 \xrightarrow{-0.2} 0.0 & \cdots \\
 & & \vdots & & \vdots &
\end{array}\right] \vec{\lambda} &\leq (\vec{\lambda})^\dagger \overbrace{\left[\begin{array}{cccccc}
 & & j & & l & \\
 & & \vdots & & \vdots & \\
i & \cdots & 0.3 & \cdots & 0.4 & \cdots \\
 & & \vdots & & \vdots & \\
k & \cdots & 0.5 & \cdots & 0.2 & \cdots \\
 & & \vdots & & \vdots &
\end{array}\right]}^{P_\Lambda} \vec{\lambda} \\
&\leq (\vec{\lambda})^\dagger \left[\begin{array}{cccccc}
 & & j & & l & \\
 & & \vdots & & \vdots & \\
i & \cdots & 0.3 \xrightarrow{+0.4} 0.7 & \cdots & 0.4 \xrightarrow{-0.4} 0.0 & \cdots \\
 & & \vdots & & \vdots & \\
k & \cdots & 0.5 \xrightarrow{-0.4} 0.1 & \cdots & 0.2 \xrightarrow{+0.4} 0.6 & \cdots \\
 & & \vdots & & \vdots &
\end{array}\right] \vec{\lambda}.
\end{aligned}
\tag{3.39}
$$

After a finite number of steps, we will finally obtain

$$
\sum_{i=1}^{2^{n_{qsc}}} \lambda_i \lambda_{2^{n_{qsc}}-i+1} = (\vec{\lambda})^\dagger \begin{bmatrix} & & & 1 \\ & & 1 & \\ & \vdots & & \\ 1 & & & \end{bmatrix} \vec{\lambda} \leq (\vec{\lambda})^\dagger P_\Lambda \vec{\lambda} \leq (\vec{\lambda})^\dagger \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix} \vec{\lambda} = \sum_{i=1}^{2^{n_{qsc}}} \lambda_i^2.
\tag{3.40}
$$

Substituting Eq. (3.40) into Eq. (3.36), we have

$$
2 \sum_{i=1}^{2^{n_{qsc}}} \lambda_i \lambda_{2^{n_{qsc}}-i+1} - 2\,\mathrm{Tr}(A^2) \leq \mathbb{E}\left[\mathrm{Tr}\left([P, U^\dagger A U]^2\right)\right] \leq 2 \sum_{i=1}^{2^{n_{qsc}}} \lambda_i^2 - 2\,\mathrm{Tr}(A^2) = 0. \tag{3.41}
$$

Now we prove the variance of the gradients.

(i) Substituting Eq. (3.41) into Eq. (3.30) with $A = \rho_i$, we define

$$C(\rho_i) \equiv \mathrm{Tr}\left(O^2\right) \mathbb{E}\left[\mathrm{Tr}\left([P_l, U_{\le l}\rho_i U_{\le l}^\dagger]^2\right)\right].$$

We have

$$-4 \times 2^{n_{qsc}} < 2^{n_{qsc}}\left(0 - 2\,\mathrm{Tr}(\rho_i^2)\right) \le C(\rho_i) \le 0. \tag{3.42}$$

(ii) Substituting Eq. (3.41) into Eq. (3.32) with $A = O = X \otimes \cdots \otimes X$, and we define

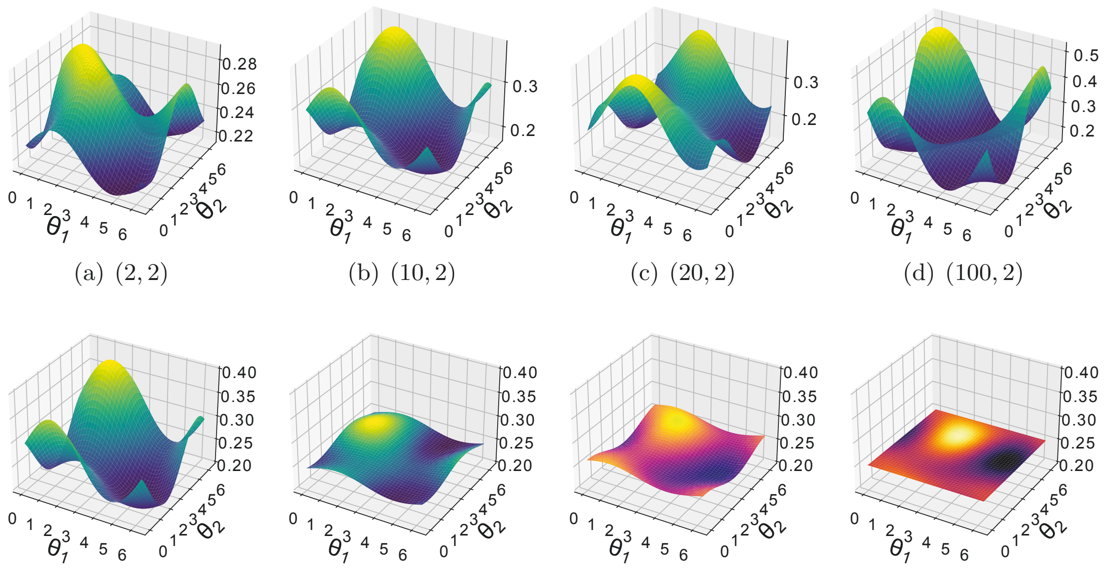
$$C(\rho_i) \equiv \mathrm{Tr}\left(\rho_i^2\right) \mathbb{E}\left[\mathrm{Tr}\left([P_l, U_{>l}^\dagger O U_{>l}]^2\right)\right] - \frac{\mathrm{Tr}^2\left(\rho_i\right) \mathbb{E}\left[\mathrm{Tr}\left([P_l, U_{>l}^\dagger O U_{>l}]^2\right)\right]}{2^{n_{qsc}}}.$$

Because O has half of the 1 eigenvalues and half of the -1 eigenvalues, we have

$$-4 \times 2^{n_{qsc}} < \left(\mathrm{Tr}\left(\rho_i^2\right) - \frac{\mathrm{Tr}^2\left(\rho_i\right)}{2^{n_{qsc}}}\right)\left(2\sum_{i=1}^{2^{n_{qsc}}}(-1) - 2\,\mathrm{Tr}(O^2)\right) \le C(\rho_i) \le 0. \tag{3.43}$$

Another point that needs to note is that for most of θ_l's, both $U_{>l}$ and $U_{\le l}$ approximate n_{qsc}-local unitary 2-design. Hence, although we give the upper bound 0, most of $C(\rho_i)$ will concentrate to $2\left(1 - 2^{n_{qsc}}\,\mathrm{Tr}(\rho_i^2)\right)$, which is far from 0 if ρ_i is close to a pure state. This completes the proof. □

From Proposition 3.8, we notice that the variance of the gradients decays exponentially with n_{qsc}, rather than the qubit number n. Hence, no matter how big the problem size n is, as long as we choose a small n_{qsc} (e.g., $n_{qsc} \le 4$) and assume $C\left(\rho_i\right) \approx -2 \times 2^{n_{qsc}}$, we can evaluate the analytical gradients efficiently via more than 1,000 repetitions derived from the Chernoff bound. In one word, VSQL could escape the barren plateaus by choosing an appropriate operating scope n_{qsc}. Moreover, [125] indicates that noise could also induce the barren plateau issue. Following this line of reasoning, the small shadow circuits in VSQL will also be beneficial for escaping the barren plateaus from a different perspective, as less noise is introduced.



(a) $(2, 2)$ (b) $(10, 2)$ (c) $(20, 2)$ (d) $(100, 2)$

(e) $(10, 2)$ (f) $(10, 4)$ (g) $(10, 6)$ (h) $(10, 10)$

FIGURE 3.2: The slice of loss landscape with respect to the first two circuit parameters by changing the system size n and operating scope n_{qsc}. Here, the binary list represents (n, n_{qsc}).

Here, we provide an illustrated example. Assume the n-qubit quantum state $\rho_{in} = |\psi_{in}\rangle\langle\psi_{in}|$ we want to classify is labeled with 0, where

$$|\psi_{in}\rangle \equiv \otimes_{j=0}^{n-1} R_y(2\pi j/n) |0\rangle . \tag{3.44}$$

The chosen shadow circuit consists of a layer of single-qubit R_y rotations and a layer of CNOT gates which only connects the adjacent qubits, followed by another layer of R_y rotations. Then, we compute the loss landscape in Eq. (3.1) with regard to the first two circuit parameters by fixing all the other parameters with $\pi/4$ and setting the bias $b = 0$ and $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$ sampled from a Gaussian distribution. The result, as shown in Fig. 3.2, is in line with the above analysis, i.e., there is no barren plateaus with $n_{qsc} = 2$, but the loss landscape shrinks dramatically with an increasing n_{qsc}.

3.3 Variational Shadow Quantum Learning for Multi-Label Classification

In this section, we simply describe the VSQL for multi-label classification, which consists of overall sketch, loss function, analytical gradients, number of parameters, number of repetitions and theoretical classification ability. Most of the settings are the same as the binary case, except for the final activation function, where the sigmoid activation function for the binary case is replaced by the softmax activation function for the multi-label case.

3.3.1 Sketch of VSQL for multi-label classification

The sketch of VSQL for multi-label classification is illustrated in Fig. 3.3, and the corresponding training and inference processes are described in Algorithms 3 and 4, respectively.

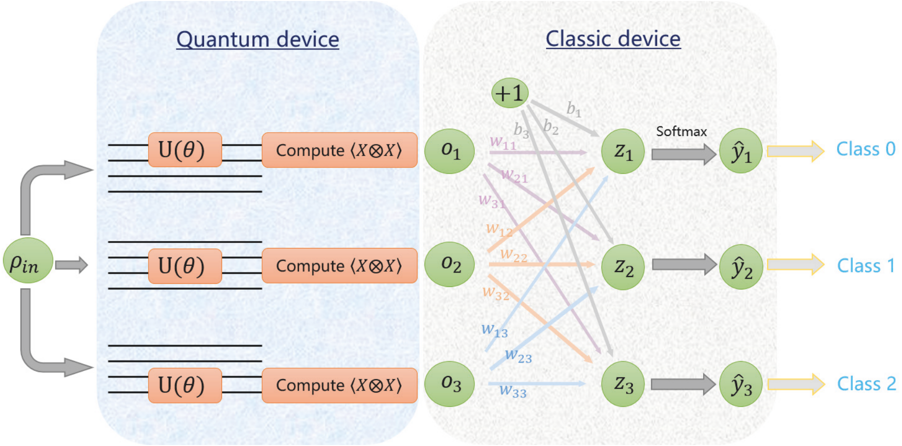


FIGURE 3.3: Sketch of variational shadow quantum learning (VSQL) for multi-label classification with $n = 4$, $n_{qsc} = 2$ and $K = 3$. In the quantum device, the shadow circuit is implemented on the subspace of input state ρ_{in}. Sliding through the whole system to collect the Pauli-$(X \otimes X)$ expectations, i.e., shadow features. In the classic device, the resulting shadow features o_i's are fed into a fully-connected neural network (FCNN). Here, the softmax activation function is employed and the output $\hat{y}$ is a K-dimensional vector for the multi-label case.

3.3.2 Loss function

Given the data set $\mathcal{D} \equiv \{(\rho_{in}^{(m)}, y^{(m)})\}_{m=1}^{N} \subset \mathbb{C}^{2^n \times 2^n} \times \mathbb{R}^K$ and n_{qsc}-local shadow circuits, where $y^{(m)}$ is a one-hot vector which indicates the category to which the m^{th} data sample

Algorithm 3 VSQL for multi-label classification: the training process

```
Input: The training data set $\mathcal{D}^{(train)} \equiv \{(\rho_{in}^{(m)}, y^{(m)} \in \mathbb{R}^K)\}_{m=1}^{N_{train}}$, $EPOCH$, optimization procedure
Output: The final parameters $\boldsymbol{\theta}^*$, $\boldsymbol{W}^*$ and $\boldsymbol{b}^*$, and the list of losses
Initialize the parameters $\boldsymbol{\theta}$ of the 2-local (for example) shadow circuit $U(\boldsymbol{\theta})$ from uniform distribution $\text{Uni}[0, 2\pi]$ and $\boldsymbol{W}, \boldsymbol{b}$ from Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbb{I})$
for $ep = 1, \ldots, EPOCH$ do
    for $m = 1, \ldots, N_{train}$ do
        Apply multi-times the shadow circuit $U(\boldsymbol{\theta})$ to the input density matrix $\rho_{in}^{(m)}$
        Measure the subsystem and estimate a series of expectations $\langle X \otimes X \rangle$, recorded as $o_i$'s
        Feed the shadow features $o_i$'s into the classical neural network and obtain the output $\hat{y}^{(m)}$
        Compute the accumulated loss $\sum_{k=1}^{K} y_k^{(m)} \log \hat{y}_k^{(m)}$ and update accordingly the parameters $\boldsymbol{\theta}$, $\boldsymbol{W}$ and $\boldsymbol{b}$ via gradient-based optimization procedure
    end for
    if the stopping criterion is satisfied then
        Break
    end if
end for
```

$\rho_{in}^{(m)}$ belongs. For example, if $K = 3$, $y^{(m)} = [1, 0, 0]^\top$ indicates the m^{th} sample belongs to class 0, $y^{(m)} = [0, 1, 0]^\top$ for class 1 and $y^{(m)} = [0, 0, 1]^\top$ for class 2. The loss function of VSQL for multi-label classification is derived from cross-entropy [114]:

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b}; \mathcal{D}) \equiv -\frac{1}{N} \sum_{m=1}^{N} \sum_{k=1}^{K} y_k^{(m)} \log \hat{y}_k^{(m)} \left(\rho_{in}^{(m)}; \boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b} \right). \tag{3.45}$$

Here, the output K-dimensional vector $\hat{y}^{(m)}$ of VSQL is defined as follows:

$$\hat{y}^{(m)} \left(\rho_{in}^{(m)}; \boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b} \right) \equiv \sigma \left(\sum_{i=1}^{n-n_{qsc}+1} \boldsymbol{w}_i o_i^{(m)} \left(\rho_{in}^{(m)}; \boldsymbol{\theta} \right) + \boldsymbol{b} \right), \tag{3.46}$$

where $\boldsymbol{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{n-n_{qsc}+1}] \in \mathbb{R}^{K \times (n-n_{qsc}+1)}$, $\boldsymbol{b} \in \mathbb{R}^{K \times 1}$, $\sigma(\boldsymbol{z}) = \frac{\mathrm{e}^{\boldsymbol{z}}}{\sum_j \mathrm{e}^{z_j}}$ denotes the softmax activation function and the shadow features o_i are calculated through

$$o_i^{(m)} \left(\rho_{in}^{(m)}; \boldsymbol{\theta} \right) = \operatorname{Tr} \left(\rho_{in}^{(m)} (\mathbb{I} \otimes \cdots \otimes U^\dagger(\boldsymbol{\theta}) O U(\boldsymbol{\theta}) \otimes \cdots \otimes \mathbb{I}) \right). \tag{3.47}$$

Note that the shadow circuit $U(\boldsymbol{\theta})$ and the Hermitian operator $O = X \otimes \cdots \otimes X$ are applied on the same local qubits. We also note that the calculation of these shadow features and

Algorithm 4 VSQL for multi-label classification: the inference process

```
Input: The test data set $\mathcal{D}^{(test)} \equiv \{(\rho_{in}^{(m)}, y^{(m)} \in \mathbb{R}^K)\}_{m=1}^{N_{test}}$, the parameters $\boldsymbol{\theta}$, $\boldsymbol{W}$ and $\boldsymbol{b}$ from the training process
Output: The list of predicted labels and the test accuracy
Set the counter $n\_c = 0$, denoting the number of correct predicted labels
for $m = 1, \ldots, N_{test}$ do
    Apply multi-times the shadow circuit $U(\boldsymbol{\theta})$ to the input density matrix $\rho_{in}^{(m)}$
    Measure and estimate a series of expectations $\langle X \otimes X \rangle$, recorded as $o_i$'s
    Feed these shadow features $o_i$'s into the classical neural network and obtain the output $\hat{y}^{(m)} \in \mathbb{R}^K$
    if $l = \underset{k}{\text{argmax}}\{\hat{y}_k^{(m)}\}$ then
        Set the predicted label as '$l - 1$'
    end if
    if $\underset{k}{\text{argmax}}\{\hat{y}_k^{(m)}\} == \underset{k}{\text{argmax}}\{y_k^{(m)}\}$ then
        $n\_c = n\_c + 1$
    end if
end for
Compute the test accuracy as $n\_c/N_{test}$
```

the construction of shadow circuits are the same as the binary case, i.e.,

$$U(\boldsymbol{\theta}) = \prod_{l=L}^{1} U_l(\theta_l) V_l, \tag{3.48}$$

where $U_l(\theta_l) = \exp(-i\theta_l/2P_l)$ with the Pauli product operator P_l and V_l denotes a fixed operator such as Identity, CNOT and so on.

3.3.3 Analytical gradients

For each data sample $(\rho_{in}^{(m)}, y^{(m)})$ in the data set $\mathcal{D}$ and assume $y_k^{(m)} = 1$, the partial derivatives with respect to the parameters w_{ji}, b_j and θ_l are calculated as follows:

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b}; \rho_{in}^{(m)}, y^{(m)})}{\partial w_{ji}} = \begin{cases} \left(\hat{y}_k^{(m)} - 1\right) \cdot o_i^{(m)}, & j = k \\ \hat{y}_j^{(m)} \cdot o_i^{(m)}, & j \neq k \end{cases} \tag{3.49}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b}; \rho_{in}^{(m)}, y^{(m)})}{\partial b_j} = \begin{cases} \left(\hat{y}_k^{(m)} - 1\right), & j = k \\ \hat{y}_j^{(m)}, & j \neq k \end{cases} \tag{3.50}$$

$$\frac{\partial \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{W}, \boldsymbol{b}; \rho_{in}^{(m)}, y^{(m)})}{\partial \theta_l} = \sum_{i=1}^{n-n_{qsc}+1} \sum_{j=1}^{K} \left(\hat{y}_j^{(m)} w_{ji} - w_{ki}\right) \frac{\partial o_i^{(m)}\left(\boldsymbol{\theta}; \rho_{in}^{(m)}\right)}{\partial \theta_l}, \tag{3.51}$$

where $\hat{y}^{(m)}$ and $o_i^{(m)}$ are the corresponding abbreviations. It should be noted that the last term in Eq. (3.51), i.e., the partial gradient $\partial o_i^{(m)}/\partial \theta_l$, is resolved the same as Eq. (3.8).

3.3.4 Number of parameters

The number of parameters of VSQL for multi-label classification is summarized in the following proposition.

Proposition 3.11. *For an n-qubit quantum system, if we use n_s shadow circuits, then the number of parameters of VSQL for K-label classification is*

$$\begin{aligned} \#\ \textit{Params} &= \#\ \textit{Params}\Big|_{\textit{in shadow circuits}} + \#\ \textit{Params}\Big|_{\textit{in NN}} \\ &= n_s n_{qsc} D + \left[n_s \left(n - n_{qsc} + 1\right) + 1\right] K, \end{aligned} \tag{3.52}$$

where we denote by n_{qsc} the ***n****umber of* ***q****ubits of the* ***s****hadow* ***c****ircuits and assume each shadow circuit consists of D layers with n_{qsc} parameters in each layer.*

3.3.5 Number of repetitions for computing each shadow feature

Since the number of repetitions to estimate the shadow features is the same as the binary case, here we merely rewrite it simply.

Proposition 3.12. *(Same as Proposition 3.2.) Given a precision ϵ, the number of repetitions of the shadow circuit for computing each shadow feature at error ϵ, with probability at least $1-\eta$, scales as $O\left(\log(1/\eta)/\epsilon^2\right)$.*

Furthermore, by utilizing these estimated shadow features, VSQL outputs the prediction vector $\hat{y}$ and gives a label according to the following prediction rule

$$\text{predicted label} = \operatorname*{argmax}_k\{\hat{y}_k\} - 1. \tag{3.53}$$

Therefore, in the inference process of VSQL for multi-label classification, for an input state with the label $y \in \mathbb{R}^K$, if the predicted label is correct and the gap between the largest two values of the prediction vector $\hat{y}$ is τ under an infinite number of repetitions of the shadow circuits, then the actual number of repetitions, required to ensure that the input state is still correctly classified, will be similarly related to the gap τ. And an analogous result is concluded in Proposition 3.13.

Proposition 3.13. *For an n-qubit quantum system, if we use n_s shadow circuits and assume the final weights w_{ji} of the neural networks in VSQL are bounded as $|w_{ji}| \leq C_w$ for all i, j, and the prediction gap is $\tau \in (0,1)$, then the actual number of repetitions for computing each shadow feature, with probability at least $1-\eta$, scales as $O\left(n_s^2 n^2 C_w^2 \log(1/\eta)/\tau^2\right)$.*

Proof. If the estimated error of each shadow feature o_i is δ, then from Proposition 3.2, the number of repetitions, with probability at least $1-\eta$, is $O\left(\log(1/\eta)/\delta^2\right)$. What's more, due to

$$\frac{\partial \hat{y}_j}{\partial o_i} \equiv \frac{\partial}{\partial o_i}\left(\frac{\mathrm{e}^{\sum_i w_{ji}o_i+b_j}}{\sum_l \mathrm{e}^{\sum_i w_{li}o_i+b_l}}\right) = \hat{y}_j\left(1-\hat{y}_j\right)\cdot w_{ji} + \sum_{l,l\neq j} -\hat{y}_j\hat{y}_l \cdot w_{li} \tag{3.54}$$

$$\leq \hat{y}_j C_w\left(1-\hat{y}_j + \sum_{l,l\neq j}\hat{y}_l\right) = 2\hat{y}_j C_w\left(1-\hat{y}_j\right) \leq \frac{1}{2}C_w, \tag{3.55}$$

for all $j = 1, \ldots, K$, the error of each value of $\hat{y}$ could be bounded as $\frac{1}{2} n_s n C_w \delta$. If we let $\frac{1}{2} n_s n C_w \delta \leq \frac{\tau}{2}$, the number of repetitions for computing each shadow feature is obtained. □

3.3.6 Theoretical classification ability

Since it is too complex to explore the theoretical classification ability of VSQL for multi-label classification, here, we merely give a sufficient condition which is concluded in Theorem 3.14. From Theorem 3.14, we could also directly induce the corresponding results analogous to Corollary 3.5 and Theorem 3.6, here we omit them.

Theorem 3.14. *Given K types of input density matrices $\rho_{in}^{(0)}$, $\rho_{in}^{(1)}$ and up to $\rho_{in}^{(K-1)}$ with labels 0, 1 up to $K-1$, respectively. If there exists a group of $\boldsymbol{\theta}$ that makes at least one group of shadow features $o_i^{(0)}, o_i^{(1)}, \ldots, o_i^{(K-1)}$ different, i.e., $|o_i^{(k)} - o_i^{(k')}| > 0$ for all $k \neq k'$, then VSQL is theoretically capable of distinguishing them.*

Proof. Without loss of generality, we assume $i = 1$ and $o_1^{(0)} < o_1^{(1)} < \cdots < o_1^{(K-1)}$. By simply setting $w_{ji} = 0$ for $i \neq 1, j = 1, 2, \ldots, K$, we have

$$z_1 = w_{11} o_1^{(k)} + b_1, \quad z_2 = w_{21} o_1^{(k)} + b_2, \quad \cdots \quad z_K = w_{K1} o_1^{(k)} + b_K. \tag{3.56}$$

Our goal is to prove that z_{k+1} is the largest one for any $o_1^{(k)}$, $k = 0, 1, \ldots, K-1$, via adjusting $\boldsymbol{w}_1$ and $\boldsymbol{b}$.

Now if we define $o_1^{(-1)} = o_1^{(0)} - 1$, set $b_j = -w_{j1} o_1^{(j-2)}$ for all $j = 1, 2, \ldots, K$, and set $0 < w_{11} < \cdots < w_{j1} < \cdots < w_{K1}$ such that

$$w_{j1} > w_{j-1,1} \frac{o_1^{(j-1)} - o_1^{(j-3)}}{o_1^{(j-1)} - o_1^{(j-2)}}, \qquad \text{for} \quad j > 1. \tag{3.57}$$

Then we could easily verify that for any $o_1^{(k)}$,

1. if $l \geq k+2$, then

$$z_l = w_{l1} o_1^{(k)} + b_l = w_{l1} o_1^{(k)} - w_{l1} o_1^{(l-2)} = w_{l1} \left(o_1^{(k)} - o_1^{(l-2)} \right) \leq 0; \tag{3.58}$$

2. if $l = k + 1$, then

$$z_l = w_{l1} o_1^{(k)} + b_l = w_{l1} o_1^{(k)} - w_{l1} o_1^{(l-2)} = w_{l1} \left(o_1^{(k)} - o_1^{(k-1)} \right) > 0; \tag{3.59}$$

3. from Eq. (3.57), we have

$$z_{k+1} = w_{k+1,1} \left(o_1^{(k)} - o_1^{(k-1)} \right) > w_{k1} \left(o_1^{(k)} - o_1^{(k-2)} \right) > w_{k1} \left(o_1^{(k-1)} - o_1^{(k-2)} \right) = z_k, \tag{3.60}$$

   and go on we have $z_{k+1} > z_k > z_{k-1} > \cdots > z_1 = w_{11} \left(o_1^{(0)} - o_1^{(-1)} \right) = w_{11} > 0$.

Based on the above three cases, we obtain that z_{k+1} is the largest one, i.e., $\hat{y}_{k+1}$ is the largest. That is to say, for any input density matrix $\rho_{in}^{(k)}$, VSQL outputs the predicted label $= \underset{k}{\text{argmax}}\{\hat{y}_k\} - 1 = k$, which means classifying correctly. □

3.4 Numerical Experiments

We supplement our theoretical results with numerical experiments by classical simulation of VSQL. Specifically, our numerical experiments include distinguishing two (and three) families of 2-qubit quantum states and classifying handwritten digit images taken from the MNIST data set. We also conduct experiments on classifying noisy quantum states to exhibit the robustness of VSQL. All the simulations and optimization loops are implemented via Paddle Quantum[2] on the PaddlePaddle Deep Learning Platform [126].

3.4.1 Classification of Quantum States

Quantum state discrimination (QSD) is fundamental to the theory of quantum cryptography [127] and quantum communications [79, 106, 128]. It is usually defined as follows: can we recognize a quantum state ρ_k from a set of quantum states $\{\rho_i\}_{i=1}^N$ with known probability distribution $\{q_i\}_{i=1}^N$ for the quantum system to be in each corresponding state, via certain measurements? This is non-trivial since arbitrary pre-measurement manipulations

[2] https://github.com/paddlepaddle/Quantum

and measurement does not always extract useful classical information from the quantum system. Although, in principle, an optimal projective measurement can be designed according to the Helstrom bound [129] by minimizing the average guessing error, this kind of strategy is difficult to find in general and the optimal strategy is only known for limited cases. Furthermore, even if we could obtain this optimal measure, the amount of information that we can extract is still limited by the Holevo bound [130] and the physical realization of those measures remains challenging given the hardware restrictions mentioned before. From our perspective, it is natural to think of the combination of variationally searching appropriate pre-measurement manipulations and hardware-efficient measures instead of directly finding the optimal measure. In particular, those locally-operated shadow circuits $U(\boldsymbol{\theta})$ will function as the pre-measurement manipulation in VSQL and the Pauli measure X on each qubit is indeed hardware-efficient. In general, finite copies of the given states are considered in the study of distinguishing quantum states [72, 131].

3.4.1.1 Classification of Binary Quantum States

We choose two canonical families of non-orthogonal 2-qubit quantum states as proof of principle. These states are well-studied in Refs. [107, 108, 132], and are parametrized by real numbers u and v. Here, we use the Dirac (bra-ket) notation to represent the quantum states as

$$|\psi_u\rangle = [\sqrt{1-u^2}, 0, u, 0]^\top, \tag{3.61}$$

$$|\psi_{v\pm}\rangle = [0, \pm\sqrt{1-v^2}, v, 0]^\top, \tag{3.62}$$

where $u, v \in [0,1]$. Then, we can write these two sets of quantum states as a mixed quantum state ρ:

$$\rho(u,v) \equiv q_1 \underbrace{|\psi_u\rangle\langle\psi_u|}_{\rho_1(u)} + \frac{q_2}{2}(\underbrace{|\psi_{v+}\rangle\langle\psi_{v+}| + |\psi_{v-}\rangle\langle\psi_{v-}|}_{\rho_2(v)}), \tag{3.63}$$

with probability distribution $\{q_1 = \frac{1}{3}, q_2 = \frac{2}{3}\}$. These choices are consistent with the existing literature [107, 132].

3.4.1.2 Theoretical Distinguishability

We first analyze the ability of our method for classifying these two families of quantum states. The result is summarized in Theorem 3.15.

Theorem 3.15. *Suppose we have two families of non-orthogonal 2-qubit quantum states, shown in Eq.* (3.63)*. We further assume that each state has multiple copies. VSQL could exactly distinguish them by using only one 1-local shadow circuit, which consists of only one R_y rotation gate applied on a qubit.*

Proof. Without loss of generality, we assume $|\psi_u\rangle$ is labelled as '0' and $|\psi_v\rangle$ is labeled as '1'. Thus our goal is to show $\hat{y}\left(|\psi_u\rangle\langle\psi_u|\,;\theta,\boldsymbol{w},b\right) < 0.5$ and $\hat{y}\left(|\psi_v\rangle\langle\psi_v|\,;\theta,\boldsymbol{w},b\right) \geq 0.5$ for any $u, v \in [0,1]$, i.e.:

$$z_u \equiv w_1 o_1\left(|\psi_u\rangle\langle\psi_u|\,;\theta\right) + w_2 o_2\left(|\psi_u\rangle\langle\psi_u|\,;\theta\right) + b < 0 \tag{3.64}$$

$$z_v \equiv w_1 o_1\left(|\psi_v\rangle\langle\psi_v|\,;\theta\right) + w_2 o_2\left(|\psi_v\rangle\langle\psi_v|\,;\theta\right) + b \geq 0 \tag{3.65}$$

could be always satisfied with suitable w_1, w_2, θ and b.

Now we compute these 1-local shadow features from Eq. (3.3) as follows:

$$o_1\left(|\psi_x\rangle\langle\psi_x|\,;\theta\right) = \langle\psi_x|\left(U^\dagger(\theta) X U(\theta) \otimes \mathbb{I}\right)|\psi_x\rangle \tag{3.66}$$

$$o_2\left(|\psi_x\rangle\langle\psi_x|\,;\theta\right) = \langle\psi_x|\left(\mathbb{I} \otimes U^\dagger(\theta) X U(\theta)\right)|\psi_x\rangle\,, \tag{3.67}$$

where $x \in \{u, v\}$ and the shadow circuit $U(\theta)$ is set as $R_y(\theta)$. Since

$$R_y^\dagger(\theta) X R_y(\theta) = \begin{bmatrix} \cos\frac{\theta_2}{2} & \sin\frac{\theta_2}{2} \\ -\sin\frac{\theta_2}{2} & \cos\frac{\theta_2}{2} \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \cos\frac{\theta_2}{2} & -\sin\frac{\theta_2}{2} \\ \sin\frac{\theta_2}{2} & \cos\frac{\theta_2}{2} \end{bmatrix} = \begin{bmatrix} \sin\theta & \cos\theta \\ \cos\theta & -\sin\theta \end{bmatrix}, \tag{3.68}$$

we obtain the observables

$$\begin{aligned}
o_1\left(|\psi_u\rangle\langle\psi_u|\,;\theta\right) &= \begin{bmatrix}\sqrt{1-u^2} & 0 & u & 0\end{bmatrix}\cdot\begin{bmatrix}\sin\theta & 0 & \cos\theta & 0\\ 0 & \sin\theta & 0 & \cos\theta\\ \cos\theta & 0 & -\sin\theta & 0\\ 0 & \cos\theta & 0 & -\sin\theta\end{bmatrix}\cdot\begin{bmatrix}\sqrt{1-u^2}\\ 0\\ u\\ 0\end{bmatrix}\\
&= (1-2u^2)\sin\theta + 2u\sqrt{1-u^2}\cos\theta, \qquad (3.69)\\
o_2\left(|\psi_u\rangle\langle\psi_u|\,;\theta\right) &= \begin{bmatrix}\sqrt{1-u^2} & 0 & u & 0\end{bmatrix}\cdot\begin{bmatrix}\sin\theta & \cos\theta & 0 & 0\\ \cos\theta & -\sin\theta & 0 & 0\\ 0 & 0 & \sin\theta & \cos\theta\\ 0 & 0 & \cos\theta & -\sin\theta\end{bmatrix}\cdot\begin{bmatrix}\sqrt{1-u^2}\\ 0\\ u\\ 0\end{bmatrix}\\
&= \sin\theta; \qquad (3.70)
\end{aligned}$$

And similarly

$$o_1\left(|\psi_v\rangle\langle\psi_v|\,;\theta\right) = (1-2v^2)\sin\theta, \qquad (3.71)$$

$$o_2\left(|\psi_v\rangle\langle\psi_v|\,;\theta\right) = (2v^2-1)\sin\theta. \qquad (3.72)$$

Substituting Eqs. (3.69), (3.70), (3.71) and (3.72) into Eqs. (3.64) and (3.65), we have

$$z_u = w_1\left[(1-2u^2)\sin\theta + 2u\sqrt{1-u^2}\cos\theta\right] + w_2\sin\theta + b, \qquad (3.73)$$

$$z_v = (w_1 - w_2)\,(1-2v^2)\sin\theta + b. \qquad (3.74)$$

As w_1, w_2, θ and b are chosen arbitrarily and $u, v \in [0,1]$, without loss of generality, we assume $0 < \sin\theta \leq \cos\theta < 1$, then

$$(1-2u^2)\sin\theta + 2u\sqrt{1-u^2}\cos\theta \geq \left(1-2u^2+2u\sqrt{1-u^2}\right)\sin\theta \geq -\sin\theta. \qquad (3.75)$$

If we set $w_2 < w_1 < 0$, combining with Eq. (3.75), we have

$$z_u \leq (w_2 - w_1)\sin\theta + b, \qquad (3.76)$$

$$z_v \geq (w_2 - w_1)\sin\theta + b, \qquad (3.77)$$

where both the equal signs ('=') occur only if $u = v = 1$. Therefore, if we want $z_u < 0$ and $z_v \geq 0$ all the time, it's sufficient to have the following conditions:

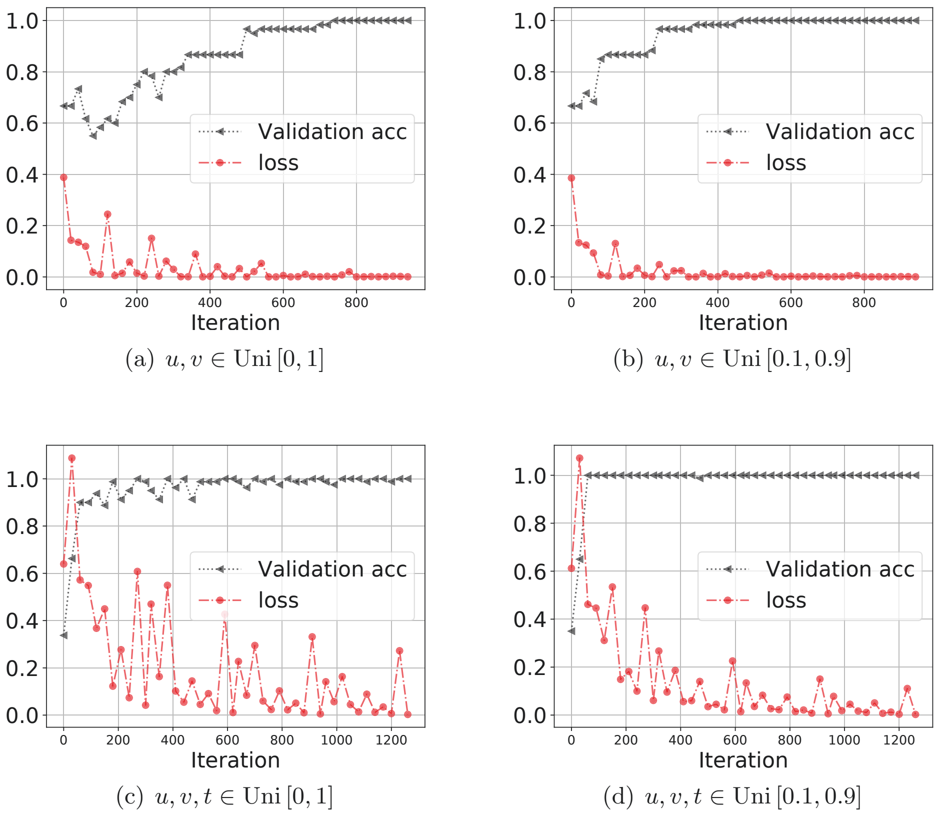
$$\begin{cases} 0 < \sin\theta \leq \cos\theta < 1, \\ w_2 < w_1 < 0, \\ (w_2 - w_1)\sin\theta + b = 0. \end{cases} \tag{3.78}$$

Of course, we could also have other settings for θ, w_1, w_2, b that satisfy our requirements, but here, one is enough. This completes the proof. □

This theorem shows that VSQL could theoretically distinguish these two different families of quantum states. We further evaluate the performance of VSQL via numerical experiments.

3.4.1.3 Experimental Setting and Results

300 density matrices with 100 $\rho_1(u)$ (labeled as '0') and 200 $\rho_2(v)$ (labeled as '1') are sampled according to Eq. (3.63), where the parameters u and v are uniformly taken from $[0, 1]$. Then, we randomly select 80% of them as the training set and the rest 20% as the validation set. Consistent with Theorem 3.15, one 1-local shadow circuit, which consists of an R_y gate only, is used to extract local features. The parameters of the shadow circuit $\boldsymbol{\theta}$ and the FCNN $\{\boldsymbol{w}, b\}$ are initialized from the uniform distribution $\text{Uni}[0, 2\pi]$ and the Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbb{I})$, respectively. During the optimization loop, we choose the Adam [133] optimizer with a learning rate $\text{LR} = 0.03$. Learning curves for the training loss and the validation accuracy are illustrated in Fig. 3.4(a), where the distinguishability shown coincides with Theorem 3.15. We conclude VSQL could perfectly recognize the two families of quantum states defined in Eq. (3.63) after about 700 iterations. We find that the classification task becomes very difficult when u, v are both close to 1. This makes sense because $\rho_1(u = 1) \to \rho_2(v = 1)$ on the extreme case. This experimental result highlights the strength of our method. As a comparison, we adjust the sample range such that e.g., $u, v \in [0.1, 0.9]$, and the results are shown in Fig. 3.4(b). As expected, this modification leads to faster convergence.



(a) $u, v \in \text{Uni}\,[0,1]$

(b) $u, v \in \text{Uni}\,[0.1, 0.9]$

(c) $u, v, t \in \text{Uni}\,[0,1]$

(d) $u, v, t \in \text{Uni}\,[0.1, 0.9]$

Figure 3.4: Learning curves that record the training loss and the validation accuracy of VSQL with different experimental settings. (a) and (b) are binary classification with different parameter range $u, v \in [0,1]$ and $u, v \in [0.1, 0.9]$. By adjusting the sample range, the training loss and the validation accuracy reach the optimal values faster. (c) and (d) describe a similar experimental setting but for a three-class classification of quantum states.

3.4.1.4 Classification of Multi-Class Quantum States

As declared before, our method can be easily extended to multi-class classification and numerically verified. Here, we take three different categories as an example. The data set we choose is again taken from Eq. (3.63), but adding the following third family $\rho_3(t) = |\psi_t\rangle\langle\psi_t|$ with a new probability distribution $\{q_i\} = \{\frac{1}{4}, \frac{1}{2}, \frac{1}{4}\}$,

$$|\psi_t\rangle = \left[\sqrt{1-t^2}, t, 0, 0\right]^\top, \tag{3.79}$$

where $t \in [0,1]$. We shall note that these three families of states $|\psi_u\rangle, |\psi_v\rangle$ and $|\psi_t\rangle$ are mutually non-orthogonal unless u, v, t are taken as 0 or 1. Hence, it's difficult to distinguish them via POVM method [106]. Now we use VSQL to distinguish them. Similarly, we generate another 100 density matrices ρ_3 which are labeled as '[0,0,1]' (here we use one-hot vectors to denote the labels, i.e., '[1,0,0]' for ρ_1 and '[0,1,0]' for ρ_2). The other experimental settings are identical to the binary case except for the softmax activation function used in FCNN. Similar learning curves of the training process for the loss and the validation accuracy are demonstrated in Fig. 3.4(c), which shows VSQL could perfectly distinguish multi-class quantum states by reaching 100% validation accuracy. The fluctuations on the loss curve are probably due to the design of the cross-entropy loss function and the existence of the highly non-orthogonal data samples. As a consequence, the validation accuracy is also jiggling around but gradually converges to the theoretical maximum. Similar to the binary case, we repeat the simulation by sampling $u, v, t \in \text{Uni}\,[0.1, 0.9]$ and summarize the results in Fig. 3.4(d). This eliminates the extreme cases $u, v, t \in \{0, 1\}$ which reduce the multi-class to binary classification. As expected, smaller fluctuations are observed which means our method could unambiguously distinguish multi-class quantum states.

3.4.2 MNIST Classification

Next, we apply VSQL to classify handwritten digits taken from a public benchmark dataset MNIST [134], which consists of 60,000 train examples and 10,000 test examples. The MNIST data set contains 10 different classes labeled from '0' to '9'. Each image contains 28×28 grayscale pixels valued in $0 \sim 255$. In order to match the input of VSQL, these pictures are normalized and unfolded into 784-dimensional vectors. Then, we expand their dimension to 10-qubit pure quantum states $\{|\psi_i\rangle\}$ (1024-dimensional vectors) via zero-padding and represent them in the density matrix formulation $\{\rho_{in}^{(i)}\} = \{|\psi_i\rangle\langle\psi_i|\}$. By doing so, the pre-processing is complete and we obtain the training set $\mathcal{D}^{(train)} \equiv \{(\rho_{in}^{(m)}, y^{(m)})\}_{m=1}^{N_{train}} \subset \mathbb{C}^{1024\times1024} \times \mathbb{R}^{10}$. We first select two classes ('0' and '1') to verify the binary classification ability of VSQL, which contains 12,665 training samples (5923 0-label and 6742 1-label) and 2115 test samples (980 0-label and 1135 1-label). Then, we use the whole data set to evaluate the 10-class classification performance.

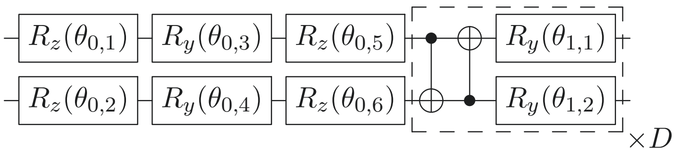


FIGURE 3.5: The 2-local shadow circuit design for MNIST classification (binary case). The first part uses $R_z - R_y - R_z$ combination to represent general rotations on each single-qubit subspace. The following repeated block consists of CNOT gates and two single-qubit R_y rotations. The block circuit in the dashed box is repeated D times to extend the expressive power of quantum circuits.

3.4.2.1 Experimental Setting

For the binary case, the 2-local shadow circuit (ansatz) used to extract local features is shown in Fig. 3.5. The number of repetitions of the dashed block structure is denoted as the circuit depth D and this ansatz has $2(D+3)$ parameters in total. The parameters $\boldsymbol{\theta}$ and $\{\boldsymbol{w}, b\}$ are initialized from a uniform distribution in $[0, 2\pi]$ and a Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbb{I})$, respectively. During the optimization, we choose the Adam optimizer with a batch size of 20 samples and a learning rate of LR $= 0.02$. Each experiment is repeated 10 times to collect the mean accuracy and the corresponding fluctuations. For the 10-class case, the classification task becomes much more difficult and hence we choose 4-local shadow circuits to extract shadow features, which can be extended from the 2-local design in Fig. 3.5. There will be $4(D+3)$ parameters in each shadow circuit. All the other settings are identical to the binary case, except for a new batch size of 200 samples.

3.4.2.2 Results

The results for the binary case are summarized in Table 3.1. Our method VSQL easily achieves an average test accuracy above 99% with only $n_s = 1$ shadow circuit and depth $D = 1$, which has 8 rotation angles in the shadow circuit and 9 weights and 1 bias in FCNN. This result demonstrates the powerful capacity of VSQL to classify handwritten digits. By adding another shadow circuit to $n_s = 2$ with 35 parameters, one could obtain an average test accuracy above 99.5%. As a comparison, we list the results of existing methods: Circuit-centric classifier [60] and QNN classifier [77]. Our method outperforms these variational quantum classifiers in terms of the number of parameters and test accuracy. Here, we should note that the details of their data preprocessing are slightly

Methods	n_s	D	# Ps	Test acc (%)
Circuit-centric classifier [60]	/	/	124	96.70
QNN classifier [77]	/	/	96	98.00
VSQL (this chapter)	1	1	18	99.43 ± 0.14
	2	1	35	**99.52 ± 0.18**

Table 3.1: Summary of the existing variational quantum classifiers on MNIST binary classification. VSQL outperforms other classifiers in terms of the number of parameters and test accuracy by reaching 99.52% average test accuracy among 10 random experiments. # Ps denotes # Params.

different from us, i.e., the Circuit-centric classifier uses the MNIST256 dataset with an 8-qubit quantum system, and the QNN classifier uses a 4×4 downsampled version of the MNSIT dataset with a 17-qubit quantum system. Although we have adopted a different data preprocessing strategy, such excellent results (test error less than 1%) are sufficient to illustrate the effectiveness of our method.

For multi-class classification, it is rarely discussed and tested in the literature of variational quantum classifiers. The one-vs.-all method is mentioned in Schuld et al. but is troublesome to implement. Therefore, we only compare the performance of VSQL with a single-layered classical neural network (NN). The experimental settings of the classical neural network are similar to VSQL, and it contains 7840 weights and 10 biases to map the 784-dimensional input vectors to 10-dimensional output vectors. The results are summarized in Table 3.2. When using 9 different shadow circuits with each circuit depth $D = 5$, VSQL could reach almost the same test accuracy with the single layer NN, but requiring much fewer parameters. Although this accuracy is not quite satisfactory, it can still compete with the simplest classical NN. Notably, we find that if we select 1k samples (about 100 samples for each class) for training from 60k examples and choose the same size of test examples (10k), VSQL could achieve a higher test accuracy than NN (cf. the bottom half of Table 3.2). The above finding indicates that VSQL could extract high-level features from fewer training samples than NN, which may be a potential advantage of VSQL for future practical applications in the NISQ era.

Methods	n_s	D	# Ps	Test acc (%)
NN (60k samples)	/	1	7850	**91.63 ± 0.15**
VSQL (60k samples)	5	5	520	87.69 ± 0.98
VSQL (60k samples)	9	5	928	91.13 ± 0.51
NN (1k samples)	/	1	7850	86.36 ± 0.23
VSQL (1k samples)	5	5	520	83.92 ± 1.20
VSQL (1k samples)	9	5	928	**87.39 ± 0.40**

Table 3.2: MNIST 10-class classification

3.4.3 Distinguishing Noisy Quantum States

In practice, it is inevitable to deal with noise on the current quantum hardware which leads to noisy quantum states. Thus, it is essential to verify whether VSQL could distinguish noisy quantum states if we want to realize VSQL on the hardware in near future. In this subsection, we will run simulations on a pair of constructed noisy quantum states with high fidelity.

The procedure for generating this pair of simulated quantum states is as follows:

(i) We first construct two pure states (in 3 qubits as an example) with high fidelity: $\rho^{(0)} = |\psi_0\rangle\langle\psi_0|$ and $\rho^{(1)} = |\psi_1\rangle\langle\psi_1|$, which are labeled 0 and 1 respectively, where

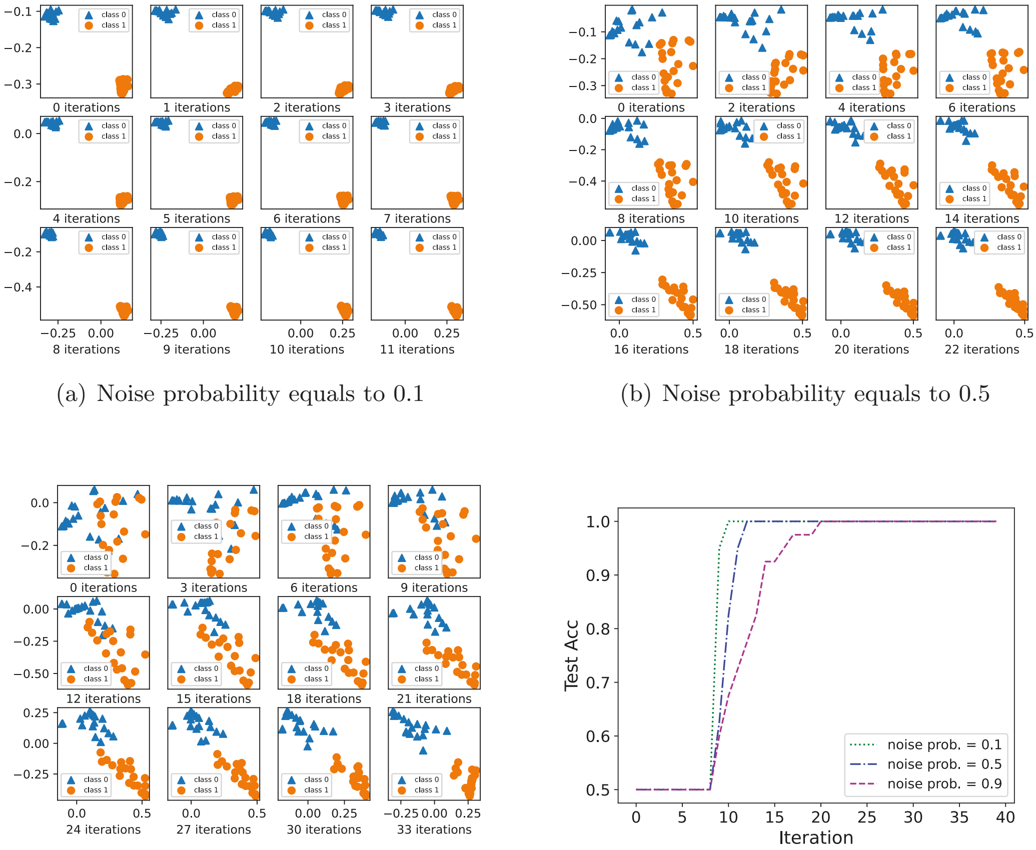
$$|\psi_0\rangle = 1/2\,(|000\rangle + |001\rangle + |010\rangle + |011\rangle)\,, \tag{3.80}$$

$$|\psi_1\rangle = 1/\sqrt{3}\,(|000\rangle + |001\rangle + |010\rangle)\,. \tag{3.81}$$

(ii) Then we sample a unitary U_s from matrix QR decomposition and apply it to these two pure states: $U_s\rho^{(0)}U_s^\dagger$ and $U_s\rho^{(1)}U_s^\dagger$.

(iii) Last we imposed a Pauli noise on the states, i.e.,

$$\rho_{in}^{(i)} = (1-p_i)U_s\rho^{(i)}U_s^\dagger + \frac{p_i}{3}\sum_{j=1}^{3} E_j U_s\rho^{(i)}U_s^\dagger E_j^\dagger, \tag{3.82}$$



(a) Noise probability equals to 0.1

(b) Noise probability equals to 0.5

(c) Noise probability equals to 0.9

(d) Test accuracy curves

FIGURE 3.6: Shadow features changing with the number of iterations under different noise probabilities, and the corresponding test accuracy curves.

where $i \in \{0, 1\}$, $E_1 = P \otimes \mathbb{I} \otimes \mathbb{I}, E_2 = \mathbb{I} \otimes P \otimes \mathbb{I}, E_3 = \mathbb{I} \otimes \mathbb{I} \otimes P$, $P \in \{X, Y, Z\}$, and the noise probability p_i is sampled from a uniform distribution $[0, \mathcal{P}]$ with a constant $\mathcal{P}$ between 0 and 1.

In our experiment, given a fixed $\mathcal{P}$, we sample 40 probabilities p_0's and 40 p_1's and thus generate 40 noisy quantum states for each class. Amongst these states, 50% is for training and the remaining 50% for testing. We employ one 2-local shadow circuit, which is similar to Fig. 3.5 with depth $D = 1$. The learning rate is set to 0.1 and the other experimental settings are similar to the above two experiments.

In order to explore the sensitivity of VSQL to the noise level, we conduct multiple experiments by setting $\mathcal{P}$ as $0.1, 0.5, 0.9$, respectively. The test accuracy curves in the training process are illustrated in Fig. 3.6(d), where we see intuitively that all the test accuracy could reach 100% after 20 iterations also, even though given a higher noise level. It also shows that the lower the noise level is, the faster the test accuracy increases, which is in line with our intuition. Furthermore, for the sake of figuratively understanding the classification ability of VSQL, we record the two shadow features (in Eq. (3.3)) of the 40 test quantum states in each training iteration. The results with different noise probabilities are illustrated in Figs. 3.6(a), 3.6(b) and 3.6(c), respectively. We observe that it is easier to distinguish when the noise probability equals 0.1 or 0.5, as the corresponding two classes of points are distributed in two clusters initially. However, even if the two classes of points are interlaced initially when the noise probability equals 0.9, they will be gradually separated into two clusters with the training process going on.

3.5 Discussions

We proposed the VSQL framework, which adopts a similar idea of obtaining classical shadows to distinguish quantum data. With theoretical justifications and numerical experiments, we have shown that VSQL for classification outperforms many other variational classifiers on the benchmark test of binary MNIST handwritten digit recognition with much fewer network parameters. In particular, in our framework, less noise will be introduced during the quantum-classical hybrid information processing as the number of quantum gates used is independent of the problem size. By sampling a slice of the loss landscape, we also briefly introduced the barren plateau problem and showed the solution to escape from it. By adjusting the operating scope of shadow circuits, our approach can be easily implemented on existing quantum devices with topological connectivity limitations.

We believe that VSQL would open the possibility for many future directions. For example, in our VSQL, we set the local quantum circuits in a convolutional way. However, the best combination of these local circuits deserves further exploration for practical problems, especially on NISQ devices. It would also be interesting to explore the applications of VSQL for generative models and unsupervised quantum machine learning tasks such as clustering. Furthermore, the online learning version of VSQL may also be a good future direction, see [135–137]. We also expect that VSQL may shed light on other quantum applications on near-term quantum devices.

Chapter 4

Quantum Self-Attention Neural Networks

4.1 Introduction

In recent years, plenty of NISQ algorithms [122, 138–140] dealing with machine learning problems, by employing parameterized quantum circuits [74] (also called quantum neural networks [77]), show great potential in the field of quantum machine learning. However, in artificial intelligence, the study of quantum machine learning in the NISQ era is still in its infancy. Thus it is desirable to explore more quantum machine learning algorithms exploiting the power that lies within the NISQ devices.

This chapter mainly focuses on quantum natural language processing, and the remainder of this chapter is organized as follows: in Sec. 4.2, we introduce the quantum self-attention neural networks, which includes quantum self-attention layer, ansatz selection, loss function, analytical gradients and complexity analysis. In Sec. 4.3, some numerical experiments are conducted on MC, RP, Yelp, IMDb and Amazon data sets, and also some visualization and noisy experiments are performed on the Yelp data set. Finally, some discussions about future research are included in Sec. 4.4.

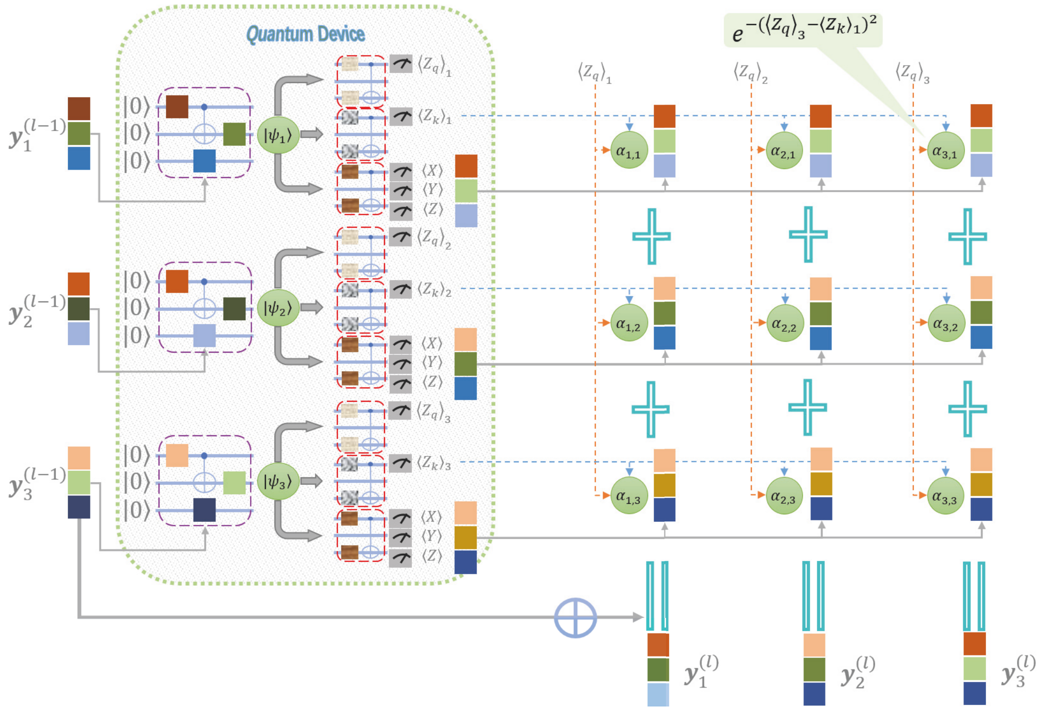


Figure 4.1: Sketch of a quantum self-attention layer (QSAL). On quantum devices, the classical inputs $\{\boldsymbol{y}_s^{(l-1)}\}$ are used as the rotation angles of quantum ansatzes (purple dashed boxes) to encode them into their corresponding quantum states $\{|\psi_s\rangle\}$. Then for each state, there are three different classes of ansatzes (red dashed boxes) that need to be executed, where the top two classes denote the query and key parts, and the bottom one denotes the value part. On classical computers, the measurement outputs of the query part $\langle Z_q\rangle_s$ and the key part $\langle Z_k\rangle_j$ are computed through a Gaussian function to obtain the quantum self-attention coefficients $\alpha_{s,j}$ (green circles); we calculate classically weighted sums of the measurement outputs of the value part (small colored squares) and add the inputs to get the outputs $\{\boldsymbol{y}_s^{(l)}\}$, where the weights are the normalized coefficients $\tilde{\alpha}_{s,j}$, cf. Eq. (4.5).

4.2 Method

In this section, we introduce the QSANN in detail, which mainly consists of *quantum self-attention layer* (QSAL), loss function, analytical gradients and analysis.

4.2.1 Quantum Self-Attention Layer

In the classical self-attention mechanism [33], there are mainly three components (vectors), i.e., queries, keys and values, where queries and keys are computed as weights assigned to corresponding values to obtain final outputs. Inspired by this mechanism, in QSAL we design the quantum analogs of these components. The overall picture of QSAL is illustrated in Fig. 4.1.

For the classical input data $\{\boldsymbol{y}_s^{(l-1)} \in \mathbb{R}^d\}$ of the l-th QSAL, we first use a quantum ansatz U_{enc} to encode them into an n-qubit quantum Hilbert space, i.e.,

$$|\psi_s\rangle = U_{enc}(\boldsymbol{y}_s^{(l-1)})H^{\otimes n}|0^n\rangle, \quad 1 \leq s \leq S, \tag{4.1}$$

where H denotes the Hadamard gate and S denotes the number of input vectors in a data sample.

Then we use another three quantum ansatzes, i.e., U_q, U_k, U_v with parameters $\boldsymbol{\theta}_q$, $\boldsymbol{\theta}_k$, $\boldsymbol{\theta}_v$, to represent the query, key and value parts, respectively. Concretely, for each input state $|\psi_s\rangle$, we denote by $\langle Z_q\rangle_s$ and $\langle Z_k\rangle_s$ the Pauli-Z_1 measurement outputs of the query and key parts, respectively, where

$$\begin{aligned}\langle Z_q\rangle_s &\equiv \langle\psi_s| U_q^\dagger(\boldsymbol{\theta}_q)Z_1U_q(\boldsymbol{\theta}_q)|\psi_s\rangle, \\ \langle Z_k\rangle_s &\equiv \langle\psi_s| U_k^\dagger(\boldsymbol{\theta}_k)Z_1U_k(\boldsymbol{\theta}_k)|\psi_s\rangle.\end{aligned} \tag{4.2}$$

The measurement outputs of the value part are represented by a d-dimensional vector

$$\boldsymbol{o}_s \equiv \begin{bmatrix}\langle P_1\rangle_s & \langle P_2\rangle_s & \cdots & \langle P_d\rangle_s\end{bmatrix}^\top, \tag{4.3}$$

where $\langle P_j\rangle_s = \langle\psi_s| U_v^\dagger(\boldsymbol{\theta}_v)P_jU_v(\boldsymbol{\theta}_v)|\psi_s\rangle$. Here, each $P_j \in \{I, X, Y, Z\}^{\otimes n}$ denotes a Pauli observable.

Finally, by combining Eqs. (4.2) and (4.3), the classical outputs $\{\boldsymbol{y}_s^{(l)} \in \mathbb{R}^d\}$ of the l-th QSAL are computed as follows:

$$\boldsymbol{y}_s^{(l)} = \boldsymbol{y}_s^{(l-1)} + \sum_{j=1}^{S} \tilde{\alpha}_{s,j} \cdot \boldsymbol{o}_j, \tag{4.4}$$

where $\tilde{\alpha}_{s,j}$ denotes the normalized quantum self-attention coefficient between the s-th and the j-th input vectors and is calculated by the corresponding query and key parts:

$$\tilde{\alpha}_{s,j} = \frac{\alpha_{s,j}}{\sum_{m=1}^{S} \alpha_{s,m}} \qquad \text{with} \qquad \alpha_{s,j} \equiv \mathrm{e}^{-(\langle Z_q \rangle_s - \langle Z_k \rangle_j)^2}. \tag{4.5}$$

Here in Eq. (4.4), we adopt a residual scheme when computing the output, which is analogous to [33].

4.2.1.1 Gaussian Projected Quantum Self-Attention

When designing a quantum version of self-attention, a natural and direct extension of the inner-product self-attention to consider is $\alpha_{s,j} \equiv | \langle \psi_s | U_q^\dagger U_k | \psi_j \rangle |^2$. However, due to the unitary nature of quantum circuits, $\langle \psi_s | U_q^\dagger U_k$ can be regarded as rotating $|\psi_s\rangle$ by an angle, which makes it difficult for $|\psi_s\rangle$ to simultaneously correlate those $|\psi_j\rangle$ that are far away. In a word, this direct extension is not suitable or reasonable for working as the quantum self-attention. Instead, the particular quantum self-attention proposed in Eq. (4.5), which we call *Gaussian projected quantum self-attention* (GPQSA), could overcome the above drawback. In GPQSA, the states $U_q |\psi_s\rangle$ (and $U_k |\psi_j\rangle$) in large quantum Hilbert space are projected to classical representations $\langle Z_q \rangle_s$ (and $\langle Z_k \rangle_j$) in one-dimensional[1] classical space via quantum measurements, and a Gaussian function is applied to these classical representations. As U_q and U_k are separated, it's pretty easier to correlate $|\psi_s\rangle$ to any $|\psi_j\rangle$, making GPQSA more suitable to serve as a quantum self-attention. Here, we utilize the Gaussian function [141] mainly because it contains infinite-dimensional feature space and is well-studied in classical machine learning. Numerical experiments also confirm our choice of Gaussian function. We also note that other choices for building quantum self-attention are also worth future study.

Remark. During the preparation of this manuscript as well as after submitting our work to a peer-review conference, we became aware that Ref. [142] also made initial attempts to employ the attention mechanism in QNNs. In that work, the authors mentioned a possible quantum extension towards a quantum Transformer where the straightforward inner-product self-attention is adopted. As discussed above, the inner-product self-attention

[1] Multi-dimension is also possible by choosing multiple measurement results, like the value part.

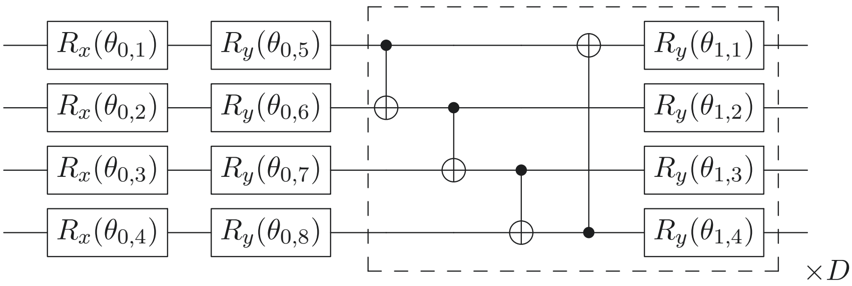


FIGURE 4.2: The ansatz used in QSANN. The first two columns denote the R_x-R_y rotations on each single-qubit subspace, then followed by repeated CNOT gates and single-qubit R_y rotations. The block circuit in the dashed box is repeated D times to enhance the expressive power of the ansatz.

may not be reasonable for dealing with quantum data. In this chapter, we present that GPQSA is more suitable for the quantum version of self-attention and show the validity of our method via numerical experiments on several public data sets.

4.2.2 Ansatz Selection

In QSAL, we employ multiple ansatzes for the various components, i.e., data encoding, query, key and value. Hence, we give a brief review of it here.

In general, an ansatz, a.k.a. parameterized quantum circuit [74], has the form $U(\boldsymbol{\theta}) = \prod_j U_j(\theta_j)V_j$, where $U_j(\theta_j) = \exp(-i\theta_j P_j/2)$ and V_j denotes a fixed operator such as Identity, CNOT and so on. Here, P_j denotes a Pauli operator. Due to the numerous choices of the form of V_j, various kinds of ansatzes can be used. In this chapter, we use the strongly entangling ansatz [60] shown in Fig. 4.2 in QSAL. This circuit has $n(D+2)$ parameters in total for n qubits and D repeated layers.

4.2.3 Loss Function

Consider the data set $\mathcal{D} \equiv \{({}^{(m)}\boldsymbol{x}_1,\ {}^{(m)}\boldsymbol{x}_2,\ \ldots,\ {}^{(m)}\boldsymbol{x}_{S_m}),\ {}^{(m)}y\}_{m=1}^{N_s}$, where there are in total N_s sequences or samples and each has S_m words with a label ${}^{(m)}y \in \{0,1\}$. Here, we assume each word is embedded as a d-dimensional vector, i.e., ${}^{(m)}\boldsymbol{x}_s \in \mathbb{R}^d$. The whole procedure of QSANN is depicted in Fig. 4.3, which mainly consists of L QSALs to extract

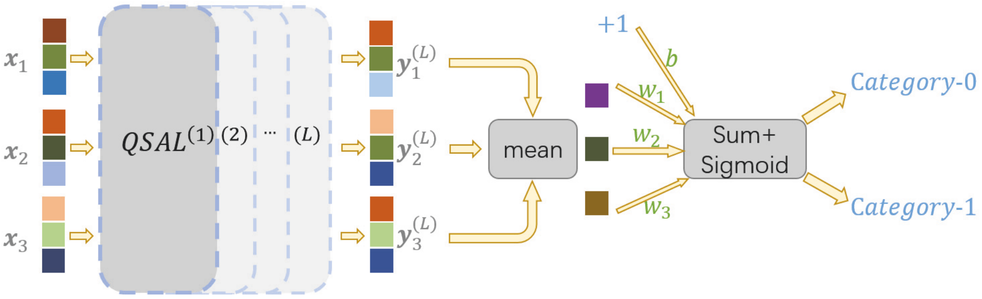


FIGURE 4.3: Sketch of QSANN, where a sequence of classical vectors $\{\boldsymbol{x}_s\}$ firstly goes through L QSALs to obtain the corresponding sequence of feature vectors $\{\boldsymbol{y}_s^{(L)}\}$, then through the average operation, and finally through the fully-connected layer for the binary prediction task.

hidden features and one fully-connected layer to complete the binary prediction task. Here, the mean squared error [118] is employed as the loss function:

$$\mathcal{L}\left(\boldsymbol{\Theta}, \boldsymbol{w}, b; \mathcal{D}\right) = \frac{1}{2N_s} \sum_{m=1}^{N_s} \left({}^{(m)}\hat{y} - {}^{(m)}y \right)^2 + \text{RegTerm}, \tag{4.6}$$

where the predicted value ${}^{(m)}\hat{y}$ is defined as ${}^{(m)}\hat{y} \equiv \sigma\left(\boldsymbol{w}^\top \cdot \frac{1}{S_m}\sum_{s=1}^{S_m} {}^{(m)}\boldsymbol{y}_s^{(L)} + b\right)$ with $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ denoting the weights and bias of the final fully-connected layer, $\boldsymbol{\Theta}$ denoting all parameters in the ansatz, σ denoting the sigmoid activation function and 'RegTerm' being the regularization term to avoid overfitting in the training process.

Combining Eqs. (4.1) - (4.5), we know each output of QSAL is dependent on all its inputs, i.e.,

$$\begin{aligned} {}^{(m)}\boldsymbol{y}_s^{(l)} &\equiv {}^{(m)}\boldsymbol{y}_s^{(l)} \left(\boldsymbol{\theta}_q^{(l)}, \boldsymbol{\theta}_k^{(l)}, \boldsymbol{\theta}_v^{(l)}; \{{}^{(m)}\boldsymbol{y}_i^{(l-1)}\}_{i=1}^{S_m}\right) \\ &= {}^{(m)}\boldsymbol{y}_s^{(l-1)} + \sum_{j=1}^{S_m} \tilde{\alpha}_{s,j}^{(l)} \left(\boldsymbol{\theta}_q^{(l)}, \boldsymbol{\theta}_k^{(l)}; \{{}^{(m)}\boldsymbol{y}_i^{(l-1)}\}_{i=1}^{S_m}\right) \cdot \boldsymbol{o}_j^{(l)} \left(\boldsymbol{\theta}_v^{(l)}; {}^{(m)}\boldsymbol{y}_j^{(l-1)}\right), \end{aligned} \tag{4.7}$$

where ${}^{(m)}\boldsymbol{y}_s^{(0)} = {}^{(m)}\boldsymbol{x}_s$ and $1 \leq s \leq S_m$, $1 \leq l \leq L$. Here, the regularization term is defined as

$$\text{RegTerm} \equiv \frac{\lambda}{2d}\|\boldsymbol{w}\|^2 + \frac{\gamma}{2d}\sum_{s=1}^{S_m} \|\,{}^{(m)}\boldsymbol{x}_s\|^2, \tag{4.8}$$

where $\lambda, \gamma \geq 0$ are two regularization coefficients.

With the loss function defined in Eq. (4.6), we can optimize its parameters by (stochastic) gradient-descent [119]. The analytical gradient analysis can be found in Subsection 4.2.4. Finally, with the above preparation, we could train our QSANN to get the optimal (or sub-optimal) parameters. See Algorithm 5 for details on the training procedure. We remark that if the loss converges during training or the maximum number of iterations is reached, the optimization stops.

4.2.4 Analytical Gradients

Here, we give the stochastic analytical partial gradients of the loss function with regard to its parameters as follows.

We first consider the parameters in the last quantum self-attention neural network layer, i.e., $\boldsymbol{\theta}_q^{(L)}, \boldsymbol{\theta}_k^{(L)}, \boldsymbol{\theta}_v^{(L)}$, and the final fully-connected layer, i.e., $\boldsymbol{w}, b$, and then the parameters in the front layers could be evaluated in a similar way and be updated through back-propagation algorithm [114]. Given the m-th data sample $\{(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{S_m}), y\}$ (here, we omit (m) in the left superscript for writing convenience, the same below in this subsection), we have

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \tilde{\sigma} \cdot \frac{1}{S_m} \sum_{s=1}^{S_m} \boldsymbol{y}_s^{(L)} + \frac{\lambda}{d} \boldsymbol{w}, \qquad \frac{\partial \mathcal{L}}{\partial b} = \tilde{\sigma}, \tag{4.9}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} = \tilde{\sigma} \cdot \frac{1}{S_m} \cdot \boldsymbol{w}, \tag{4.10}$$

where $\tilde{\sigma} = (\sigma - y) \cdot \sigma (1 - \sigma)$ and σ denotes the abbreviation of $\sigma \left(\boldsymbol{w}^\top \cdot \frac{1}{S_m} \sum_{s=1}^{S_m} \boldsymbol{y}_s^{(L)} + b \right)$. And we also have

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_v^{(L)}} = \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^\top \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \boldsymbol{o}_j^{(L)}} \cdot \frac{\partial \boldsymbol{o}_j^{(L)}}{\partial \boldsymbol{\theta}_v^{(L)}}, \tag{4.11}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_q^{(L)}} = \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^\top \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \alpha_{s,j}^{(L)}} \cdot \frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_q \rangle_s} \cdot \frac{\partial \langle Z_q \rangle_s}{\partial \boldsymbol{\theta}_q^{(L)}}, \tag{4.12}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k^{(L)}} = \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^\top \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \alpha_{s,j}^{(L)}} \cdot \sum_{i=1}^{S_m} \frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_k \rangle_i} \cdot \frac{\partial \langle Z_k \rangle_i}{\partial \boldsymbol{\theta}_k^{(L)}}, \tag{4.13}$$

where $\partial \boldsymbol{y}_s^{(L)} / \partial \boldsymbol{o}_j^{(L)} = \alpha_{s,j}^{(L)}$, $\partial \boldsymbol{y}_s^{(L)} / \partial \alpha_{s,j}^{(L)} = \boldsymbol{o}_j^{(L)}$, $\partial \alpha_{s,j}^{(L)} / \partial \langle Z_q \rangle_s = -\sum_{i=1}^{S_m} \partial \alpha_{s,j}^{(L)} / \partial \langle Z_k \rangle_i$ and

$$\frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_k \rangle_i} = -\alpha_{s,j}^{(L)} \left(\alpha_{s,i}^{(L)} - \delta_{ij} \right) \cdot 2 \left(\langle Z_q \rangle_s - \langle Z_k \rangle_i \right),$$

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & \text{otherwise.} \end{cases} \tag{4.14}$$

Furthermore, the last three partial derivatives of Eqs. (4.11), (4.12) and (4.13) could be evaluated exactly on the quantum computers via the parameter shift rule [98]. For example, by combining Eq. (4.2),

$$\frac{\partial \langle Z_q \rangle_s}{\partial \theta_{q,j}^{(L)}} = \frac{1}{2} \left(\langle Z_q \rangle_{s,+} - \langle Z_q \rangle_{s,-} \right), \tag{4.15}$$

where $\langle Z_q \rangle_{s,\pm} = \langle \psi_s | U_{q,\pm}^\dagger Z U_{q,\pm} | \psi_s \rangle$ and $U_{q,\pm} \equiv U_q \left(\boldsymbol{\theta}_{q,-j}^{(L)}, \theta_{q,j}^{(L)} \pm \frac{\pi}{2} \right)$.

Finally, in order to derive the partial derivatives of the parameters in the front layers, we also need the following

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_i^{(L-1)}} = & \frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_i^{(L)}} + \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^\top \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \boldsymbol{o}_i^{(L)}} \cdot \frac{\partial \boldsymbol{o}_i^{(L)}}{\partial \boldsymbol{y}_i^{(L-1)}} \\ & + \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_i^{(L)}} \right)^\top \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_i^{(L)}}{\partial \alpha_{i,j}^{(L)}} \cdot \frac{\partial \alpha_{i,j}^{(L)}}{\partial \langle Z_q \rangle_i} \cdot \frac{\partial \langle Z_q \rangle_i}{\partial \boldsymbol{y}_i^{(L-1)}} \\ & + \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^\top \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \alpha_{s,j}^{(L)}} \cdot \frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_k \rangle_i} \cdot \frac{\partial \langle Z_k \rangle_i}{\partial \boldsymbol{y}_i^{(L-1)}}, \end{aligned} \tag{4.16}$$

where the four terms denote the residual, value, query and key parts, respectively, and each sub-term can be evaluated similarly to the above analysis. With the above preparation, we could easily calculate every parameter's gradients and update these parameters accordingly.

4.2.5 Analysis of QSANN

According to the definition of the Quantum Self-Attention Layer, for a sequence with S words, we need $S(d+2)$ Pauli measurements to obtain the d-dimensional value vectors

Algorithm 5 QSANN training for text classification

```
Input: The training data set $\mathcal{D} \equiv \{({}^{(m)}\boldsymbol{x}_1,\ {}^{(m)}\boldsymbol{x}_2,\ \ldots,\ {}^{(m)}\boldsymbol{x}_{S_m}),\ {}^{(m)}y\}_{m=1}^{N_s}$, $EPOCH$,
    number of QSALs $L$ and optimization procedure
Output: The final ansatz parameters $\Theta^*$, weight $\boldsymbol{w}^*, b^*$
 1: Initialize the ansatz parameters $\Theta$, weight $\boldsymbol{w}$ from Gaussian distribution $\mathcal{N}(0, 0.01)$
    and the bias $b$ to 0.
 2: for $ep = 1, \ldots, EPOCH$ do
 3:     for $m = 1, \ldots, N_s$ do
 4:         Apply the encoder ansatz $U_{enc}$ to each of ${}^{(m)}\boldsymbol{x}_s$ to get the corresponding quantum
            state $|\psi_s\rangle$, cf. (4.1).
 5:         Apply $U_q$ and $U_k$ to $|\psi_s\rangle$ and measure the Pauli-Z expectations to get $\langle Z_q\rangle_s, \langle Z_k\rangle_s$,
            cf. (4.2), and then calculate the quantum self-attention coefficients $\alpha_{s,j}$, cf. (4.5).

 6:         Apply $U_v$ and measure a series of Pauli expectations to get $\boldsymbol{o}_s$, cf. (4.3), and then
            compute the output $\{\boldsymbol{y}_s^{(l)}\}$ of the $l$-th QSAL, cf. (4.4).
 7:         Repeat 4-6 $L$ times to get the output $\{\boldsymbol{y}_s^{(L)}\}$ of the $L$-th QSAL.
 8:         Average $\{\boldsymbol{y}_s^{(L)}\}$ and through a fully-connected layer to obtain the predicted value
            ${}^{(m)}\hat{y}$.
 9:         Calculate the mean squared error in (4.6) and update the parameters through the
            optimization procedure.
10:     end for
11:     if the stopping condition is met then
12:         Break.
13:     end if
14: end for
```

as well as the queries and keys for all words from the quantum device. After that, we need to compute S^2 self-attention coefficients for all S^2 pairs of words on the classical computer. In general, QSANN takes advantage of quantum devices' efficiency in processing high-dimensional data while outsourcing some calculations to classical computers. This approach keeps the quantum circuit depth low and thus makes QSANN robust to low-level noise common in near-term quantum devices. This beneficial attribute is further verified by numerical results in the next section, where we test QSANN against noise.

In short, our QSANN first encodes words into a large quantum Hilbert space as the feature space and then projects them back to low-dimensional classical feature space by quantum measurement. Recent works have proved rigorous quantum advantages on some classification tasks by utilizing high-dimensional quantum feature space [69] and projected quantum models [95]. Thus, we expect that our QSANN might also have the potential advantage of digging out some hidden features that are classically intractable. In the following section, we carry out numerical simulations of QSANN on several data sets to evaluate its performance on binary text classification tasks.

4.3 Numerical Results

In order to demonstrate the performance of our proposed QSANN, we have conducted numerical experiments on public data sets, where the quantum part was accomplished via classical simulation. Concretely, we first exhibit the better performance of QSANN by comparing it with i) the syntactic analysis-based quantum model [78] on two simple tasks, i.e., MC and RP, ii) the *classical self-attention neural network* (CSANN) and the naive method on three public sentiment analysis data sets, i.e., Yelp, IMDb and Amazon [143]. Then we show the reasonableness of our particular quantum self-attention GPQSA via visualization of self-attention coefficients. Finally, the noisy experiments are performed to show the robustness of QSANN to noisy quantum channels. All the simulations and optimization loops are implemented via Paddle Quantum[2] on the PaddlePaddle Deep Learning Platform [126].

4.3.1 Data Sets

The two simple synthetic data sets we employed come directly from [78], which are named as MC and RP, respectively. MC contains 17 words and 130 sentences (70 train + 30 development + 30 test) with 3 or 4 words each; RP has 115 words and 105 sentences (74 train + 31 test) with 4 words in each one. The other three data sets we use are real-world data sets available at [144] as the Sentiment Labelled Sentences Data Set. These data sets consist of reviews of restaurants, movies and products selected from Yelp, IMDb and Amazon, respectively. Each of the three data sets contains 1000 sequences, where half are labeled as '0' (for negative) and the other half as '1' (for positive). And each sequence contains several to dozens of words. We randomly select 80% as training sequences and the rest 20% as test ones.

[2] https://github.com/paddlepaddle/Quantum

Data set	n	d	D_{enc}	$D_{q/k/v}$	λ	γ	LR
MC	2	6	1	1	0	0	0.008
RP	4	24	4	5	0.2	0.4	0.008
Yelp	4	12	1	1	0.2	0.2	0.008
IMDb	4	12	1	1	0.002	0.002	0.002
Amazon	4	12	1	2	0.2	0.2	0.008

Table 4.1: Overview of hyper-parameter settings. Here, 'LR' denotes learning rate, D_{enc}, D_q, D_k, D_v denote the depths of the corresponding ansatzes and $d = n(D_{enc} + 2)$.

4.3.2 Experimental Setting

In the experiments, we use a single self-attention layer for both QSANN and CSANN. As a comparison, we also perform the most straightforward method, i.e., averaging directly the embedded vectors of a sequence, followed by a fully-connected layer, which we call the 'Naive' method, on the three data sets of reviews.

In QSANN, all the encoder, query, key and value ansatzes have the same qubit number and are constructed according to Fig. 4.2, which are easily implementable on the NISQ devices. Specifically, assuming the n-qubit encoder ansatz has D_{enc} layers with $n(D_{enc}+2)$ parameters, we could just set the dimension of the input vectors as $d = n(D_{enc} + 2)$. The depths of the query, key and value ansatzes are set to the same, and are at most the polynomial size of the qubit number n. The actual hyper-parameter settings on different data sets are concluded in Table 4.1. In addition, we choose $Z_1, \ldots, Z_n$, $X_1, \ldots, X_n$, $Y_1, \ldots, Y_n$ as the Pauli observables P_j in Eq. (4.3). For example, it is just required $3n$ observables when $D_{enc} = 1$. However, if $D_{enc} > 1$, we could also choose two-qubit observables Z_{12}, Z_{23} and so on. All the ansatz parameters $\boldsymbol{\Theta}$ and weight $\boldsymbol{w}$ are initialized from a Gaussian distribution with zero mean and 0.01 standard deviation, and the bias b is initialized to zero. Here, the ansatz parameters are not initialized uniformly from $[0, 2\pi)$ is mainly due to the residual scheme applied in Eq. (4.4). During the optimization iteration, we use Adam optimizer [133]. And we repeat each experiment 9 times with different parameter initializations to collect the average accuracy and the corresponding fluctuations.

Method	MC			RP		
	# Paras	TrainAcc(%)	TestAcc(%)	# Paras	TrainAcc(%)	TestAcc(%)
DisCoCat [78]	40	83.10	79.80	168	90.60	**72.30**
QSANN	25	**100.00**	**100.00**	109	**95.35±1.95**	67.74±0.00

Table 4.2: Training accuracy and test accuracy of QSANN as well as DisCoCat on MC and RP tasks.

In CSANN, we set $d = 16$ and the classical query, key and value matrices are also initialized from a Gaussian distribution with zero mean and 0.01 standard deviation. Except for these, almost all other parameters are set the same as QSANN. These settings and initializations are the same in the naive method as well.

4.3.3 Results on MC and RP Tasks

The results on MC and RP tasks are summarized in Table 4.2. In the MC task, our method QSANN could easily achieve a 100% test accuracy while requiring only 25 parameters (18 in the query-key-value part and 7 in the fully-connected part). However, in DisCoCat, the authors use 40 parameters but get a test accuracy lower than 80%. This result strongly demonstrates the powerful ability of QSANN for binary text classification. Here, the parameters in the encoder part are not counted as they could be replaced by fixed representations such as pre-trained word embeddings. In the RP task, we get a higher training accuracy but a slightly lower test accuracy. However, we observe that both test accuracies are pretty low when compared with the training accuracy. It is mainly because there is a massive bias between the training set and the test set, i.e., more than half of the words in the test set have not appeared in the training one. Hence, the test accuracy highly depends on random guessing.

4.3.4 Results on Yelp, IMDb and Amazon Data Sets

As there are no quantum algorithms for text classification on these three data sets before, we benchmark our QSANN with the classical self-attention neural network (CSANN). The naive method is also listed for comparison. The results on Yelp, IMDb and Amazon data sets are summarized in Table 4.3. We can intuitively see that QSANN outperforms CSANN

Method	Yelp		IMDb		Amazon	
	# Paras	TestAcc (%)	# Paras	TestAcc (%)	# Paras	TestAcc (%)
Naive	17	82.78±0.78	17	79.33±0.67	17	80.39±0.61
CSANN	785	83.11±0.89	785	79.67±0.83	785	83.22±1.28
QSANN	49	**84.79±1.29**	49	**80.28±1.78**	61	**84.25±1.75**

Table 4.3: Test accuracy of QSANN compared to CSANN and the naive method on Yelp, IMDb, and Amazon data sets. The highest accuracy in each column is indicated in bold font. On all three data sets, QSANN achieves the highest accuracies among the three methods while using much fewer parameters than CSANN.

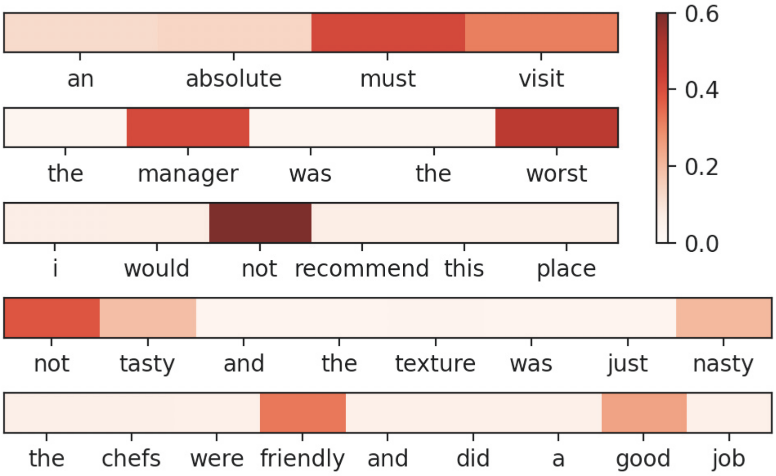
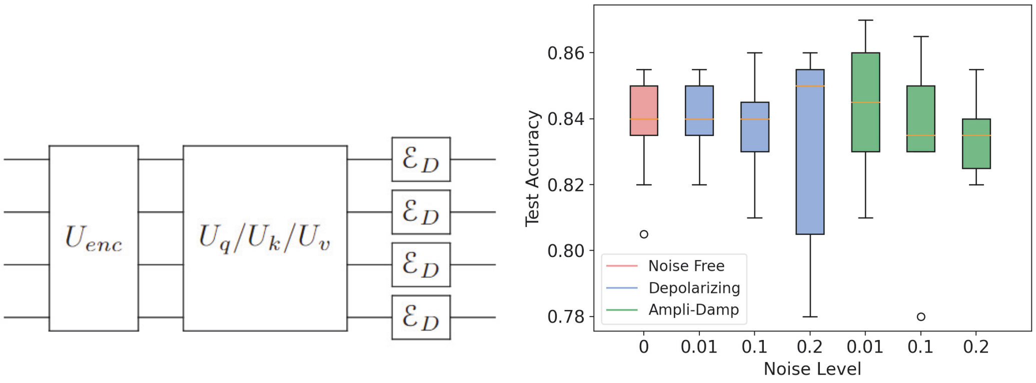


Figure 4.4: Heat maps of the averaged quantum self-attention coefficients for some selected test sequences from the Yelp data set, where a deeper color indicates a higher coefficient. Words that are more sentiment-related are generally assigned higher self-attention coefficients by our Gaussian projected quantum self-attention, implying the validity and interpretability of QSANN.

and the naive method on all three data sets. Specifically, CSANN has 785 parameters (768 in the classical query-key-value part and 17 in the fully-connected part) on all data sets. In comparison, QSANN has only 49 parameters (36 in the query-key-value part and 13 in the fully-connected part) on the Yelp and IMDb data sets and 61 parameters (48 in the query-key-value part and 13 in the fully-connected part) on the Amazon data set, improving the test accuracy by about 1% as well as saving more than 10 times the number of parameters. Therefore, QSANN could have a potential advantage for text classification.



(a) (b)

FIGURE 4.5: (a) The diagram for adding depolarizing channels in our simulated experiments. The amplitude-damping channels are added in the same way. (b) Box plots of test accuracy on Yelp data set with depolarizing and amplitude damping noises. Each box contains 9 repeated experiments. The absence of a notable decrease in accuracy implies the noise-resilience attribute of QSANN.

4.3.5 Visualization of Self-Attention Coefficient

To intuitively demonstrate the rationale of the Gaussian projected quantum self-attention, in Fig. 4.4 we visualize the averaged quantum self-attention coefficients of some selected test sequences from the Yelp data set. Concretely, for a sequence, we calculate $\frac{1}{S}\sum_{s=1}^{S}\tilde{\alpha}_{s,j}$ for $j = 1, \ldots, S$ and visualize them via a heat map, where S is the number of words in this sequence and $\tilde{\alpha}_{s,j}$ is the quantum self-attention coefficient. As shown in the figure, words with higher quantum self-attention coefficients are indeed those that determine the emotion of a sequence, implying the power of QSANN for capturing the most relevant words in a sequence on text classification tasks.

4.3.6 Noisy Experimental Results on Yelp Data Set

Due to the limitations of the near-term quantum computers, we add experiments with noisy quantum circuits to demonstrate the robustness of QSANN on the Yelp data set. We consider the representative channels [79] such as the depolarizing channel $\mathcal{E}_D(\rho)$ and the amplitude damping channel $\mathcal{E}_{AD}(\rho)$

$$\mathcal{E}_D(\rho) \equiv (1-p)\,\rho + \frac{p}{3}\,(X\rho X + Y\rho Y + Z\rho Z) \tag{4.17}$$

$$\mathcal{E}_{AD}(\rho) \equiv E_0\rho E_0^\dagger + E_1\rho E_1^\dagger, \tag{4.18}$$

with $E_0 = |0\rangle\langle 0| + \sqrt{1-p}\,|1\rangle\langle 1|$ and $E_1 = \sqrt{p}\,|0\rangle\langle 1|$ denoting the Kraus operators, i.e.,

$$E_0 = \begin{bmatrix} 1 & 0 \\ 0 & \sqrt{1-p} \end{bmatrix}, \quad E_1 = \begin{bmatrix} 0 & \sqrt{p} \\ 0 & 0 \end{bmatrix}. \tag{4.19}$$

Here, $\rho = |\psi\rangle\langle\psi|$ for a pure quantum state $|\psi\rangle$ and p denotes the noise level. As a regular way to analyze the effect of quantum noises, we add these single-qubit noisy channels in the final circuit layer to represent the whole system's noise, which is illustrated in Fig. 4.5(a).

We take the noise level p as 0.01, 0.1, and 0.2 for these two noisy channels, respectively, and the box plots of test accuracies are depicted in Fig. 4.5(b). From the picture, we see the test accuracy of our QSANN almost does not decrease when the noise level is less than 0.1, and even when the noise level is up to 0.2, the overall test accuracy has only decreased a little, showing that QSANN is robust to these quantum noises.

4.4 Discussions

We have proposed a quantum self-attention neural network (QSANN) by introducing the self-attention mechanism to quantum neural networks. Specifically, the adopted Gaussian projected quantum self-attention exploits the exponentially large quantum Hilbert space as the quantum feature space, making QSANN have the potential advantage of mining some hidden correlations between words that are difficult to dig out classically. Numerical results show that QSANN outperforms the best-known QNLP method and a simple classical self-attention neural network for text classification on several public data sets. Moreover, using only shallow quantum circuits and Pauli measurements, QSANN can be easily implemented on near-term quantum devices and is noise-resilient, as implied by simulation results.

We believe that this attempt to combine self-attention and quantum neural networks would open up new avenues for QNLP as well as QML. As a future direction, more advanced techniques such as positional encoding and multi-head attention can be employed in quantum neural networks for generative models and other more complicated tasks.

Chapter 5

Concentration of Data Encoding in Parameterized Quantum Circuits

5.1 Introduction

In this chapter, we present our main results from the perspective of quantum divergence between the average encoded quantum state and the maximally mixed state with respect to the width and depth of the encoding PQCs. A cartoon illustration summarizing our main result is depicted in Fig. 5.1. We show that for the common PQC-based encoding strategies with a fixed width, the average encoded state is close to the maximally mixed state at an exponential speed on depth.

The main content of this chapter was published in NeurIPS 2022, titled "Concentration of Data Encoding in Parameterized Quantum Circuits" [145], and the remainder is scheduled as follows: in the remainder of this section, we introduce some related works and necessary background. In Sec. 5.2, we introduce the main results of data encoding concentration, which includes a warm-up case that consists of only R_y rotation gates, a specific case that consists of only R_y rotation and CNOT gates, and general case which consists of $U3$ gates and arbitrary entangling gates. In Sec. 5.3, we present some applications of these encoded states in quantum supervised learning, such as quantum classification and quantum state discrimination. Several numerical experiments are conducted to verify the main results

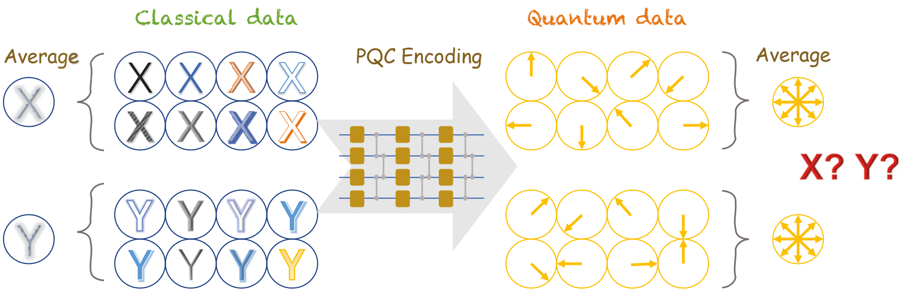


Figure 5.1: Cartoon illustrating the concentration of PQC-based data encoding. The average encoded quantum states concentrate on the maximally mixed state at an exponential rate on the encoding depth. This concentration implies the theoretical indistinguishability of the encoded quantum data.

on both synthetic and public data sets in Sec. 5.4. Lastly, in Sec. 5.5, we conclude some future discussions about data encoding.

5.1.1 Related Work

Ref. [66] derived generalization bounds of PQC-based data encoding, which mainly depends on the total number of circuit gates, while we derive quantum divergence bound that depends on the width and depth of PQCs. The works [73, 146] explored the effects of data encoding from the perspective of data re-uploading. [147] studied the robustness of data encoding for quantum classifiers. Data encoding strategies with discrete features were proposed for variational quantum classifiers [148].

5.1.2 Background

5.1.2.1 Quantum Divergence

Similar to Kullback-Leibler divergence in machine learning, we use quantum divergence to quantify the difference between quantum states or quantum data. Two widely-used quantum divergences are quantum sandwiched Rényi divergence [149, 150]

$$\widetilde{D}_\alpha(\rho\|\sigma) \equiv \frac{1}{\alpha-1}\log \operatorname{Tr}\left[\sigma^{\frac{1-\alpha}{2\alpha}}\rho\sigma^{\frac{1-\alpha}{2\alpha}}\right]^\alpha \tag{5.1}$$

and the Petz-Rényi divergence [151]

$$D_\alpha(\rho\|\sigma) \equiv \frac{1}{\alpha-1} \log \operatorname{Tr}\left[\rho^\alpha \sigma^{1-\alpha}\right], \tag{5.2}$$

where $\alpha \in (0,1)\cup(1,\infty)$ and the latter has an operational significance in quantum hypothesis testing [152–154]. In this chapter, for the purpose of analyzing quantum encoding, we focus on the Petz-Rényi divergence with $\alpha = 2$, i.e.,

$$D_2(\rho\|\sigma) = \log \operatorname{Tr}\left[\rho^2 \sigma^{-1}\right], \tag{5.3}$$

which also plays an important role in training quantum neural networks [155] as well as quantum communication [156]. Throughout this chapter, when we mention the quantum divergence, we mean the Petz-Rényi divergence D_2 if not otherwise specified; log denotes $\log_2$ if not otherwise specified.

5.1.2.2 Parameterized Quantum Circuit

In general, a parameterized quantum circuit [74] has the form $U(\boldsymbol{\theta}) = \prod_j U_j(\theta_j)V_j$, where $\boldsymbol{\theta}$ is its parameter vector, $U_j(\theta_j) = \mathrm{e}^{-i\theta_j P_j/2}$ with P_j denoting a Pauli gate, and V_j denotes a fixed gate such as Identity, CNOT and so on. In this chapter, PQCs are utilized as both data encoding strategies and quantum neural networks. Specifically, when used for data encoding, an n-qubit PQC takes a classical input vector $\boldsymbol{x}$ as its parameters and acts on an initial state $|0\rangle^{\otimes n}$ to obtain the encoded state $|\boldsymbol{x}\rangle$. Here, $|0\rangle^{\otimes n}$ is a 2^n-dimensional vector whose first element is 1 and all other elements are 0.

5.2 Main Results

5.2.1 A Warm-up Case

For quick access, we first consider one of the most straightforward PQC-based data encoding strategies, i.e., consisting of R_y rotations only, cf. Fig. 5.2. It can be viewed as a generalized angle encoding. For a classical input vector $\boldsymbol{x}$ with nD components, the output

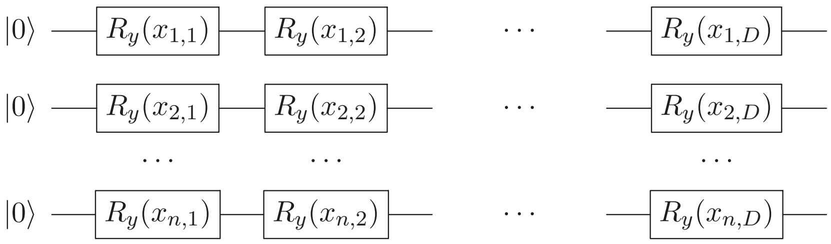


FIGURE 5.2: Circuit for the data encoding strategy with R_y rotations only.

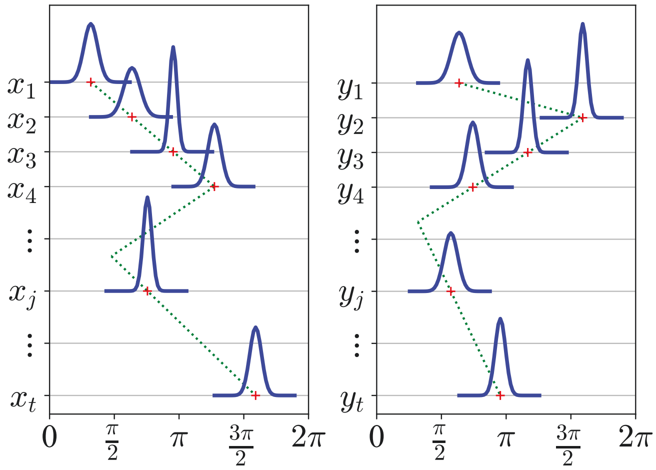


FIGURE 5.3: An example of a binary data set with two classes of t-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{y}$. Here, it is assumed that each x_j (or y_j) obeys an independent Gaussian distribution (IGD), i.e., $x_j \sim \mathcal{N}(\mu_{x,j}, \sigma^2_{x,j})$ (or $y_j \sim \mathcal{N}(\mu_{y,j}, \sigma^2_{y,j})$), where these mean values (small red cross symbols) range in $[0, 2\pi)$ and form the green dotted lines. Note that the difference between these two lines determines that they belong to different classes.

of this data encoding circuit is a pure state $|\boldsymbol{x}\rangle \in \mathbb{C}^{2^n}$ expanded in a 2^n-dimensional Hilbert space. We denote the density matrix of the output state by $\rho(\boldsymbol{x}) = |\boldsymbol{x}\rangle\langle\boldsymbol{x}|$. Assuming each element of the input vector obeys an *independent Gaussian distribution* (IGD, see Fig. 5.3 for an intuitive illustration), we have the following theorem.

Theorem 5.1. *Assume each element of an nD-dimensional vector $\boldsymbol{x}$ obeys an IGD, i.e., $x_{j,d} \sim \mathcal{N}(\mu_{j,d}, \sigma^2_{j,d})$, where $\sigma_{j,d} \geq \sigma$ for some constant σ and $1 \leq j \leq n, 1 \leq d \leq D$. If $\boldsymbol{x}$ is encoded into an n-qubit pure state $\rho(\boldsymbol{x})$ according to the circuit in Fig. 5.2, then the quantum divergence between the average encoded state $\bar{\rho}$ and the maximally mixed state $\mathbb{1}$*

is upper-bounded as

$$D_2\left(\bar{\rho}\|\mathbb{1}\right) \leq n \log\left(1 + \mathrm{e}^{-D\sigma^2}\right), \tag{5.4}$$

where $\bar{\rho}$ is defined as $\bar{\rho} \equiv \mathbb{E}\left[\rho(\boldsymbol{x})\right]$.

This theorem shows that the upper bound of the quantum divergence between $\bar{\rho}$ and $\mathbb{1}$ explicitly depends on the qubit number n and the encoding depth D under certain conditions. By approximating Eq. (5.4) as

$$D_2\left(\bar{\rho}\|\mathbb{1}\right) \leq n \log(1 + \mathrm{e}^{-D\sigma^2}) \approx \begin{cases} n\left(1 - \frac{\sigma^2}{2\ln 2}D\right), & D \in O(1) \\ n\mathrm{e}^{-D\sigma^2}, & D \in \Omega(\operatorname{poly}\log(n)) \end{cases}, \tag{5.5}$$

we easily find that for a fixed n, the upper bound decays exponentially with D growing in $\Omega(\operatorname{poly}\log(n))$. This means that the average encoded state will quickly approach the maximally mixed state with an arbitrarily small distance under reasonable depths.

To prove this theorem, we need the following lemma.

Lemma 5.2. *Assume a variable $x \sim \mathcal{N}(\mu, \sigma^2)$. Then*

$$\mathbb{E}\left[\cos(x)\right] = \mathrm{e}^{-\frac{\sigma^2}{2}} \cos(\mu); \qquad \mathbb{E}\left[\sin(x)\right] = \mathrm{e}^{-\frac{\sigma^2}{2}} \sin(\mu). \tag{5.6}$$

Proof. (Proof of Theorem 5.1.) Let $\rho(\boldsymbol{x}_j) \equiv R_y(x_{j,1} + \cdots + x_{j,D}) |0\rangle\langle 0| R_y^\dagger(x_{j,1} + \cdots + x_{j,D})$. Then

$$\rho(\boldsymbol{x}) = \rho(\boldsymbol{x}_1) \otimes \rho(\boldsymbol{x}_2) \otimes \cdots \otimes \rho(\boldsymbol{x}_n). \tag{5.7}$$

Due to the independence of each $\rho(\boldsymbol{x}_j)$, we have

$$\bar{\rho} = \mathbb{E}\left[\rho(\boldsymbol{x})\right] = \mathbb{E}\left[\rho(\boldsymbol{x}_1)\right] \otimes \mathbb{E}\left[\rho(\boldsymbol{x}_2)\right] \otimes \cdots \otimes \mathbb{E}\left[\rho(\boldsymbol{x}_n)\right]. \tag{5.8}$$

What's more, for $j = 1, \ldots, n$,

$$\mathbb{E}\left[\rho(\boldsymbol{x}_j)\right] = \frac{1}{2}\mathbb{E}\begin{bmatrix} 1+\cos\left(\sum_d x_{j,d}\right) & \sin\left(\sum_d x_{j,d}\right) \\ \sin\left(\sum_d x_{j,d}\right) & 1-\cos\left(\sum_d x_{j,d}\right) \end{bmatrix} \tag{5.9}$$

$$= \frac{1}{2}\begin{bmatrix} 1+\mathbb{E}\left[\cos\left(\sum_d x_{j,d}\right)\right] & \mathbb{E}\left[\sin\left(\sum_d x_{j,d}\right)\right] \\ \mathbb{E}\left[\sin\left(\sum_d x_{j,d}\right)\right] & 1-\mathbb{E}\left[\cos\left(\sum_d x_{j,d}\right)\right] \end{bmatrix}. \tag{5.10}$$

We know $\sum_d x_{j,d} \sim \mathcal{N}(\sum_d \mu_{j,d}, \sqrt{\sum_d \sigma_{j,d}^2})$, and combining Eq. (5.10) with Lemma 5.2, we have

$$\left|\left|\mathbb{E}\left[\rho(\boldsymbol{x}_j)\right]\right|\right|_F^2 = \frac{1}{2} + \frac{1}{2}\mathbb{E}^2\left[\cos\left(\sum_d x_{j,d}\right)\right] + \frac{1}{2}\mathbb{E}^2\left[\sin\left(\sum_d x_{j,d}\right)\right] \tag{5.11}$$

$$= \frac{1}{2} + \frac{1}{2}\left(\mathrm{e}^{-\frac{\sum_d \sigma_{j,d}^2}{2}}\cos\left(\sum_d \mu_{j,d}\right)\right)^2 + \frac{1}{2}\left(\mathrm{e}^{-\frac{\sum_d \sigma_{j,d}^2}{2}}\sin\left(\sum_d \mu_{j,d}\right)\right)^2 \tag{5.12}$$

$$= \frac{1}{2} + \frac{1}{2}\mathrm{e}^{-\sum_d \sigma_{j,d}^2} \tag{5.13}$$

$$\leq \frac{1}{2} + \frac{1}{2}\mathrm{e}^{-D\sigma^2}. \tag{5.14}$$

Finally, from Eq. (5.8), we have

$$\log \mathrm{Tr}\left(\bar{\rho}^2 \cdot \left(\frac{I}{2^n}\right)^{-1}\right) = \log\left(2^n \|\bar{\rho}\|_F^2\right) \tag{5.15}$$

$$= \log\left(2^n \prod_{j=1}^{n} \left|\left|\mathbb{E}\left[\rho(\boldsymbol{x}_j)\right]\right|\right|_F^2\right) \tag{5.16}$$

$$\leq \log\left(2^n \left(\frac{1+\mathrm{e}^{-D\sigma^2}}{2}\right)^n\right) \tag{5.17}$$

$$= n\log(1+\mathrm{e}^{-D\sigma^2}). \tag{5.18}$$

This completes the proof. □

5.2.2 A Specific Case

In this subsection, we consider a specific case where the data encoding strategy is slightly more complex. That is, some CNOT gates are inserted between R_y rotation gates. Here,

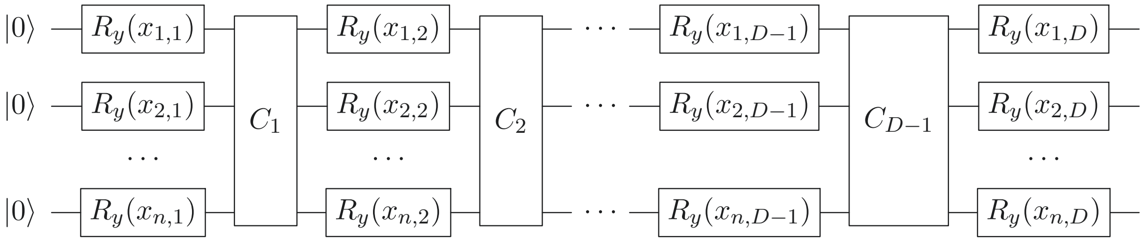


FIGURE 5.4: Circuit for the data encoding strategy with D layers of R_y rotations and $D-1$ layers of CNOTs entanglement. Here C_i denotes a group of CNOT gates.

we also assume the *IGDs* have zero means.

Theorem 5.3. *Assume each element of an nD-dimensional vector $\boldsymbol{x}$ obeys an IGD with zero mean, i.e., $x_{j,d} \sim \mathcal{N}(0, \sigma_{j,d}^2)$, where $\sigma_{j,d} \geq \sigma$ for some constant σ and $1 \leq j \leq n, 1 \leq d \leq D$. If $\boldsymbol{x}$ is encoded into an n-qubit pure quantum state $\rho(\boldsymbol{x})$ according to the encoding circuit in Fig. 5.4, then the quantum divergence between the average encoded state $\bar{\rho}$ and the maximally mixed state $\mathbb{1}$ is upper bounded as*

$$D_2\left(\bar{\rho} \| \mathbb{1}\right) \leq n \log(1 + \mathrm{e}^{-D\sigma^2}). \tag{5.19}$$

This theorem shows that the upper bound of the quantum divergence in this specific case is the same as the one in the warm-up case. Therefore, the average encoded state under this specific encoding strategy will also approach the maximally mixed state at an exponential speed.

Proof. We first consider the case of two qubits as an example. The state after the first column of R_y rotations becomes

$$\rho_1 = \begin{bmatrix} \cos^2(\frac{x_{1,1}}{2}) & \cos(\frac{x_{1,1}}{2})\sin(\frac{x_{1,1}}{2}) \\ \cos(\frac{x_{1,1}}{2})\sin(\frac{x_{1,1}}{2}) & \sin^2(\frac{x_{1,1}}{2}) \end{bmatrix} \otimes \begin{bmatrix} \cos^2(\frac{x_{2,1}}{2}) & \cos(\frac{x_{2,1}}{2})\sin(\frac{x_{2,1}}{2}) \\ \cos(\frac{x_{2,1}}{2})\sin(\frac{x_{2,1}}{2}) & \sin^2(\frac{x_{2,1}}{2}) \end{bmatrix} \tag{5.20}$$

$$= \frac{1}{4}\begin{bmatrix} 1+\cos(x_{1,1}) & \sin(x_{1,1}) \\ \sin(x_{1,1}) & 1-\cos(x_{1,1}) \end{bmatrix} \otimes \begin{bmatrix} 1+\cos(x_{2,1}) & \sin(x_{2,1}) \\ \sin(x_{2,1}) & 1-\cos(x_{2,1}) \end{bmatrix}. \tag{5.21}$$

Since $x_{1,1}$, $x_{2,1}$ have 0 means, by combining it with Lemma 5.2, we have

$$\mathbb{E}\left[\rho_1\right] = \frac{1}{4}\begin{bmatrix} 1+A_{1,1} & 0 \\ 0 & 1-A_{1,1} \end{bmatrix} \otimes \begin{bmatrix} 1+A_{2,1} & 0 \\ 0 & 1-A_{2,1} \end{bmatrix}, \tag{5.22}$$

where we define $A_{j,d} \equiv \mathrm{e}^{-\frac{\sigma_{j,d}^2}{2}}$ for writing convenience. Also, because $x_{1,1}$ and $x_{2,1}$ are independent of others, computing the expectation of ρ_1 now has no influence on the calculation of the expectation of the final state. We find that $\mathbb{E}\left[\rho_1\right]$ is diagonal and so it is when C_1 is applied.

If we record $C_1\mathbb{E}\left[\rho_1\right]C_1^\dagger$ as follows

$$C_1\mathbb{E}\left[\rho_1\right]C_1^\dagger \equiv \begin{bmatrix} E_1 & & & \\ & E_2 & & \\ & & E_3 & \\ & & & E_4 \end{bmatrix}, \tag{5.23}$$

then the state after the second column of R_y rotations becomes

$$\rho_2 = \left(R_y(x_{1,2}) \otimes R_y(x_{2,2})\right) C_1\mathbb{E}\left[\rho_1\right]C_1^\dagger \left(R_y^\dagger(x_{1,2}) \otimes R_y^\dagger(x_{2,2})\right). \tag{5.24}$$

By combining Lemma 5.2 again, we have

$$\begin{aligned}
\mathbb{E}\left[\rho_2\right] =& \frac{1}{4}\Big[\left(1+A_{1,2}\right)\left(1+A_{2,2}\right)E_1 + \left(1+A_{1,2}\right)\left(1-A_{2,2}\right)E_2 \\
&\qquad + \left(1-A_{1,2}\right)\left(1+A_{2,2}\right)E_1 + \left(1-A_{1,2}\right)\left(1-A_{2,2}\right)E_4\Big] |00\rangle\langle 00| \\
&+\frac{1}{4}\Big[\left(1+A_{1,2}\right)\left(1-A_{2,2}\right)E_1 + \left(1+A_{1,2}\right)\left(1+A_{2,2}\right)E_2 \\
&\qquad + \left(1-A_{1,2}\right)\left(1-A_{2,2}\right)E_1 + \left(1-A_{1,2}\right)\left(1+A_{2,2}\right)E_4\Big] |01\rangle\langle 01| \\
&+\frac{1}{4}\Big[\left(1-A_{1,2}\right)\left(1+A_{2,2}\right)E_1 + \left(1-A_{1,2}\right)\left(1-A_{2,2}\right)E_2 \\
&\qquad + \left(1+A_{1,2}\right)\left(1+A_{2,2}\right)E_1 + \left(1+A_{1,2}\right)\left(1-A_{2,2}\right)E_4\Big] |10\rangle\langle 10| \\
&+\frac{1}{4}\Big[\left(1-A_{1,2}\right)\left(1-A_{2,2}\right)E_1 + \left(1-A_{1,2}\right)\left(1+A_{2,2}\right)E_2 \\
&\qquad + \left(1+A_{1,2}\right)\left(1-A_{2,2}\right)E_1 + \left(1+A_{1,2}\right)\left(1+A_{2,2}\right)E_4\Big] |11\rangle\langle 11|.
\end{aligned} \tag{5.25}$$

We find that $\mathbb{E}\left[\rho_2\right]$ is still diagonal. Then we can similarly derive that $\mathbb{E}\left[\rho_3\right]$, $\mathbb{E}\left[\rho_4\right]$, ..., up to $\mathbb{E}\left[\rho_D\right]$ are all diagonal. What's more, if we think of every $\mathbb{E}\left[\rho_d\right]$ as a column vector, then we have the following relations for $2 \leq d \leq D$,

$$\mathbb{E}\left[\rho_d\right] = \left(\frac{1}{2}\begin{bmatrix} 1+A_{1,d} & 1-A_{1,d} \\ 1-A_{1,d} & 1+A_{1,d} \end{bmatrix} \otimes \frac{1}{2}\begin{bmatrix} 1+A_{2,d} & 1-A_{2,d} \\ 1-A_{2,d} & 1+A_{2,d} \end{bmatrix}\right) C_{d-1}\mathbb{E}\left[\rho_{d-1}\right]. \tag{5.26}$$

Now we focus on the matrix in Eq. (5.26), i.e.,

$$R_{A_{j,d}} \equiv \frac{1}{2}\begin{bmatrix} 1+A_{j,d} & 1-A_{j,d} \\ 1-A_{j,d} & 1+A_{j,d} \end{bmatrix}. \tag{5.27}$$

It has the spectral decomposition

$$R_{A_{j,d}} = 1 \cdot \boldsymbol{u}_0\boldsymbol{u}_0^\dagger + A_{j,d} \cdot \boldsymbol{u}_1\boldsymbol{u}_1^\dagger = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}\begin{bmatrix} 1 & \\ & A_{j,d} \end{bmatrix}\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \tag{5.28}$$

where $\boldsymbol{u}_0 \equiv \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 1 \end{bmatrix}$ and $\boldsymbol{u}_1 \equiv \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ -1 \end{bmatrix}$. Then

$$\begin{aligned} R_{A_{1,d}} \otimes R_{A_{2,d}} &= \left(1 \cdot \boldsymbol{u}_0\boldsymbol{u}_0^\dagger + A_{1,d} \cdot \boldsymbol{u}_1\boldsymbol{u}_1^\dagger\right) \otimes \left(1 \cdot \boldsymbol{u}_0\boldsymbol{u}_0^\dagger + A_{2,d} \cdot \boldsymbol{u}_1\boldsymbol{u}_1^\dagger\right) & (5.29) \\ &= 1 \cdot \boldsymbol{u}_{00}\boldsymbol{u}_{00}^\dagger + A_{2,d} \cdot \boldsymbol{u}_{01}\boldsymbol{u}_{01}^\dagger + A_{1,d} \cdot \boldsymbol{u}_{10}\boldsymbol{u}_{10}^\dagger + A_{1,d}A_{2,d} \cdot \boldsymbol{u}_{11}\boldsymbol{u}_{11}^\dagger, & (5.30) \end{aligned}$$

where $\boldsymbol{u}_{00} \equiv \boldsymbol{u}_0 \otimes \boldsymbol{u}_0$ and other terms are similarly defined.

From Eq. (5.22), we know $\mathbb{E}\left[\rho_1\right]$ can be decomposed as

$$\mathbb{E}\left[\rho_1\right] = \frac{1}{2}\left(1 \cdot \boldsymbol{u}_{00} + A_{2,1} \cdot \boldsymbol{u}_{01} + A_{1,1} \cdot \boldsymbol{u}_{10} + A_{1,1}A_{2,1} \cdot \boldsymbol{u}_{11}\right), \tag{5.31}$$

and acting C_1 means performing a permutation of the coefficients except the first one "1". For example, if C_1 is just the CNOT gate, then

$$C_1\mathbb{E}\left[\rho_1\right] = \frac{1}{2}\left(1 \cdot \boldsymbol{u}_{00} + A_{1,1}A_{2,1} \cdot \boldsymbol{u}_{01} + A_{1,1} \cdot \boldsymbol{u}_{10} + A_{2,1} \cdot \boldsymbol{u}_{11}\right). \tag{5.32}$$

Next let us consider the D sequences

$$\left\{\alpha_1^{(d)} \equiv A_{2,d}, \quad \alpha_2^{(d)} \equiv A_{1,d}, \quad \alpha_3^{(d)} \equiv A_{1,d}A_{2,d}\right\}_{d=1}^{D} \tag{5.33}$$

and another D sequences $\{p_d\}_{d=1}^{D}$ with each p_d denoting a permutation of $\{1,2,3\}$. For example, p_1 may represent $\{p_{1,1} \equiv 2, p_{1,2} \equiv 3, p_{1,3} \equiv 1\}$. Then we rewrite each $\mathbb{E}[\rho_d]$ by using these sequences and the relationships in Eq. (5.26),

$$\mathbb{E}[\rho_1] = \frac{1}{2}\left(1 \cdot \boldsymbol{u}_{00} + \alpha_{p_{1,1}}^{(1)} \cdot \boldsymbol{u}_{01} + \alpha_{p_{1,2}}^{(1)} \cdot \boldsymbol{u}_{10} + \alpha_{p_{1,3}}^{(1)} \cdot \boldsymbol{u}_{11}\right) \tag{5.34}$$

$$\mathbb{E}[\rho_2] = \frac{1}{2}\left(1 \cdot \boldsymbol{u}_{00} + \alpha_{p_{1,1}}^{(1)}\alpha_{p_{2,1}}^{(2)} \cdot \boldsymbol{u}_{01} + \alpha_{p_{1,2}}^{(1)}\alpha_{p_{2,2}}^{(2)} \cdot \boldsymbol{u}_{10} + \alpha_{p_{1,3}}^{(1)}\alpha_{p_{2,3}}^{(2)} \cdot \boldsymbol{u}_{11}\right) \tag{5.35}$$

$$\cdots\cdots$$

$$\mathbb{E}[\rho_D] = \frac{1}{2}\left(1 \cdot \boldsymbol{u}_{00} + \prod_{d=1}^{D}\alpha_{p_{d,1}}^{(d)} \cdot \boldsymbol{u}_{01} + \prod_{d=1}^{D}\alpha_{p_{d,2}}^{(d)} \cdot \boldsymbol{u}_{10} + \prod_{d=1}^{D}\alpha_{p_{d,3}}^{(d)} \cdot \boldsymbol{u}_{11}\right). \tag{5.36}$$

Further from Eq. (5.36), we have

$$\begin{aligned}\left|\left|\mathbb{E}[\rho_D]\right|\right|_F^2 &= \frac{1}{4}\left[1 + \prod_{d=1}^{D}\left(\alpha_{p_{d,1}}^{(d)}\right)^2 + \prod_{d=1}^{D}\left(\alpha_{p_{d,2}}^{(d)}\right)^2 + \prod_{d=1}^{D}\left(\alpha_{p_{d,3}}^{(d)}\right)^2\right] &(5.37)\\ &\leq \frac{1}{4}\left[1 + \prod_{d=1}^{D}\left(\tilde{\alpha}_{p_{d,1}}^{(d)}\right)^2 + \prod_{d=1}^{D}\left(\tilde{\alpha}_{p_{d,2}}^{(d)}\right)^2 + \prod_{d=1}^{D}\left(\tilde{\alpha}_{p_{d,3}}^{(d)}\right)^2\right], &(5.38)\end{aligned}$$

where $\left\{\tilde{\alpha}_1^{(d)} \equiv \mathrm{e}^{-\frac{\sigma^2}{2}}, \tilde{\alpha}_2^{(d)} \equiv \mathrm{e}^{-\frac{\sigma^2}{2}}, \tilde{\alpha}_3^{(d)} \equiv \mathrm{e}^{-\sigma^2}\right\}_{d=1}^{D}$, and the inequality comes from $\sigma_{j,d} \geq \sigma$. Furthermore, from the rearrangement inequalities for multiple sequences [157], we know similarly ordered sequences provide the largest sum of products. Hence,

$$\begin{aligned}\left|\left|\mathbb{E}[\rho_D]\right|\right|_F^2 &\leq \frac{1}{4}\left[1 + \prod_{d=1}^{D}\left(\tilde{\alpha}_1^{(d)}\right)^2 + \prod_{d=1}^{D}\left(\tilde{\alpha}_2^{(d)}\right)^2 + \prod_{d=1}^{D}\left(\tilde{\alpha}_3^{(d)}\right)^2\right] &(5.39)\\ &= \frac{1}{4}\left[1 + \prod_{d=1}^{D}\mathrm{e}^{-\sigma^2} + \prod_{d=1}^{D}\mathrm{e}^{-\sigma^2} + \prod_{d=1}^{D}\mathrm{e}^{-2\sigma^2}\right] &(5.40)\\ &= \frac{1}{4}\left(1 + \mathrm{e}^{-D\sigma^2}\right)^2. &(5.41)\end{aligned}$$

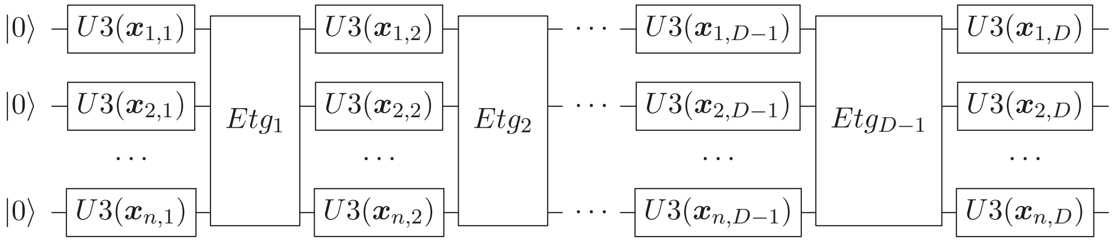

$|0\rangle$
$|0\rangle$
$|0\rangle$
$U3(\boldsymbol{x}_{1,1})$
$U3(\boldsymbol{x}_{2,1})$
$U3(\boldsymbol{x}_{n,1})$
Etg_1
$U3(\boldsymbol{x}_{1,2})$
$U3(\boldsymbol{x}_{2,2})$
$U3(\boldsymbol{x}_{n,2})$
Etg_2
$U3(\boldsymbol{x}_{1,D-1})$
$U3(\boldsymbol{x}_{2,D-1})$
$U3(\boldsymbol{x}_{n,D-1})$
Etg_{D-1}
$U3(\boldsymbol{x}_{1,D})$
$U3(\boldsymbol{x}_{2,D})$
$U3(\boldsymbol{x}_{n,D})$

FIGURE 5.5: Circuit for the data encoding strategy with D layers of $U3$ gates and $D-1$ layers of entanglements. Here, each $\boldsymbol{x}_{j,d}$ represents three elements $x_{j,d,1}, x_{j,d,2}, x_{j,d,3}$, and each Etg_i denotes an arbitrary group of entangled two-qubit gates, such as CNOT or CZ, where $1 \leq j \leq n, 1 \leq d \leq D, 1 \leq i \leq D-1$.

Finally, we have

$$\log \operatorname{Tr}\left(\bar{\rho}^2 \cdot \left(\frac{I}{2^2}\right)^{-1}\right) = \log\left(2^2 \cdot \left|\left|\mathbb{E}\left[\rho_D\right]\right|\right|_F^2\right) \tag{5.42}$$

$$\leq 2\log\left(1 + \mathrm{e}^{-D\sigma^2}\right). \tag{5.43}$$

This completes the proof for the case of two qubits. However, due to all the derivations can be directly extended to the multi-qubit case, we say this theorem is valid for the arbitrary-qubit case. □

5.2.3 General Case

Next, we consider the general PQC-based data encoding strategies shown in Fig. 5.5, where a column of $U3$ gates and a column of entangled gates spread out alternately.

Theorem 5.4. ***(Data Encoding Concentration)*** *Assume each element of a $3nD$-dimensional vector $\boldsymbol{x}$ obeys an IGD, i.e., $x_{j,d,k} \sim \mathcal{N}(\mu_{j,d,k}, \sigma^2_{j,d,k})$, where $\sigma_{j,d,k} \geq \sigma$ for some constant σ and $1 \leq j \leq n, 1 \leq d \leq D, 1 \leq k \leq 3$. If $\boldsymbol{x}$ is encoded into an n-qubit pure state $\rho(\boldsymbol{x})$ according to the circuit in Fig. 5.5, the quantum divergence between the average encoded state $\bar{\rho}$ and the maximally mixed state $\mathbb{1}$ is upper-bounded as*

$$D_2\left(\bar{\rho}\|\mathbb{1}\right) \leq \log(1 + (2^n - 1)\mathrm{e}^{-D\sigma^2}). \tag{5.44}$$

This theorem shows that, when employing general PQC-based encoding strategies, we could also have an upper bound of the quantum divergence $D_2\left(\bar{\rho}\|\mathbb{1}\right)$ which explicitly depends on n and D. By approximating the upper bound in Eq. (5.44) as follows

$$D_2\left(\bar{\rho}\|\mathbb{1}\right) \leq \log\left(1+(2^n-1)\mathrm{e}^{-D\sigma^2}\right) \approx \begin{cases} n-\frac{\sigma^2}{\ln 2}D, & D \in O(\mathrm{poly}\log(n)) \\ (2^n-1)\mathrm{e}^{-D\sigma^2}, & D \in \Omega(\mathrm{poly}(n)) \end{cases}, \tag{5.45}$$

we observe similarly that for some fixed n, the upper bound decays at an exponential speed as D grows in $\Omega(\mathrm{poly}(n))$. In addition, according to our proof analysis, even if each $U3$ gate is replaced by a $U2$ gate containing only two different kinds of Pauli rotations or even a $U1$ gate with only one proper Pauli rotation, we still get similar bound as Eq. (5.44). Therefore, we conclude that as long as D grows within a reasonable scope, the average of the quantum states encoded by a wide family of PQCs will quickly concentrate on the maximally mixed state. Unlike the warm-up and the specific cases, the proof for this theorem is quite non-straightforward due to the tricky entangled gates.

Proof. In this proof, we consider the $U3(x_{j,d,1}, x_{j,d,2}, x_{j,d,3})$ gate as $R_z(x_{j,d,3}) \cdot R_y(x_{j,d,2}) \cdot R_z(x_{j,d,1})$, which is one of the most commonly used ones. Of course, other forms of U3 gate are similar.

Outline of Proof. 1) Decomposing initial state. Firstly, we decompose the initial state according to Pauli bases; **2) Vectors transition.** Then by taking the corresponding coefficients as a row vector, we state that each action of a group of entangled gates Etg_i or a column of $U3$ gates is equivalent to multiplying the previous coefficient vector by a transition matrix; **3) Bound by singular value.** Finally, we get the upper bound by investigating the singular values of these transition matrices.

1) Decomposing initial state. The state after the first column of $U3$ gates becomes

$$\rho_1 = \frac{1}{2}\begin{bmatrix} 1+\cos(x_{1,1,2}) & \mathrm{e}^{-ix_{1,1,3}}\sin(x_{1,1,2}) \\ \mathrm{e}^{ix_{1,1,3}}\sin(x_{1,1,2}) & 1-\cos(x_{1,1,2}) \end{bmatrix} \otimes \frac{1}{2}\begin{bmatrix} 1+\cos(x_{2,1,2}) & \mathrm{e}^{-ix_{2,1,3}}\sin(x_{2,1,2}) \\ \mathrm{e}^{ix_{2,1,3}}\sin(x_{2,1,2}) & 1-\cos(x_{2,1,2}) \end{bmatrix} \otimes \cdots\cdots \otimes \frac{1}{2}\begin{bmatrix} 1+\cos(x_{n,1,2}) & \mathrm{e}^{-ix_{n,1,3}}\sin(x_{n,1,2}) \\ \mathrm{e}^{ix_{n,1,3}}\sin(x_{n,1,2}) & 1-\cos(x_{n,1,2}) \end{bmatrix}. \tag{5.46}$$

Now we define $\rho_1 \equiv \rho_{1,1} \otimes \rho_{1,2} \otimes \cdots \otimes \rho_{1,n}$, where

$$\rho_{1,j} \equiv \frac{1}{2}\begin{bmatrix} 1+\cos(x_{j,1,2}) & \mathrm{e}^{-ix_{j,1,3}}\sin(x_{j,1,2}) \\ \mathrm{e}^{ix_{j,1,3}}\sin(x_{j,1,2}) & 1-\cos(x_{j,1,2}) \end{bmatrix}. \tag{5.47}$$

And from Lemma 5.2, we have

$$\mathbb{E}\left[\rho_{1,j}\right] = \frac{1}{2}\begin{bmatrix} 1+A_{j,1,2}\cos(\mu_{j,1,2}) & A_{j,1,3}\mathrm{e}^{-i\mu_{j,1,3}}A_{j,1,2}\sin(\mu_{j,1,2}) \\ A_{j,1,3}\mathrm{e}^{i\mu_{j,1,3}}A_{j,1,2}\sin(\mu_{j,1,2}) & 1-A_{j,1,2}\cos(\mu_{j,1,2}) \end{bmatrix}, \tag{5.48}$$

where we define $A_{j,d,k} = \mathrm{e}^{-\frac{\sigma^2_{j,d,k}}{2}}$ for writing convenience. Here we note that due to all $x_{j,d,k}$'s being independent of each other, calculating the expectation of $\rho_{1,j}$ in advance does not affect the following computations. Next we decompose $\mathbb{E}\left[\rho_{1,j}\right]$ according to the Pauli bases, i.e., I, Z, X, Y,

$$\mathbb{E}\left[\rho_{1,j}\right] = \frac{1}{2}I + \frac{A_{j,1,2}\cos(\mu_{j,1,2})}{2}Z + \frac{A_{j,1,3}\cos(\mu_{j,1,3})A_{j,1,2}\sin(\mu_{j,1,2})}{2}X + \frac{A_{j,1,3}\sin(\mu_{j,1,3})A_{j,1,2}\sin(\mu_{j,1,2})}{2}Y. \tag{5.49}$$

Then from Eq. (5.46), we could derive that $\mathbb{E}\left[\rho_1\right] = \bigotimes_{j=1}^{n}\mathbb{E}\left[\rho_{1,j}\right]$ can also be decomposed in accordance with various tensor products of Pauli bases. Therefore, studying the state after the gate Etg_1 could be transferred to what performance it will be when entangled two-qubit gates act on the tensor products of Pauli bases.

2) Vectors transition. Here, we focus on the two widely employed two-qubit entangled gates CNOT and CZ, and the calculations are concluded in Table 5.1. The results in the

Pauli bases	Apply CNOT	Apply CZ
$I \otimes I$	$I \otimes I$	$I \otimes I$
$I \otimes Z$	$Z \otimes Z$	$I \otimes Z$
$I \otimes X$	$I \otimes X$	$Z \otimes X$
$I \otimes Y$	$Z \otimes Y$	$Z \otimes Y$
$Z \otimes I$	$Z \otimes I$	$Z \otimes I$
$Z \otimes Z$	$I \otimes Z$	$Z \otimes Z$
$Z \otimes X$	$Z \otimes X$	$I \otimes X$
$Z \otimes Y$	$I \otimes Y$	$I \otimes Y$
$X \otimes I$	$X \otimes X$	$X \otimes Z$
$X \otimes Z$	-$Y \otimes Y$	$X \otimes I$
$X \otimes X$	$X \otimes I$	$Y \otimes Y$
$X \otimes Y$	$Y \otimes Z$	-$Y \otimes X$
$Y \otimes I$	$Y \otimes X$	$Y \otimes Z$
$Y \otimes Z$	$X \otimes Y$	$Y \otimes I$
$Y \otimes X$	$Y \otimes I$	-$X \otimes Y$
$Y \otimes Y$	-$X \otimes Z$	$X \otimes X$

TABLE 5.1: The transition table for tensor products of Pauli bases when applying CNOT or CZ gates.

table show that the transitions are closed for tensor products of Pauli bases. Here we note that other entangled two-qubit gates will have a similar effect.

Next, we consider the effects of applying the gate $U3(x_1, x_2, x_3) = R_z(x_3)R_y(x_2)R_z(x_1)$ to four Pauli matrices. And the results of the calculations are as follows:

$$\mathbb{E}\left[U3 \cdot I \cdot U3^\dagger\right] = I \tag{5.50}$$

$$\mathbb{E}\left[U3 \cdot Z \cdot U3^\dagger\right] = p_{zz}Z + p_{zx}X + p_{zy}Y \tag{5.51}$$

$$\mathbb{E}\left[U3 \cdot X \cdot U3^\dagger\right] = p_{xz}Z + p_{xx}X + p_{xy}Y \tag{5.52}$$

$$\mathbb{E}\left[U3 \cdot Y \cdot U3^\dagger\right] = p_{yz}Z + p_{yx}X + p_{yy}Y, \tag{5.53}$$

where

$$
\begin{aligned}
p_{zz} &= A_2 \cos(\mu_2) && (5.54)\\
p_{zx} &= A_2 \sin(\mu_2) A_3 \cos(\mu_3) && (5.55)\\
p_{zy} &= A_2 \sin(\mu_2) A_3 \sin(\mu_3) && (5.56)\\
p_{xz} &= -A_2 \sin(\mu_2) A_1 \cos(\mu_1) && (5.57)\\
p_{xx} &= A_2 \cos(\mu_2) A_1 \cos(\mu_1) A_3 \cos(\mu_3) - A_1 \sin(\mu_1) A_3 \sin(\mu_3) && (5.58)\\
p_{xy} &= A_2 \cos(\mu_2) A_1 \cos(\mu_1) A_3 \sin(\mu_3) + A_1 \sin(\mu_1) A_3 \cos(\mu_3) && (5.59)\\
p_{yz} &= A_2 \sin(\mu_2) A_1 \sin(\mu_1) && (5.60)\\
p_{yx} &= -A_2 \cos(\mu_2) A_1 \sin(\mu_1) A_3 \cos(\mu_3) - A_1 \cos(\mu_1) A_3 \sin(\mu_3) && (5.61)\\
p_{yy} &= -A_2 \cos(\mu_2) A_1 \sin(\mu_1) A_3 \sin(\mu_3) + A_1 \cos(\mu_1) A_3 \cos(\mu_3). && (5.62)
\end{aligned}
$$

Here, x_k, A_k, μ_k are the abbreviations for $x_{j,d,k}, A_{j,d,k}, \mu_{j,d,k}$, respectively.

Now we record Eqs. (5.50)-(5.53) as a matrix T, which we call *transition matrix*, i.e.,

$$
T \equiv \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & p_{zz} & p_{zx} & p_{zy} \\ 0 & p_{xz} & p_{xx} & p_{xy} \\ 0 & p_{yz} & p_{yx} & p_{yy} \end{bmatrix}. \tag{5.63}
$$

By carefully calculating Eqs. (5.50)-(5.63) again, we also have

$$
T = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & A_1 \cos(\mu_1) & A_1 \sin(\mu_1) \\ 0 & 0 & -A_1 \sin(\mu_1) & A_1 \cos(\mu_1) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & A_2 \cos(\mu_2) & A_2 \sin(\mu_2) & 0 \\ 0 & -A_2 \sin(\mu_2) & A_2 \cos(\mu_2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & A_3 \cos(\mu_3) & A_3 \sin(\mu_3) \\ 0 & 0 & -A_3 \sin(\mu_3) & A_3 \cos(\mu_3) \end{bmatrix}, \tag{5.64}
$$

where the three matrices correspond to the effects of applying $R_z(x_1)$, $R_y(x_2)$ and $R_z(x_3)$, respectively.

If we further record an arbitrary input $\rho_{in} \equiv \alpha_1 I + \alpha_2 Z + \alpha_3 X + \alpha_4 Y$ as a row vector $\pi_{in} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{bmatrix}$, then applying the gate $U3(x_1, x_2, x_3)$ to ρ_{in} will result in the output $\rho_{out} \equiv \beta_1 I + \beta_2 Z + \beta_3 X + \beta_4 Y$, where

$$\pi_{out} \equiv \begin{bmatrix} \beta_1 & \beta_2 & \beta_3 & \beta_4 \end{bmatrix} = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & p_{zz} & p_{zx} & p_{zy} \\ 0 & p_{xz} & p_{xx} & p_{xy} \\ 0 & p_{yz} & p_{yx} & p_{yy} \end{bmatrix} = \pi_{in} T. \tag{5.65}$$

This is a fundamental relationship in this proof, which can be easily verified in multi-qubit and multi-depth cases. Hence, we could rewrite each $\mathbb{E}\left[\rho_d\right]$, $0 \leq d \leq D$, as follows

$$\mathbb{E}\left[\rho_0\right] \quad \longleftrightarrow \quad \pi_0 = \otimes_{j=1}^{n} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & 0 & 0 \end{bmatrix} \tag{5.66}$$

$$\mathbb{E}\left[\rho_1\right] \quad \longleftrightarrow \quad \pi_1 = \pi_0 \cdot \otimes_{j=1}^{n} T_{j,1} \cdot \widetilde{Etg_1} \tag{5.67}$$

$$\cdots$$

$$\mathbb{E}\left[\rho_d\right] \quad \longleftrightarrow \quad \pi_d = \pi_{d-1} \cdot \otimes_{j=1}^{n} T_{j,d} \cdot \widetilde{Etg_d} \tag{5.68}$$

$$\cdots$$

$$\mathbb{E}\left[\rho_{D-1}\right] \quad \longleftrightarrow \quad \pi_{D-1} = \pi_{D-2} \cdot \otimes_{j=1}^{n} T_{j,D-1} \cdot \widetilde{Etg_{D-1}} \tag{5.69}$$

$$\mathbb{E}\left[\rho_D\right] \quad \longleftrightarrow \quad \pi_D = \pi_{D-1} \cdot \otimes_{j=1}^{n} T_{j,D}, \tag{5.70}$$

where each $T_{j,d}$ represents that this transition matrix is constructed based on the gate $U3(x_{j,d,1},\ x_{j,d,2},\ x_{j,d,3})$ and each $\widetilde{Etg_i}$ means rearranging the elements of the previously multiplied row vector, which is equivalent to the effect after applying Etg_i, $1 \leq i \leq D-1$. Here, please note that we omit the possible negative sign described in Table 5.1, because in the following proof, it has no influence.

From the fact that $\operatorname{Tr}\left(P_i^2\right) = 2$, $\operatorname{Tr}\left(P_i P_j\right) = 0$, where P_i, P_j denote different Pauli matrices, and combining the relationship in Eq. (5.70), we have

$$\operatorname{Tr}\left(\mathbb{E}\left[\rho_D\right]\right)^2 = 2^n \cdot \pi_D \left(\pi_D\right)^\top . \tag{5.71}$$

What's more, we also find that every $\otimes_{j=1}^{n} T_{j,d}$ and $\widetilde{Etg_i}$ always have an element 1 in the top left corner, i.e.,

$$\otimes_{j=1}^{n} T_{j,d} \equiv \begin{bmatrix} 1 & \\ & \mathcal{T}_d \end{bmatrix}, \qquad \widetilde{Etg_i} \equiv \begin{bmatrix} 1 & \\ & \mathcal{E}_i \end{bmatrix}, \tag{5.72}$$

where $\mathcal{T}_d, \mathcal{E}_i \in \mathbb{R}^{(4^n-1)\times(4^n-1)}$ and $1 \leq d \leq D, 1 \leq i \leq D-1$. Therefore,

$$\begin{aligned}
\operatorname{Tr}\left(\mathbb{E}\left[\rho_D\right]\right)^2 &= 2^n \cdot \pi_D \left(\pi_D\right)^\top & (5.73)\\
&= 2^n \cdot \pi_0 \begin{bmatrix} 1 & \\ & \mathcal{T}_1\mathcal{E}_1\mathcal{T}_2\mathcal{E}_2\cdots\mathcal{T}_D \end{bmatrix} \begin{bmatrix} 1 & \\ & \mathcal{T}_D^\top\cdots\mathcal{E}_2^\top\mathcal{T}_2^\top\mathcal{E}_1^\top\mathcal{T}_1^\top \end{bmatrix} \left(\pi_0\right)^\top & (5.74)\\
&= 2^n \cdot \begin{bmatrix} \frac{1}{2^n} & \mathring{\pi}_0 \end{bmatrix} \begin{bmatrix} 1 & \\ & \mathcal{T}_1\mathcal{E}_1\mathcal{T}_2\mathcal{E}_2\cdots\mathcal{T}_D\mathcal{T}_D^\top\cdots\mathcal{E}_2^\top\mathcal{T}_2^\top\mathcal{E}_1^\top\mathcal{T}_1^\top \end{bmatrix} \begin{bmatrix} \frac{1}{2^n} \\ \mathring{\pi}_0^\top \end{bmatrix} & (5.75)\\
&= \frac{1}{2^n} + 2^n \cdot \mathring{\pi}_0\mathcal{T}_1\mathcal{E}_1\cdots\mathcal{T}_{D-1}\mathcal{E}_{D-1}\mathcal{T}_D\mathcal{T}_D^\top\mathcal{E}_{D-1}^\top\mathcal{T}_{D-1}^\top\cdots\mathcal{E}_1^\top\mathcal{T}_1^\top\mathring{\pi}_0^\top, & (5.76)
\end{aligned}$$

where $\mathring{\pi}_0$ means that the row vector π_0 removes the first element.

3) Bound by singular value. Next, in order to further calculate it, we need to prove first the following two Lemmas.

Lemma 5.5. *Given a Hermitian matrix $H \in \mathbb{C}^{n\times n}$ with all its eigenvalues no larger than λ, and an n-dimensional vector $\boldsymbol{x}$, then*

$$\boldsymbol{x}^\dagger H \boldsymbol{x} \leq \|\boldsymbol{x}\|_2^2 \lambda, \tag{5.77}$$

where $\|\cdot\|_2$ denotes the l_2-norm.

Proof. Assume H has the spectral decomposition

$$H = \sum_{i=1}^{n} \lambda_i \boldsymbol{u}_i \boldsymbol{u}_i^\dagger, \tag{5.78}$$

then $\boldsymbol{x}$ can be uniquely decomposed as $\boldsymbol{x} = \sum_{i=1}^{n} \alpha_i \boldsymbol{u}_i$ with $\sum_{i=1}^{n} |\alpha_i|^2 = \|\boldsymbol{x}\|_2^2$. Finally, we have

$$\boldsymbol{x}^\dagger H \boldsymbol{x} = \sum_{i=1}^{n} |\alpha_i|^2 \lambda_i \leq \sum_{i}^{n} |\alpha_i|^2 \lambda = \|\boldsymbol{x}\|_2^2 \lambda. \tag{5.79}$$

□

Lemma 5.6. *Given a Hermitian matrix $H \in \mathbb{C}^{n\times n}$ with all its eigenvalues no larger than λ, and an arbitrary matrix $Q \in \mathbb{C}^{n\times n}$ with all its singular values no larger than s, then the largest eigenvalue of $QHQ^\dagger$ is no larger than $s^2\lambda$.*

Proof. The largest eigenvalue of $QHQ^\dagger$ can be computed as $\lambda_{max} \equiv \max_{\boldsymbol{x}} \boldsymbol{x}^\dagger QHQ^\dagger \boldsymbol{x}$, where $\boldsymbol{x}$ denotes a unit vector. Assume Q has the singular value decomposition

$$Q = USV^\dagger = \sum_{i=1}^{n} s_i \boldsymbol{u}_i \boldsymbol{v}_i^\dagger = \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{u}_2 & \cdots & \boldsymbol{u}_n \end{bmatrix} \begin{bmatrix} s_1 & 0 & 0 & 0 \\ 0 & s_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & s_n \end{bmatrix} \begin{bmatrix} \boldsymbol{v}_1^\dagger \\ \boldsymbol{v}_2^\dagger \\ \vdots \\ \boldsymbol{v}_n^\dagger \end{bmatrix} \tag{5.80}$$

and $\boldsymbol{x} = \sum_{i=1}^{n} \alpha_i \boldsymbol{u}_i$ with $\sum_{i=1}^{n} |\alpha_i|^2 = 1$, then

$$\boldsymbol{x}^\dagger QHQ^\dagger \boldsymbol{x} = \boldsymbol{x}^\dagger USV^\dagger HVSU^\dagger \boldsymbol{x} = \begin{bmatrix} \alpha_1^\dagger s_1 & \alpha_2^\dagger s_2 & \cdots & \alpha_n^\dagger s_n \end{bmatrix} V^\dagger HV \begin{bmatrix} \alpha_1 s_1 \\ \alpha_2 s_2 \\ \cdots \\ \alpha_n s_n \end{bmatrix}. \tag{5.81}$$

Consider $V^\dagger HV$ as a new Hermitian matrix and $SU^\dagger \boldsymbol{x}$ as a new vector $\tilde{\boldsymbol{x}}$, then all the eigenvalues of $V^\dagger HV$ are still no larger than λ and the square of the l_2-norm of $\tilde{\boldsymbol{x}}$ is computed as

$$\|\tilde{\boldsymbol{x}}\|_2^2 = \sum_{i=1}^{n} |\alpha_i|^2 s_i^2 \leq \sum_{i=1}^{n} |\alpha_i|^2 s^2 = s^2. \tag{5.82}$$

From Lemma 5.5, we have

$$\boldsymbol{x}^\dagger Q H Q^\dagger \boldsymbol{x} \le \|\tilde{\boldsymbol{x}}\|_2^2 \lambda \le s^2 \lambda. \tag{5.83}$$

As $\boldsymbol{x}$ is arbitrary, we can obtain that $\lambda_{max} \equiv \max_{\boldsymbol{x}} \boldsymbol{x}^\dagger Q H Q^\dagger \boldsymbol{x}$ is no larger than $s^2\lambda$ as well. □

Now, let us investigate the singular values of $T_{j,d}$. From Eq. (5.63), we know it always has the trivial biggest singular value 1. The second-biggest singular value s_m can be derived from

$$s_m^2 = \max_{\boldsymbol{u}} \boldsymbol{u}^\dagger \begin{bmatrix} p_{zz} & p_{zx} & p_{zy} \\ p_{xz} & p_{xx} & p_{xy} \\ p_{yz} & p_{yx} & p_{yy} \end{bmatrix} \begin{bmatrix} p_{zz} & p_{xz} & p_{yz} \\ p_{zx} & p_{xx} & p_{yx} \\ p_{zy} & p_{xy} & p_{yy} \end{bmatrix} \boldsymbol{u}, \tag{5.84}$$

where $\boldsymbol{u} \in \mathbb{C}^3$ denotes a unit column vector. From Eq. (5.64), we derive that

$$\begin{aligned} \begin{bmatrix} p_{zz} & p_{zx} & p_{zy} \\ p_{xz} & p_{xx} & p_{xy} \\ p_{yz} & p_{yx} & p_{yy} \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & A_1\cos(\mu_1) & A_1\sin(\mu_1) \\ 0 & -A_1\sin(\mu_1) & A_1\cos(\mu_1) \end{bmatrix} \begin{bmatrix} A_2\cos(\mu_2) & A_2\sin(\mu_2) & 0 \\ -A_2\sin(\mu_2) & A_2\cos(\mu_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & A_3\cos(\mu_3) & A_3\sin(\mu_3) \\ 0 & -A_3\sin(\mu_3) & A_3\cos(\mu_3) \end{bmatrix} &(5.85)\\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & A_1\cos(\mu_1) & A_1\sin(\mu_1) \\ 0 & -A_1\sin(\mu_1) & A_1\cos(\mu_1) \end{bmatrix} \begin{bmatrix} \cos(\mu_2) & \sin(\mu_2) & 0 \\ -\sin(\mu_2) & \cos(\mu_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_2 & & \\ & A_2A_3 & 0 \\ 0 & 0 & A_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\mu_3) & \sin(\mu_3) \\ 0 & -\sin(\mu_3) & \cos(\mu_3) \end{bmatrix}, &(5.86) \end{aligned}$$

hence,

$$\begin{bmatrix} p_{zz} & p_{zx} & p_{zy} \\ p_{xz} & p_{xx} & p_{xy} \\ p_{yz} & p_{yx} & p_{yy} \end{bmatrix} \begin{bmatrix} p_{zz} & p_{xz} & p_{yz} \\ p_{zx} & p_{xx} & p_{yx} \\ p_{zy} & p_{xy} & p_{yy} \end{bmatrix} = Q \begin{bmatrix} \cos(\mu_2) & \sin(\mu_2) & 0 \\ -\sin(\mu_2) & \cos(\mu_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} A_2^2 & & \\ & (A_2A_3)^2 & 0 \\ 0 & 0 & A_3^2 \end{bmatrix} \begin{bmatrix} \cos(\mu_2) & -\sin(\mu_2) & 0 \\ \sin(\mu_2) & \cos(\mu_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} Q^\top, \tag{5.87}$$

where $Q \equiv \begin{bmatrix} 1 & 0 & 0 \\ 0 & A_1\cos(\mu_1) & A_1\sin(\mu_1) \\ 0 & -A_1\sin(\mu_1) & A_1\cos(\mu_1) \end{bmatrix}$ has the largest singular value 1.

From Lemma 5.6, we deduce that the largest eigenvalue of the matrix in Eq. (5.87) is $\max\left\{A_2^2, A_3^2\right\}$, which is no larger than $\mathrm{e}^{-\sigma^2}$. Further combining Eq. (5.84), we infer that s_m is no larger than $\mathrm{e}^{-\frac{\sigma^2}{2}}$, i.e., the second-biggest singular value of each $T_{j,d}$ is no larger than $\mathrm{e}^{-\frac{\sigma^2}{2}}$. What's more, we could derive that their tensor product $\otimes_{j=1}^n T_{j,d}$ also has the trivial largest singular value 1 and the second-largest singular value which is no larger than $\mathrm{e}^{-\frac{\sigma^2}{2}}$.

From the definition of $\mathcal{T}_d$ in Eq. (5.72), we declare that the largest singular value of each $\mathcal{T}_d$ is no larger than $\mathrm{e}^{-\frac{\sigma^2}{2}}$. Let's go back to the following formula in Eq. (5.76) to continue estimating $\operatorname{Tr}\left(\mathbb{E}\left[\rho_D\right]\right)^2$, i.e.,

$$\mathring{\pi}_0 \mathcal{T}_1 \mathcal{E}_1 \cdots \mathcal{T}_{D-1} \mathcal{E}_{D-1} \mathcal{T}_D \mathcal{T}_D^\top \mathcal{E}_{D-1}^\top \mathcal{T}_{D-1}^\top \cdots \mathcal{E}_1^\top \mathcal{T}_1^\top \mathring{\pi}_0^\top. \tag{5.88}$$

Since the largest eigenvalue of $\mathcal{T}_D \mathcal{T}_D^\top$ is no larger than $\mathrm{e}^{-\sigma^2}$, and each $\mathcal{E}_i$, defined in Eq. (5.72), is a unitary matrix, by repeatedly applying Lemma 5.6, we obtain that largest eigenvalue of $\mathcal{T}_1 \mathcal{E}_1 \cdots\ \mathcal{T}_D \mathcal{T}_D^\top \cdots\ \mathcal{E}_1^\top \mathcal{T}_1^\top$ is no larger than $\mathrm{e}^{-D\sigma^2}$. Furthermore, from Eq. (5.66) and the definition of $\mathring{\pi}_0$, we know $\mathring{\pi}_0$ has $4^n - 1$ dimensions, where $2^n - 1$ elements are $\frac{1}{2^n}$ and the others are 0. Hence, $\|\mathring{\pi}_0\|_2^2 = \frac{2^n-1}{2^{2n}}$. Combining these with Lemma 5.5, we have

$$\mathring{\pi}_0 \mathcal{T}_1 \mathcal{E}_1 \cdots \mathcal{T}_{D-1} \mathcal{E}_{D-1} \mathcal{T}_D \mathcal{T}_D^\top \mathcal{E}_{D-1}^\top \mathcal{T}_{D-1}^\top \cdots \mathcal{E}_1^\top \mathcal{T}_1^\top \mathring{\pi}_0^\top \leq \|\mathring{\pi}_0\|_2^2 \mathrm{e}^{-D\sigma^2} = \frac{2^n-1}{2^{2n}} \mathrm{e}^{-D\sigma^2}. \tag{5.89}$$

Go further, and we have, together with Eq. (5.76),

$$\begin{aligned}
\operatorname{Tr}\left(\mathbb{E}\left[\rho_D\right]\right)^2 &= \frac{1}{2^n} + 2^n \cdot \mathring{\pi}_0 \mathcal{T}_1 \mathcal{E}_1 \cdots \mathcal{T}_{D-1} \mathcal{E}_{D-1} \mathcal{T}_D \mathcal{T}_D^\top \mathcal{E}_{D-1}^\top \mathcal{T}_{D-1}^\top \cdots \mathcal{E}_1^\top \mathcal{T}_1^\top \mathring{\pi}_0^\top && (5.90)\\
&\leq \frac{1}{2^n} + 2^n \cdot \frac{2^n-1}{2^{2n}} \mathrm{e}^{-D\sigma^2} && (5.91)\\
&= \frac{1 + \left(2^n - 1\right) \mathrm{e}^{-D\sigma^2}}{2^n}. && (5.92)
\end{aligned}$$

Finally, we have

$$\begin{aligned}
\log \operatorname{Tr}\left(\bar{\rho}^2 \cdot \left(\frac{I}{2^n}\right)^{-1}\right) &= \log\left(2^n \cdot \operatorname{Tr}\left(\mathbb{E}\left[\rho_D\right]\right)^2\right) && (5.93)\\
&\leq \log\left(1 + (2^n - 1)\mathrm{e}^{-D\sigma^2}\right). && (5.94)
\end{aligned}$$

This completes the proof of Theorem 5.4.

Without stopping here, we also analyze some generalizations of Theorem 5.4.

(I) From Eqs. (5.84)-(5.87), we find that removing the matrix Q will have no influence on the final result. Hence, we directly generalize that if there is only one column of R_zR_y or R_yR_z gates in each layer, we will get the same upper bound. In fact, according to our proof method, we infer that as long as there are two different kinds of rotation gates in each encoding layer, this upper bound is valid.

(II) What is the result for the case with only R_y rotation gates in each encoding layer? Since each $\mathcal{T}_d$ has the largest singular value 1, it is not suitable for the above proof. However, through analyzing the transition rule in Table 5.1, we find that the largest singular value of $\mathcal{T}_{d-1}\mathcal{E}_{d-1}\mathcal{T}_d$ is still no larger than $\mathrm{e}^{-\frac{\sigma^2}{2}}$, which means every two encoding layers have the same effect as above with one layer. Therefore, the final upper bound can be changed to $\log\left(1+(2^n-1)\mathrm{e}^{-\lfloor\frac{D}{2}\rfloor\sigma^2}\right)$. Since it has the same trend as the original bound, it has no impact on our final analysis. □

The average encoded state $\bar{\rho}$ in Theorem 5.4 is built on infinite data samples, but in practice, we do not have infinite ones. Therefore, we provide the following helpful corollary.

Corollary 5.7. *Assume there are M classical vectors $\{\boldsymbol{x}^{(m)}\}_{m=1}^{M}$ sampled from the distributions described in Theorem 5.4 and define $\bar{\rho}_M \equiv \frac{1}{M}\sum_{m=1}^{M}\rho(\boldsymbol{x}^{(m)})$. Let H be a Hermitian matrix with its eigenvalues ranging in $[-1,1]$, then given an arbitrary $\epsilon \in (0,1)$, as long as the encoding depth $D \geq \frac{1}{\sigma^2}\left[(n+4)\ln 2 + 2\ln(1/\epsilon)\right]$, we have*

$$\Big|\mathrm{Tr}\left[H\left(\bar{\rho}_M - \mathbb{1}\right)\right]\Big| \leq \epsilon \tag{5.95}$$

with a probability of at least $1-2\mathrm{e}^{-M\epsilon^2/8}$.

This corollary implies that for a reasonable encoding depth D and a number of samples M, the practical average encoded state $\bar{\rho}_M$ will also be infinitely close to the maximally mixed state with a high probability. The proof is mainly derived from *Hoeffding's inequality* [158] and the relationships between quantum divergence and trace norm.

Proof. Let $\bar{\rho} \equiv \mathbb{E}\left[\rho\left(\boldsymbol{x}^{(m)}\right)\right]$, then we have

$$\left|\operatorname{Tr}\left[H\left(\bar{\rho}_M - \frac{I}{2^n}\right)\right]\right| = \left|\operatorname{Tr}\left[H\left(\bar{\rho}_M - \bar{\rho} + \bar{\rho} - \frac{I}{2^n}\right)\right]\right| \tag{5.96}$$

$$\leq \left|\operatorname{Tr}\left[H\left(\bar{\rho}_M - \bar{\rho}\right)\right]\right| + \left|\operatorname{Tr}\left[H\left(\bar{\rho} - \frac{I}{2^n}\right)\right]\right|, \tag{5.97}$$

where the inequality is due to triangle inequality.

Now we first consider the first term in Eq. (5.97). Since

$$\frac{1}{M}\operatorname{Tr}\left(H\rho(\boldsymbol{x}^{(1)})\right), \ldots, \frac{1}{M}\operatorname{Tr}\left(H\rho(\boldsymbol{x}^{(M)})\right) \tag{5.98}$$

are i.i.d. and $\frac{-1}{M} \leq \frac{1}{M}\operatorname{Tr}\left(H\rho(\boldsymbol{x}^{(m)})\right) \leq \frac{1}{M}$, through *Hoeffding's inequality* [158], we have

$$P\left(\left|\sum_{m=1}^{M}\frac{1}{M}\operatorname{Tr}\left(H\rho(\boldsymbol{x}^{(m)})\right) - \mathbb{E}\left[\operatorname{Tr}\left(H\rho(\boldsymbol{x}^{(m)})\right)\right]\right| \leq t\right) \geq 1 - 2\mathrm{e}^{-\frac{Mt^2}{2}}. \tag{5.99}$$

From the fact that

$$\sum_{m=1}^{M}\frac{1}{M}\operatorname{Tr}\left(H\rho(\boldsymbol{x}^{(m)})\right) = \operatorname{Tr}\left(H\bar{\rho}_M\right) \quad \text{and} \quad \mathbb{E}\left[\operatorname{Tr}\left(H\rho(\boldsymbol{x}^{(m)})\right)\right] = \operatorname{Tr}\left(H\bar{\rho}\right), \tag{5.100}$$

we obtain

$$\left|\operatorname{Tr}\left(H\bar{\rho}_M\right) - \operatorname{Tr}\left(H\bar{\rho}\right)\right| \leq \frac{\epsilon}{2} \tag{5.101}$$

with a probability of at least $1 - 2\mathrm{e}^{-\frac{M\epsilon^2}{8}}$.

Next, we consider the second term in Eq. (5.97). Since the eigenvalues of H range in $[-1, 1]$, we obtain

$$\left|\operatorname{Tr}\left(H\left(\bar{\rho} - \frac{I}{2^n}\right)\right)\right| \leq \left\|\bar{\rho} - \frac{I}{2^n}\right\|_{\mathrm{tr}} \leq 2\sqrt{1 - F\left(\bar{\rho}, \frac{I}{2^n}\right)}, \tag{5.102}$$

where $\|\cdot\|_{\mathrm{tr}}$ denotes the trace norm and the second inequality is from the *Fuchs–van de Graaf inequalities* [159], i.e., $1 - \sqrt{F(\rho, \rho')} \leq \frac{1}{2}\|\rho - \rho'\|_{\mathrm{tr}} \leq \sqrt{1 - F(\rho, \rho')}$. By combining the upper bound in Theorem 5.4 with the fact that $-\log F(\rho, \rho') \leq D_2(\rho, \rho')$ [151], we have

$$F\left(\bar{\rho}, \frac{I}{2^n}\right) \geq \frac{1}{2^{D_2\left(\bar{\rho}\| \frac{I}{2^n}\right)}} \geq \frac{1}{1+(2^n-1)\mathrm{e}^{-D\sigma^2}} \tag{5.103}$$

$$\geq \frac{1}{1+\frac{(2^n-1)\epsilon^2}{2^{n+4}}} \tag{5.104}$$

$$\geq \frac{1}{1+\frac{2^n\epsilon^2}{16\cdot 2^n}} \tag{5.105}$$

$$= \frac{16}{16+\epsilon^2}, \tag{5.106}$$

where in Eq. (5.104) we use the condition $D \geq \frac{1}{\sigma^2}\left[(n+4)\ln 2 + 2\ln\left(1/\epsilon\right)\right]$. By inserting Eq. (5.106) into Eq. (5.102), we can get

$$\left|\operatorname{Tr}\left(H\left(\bar{\rho}-\frac{I}{2^n}\right)\right)\right| \leq 2\sqrt{1-\frac{16}{16+\epsilon^2}} = \frac{2\epsilon}{\sqrt{16+\epsilon^2}} \leq \frac{\epsilon}{2}. \tag{5.107}$$

Bringing Eqs. (5.101) and (5.107) into Eq. (5.97), we complete the proof of Corollary 5.7. □

5.3 Applications in Quantum Supervised Learning

In this section, we show that the concentrated quantum states encoded by the above PQC-based data encoding strategies will severely limit the performances of quantum supervised learning tasks. To this end, we use the following necessary definition.

Definition 5.8. (Data Set) The K-class data set $\mathcal{D} \equiv \{(\boldsymbol{x}^{(m)}, \boldsymbol{y}^{(m)})\}_{m=1}^{KM} \subset \mathbb{R}^{3nD} \times \mathbb{R}^K$ totally has KM data samples, including M samples in each category. Here, suppose elements in the same entry of all input vectors from the same category are sampled from the same IGD with a variance of at least σ^2 and each $\boldsymbol{x}^{(m)}$ is encoded into the corresponding pure state $\rho(\boldsymbol{x}^{(m)})$ according to the circuit in Fig. 5.5 with n qubits and D layers of $U3$ gates. The label $\boldsymbol{y}^{(m)}$ is a one-hot vector that indicates which of the K classes $\boldsymbol{x}^{(m)}$ belongs to.

5.3.1 Quantum Classification

Quantum classification, as one of the most significant branches in quantum machine learning, is widely studied nowadays. More details about the introduction of quantum classifiers are referred to Subsec. 2.4.2. In general, a quantum classifier aims to learn a map from input to label by optimizing a loss function constructed through QNNs to predict the label of an unseen input as accurately as possible. Now, we demonstrate the performance of a quantum classifier on the data set $\mathcal{D}$ defined in Def. 5.8.

In this chapter, the loss function is defined from the cross-entropy loss with softmax function [16]:

$$L\left(\boldsymbol{\theta};\mathcal{D}\right)\equiv\frac{1}{KM}\sum_{m=1}^{KM}L^{(m)}\quad\text{with}\quad L^{(m)}\left(\boldsymbol{\theta};(\boldsymbol{x}^{(m)},\boldsymbol{y}^{(m)})\right)\equiv-\sum_{k=1}^{K}y_k^{(m)}\ln\frac{\mathrm{e}^{h_k}}{\sum_{j=1}^{K}\mathrm{e}^{h_j}},\tag{5.108}$$

where $y_k^{(m)}$ denotes the k-th element of the label $\boldsymbol{y}^{(m)}$ and

$$h_k\left(\boldsymbol{x}^{(m)},\boldsymbol{\theta}\right)=\operatorname{Tr}\left[H_kU(\boldsymbol{\theta})\rho(\boldsymbol{x}^{(m)})U^\dagger(\boldsymbol{\theta})\right],\tag{5.109}$$

which means the Hermitian operator H_k is finally measured after the quantum neural network $U(\boldsymbol{\theta})$. Here, each H_k is chosen from tensor products of various Pauli matrices, such as $Z\otimes I$, $X\otimes Y\otimes Z$ and so on. By minimizing the loss function with a gradient descent method, we could obtain the final trained model $U(\boldsymbol{\theta}^*)$ with the optimal or sub-optimal parameters $\boldsymbol{\theta}^*$. After that, when provided a new input quantum state $\rho(\boldsymbol{x}')$, we compute each h_k' with parameters $\boldsymbol{\theta}^*$ according to Eq. (5.109), and the index of the largest h_k' is exactly our designated label.

However, all these graceful expectations can only be established on gradients with relatively large absolute values. On the contrary, gradients with significantly small absolute values will cause a severe training problem, for example, the barren plateau issue [44]. Therefore, in the following, we investigate the partial gradient of the cost defined in Eq. (5.108) with regard to its parameters. The results are exhibited in Proposition 5.9.

Proposition 5.9. *Consider a K-classification task with the data set $\mathcal{D}$ defined in Def. 5.8. If the encoding depth $D \geq \frac{1}{\sigma^2}\left[(n+4)\ln 2 + 2\ln(1/\epsilon)\right]$ for some $\epsilon \in (0,1)$, then the partial gradient of the loss function defined in Eq.* (5.108) *with respect to each parameter θ_i of the employed QNN is bounded as*

$$\left|\frac{\partial L\left(\boldsymbol{\theta};\mathcal{D}\right)}{\partial \theta_i}\right| \leq K\epsilon \tag{5.110}$$

with a probability of at least $1 - 2\mathrm{e}^{-M\epsilon^2/8}$.

From this proposition, we observe that no matter what QNN structures are selected, the absolute gradient value can be arbitrarily small with a very high probability for the above data set $\mathcal{D}$, provided that the encoding depth D and the number of data samples M are sufficiently large. This vanishing of the gradients will severely restrict the trainability of QNNs. Moreover, if before training $U(\boldsymbol{\theta})$ is initialized to satisfy certain randomness, such as unitary 2-design [123], then each h_k in Eq. (5.109) will concentrate on 0 with a high probability, thus the loss in Eq. (5.108) will concentrate on $\ln K$. This concentration of loss is also verified through numerical simulations, as presented in Sec. 5.4. This phenomenon, together with Proposition 5.9, implies that large encoding depth will significantly hinder the training of a quantum classifier and probably lead to poor classification accuracy.

Proof. From the chain rule, we know

$$\frac{\partial L\left(\boldsymbol{\theta};\mathcal{D}\right)}{\partial \theta_i} = \frac{1}{KM}\sum_{m=1}^{KM}\frac{\partial L^{(m)}}{\partial \theta_i} = \frac{1}{KM}\sum_{m=1}^{KM}\sum_{l=1}^{K}\frac{\partial L^{(m)}}{\partial h_l}\frac{\partial h_l}{\partial \theta_i}. \tag{5.111}$$

We first calculate $\frac{\partial L^{(m)}}{\partial h_l}$ as follows

$$\frac{\partial L^{(m)}}{\partial h_l} = \frac{\partial \sum_{k=1}^{K} y_k^{(m)}\left(\ln\sum_{j=1}^{K}\mathrm{e}^{h_j} - h_k\right)}{\partial h_l} = \begin{cases}\sum_{k=1}^{K} y_k^{(m)}\frac{\mathrm{e}^{h_l}}{\sum_{j=1}^{K}\mathrm{e}^{h_j}}, & l \neq k\\ \sum_{k=1}^{K} y_k^{(m)}\left(\frac{\mathrm{e}^{h_l}}{\sum_{j=1}^{K}\mathrm{e}^{h_j}} - 1\right), & l = k\end{cases} \tag{5.112}$$

Since $\mathrm{e}^{h_l}/\sum_{j=1}^{K}\mathrm{e}^{h_j} \in (0,1)$ and $\boldsymbol{y}^{(m)}$ is one-hot, we can get its upper bound as $|\frac{\partial L^{(m)}}{\partial h_l}| \leq 1$. Next, from the parameter-shift rule [98], we calculate $\frac{\partial h_l}{\partial \theta_i}$ as follows

$$\frac{\partial h_l}{\partial \theta_i} = \frac{1}{2}\left[\operatorname{Tr}\left(H_l U(\boldsymbol{\theta}_{+\frac{\pi}{2}})\rho(\boldsymbol{x}^{(m)})U^\dagger(\boldsymbol{\theta}_{+\frac{\pi}{2}})\right) - \operatorname{Tr}\left(H_l U(\boldsymbol{\theta}_{-\frac{\pi}{2}})\rho(\boldsymbol{x}^{(m)})U^\dagger(\boldsymbol{\theta}_{-\frac{\pi}{2}})\right)\right], \quad (5.113)$$

where $\boldsymbol{\theta}_{+\frac{\pi}{2}}$ means adding $\frac{\pi}{2}$ to θ_i and keeping the others unchanged, and $\boldsymbol{\theta}_{-\frac{\pi}{2}}$ is similarly defined. If we define

$$\tilde{H}_l \equiv \frac{1}{2}\left[U^\dagger(\boldsymbol{\theta}_{+\frac{\pi}{2}})H_l U(\boldsymbol{\theta}_{+\frac{\pi}{2}}) - U^\dagger(\boldsymbol{\theta}_{-\frac{\pi}{2}})H_l U(\boldsymbol{\theta}_{-\frac{\pi}{2}})\right], \quad (5.114)$$

then together with Eqs. (5.111)-(5.113), we could bound the gradient as

$$\begin{aligned}
\left|\frac{\partial L(\boldsymbol{\theta};\mathcal{D})}{\partial \theta_i}\right| \leq \left|\frac{1}{KM}\sum_{m=1}^{KM}\sum_{l=1}^{K}\frac{\partial h_l}{\partial \theta_i}\right| &= \left|\frac{1}{KM}\sum_{m=1}^{KM}\sum_{l=1}^{K}\operatorname{Tr}\left(\tilde{H}_l\rho(\boldsymbol{x}^{(m)})\right)\right| && (5.115)\\
&\leq \sum_{l=1}^{K}\left|\frac{1}{KM}\sum_{m=1}^{KM}\operatorname{Tr}\left(\tilde{H}_l\rho(\boldsymbol{x}^{(m)})\right)\right| && (5.116)\\
&= \sum_{l=1}^{K}\left|\frac{1}{KM}\sum_{m=1}^{KM}\sum_{k=1}^{K}y_k^{(m)}\operatorname{Tr}\left(\tilde{H}_l\rho(\boldsymbol{x}^{(m)})\right)\right| && (5.117)\\
&\leq \sum_{l=1}^{K}\frac{1}{K}\sum_{k=1}^{K}\left|\frac{1}{M}\sum_{m=1}^{KM}y_k^{(m)}\operatorname{Tr}\left(\tilde{H}_l\rho(\boldsymbol{x}^{(m)})\right)\right| && (5.118)\\
&\leq \sum_{l=1}^{K}\frac{1}{K}\sum_{k=1}^{K}\epsilon. && (5.119)
\end{aligned}$$

Here, it could be easily verified that the eigenvalues of $\tilde{H}_l$ (defined in Eq. (5.114)) range in $[-1,1]$ and $\operatorname{Tr}(\tilde{H}_l) = 0$. Then from Corollary 5.7, we could bound Eq. (5.118) as Eq. (5.119), i.e., for any $\epsilon \in (0,1)$, provided that the encoding depth $D \geq \frac{1}{\sigma^2}[(n+4)\ln 2 + 2\ln(1/\epsilon)]$, we have $|\frac{\partial L(\boldsymbol{\theta};\mathcal{D})}{\partial \theta_i}| \leq K\epsilon$ with a probability of at least $1 - 2\mathrm{e}^{-M\epsilon^2/8}$. This completes the proof of Proposition 5.9. □

5.3.2 Quantum State Discrimination

Quantum state discrimination [106] is a central information-theoretic task and finds applications in various topics such as quantum cryptography [160], quantum error mitigation [161], and quantum data hiding [162]. It aims to distinguish quantum states using a positive operator-valued measure (POVM), a set of positive semi-definite operators that sum to the identity operator. Here, we have to seriously note that in quantum state discrimination, we can only measure each quantum state once, instead of measuring repeatedly and calculating the expectations as shown in quantum classification.

In general, perfect discrimination (i.e., a perfect POVM) can not be achieved if quantum states are non-orthogonal. A natural alternative option is by adopting some metrics such as the success probability so that the optimal POVM could be obtained via various kinds of optimization ways, e.g., Helstrom bound [163] and semi-definite programming (SDP) [164]. Recently, researchers also try to train QNNs as substitutions for optimal POVMs [107, 165].

Next, we demonstrate the impact of the encoded quantum states from the data set $\mathcal{D}$ defined in Def. 5.8 on quantum state discrimination. Our goal is to obtain the maximum success probability p_{succ} by maximizing the success probability over all POVMs with K operators:

$$p_{\text{succ}} \equiv \max_{\{\Pi_k\}_k} \frac{1}{K} \sum_{k=1}^{K} \operatorname{Tr}\left[\Pi_k \bar{\rho}_{k,M}\right] \quad \text{with} \quad \bar{\rho}_{k,M} \equiv \frac{1}{M} \sum_{m=1}^{KM} y_k^{(m)} \rho(\boldsymbol{x}^{(m)}), \tag{5.120}$$

where $y_k^{(m)}$ denotes the k-th element of the label $\boldsymbol{y}^{(m)}$ and $\{\Pi_k\}_{k=1}^{K}$ denotes a POVM, which satisfies $\sum_{k=1}^{K} \Pi_k = I$.

Proposition 5.10. *Consider a K-class discrimination task with the data set $\mathcal{D}$ defined in Def. 5.8. If the encoding depth $D \geq \frac{1}{\sigma^2}\left[(n+4)\ln 2 + 2\ln(1/\epsilon)\right]$ for a given $\epsilon \in (0,1)$, then with a probability of at least $1 - 2\mathrm{e}^{-M\epsilon^2/8}$, the maximum success probability p_{succ} is bounded as*

$$p_{\text{succ}} \leq 1/K + \epsilon. \tag{5.121}$$

This proposition implies that as long as the encoding depth D and the data numbers M are large enough, the optimal success probability p_{succ} could be arbitrarily close to $\frac{1}{K}$ with a remarkably high probability for the data set $\mathcal{D}$. This nearly blind-guessing success probability shows that the concentration of the encoded quantum states in the data set

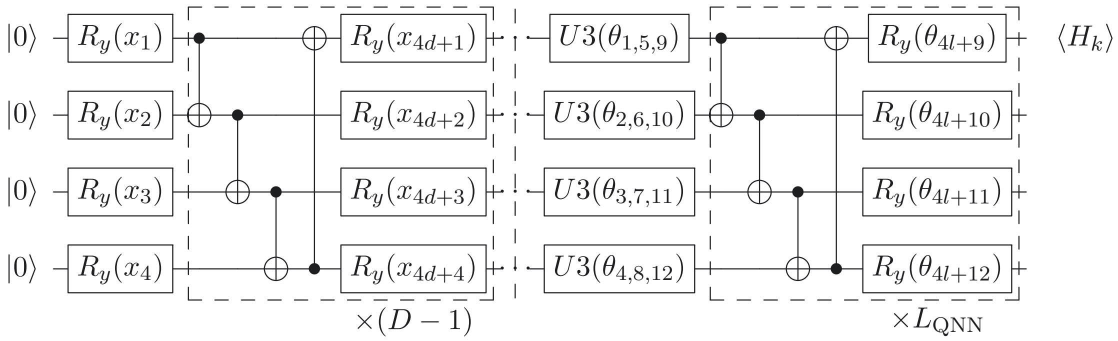


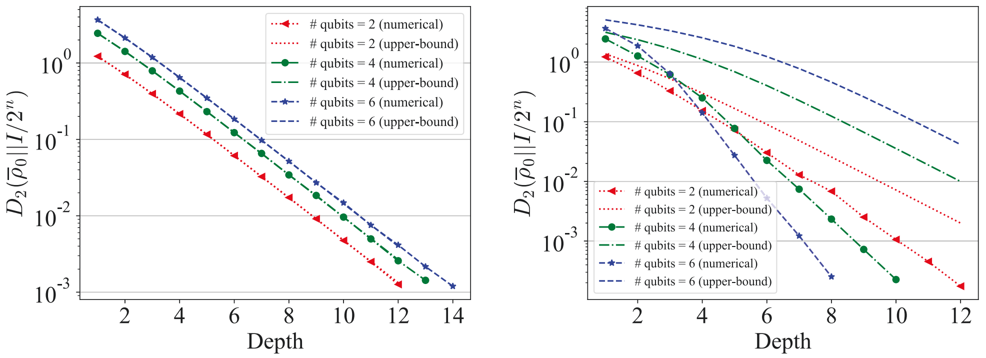
FIGURE 5.6: Circuits for data encoding (before the barrier line) and quantum neural network (after the barrier line) in the 4-qubit case. Here the input $\boldsymbol{x} \in \mathbb{R}^{4D}$ and $d \in [1, D-1]$. The QNN totally has $L_{\text{QNN}} + 1$ layers with parameters $\boldsymbol{\theta} \in \mathbb{R}^{4L_{\text{QNN}}+12}$, where the first layer $U3$ gates consists of 12 parameters. After QNN, there are K expectations $\{\langle H_k\rangle\}_{k=1}^{K}$ for K-classification tasks.

$\mathcal{D}$ will lead to the failure of state discrimination via POVM. As POVMs are the most general kind of measurements one can implement to extract classical information from quantum systems [79], we conclude that the above different classes of encoded states are indistinguishable from the perspective of quantum information. The proof of Proposition 5.10 could be derived straightforwardly by combining Eq. (5.120) with Corollary 5.7.

5.4 Numerical Experiments

Previous sections demonstrated that the average encoded state will concentrate on the maximally mixed state under PQC-based data encoding strategies with large depth. These encoded states theoretically cannot be utilized to train QNNs or distinguished by POVMs. In this section, we verify these results on both synthetic and public data sets by choosing a commonly employed strongly entangling circuit [60], which helps to understand the concentration rate intuitively for realistic encoding circuits. All the simulations and optimization loop are implemented via Paddle Quantum[1] on the PaddlePaddle Deep Learning Platform [126].

[1] https://github.com/paddlepaddle/Quantum



(a) Encoding Strategy in Fig. 5.2

(b) Encoding Strategy in Fig. 5.6

FIGURE 5.7: Exponential decay of quantum divergence $D_2(\bar{\rho}_0||\mathbb{1})$ vs. encoding depth under different qubit cases for the synthetic data set. Here, there are one million data samples for calculating average encoded state $\bar{\rho}_0$ for class 0 at each point in numerical lines. And the upper-bounds come from (a) Theorem 5.1, (b) Theorem 5.4, respectively.

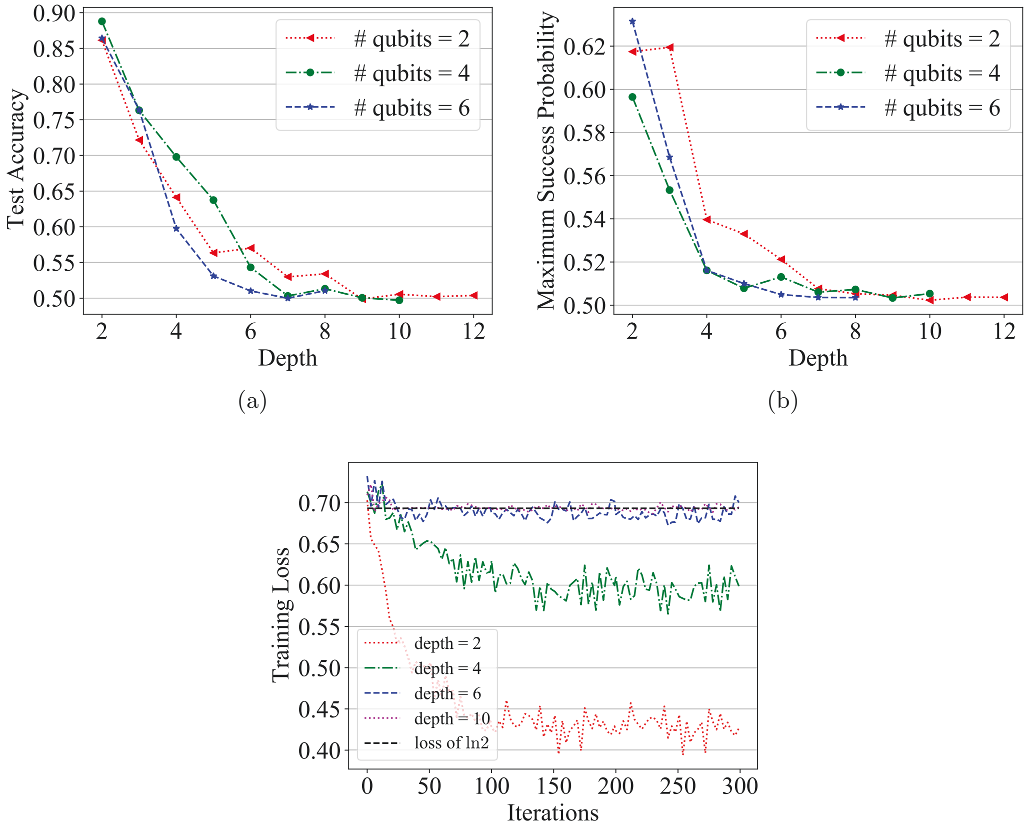
5.4.1 On Synthetic Data Set

5.4.1.1 Data Set

The synthetic two-class data set $\{(\boldsymbol{x}^{(m)}, \boldsymbol{y}^{(m)})\}_{m=1}^{M}$ is generated following the distributions depicted in Fig. 5.3, where each $x_j^{(m)} \sim \mathcal{N}(\mu_j, \sigma_j^2)$ for $1 \leq j \leq t$ and $\boldsymbol{y}^{(m)}$ denotes a one-hot label. Here, we assume all means come from two lines, i.e., $\mu_j = \frac{2\pi}{16}(j-1) \bmod 2\pi$ for class 0 and $\mu_j = \frac{2\pi}{16}(16-j) \bmod 2\pi$ for class 1, and all σ_j's are set as 0.8. Note that the same variance is selected for both classes to facilitate the demonstration of the experiment. Other choices of σ_j's would have similar effects.

5.4.1.2 Results

We first verify our two main upper bounds given in Theorems 5.1 and 5.4 by encoding the nD-dimensional inputs that belong to the same class into n-qubit quantum states with D encoding depths under the encoding strategies illustrated in Figs. 5.2 and 5.6, respectively. Here, n is set as $2, 4, 6$ and $D \in [1, 14]$. The results are displayed in Fig. 5.7, from which we can intuitively see that the divergences decrease exponentially on depth. Specifically, from Fig. 5.7(a), we know the upper bound in Theorem 5.2 is tight, which is also easily verified from our proof. From Fig. 5.7(b), we learn that the upper bound in Theorem 5.4 is quite



(a)

(b)

(c)

FIGURE 5.8: Numerical results for synthetic data sets under the encoding strategy in Fig. 5.6. In all qubit cases, (a) the test accuracy of QNN (or (b) the maximum success probability of POVM) will eventually decay to 50% or so with the depth growing; (c) In the 4 qubit case, for instance, the training losses of QNN do not decrease and stay at about $\ln 2$ in the training process when the depth becomes large enough.

loose, which suggests that the real situation is much worse than our theoretical analysis. We also notice in Fig. 5.7(b) that for this strongly entangling encoding strategy, the larger the qubit number is, the faster the divergence decreases. This unexpected phenomenon reveals the possibility that specific structures of encoding circuits may lead to more severe concentrations for larger numbers of qubits and is worthy of further studies.

Next, we examine the performance of QNNs and POVMs by generating 20k data samples for training and 4k for testing under the encoding strategy in Fig. 5.6, where half of the data belong to class 0, and the others belong to class 1. The QNNs are designed according to the right-hand side of Fig. 5.6, where the number of layers L_{QNN} is set as $n+2$, the finally measured Hermitian operators are set as $H_1 = Z$ and $H_2 = X$

on the first qubit, and all parameters $\boldsymbol{\theta}$ are initialized randomly in $[0, 2\pi]$. During the optimization, we adopt the Adam optimizer [133] with a batch size of 200 and a learning rate of 0.02. In the POVM setting, we directly employ semi-definite programming [164] to obtain the maximum success probability P_{succ} on the training data samples. The results are illustrated in Fig. 5.8. We observe that both the test accuracy of the QNNs and the maximum success probability P_{succ} of the POVMs eventually decay to about 0.5 as the encoding depth grows, indicating that the classification abilities of both the QNNs and the POVMs are no better than random guessing. In addition, the training losses of QNNs in Fig. 5.8(c) gradually approach $\ln 2$ as the depth grows and finally do not go down anymore during the whole training process, which implies that the concentration of this data set on the maximally mixed state would limit the trainability of QNNs as we predicted in Sec. 5.3.1. All these results are in line with our theoretical expectations.

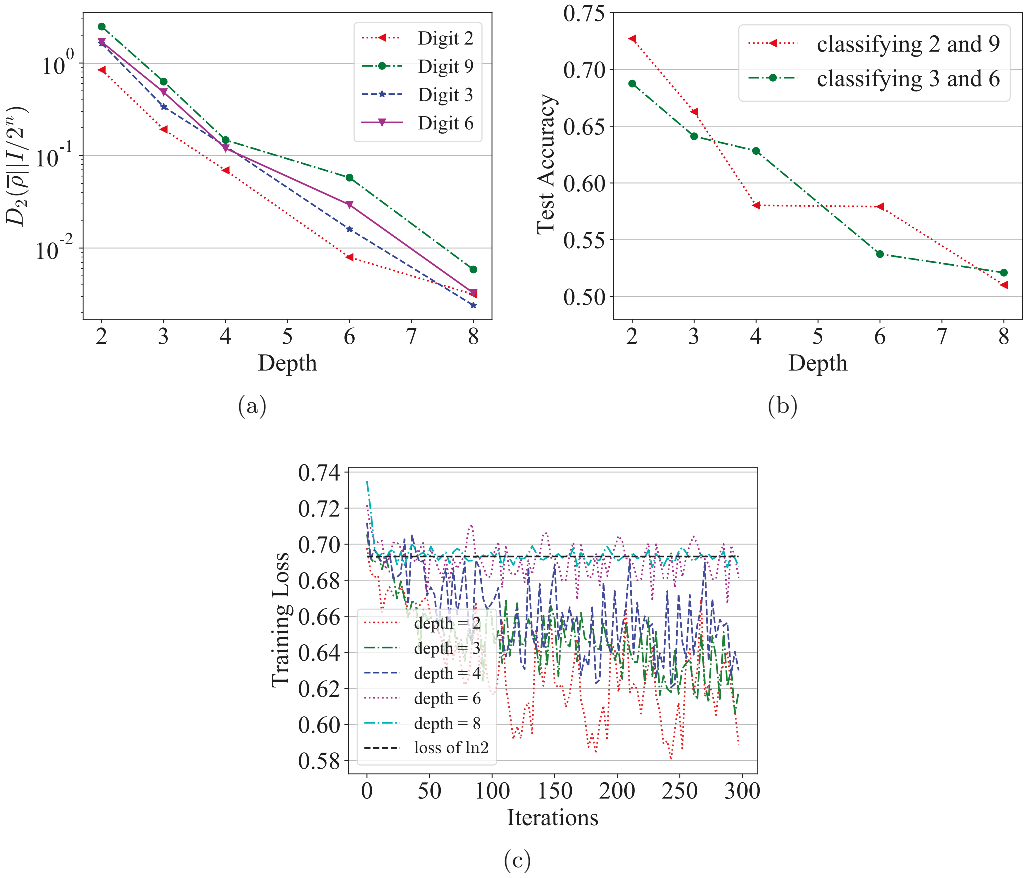
5.4.2 On Public Data Set

5.4.2.1 Data Set and Preprocessing

The handwritten digit data set MNIST [134] consists of 70k images labeled from ‘0’ to ‘9’, each of which contains 28×28 gray-scale pixels valued in $[0, 255]$. In order to facilitate encoding, these images are first resized to 4×4 and then normalized to values between 0 and π. Finally, we select all images corresponding to two pairs of labels, i.e., $(2, 9)$ and $(3, 6)$, for two binary classification tasks. For each task, there are about 12k training samples and 2k testing samples, and each category accounts for half or so.

5.4.2.2 Results

Here we mainly consider the performance of QNN on this data set because POVMs are generally not suitable for prediction. These 16-dimensional preprocessed images are first encoded into n-qubit quantum states with encoding depth D and then fed into a QNN (cf. Fig. 5.6 again). We set n as 2,3,4,6,8 and D as 8,6,4,3,2 accordingly. The settings of QNN are almost the same as those used in the synthetic case, except for a new learning rate of 0.05. From Fig. 5.9(a), we see that the average state of each digit class concentrates



(a)

(b)

(c)

Figure 5.9: Numerical results of QNN for MNIST data set under the encoding strategy in Fig. 5.6. (a) The curves for the quantum divergence between the averaged encoded state $\bar{\rho}$ of each handwritten digit and the maximally mixed state $\mathbb{1}$ decrease exponentially on depth. (b) The test accuracy reduces rapidly with a larger encoding depth; (c) In the case of classifying digits 3 and 6, when the depth is large (e.g., 8), it is difficult to keep the training loss away from $\ln 2$ in the training process.

on the maximally mixed state at an approximately exponential speed on depth, which is consistent with our main result. Furthermore, the outcomes in Figs. 5.9(b) and 5.9(c) also confirm the incapability of training of QNNs, provided that the classical inputs are encoded by a higher depth PQC.

5.5 Discussion

We have witnessed both theoretically and empirically that for usual PQC-based data encoding strategies with higher depth, the average encoded state concentrates on the maximally mixed state. We further showed that such concentration severely limits the capabilities of quantum classifiers for practical tasks. Such limitation indicates that we should pay more attention to methods encoding classical data into PQCs in quantum supervised learning.

This chapter suggests that the distance between the average encoded state and the maximally mixed state may be a reasonable metric to quantify how well the quantum encoding preserves the features in quantum supervised learning. The result on the encoding concentration also motivates us to consider how to design PQC-based encoding strategies better to avoid the exponentially decayed distance. An obvious way this chapter implies might be to keep the depth shallow while accompanied by a higher width. Still, it will render poor generalization performance [67] as well as the notorious barren plateau issue [44]. Therefore, it will be desirable to develop nontrivial quantum encoding strategies to guarantee the effectiveness and efficiency of quantum supervised learning as well as quantum kernel methods [14, 68, 75]. Recent works on data re-uploading [73, 146, 166] and pooling [67, 72] of quantum neural networks may provide potential solutions for improving quantum encoding efficiency.

Chapter 6

Conclusion and Future Directions

In this chapter, we summarize the contributions of this thesis, and give potential future research directions that can be further explored.

6.1 Conclusion

In this thesis, we explored the applications of parameterized quantum circuits in machine learning, focusing on the research of PQC's capabilities in different applications and PQC's limitations on quantum machine learning.

Specifically, in Chapter 3, we proposed a VSQL framework that used the concept of obtaining the classical shadows to do classification tasks. This framework mainly uses a local PQC similar to a convolution operation to extract information features and then feeds the obtained features into the classical fully-connected neural network to complete classification. Compared with the general methods using global PQCs to extract hidden features, VSQL can achieve similar or even higher accuracy in some quantum state classification tasks and handwritten digit recognition tasks, but it has fewer parameters. Moreover, because local PQC is simpler and easier to implement, VSQL can bring less noise. In addition, VSQL could also avoid barren plateau problems as long as the dimensions of the shadow circuit are small enough. Finally, another advantage of local PQC is that we can design the entanglement structure according to the hardware topology connectivity of quantum computers at the current stage.

In Chapter 4, we proposed the QSANN architecture, which introduces the classical self-attention mechanism into the quantum neural network. Motivation mainly comes from two aspects: on the one hand, most of the existing models for quantum natural language processing are based on syntax analysis, so it requires complicated pre-processing, and is challenging to extend to larger datasets; On the other hand, among classical models employing word-embedding technique [58], the models based on self-attention mechanism have achieved excellent results. In addition, we proposed Gaussian projected quantum self-attention, which is better than the commonly used inner product self-attention, to demonstrate potential quantum advantages. In general, the former may be able to dig out some hidden correlations between words, while the latter cannot. Through numerical experiments on some small-scale and medium-scale datasets, we find that QSANN is significantly superior to the existing syntactic parsing-based models and slightly superior to the classical self-attention neural networks under the same conditions. Since QSANN can run on existing quantum computers with a medium noise scale, it is a potential quantum natural language processing model in the future.

In Chapter 5, we have seen theoretically and experimentally that for the PQC-based data encoding strategy, the average encoded state will concentrate on the maximally mixed state and converge exponentially with the increase of depth. We further show that these encoded quantum states employing PQC-based encoding strategies will severely limit the classification ability of the quantum classifiers for downstream tasks, including quantum classification and quantum state discrimination. Finally, we also analyzed how defining and finding a good quantum encoding strategy is urgent. And the quantum divergence provided in this chapter may be a good indicator to measure the encoding scheme.

6.2 Future Directions

Quantum machine learning, as a hot research topic in quantum AI, has broad research and application prospects. Many related applications and corresponding research directions have emerged in recent years. Now we briefly introduce some research directions that this thesis can inspire.

We know that VSQL aims at supervised learning tasks, so it is worth exploring how it performs in some unsupervised learning tasks, such as clustering. In addition, it is worth looking forward to seeing whether VSQL can be applied and how effective it is in some learning tasks of generative and online models. In terms of architecture design, such as how the specific design form of shadow circuit in VSQL affects the performance of the overall model, how the features extracted from the quantum computers and the following neural network parts can be better connected to make the model optimal, etc. Regarding model complexity, can the parameter amount in VSQL be further decreased, and is there a lower bound? What is the expressibility of VSQL, and is there an upper bound? Along this line, a possible method to further reduce the model complexity of VSQL is proposed [167].

In addition to text classification, the performance of QSANN in other quantum natural language processing tasks is also worth exploring, such as machine translation, question-answering systems, etc. In terms of model design, QSANN is just a simple attempt, and many advanced technologies have not been used, such as positional encoding and multi-head attention. It will be interesting to explore the performance of the improved version of QSANN with these technologies in natural language processing tasks. In addition, GPQSA proposed in QSANN also needs to be tested by more experiments. Although we argue that it is more effective than general self-attention based on inner product, we still do not know how effective it is. Exploring other forms of self-attention to show quantum advantages is also an urgent matter in the QNLP field. Or more generally, exploring the QNLP model with quantum advantages is urgent. In terms of model complexity, it is fascinating to explore the QNLP model with higher accuracy and lower parameters.

From Chapter 5, we know that the distance between the average encoded state and the maximally mixed state will directly affect the classification ability of the downstream quantum classification model. Therefore, it will be an essential direction to design indicators to measure the quality of quantum encoding strategies according to this quantum divergence. That is because there are few guiding works on developing PQCs, and most PQCs are general circuits designed based on engineering experience. These circuits not only have a large number of parameters but also have problems such as a lack of expressibility and barren plateaus. Chapter 5 suggests we should try to use some shallow circuits with many qubits when designing encoding circuits, but these circuits have apparent disadvantages. Therefore, developing some nontrivial quantum data encoding strategies in the NISQ era has become a very urgent research direction, even as important as the research direction of designing the architectures of QNNs. Recently, the research on data re-uploading encoding strategy is a good direction, but the substantial effect needs further to be explored. In addition, there are also research directions based on quantum kernels. Although quantum kernel is the most likely direction to realize quantum advantages, PQC-based kernels still need further review.

Bibliography

[1] Richard P Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7), 1982.

[2] John Preskill. Quantum computing 40 years later. *arXiv preprint arXiv:2106.10522*, 2021.

[3] Aram W Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203–209, 2017.

[4] Andrew M Childs and Wim van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1):1–52, jan 2010. ISSN 0034-6861. doi: 10.1103/RevModPhys.82.1.

[5] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2(1):15023, nov 2016. ISSN 2056-6387. doi: 10.1038/npjqi.2015.23.

[6] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, sep 2018. ISSN 0027-8424. doi: 10.1073/pnas.1801723115.

[7] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, Sep 2017.

[8] Feihu Xu, Xiongfeng Ma, Qiang Zhang, Hoi-Kwong Lo, and Jian-Wei Pan. Secure quantum key distribution with realistic devices. *Reviews of Modern Physics*, 92(2): 25002, 2020.

[9] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. Quantum computational chemistry. *Reviews of Modern Physics*, 92(1):015003, mar 2020.

[10] Yudong Cao, Jonathan Romero, Jonathan P Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P D Sawaya, Sukin Sim, Libor Veis, and Alán Aspuru-Guzik. Quantum Chemistry in the Age of Quantum Computing. *Chemical Reviews*, 119(19):10856–10915, oct 2019. ISSN 0009-2665. doi: 10.1021/acs.chemrev.8b00803.

[11] Fernando G.S.L. Brandao and Krysta M. Svore. Quantum Speed-Ups for Solving Semidefinite Programs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 415–426. IEEE, oct 2017.

[12] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(3):130503, Sep 2014.

[13] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. Power of data in quantum machine learning. *Nature Communications*, 12(1):2631, dec 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-22539-9.

[14] Maria Schuld and Francesco Petruccione. *Machine Learning with Quantum Computers*. 2021. ISBN 9783030830977. URL `http://www.springer.com/series/10039`.

[15] Yann Lecun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, May 2015.

[16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[17] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, Apr 2015.

[18] Carlo Ciliberto, Mark Herbster, Alessandro Davide Ialongo, Massimiliano Pontil, Andrea Rocchetto, Simone Severini, and Leonard Wossnig. Quantum machine learning: A classical perspective. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2209):20170551, Jan 2018.

[19] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv:1307.0411*, Jul 2013.

[20] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum algorithm for data fitting. *Physical review letters*, 109(5):050505, 2012.

[21] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science*, pages 124–134. Ieee, 1994.

[22] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009.

[23] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Physical review letters*, 100(16):160501, 2008.

[24] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, Aug 2018.

[25] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574 (7779):505–510, 2019.

[26] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020.

[27] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'brien. A variational eigenvalue solver on a photonic quantum processor. *Nature communications*, 5(1):1–7, 2014.

[28] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, 2017.

[29] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.

[30] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. The quest for a quantum neural network. *Quantum Information Processing*, 13(11):2567–2586, 2014.

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[32] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, pages 1045–1048. Makuhari, 2010.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

[34] Iris Cong, Soonwon Choi, and Mikhail D. Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, Dec 2019.

[35] Samuel Yen-Chi Chen, Shinjae Yoo, and Yao-Lung L Fang. Quantum long short-term memory. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8622–8626. IEEE, 2022.

[36] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):4213, dec 2014.

[37] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, Sep 2017.

[38] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm. *arXiv:1411.4028*, pages 1–16, Nov 2014.

[39] Edward Farhi and Aram W Harrow. Quantum Supremacy through the Quantum Approximate Optimization Algorithm. *arXiv:1602.07674*, pages 1–22, Feb 2016.

[40] Xiao Yuan, Suguru Endo, Qi Zhao, Ying Li, and Simon C. Benjamin. Theory of variational quantum simulation. *Quantum*, 3:191, oct 2019.

[41] M. Cerezo, Kunal Sharma, Andrew Arrasmith, and Patrick J. Coles. Variational Quantum State Eigensolver. *arXiv:2004.01372*, apr 2020.

[42] Ranyiliu Chen, Zhixin Song, Xuanqiang Zhao, and Xin Wang. Variational Quantum Algorithms for Trace Distance and Fidelity Estimation. *arXiv:2012.05768*, pages 1–13, dec 2020.

[43] Xin Wang, Zhixin Song, and Youle Wang. Variational Quantum Singular Value Decomposition. *arXiv:2006.02336*, jun 2020.

[44] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, Dec 2018.

[45] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, pages 1–40, Jun 2020.

[46] Scott Aaronson. Shadow tomography of quantum States. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, pages 1088–1101, New York, New York, USA, Nov 2018. ACM Press.

[47] Ivano Basile and Fabio Tamburini. Towards quantum language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1840–1849, 2017.

[48] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[49] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8658–8665, 2019.

[50] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Xiangyang Xue, and Zheng Zhang. Multi-scale self-attention for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7847–7854, 2020.

[51] Alessandro Sordoni, Jian-Yun Nie, and Yoshua Bengio. Modeling term dependencies with quantum language models for IR. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13*, page 653, New York, New York, USA, 2013. ACM Press.

[52] Peng Zhang, Jiabin Niu, Zhan Su, Benyou Wang, Liqun Ma, and Dawei Song. End-to-End Quantum-Like Language Models with Application to Question Answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

[53] Yazhou Zhang, Dawei Song, Peng Zhang, Xiang Li, and Panpan Wang. A quantum-inspired sentiment representation model for twitter sentiment analysis. *Applied Intelligence*, 49(8):3093–3108, 2019.

[54] William Zeng and Bob Coecke. Quantum algorithms for compositional natural language processing. *arXiv preprint arXiv:1608.01406*, 2016.

[55] Konstantinos Meichanetzidis, Stefano Gogioso, Giovanni De Felice, Nicolò Chiappori, Alexis Toumi, and Bob Coecke. Quantum natural language processing on near-term quantum computers. *arXiv preprint arXiv:2005.04147*, 2020.

[56] Nathan Wiebe, Alex Bocharov, Paul Smolensky, Matthias Troyer, and Krysta M Svore. Quantum language processing. *arXiv preprint arXiv:1902.05162*, 2019.

[57] Samuel Yen-Chi Chen, Shinjae Yoo, and Yao-Lung L Fang. Quantum long short-term memory. *arXiv preprint arXiv:2009.01783*, 2020.

[58] Yoav Goldberg and Omer Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv preprint arXiv:1402.3722*, 2014.

[59] Peter Cha, Paul Ginsparg, Felix Wu, Juan Carrasquilla, Peter L McMahon, and Eun-Ah Kim. Attention-based quantum tomography. *arXiv preprint arXiv:2006.12469*, 2020.

[60] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, Mar 2020.

[61] Edward Grant, Marcello Benedetti, Shuxiang Cao, Andrew Hallam, Joshua Lockhart, Vid Stojevic, Andrew G. Green, and Simone Severini. Hierarchical quantum classifiers. *npj Quantum Information*, 4(1):65, Dec 2018.

[62] Mateusz Ostaszewski, Lea M Trenkwalder, Wojciech Masarczyk, Eleanor Scerri, and Vedran Dunjko. Reinforcement learning for optimization of variational quantum circuit architectures. *Advances in Neural Information Processing Systems*, 34:18182–18194, 2021.

[63] Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. Structure optimization for parameterized quantum circuits. *Quantum*, 5:391, 2021.

[64] Shi-Xin Zhang, Chang-Yu Hsieh, Shengyu Zhang, and Hong Yao. Differentiable Quantum Architecture Search. *arXiv:2010.08561*, oct 2020.

[65] Yuxuan Du, Tao Huang, Shan You, Min-Hsiu Hsieh, and Dacheng Tao. Quantum circuit architecture search: error mitigation and trainability enhancement for variational quantum solvers. *arXiv preprint arXiv:2010.10217*, 2020.

[66] Matthias C Caro, Elies Gil-Fuster, Johannes Jakob Meyer, Jens Eisert, and Ryan Sweke. Encoding-dependent generalization bounds for parametrized quantum circuits. *Quantum*, 5:582, 2021.

[67] Leonardo Banchi, Jason Pereira, and Stefano Pirandola. Generalization in quantum machine learning: A quantum information standpoint. *PRX Quantum*, 2(4):040321, 2021.

[68] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. Power of data in quantum machine learning. *Nature Communications*, 12(1):2631, dec 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-22539-9.

[69] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, pages 1–5, 2021.

[70] Tongyang Li, Shouvanik Chakrabarti, and Xiaodi Wu. Sublinear quantum algorithms for training linear and kernel-based classifiers. In *International Conference on Machine Learning*, pages 3815–3824, 2019.

[71] Jonas Kübler, Simon Buchholz, and Bernhard Schölkopf. The inductive bias of quantum kernels. *Advances in Neural Information Processing Systems*, 34, 2021.

[72] Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. Quantum embeddings for machine learning. *arXiv:2002.08953*, pages 1–11, Jan 2020.

[73] Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

[74] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4 (4):043001, Jun 2019.

[75] Evan Peters, João Caldeira, Alan Ho, Stefan Leichenauer, Masoud Mohseni, Hartmut Neven, Panagiotis Spentzouris, Doug Strain, and Gabriel N. Perdue. Machine learning of high dimensional data on a noisy quantum processor. *arXiv:2101.09581*, pages 1–20, jan 2021.

[76] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, Mar 2019.

[77] Edward Farhi and Hartmut Neven. Classification with Quantum Neural Networks on Near Term Processors. *arXiv:1802.06002*, pages 1–21, Feb 2018.

[78] Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. Qnlp in practice: Running compositional models of meaning on a quantum computer. *arXiv preprint arXiv:2102.12846*, 2021.

[79] Michael A. Nielsen and Isaac Chuang. Quantum Computation and Quantum Information. *American Journal of Physics*, 70(5):558–559, May 2002.

[80] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

[81] Jean-Luc Brylinski and Ranee Brylinski. Universal quantum gates. In *Mathematics of quantum computation*, pages 117–134. Chapman and Hall/CRC, 2002.

[82] Henry W Lin, Max Tegmark, and David Rolnick. Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6):1223–1247, 2017.

[83] Andrew G Taube and Rodney J Bartlett. New perspectives on unitary coupled-cluster theory. *International journal of quantum chemistry*, 106(15):3393–3401, 2006.

[84] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P Orlando, Simon Gustavsson, and William D Oliver. A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, 6(2):021318, 2019.

[85] Kenneth Wright, Kristin M Beck, Sea Debnath, JM Amini, Y Nam, N Grzesiak, J-S Chen, NC Pisenti, M Chmielewski, C Collins, et al. Benchmarking an 11-qubit quantum computer. *Nature communications*, 10(1):1–6, 2019.

[86] Maria Schuld. Supervised quantum machine learning models are kernel methods. *arXiv preprint arXiv:2101.11020*, 2021.

[87] Scott Aaronson. Read the fine print. *Nature Physics*, 11(4):291–293, Apr 2015.

[88] Edwin Stoudenmire and David J Schwab. Supervised learning with tensor networks. *Advances in Neural Information Processing Systems*, 29, 2016.

[89] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. *Advances in neural information processing systems*, 20, 2007.

[90] Jakob S Kottmann, Mario Krenn, Thi Ha Kyaw, Sumner Alperin-Lea, and Alán Aspuru-Guzik. Quantum computer-aided design of quantum optics hardware. *Quantum Science and Technology*, 6(3):035010, 2021.

[91] Mario Krenn, Manuel Erhard, and Anton Zeilinger. Computer-inspired quantum experiments. *Nature Reviews Physics*, 2(11):649–661, 2020.

[92] Mario Krenn, Jakob S Kottmann, Nora Tischler, and Alán Aspuru-Guzik. Conceptual understanding through efficient automated design of quantum optical experiments. *Physical Review X*, 11(3):031044, 2021.

[93] Jakob S Kottmann, Abhinav Anand, and Alán Aspuru-Guzik. A feasible approach for automatically differentiable unitary coupled-cluster on quantum computers. *Chemical science*, 12(10):3497–3508, 2021.

[94] Joonho Lee, William J Huggins, Martin Head-Gordon, and K Birgitta Whaley. Generalized unitary coupled cluster wave functions for quantum computation. *Journal of chemical theory and computation*, 15(1):311–324, 2018.

[95] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature communications*, 12(1):1–9, 2021.

[96] Song Cheng, Jing Chen, and Lei Wang. Information perspective to probabilistic modeling: Boltzmann machines versus born machines. *Entropy*, 20(8):583, 2018.

[97] Jonathan Romero, Ryan Babbush, Jarrod R McClean, Cornelius Hempel, Peter J Love, and Alán Aspuru-Guzik. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *Quantum Science and Technology*, 4(1): 014008, 2018.

[98] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, Sep 2018.

[99] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998.

[100] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020.

[101] David Wierichs, Christian Gogolin, and Michael Kastoryano. Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer. *Physical Review Research*, 2(4):043246, 2020.

[102] Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

[103] Artur Garcia-Saez and Jordi Riu. Quantum observables for continuous control of the quantum approximate optimization algorithm via reinforcement learning. *arXiv preprint arXiv:1911.09682*, 2019.

[104] Matteo M Wauters, Emanuele Panizon, Glen B Mbeng, and Giuseppe E Santoro. Reinforcement-learning-assisted quantum optimization. *Physical Review Research*, 2(3):033446, 2020.

[105] Ken M Nakanishi, Keisuke Fujii, and Synge Todo. Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research*, 2(4):043158, 2020.

[106] Joonwoo Bae and Leong Chuan Kwek. Quantum state discrimination and its applications. *Journal of Physics A: Mathematical and Theoretical*, 48(8):083001, Feb 2015.

[107] Hongxiang Chen, Leonard Wossnig, Simone Severini, Hartmut Neven, and Masoud Mohseni. Universal discriminative quantum neural networks. *Quantum Machine Intelligence*, 3(1):1–11, 2021.

[108] Andrew Patterson, Hongxiang Chen, Leonard Wossnig, Simone Severini, Dan Browne, and Ivan Rungger. Quantum State Discrimination Using Noisy Quantum Neural Networks. *arXiv:1911.00352*, Nov 2019.

[109] Maria Schuld and Nathan Killoran. Quantum Machine Learning in Feature Hilbert Spaces. *Physical Review Letters*, 122(4):040504, Feb 2019.

[110] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, 94(1):015004, 2022.

[111] Guangxi Li, Zhixin Song, and Xin Wang. Vsql: Variational shadow quantum learning for classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 8357–8365, 2021.

[112] Weikang Li and Dong-Ling Deng. Recent advances for quantum classifiers. *Science China Physics, Mechanics & Astronomy*, 65(2):1–23, 2022.

[113] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. Mathematical foundations for a compositional distributional model of meaning. *arXiv preprint arXiv:1003.4394*, 2010.

[114] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[115] Ashish Kapoor, Nathan Wiebe, and Krysta Svore. Quantum perceptron models. In *Advances in Neural Information Processing Systems*, pages 3999–4007, 2016.

[116] Maria Schuld and Francesco Petruccione. Quantum ensembles of quantum classifiers. *Scientific Reports*, 8(1):2772, Dec 2018.

[117] Amandeep Singh Bhatia, Mandeep Kaur Saggi, Ajay Kumar, and Sushma Jain. Matrix Product State–Based Quantum Classifier. *Neural Computation*, 31(7):1499–1517, Jul 2019.

[118] Eric R. Ziegel, E. L. Lehmann, and George Casella. Theory of Point Estimation. *Technometrics*, 41(3):274, Aug 1999.

[119] Léon Bottou. Stochastic Learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3176, pages 146–168. 2004.

[120] Aram Harrow and John Napp. Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms. *arXiv:1901.05374*, pages 1–45, Jan 2019.

[121] Wassily Hoeffding. Probability Inequalities for sums of Bounded Random Variables. *Journal of the American Statistical Association*, 58(301):13—-30, 1963.

[122] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, pages 1–29, aug 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00348-9.

[123] Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Physical Review A*, 80(1):012304, Jul 2009.

[124] Z. Puchała and J.A. Miszczak. Symbolic integration with respect to the Haar measure on the unitary groups. *Bulletin of the Polish Academy of Sciences Technical Sciences*, 65(1):21–27, Feb 2017.

[125] Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-Induced Barren Plateaus in Variational Quantum Algorithms. *arXiv:2007.14384*, 2(Theorem 1):1–20, Jul 2020.

[126] Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang. PaddlePaddle: An Open-Source Deep Learning Platform from Industrial Practice. *Frontiers of Data and Domputing*, 1(1):105–115, 2019.

[127] Charles H. Bennett. Quantum cryptography using any two nonorthogonal states. *Physical Review Letters*, 68(21):3121–3124, May 1992.

[128] Luis Roa, Juan Carlos Retamal, and Carlos Saavedra. Quantum-state discrimination. *Physical Review A - Atomic, Molecular, and Optical Physics*, 66(1):121031–121034, Apr 2002.

[129] Carl W. Helstrom. Quantum detection and estimation theory. *Journal of Statistical Physics*, 1(2):231–252, Jun 1969.

[130] A. S. Holevo. Bounds for the Quantity of Information Transmitted by a Quantum Communication Channel. *Probl. Peredachi Inf.*, 9(3):3–11, 1973.

[131] Sébastien Gambs. Quantum classification. *arXiv:0809.0444*, pages 119–123, Sep 2008.

[132] Masoud Mohseni, Aephraim M. Steinberg, and János A. Bergou. Optical realization of optimal unambiguous discrimination for pure and mixed quantum states. *Physical Review Letters*, 93(20):200403, Nov 2004.

[133] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec 2015.

[134] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[135] Scott Aaronson, Xinyi Chen, Elad Hazan, and Satyen Kale. Online learning of quantum states. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pages 8976–8986, 2018.

[136] Yifang Chen and Xin Wang. More practical and adaptive algorithms for online quantum state learning. *arXiv preprint arXiv:2006.01013*, 2020.

[137] Feidiao Yang, Jiaqing Jiang, Jialin Zhang, and Xiaoming Sun. Revisiting online quantum state learning. In *AAAI*, pages 6607–6614, 2020.

[138] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum (nisq) algorithms. *arXiv preprint arXiv:2101.08448*, 2021.

[139] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, pages 1–20, 2021.

[140] Suguru Endo, Zhenyu Cai, Simon C Benjamin, and Xiao Yuan. Hybrid Quantum-Classical Algorithms and Quantum Error Mitigation. *Journal of the Physical Society of Japan*, 90(3):032001, mar 2021. ISSN 0031-9015. doi: 10.7566/JPSJ.90.032001.

[141] Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(12), 2006.

[142] Riccardo Di Sipio, Jia-Hong Huang, Samuel Yen-Chi Chen, Stefano Mangini, and Marcel Worring. The dawn of quantum natural language processing. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8612–8616. IEEE, 2022.

[143] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 597–606, 2015.

[144] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[145] Guangxi Li, Ruilin Ye, Xuanqiang Zhao, and Xin Wang. Concentration of data encoding in parameterized quantum circuits. *arXiv preprint arXiv:2206.08273*, 2022.

[146] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

[147] Ryan LaRose and Brian Coyle. Robust data encodings for quantum classifiers. *Physical Review A*, 102(3):032420, 2020.

[148] Hiroshi Yano, Yudai Suzuki, Rudy Raymond, and Naoki Yamamoto. Efficient discrete feature encoding for variational quantum classifier. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, pages 11–21. IEEE, 2020.

[149] Martin Müller-Lennert, Frédéric Dupuis, Oleg Szehr, Serge Fehr, and Marco Tomamichel. On quantum Rényi entropies: A new generalization and some properties. *Journal of Mathematical Physics*, 54(12):122203, dec 2013. ISSN 0022-2488. doi: 10.1063/1.4838856.

[150] Mark M Wilde, Andreas Winter, and Dong Yang. Strong Converse for the Classical Capacity of Entanglement-Breaking and Hadamard Channels via a Sandwiched Rényi Relative Entropy. *Communications in Mathematical Physics*, 331(2):593–622, oct 2014. ISSN 0010-3616. doi: 10.1007/s00220-014-2122-x.

[151] Dénes Petz. Quasi-entropies for finite quantum systems. *Reports on Mathematical Physics*, 23(1):57–65, feb 1986. ISSN 00344877. doi: 10.1016/0034-4877(86)90067-4.

[152] Koenraad M R Audenaert. A sharp continuity estimate for the von Neumann entropy. *Journal of Physics A: Mathematical and Theoretical*, 40(28):8127–8136, jul 2007. ISSN 1751-8113. doi: 10.1088/1751-8113/40/28/S18.

[153] Michael Nussbaum and Arleta Szkoła. The Chernoff lower bound for symmetric quantum hypothesis testing. *The Annals of Statistics*, 37(2):1040–1057, 2009. ISSN 0090-5364.

[154] Robert Salzmann, Nilanjana Datta, Gilad Gour, Xin Wang, and Mark M. Wilde. Symmetric distinguishability as a quantum resource. *New Journal of Physics*, 23(8): 083016, aug 2021. ISSN 1367-2630. doi: 10.1088/1367-2630/ac14aa.

[155] Maria Kieferova, Ortiz Marrero Carlos, and Nathan Wiebe. Quantum Generative Training Using Rényi Divergences. *arXiv:2106.09567*, jun 2021.

[156] Kun Fang and Hamza Fawzi. Geometric Rényi Divergence and its Applications in Quantum Channel Capacities. *Communications in Mathematical Physics*, 384(3): 1615–1677, jun 2021. ISSN 0010-3616. doi: 10.1007/s00220-021-04064-4.

[157] Chai Wah Wu. On rearrangement inequalities for multiple sequences. *arXiv preprint arXiv:2002.10514*, 2020.

[158] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The collected works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

[159] Christopher A Fuchs and Jeroen Van De Graaf. Cryptographic distinguishability measures for quantum-mechanical states. *IEEE Transactions on Information Theory*, 45(4):1216–1227, 1999.

[160] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, pages 136–145. IEEE, 2001.

[161] Ryuji Takagi, Suguru Endo, Shintaro Minagawa, and Mile Gu. Fundamental limits of quantum error mitigation. *arXiv preprint arXiv:2109.04457*, 2021.

[162] DP DiVincenzo, DW Leung, and BM Terhal. Quantum data hiding. *IEEE Transactions on Information Theory*, 48(3):580–598, 2002.

[163] Carl W Helstrom. Quantum detection and estimation theory. *Journal of Statistical Physics*, 1(2):231–252, 1969.

[164] Joonwoo Bae and Won-Young Hwang. Minimum-error discrimination of qubit states: Methods, solutions, and properties. *Phys. Rev. A*, 87:012334, Jan 2013. doi: 10.1103/PhysRevA.87.012334.

[165] Andrew Patterson, Hongxiang Chen, Leonard Wossnig, Simone Severini, Dan Browne, and Ivan Rungger. Quantum state discrimination using noisy quantum neural networks. *Physical Review Research*, 3(1):013063, 2021.

[166] Sofiene Jerbi, Lukas J. Fiderer, Hendrik Poulsen Nautrup, Jonas M. Kübler, Hans J. Briegel, and Vedran Dunjko. Quantum machine learning beyond kernel methods. *arXiv:2110.13162*, pages 1–13, 2021.

[167] Afrad Basheer, Yuan Feng, Christopher Ferrie, and Sanjiang Li. Alternating layered variational quantum circuits can be classically optimized efficiently using classical shadows. *arXiv preprint arXiv:2208.11623*, 2022.