

A Thesis for the Degree of Ph.D. in Engineering

Applicability of Quantum-Enhanced Machine Learning

February 2024

Graduate School of Science and Technology, Keio University

Yudai Suzuki

Abstract

Quantum computers are next-generation computing devices that have the potential to enhance the performance of machine learning. An approach to achieving quantum enhancement is to utilize the Hilbert space, whose dimension scales exponentially in the number of qubits, as a feature space for machine learning. The use of such quantum-enhanced feature space could enable us to find data patterns more easily than conventional methods. Thus far, it has been theoretically shown that quantum-enhanced machine learning models can solve specific classification tasks that classical methods cannot efficiently solve. This motivates many researchers to pursue practical advantages of the methods. Toward practical applications, it is critical to establish guidelines for designs of quantum-enhanced machine learning models.

This thesis aims to analyze the performance of quantum-enhanced machine learning models and give insights into design principles of the models for practical applications. More specifically, we focus on two methods, quantum kernel methods and quantum reservoir computing, and then provide guidelines for building their models in practical situations.

We focus on two challenges in quantum kernel methods for real-world applications. The first is that constructing quantum feature maps is nontrivial when applied to actual machine learning problems. This thesis proposes a method for analyzing quantum feature maps, which can help screen a suitable quantum feature map among many candidates. In addition, we examine the effectiveness of a synthesis method to construct a powerful quantum kernel. Another issue is that implementation feasibility and trainability deteriorate as the number of qubits increases. We thus propose a new quantum kernel called the quantum Fisher kernel and demonstrate from analytical and numerical perspectives that our proposal can avoid the aforementioned problem when shallow alternating layered ansatzes are used.

In quantum reservoir computing, there is room for investigation in designing quantum reservoir systems that are amenable to implementation and can perform well. In this thesis, we propose a quantum reservoir computing framework that positively makes use of unavoidable quantum noise in actual quantum hardware. Our experimental demonstrations on superconducting quantum devices show that quantum noise can enhance temporal information processing. Also, numerical analysis using a tool called temporal information processing capacity elucidates that dissipation noise such as amplitude damping can induce sequential data processing capabilities.

Acknowledgement

First of all, I would like to sincerely thank all the people who have given me tremendous support and encouragement during my doctoral program. Especially I appreciate Prof. Kenji Yasuoka for his continuous support and valuable advice on my study. Not only did he provide an environment where I could immerse myself in my research, but also helped me to grow my logical thinking and problem-solving abilities. I am also grateful that he encouraged me to pursue research topics I have been curious about. Furthermore, I would like to thank Prof. Naoki Yamamoto for his technical advice and patient support. He is always kind and helpful in guiding me through research discussions and writing papers. I cannot thank him enough for his tremendous help. I also appreciate him for organizing the Keio Quantum Computing Center. The discussions at the Keio Quantum Computing Center have motivated and inspired me a lot. I would also like to thank Prof. Naoki Yamamoto, Prof. Masahiro Takeoka, and Prof. Linyu Peng for taking the time to serve on my thesis committee. Their helpful comments highly improved the quality of my thesis.

In addition, I would also like to express my gratitude to professors, researchers, and students at the Keio Quantum Computing Center for their insightful discussions: Eriko Kaminishi, Hideaki Kawaguchi, Hideo Watanabe, Hiroshi Watanabe, Hiroshi Yano, Hiroyuki Tezuka, Junpei Kato, Kenji Sugisaki, Kohei Oshio, Michihiko Sugawara, Qi Gao, Rei Sakuma, Rudy Raymond, Ruho Kondo, Shu Kanno, Shumpei Uno, Takahiko Satoh, Tomoki Tanaka, Yutaka Shikano, Yuya Ohnishi, Yohichi Suzuki. In particular, weekly discussions on quantum machine learning including my research topics have broadened my knowledge and stimulated interesting ideas. The research achievements during my doctoral program are thanks to the Keio Quantum Computing Center, and hence, sincere thanks also go to Shinichi Niimi, Shiho Aizawa, and Naoko Oue, who manage and support the organization. In addition, I appreciate Hiroshi Yano for his friendship, patient discussion, and sharing his knowledge.

Moreover, I would like to thank collaborators for their valuable discussions. Prof. Kohei Nakajima, Tomoyuki Kubota, Quoc Hoan Tran, and Shumpei Kobayashi gave me meaningful insight into our work on quantum reservoir computing and progressed the research further. I am also grateful to Prof. Mayu Muramatsu for providing me with opportunities for research collaboration. She helped me to grow and impressed me with her way of thinking. Internships at IBM (Tokyo and Yorktown Heights) also stimulated my Ph.D. study and have given me deeper knowledge and understanding of quantum machine learning. I explicitly thank Tamiya Onodera, Atsushi Matsuo, Ikko Hamamura, Muyuan Li, and Kunal Sharma for their support and helpful discussions.

Furthermore, I was lucky to join the wonderful laboratory. People in Yasuoka laboratory are kind, supportive, and enthusiastic about research. Dr. Paul Brumby, Project Associate Professor, kindly helped me with preparing presentation materials and gave me advice on writing. Dr. Yoshinori Hirano, Project Associate Professor, willingly shared his knowledge about bioinformatics with us. Also, Dr. Stephen Fitz, Project Associate Professor, gave some insight into deep learning from the mathematical perspective. I am grateful to all past and current members

for their help and for sharing their knowledge: Daisuke Yuhara, Katsufumi Tomobe, Takuma Nozawa, Kantaro Inoue, Shinjiro Nakamura, Sho Ayuba, Tomohiro Hasegawa, Katsuhiro Endo, Kenta Ogino, Kiyoshiro Okada, Yuui Ono, Arafal Rafi, Jean-Francois Cailleau, Masashi Sawa, Akie Kowaguchi, Arisa Yamada, Kan Satake, Kazuya Hiraide, Masanari Ishiyama, Yo Taniguchi, Yoshinao Itakura, Daiki Sato, Genki Miwa, Ikki Yasuda, Koki Abe, Ryo Kawada, Akinori Satake, Kenta Shobu, Naonobu Kuribayashi, Satoki Ishiai, Hirotaka Kishimoto, Rio Taniguchi, Ryota Ishioka, Yonggi Park, Honomi Kashihara, Kentaro Ashitate, Kento Mima, Koki Yano, Kota Sakaki, Ren Kobayashi, Takumi Kojima. I am also thankful to my colleagues for their friendship and help: Go Kudo, Kai Pua, Kanako Matsui, Kazuaki Hirakawa, Kenta Hirayama, Shuzo Kato, Toshitsugu Miura, Yuto Yamada.

Finally, I would like to thank my family and friends for their support during my long time in graduate school.

February 2024

Contents

| | | |
|----------|---|-----------|
| 1 | Introduction | 1 |
| 2 | Quantum Computing | 4 |
| 2.1 | Basics of Quantum Computation | 4 |
| 2.1.1 | Dirac Notation for Quantum States | 4 |
| 2.1.2 | Quantum Bits (Qubits) | 5 |
| 2.1.3 | Quantum Gates | 7 |
| 2.1.4 | Measurement | 9 |
| 2.1.5 | Quantum Circuit Models | 10 |
| 2.2 | Density Matrices | 11 |
| 2.3 | Quantum Operations and Quantum Noise | 12 |
| 2.3.1 | Quantum Operator Formalism | 12 |
| 2.3.2 | Quantum Noise | 13 |
| 2.4 | Near-Term and Long-Term Quantum Computers | 14 |
| 2.5 | Application: Machine Learning | 14 |
| 3 | Quantum-Enhanced Machine Learning | 18 |
| 3.1 | Quantum-Enhanced Feature Space for Machine Learning | 18 |
| 3.1.1 | Quantum-Enhanced Feature Space | 18 |
| 3.1.2 | Quantum Feature Maps | 19 |
| 3.1.3 | Models | 21 |
| 3.2 | Quantum Kernel Methods | 22 |
| 3.2.1 | Kernel Methods | 22 |
| 3.2.2 | Basics of Quantum Kernel Methods | 25 |
| 3.2.3 | Support Vector Machines | 27 |
| 3.3 | Quantum Reservoir Computing | 29 |
| 3.3.1 | Framework of Reservoir Computing | 30 |
| 3.3.2 | Physical Reservoir Computing | 31 |
| 3.3.3 | Quantum Reservoir Computing Models | 31 |
| 4 | Quantum Kernel-Based Learning Models | 33 |
| 4.1 | Analysis and Synthesis of Quantum Feature Maps | 33 |
| 4.1.1 | Introduction | 34 |
| 4.1.2 | A Method to Analyze Quantum Feature Maps | 35 |
| 4.1.3 | Synthesized Quantum Feature Maps | 36 |
| 4.1.4 | Numerical Demonstration | 39 |
| 4.1.5 | Conclusion & Outlook | 43 |
| 4.2 | A Remedy to the Vanishing Similarity Issue: Quantum Fisher Kernel | 48 |
| 4.2.1 | Introduction | 48 |

| | | |
|----------|---|------------|
| 4.2.2 | Preliminary | 49 |
| 4.2.3 | Vanishing Similarity Issue in Fidelity-Based Quantum Kernel | 50 |
| 4.2.4 | Quantum Fisher Kernel | 54 |
| 4.2.5 | Vanishing Similarity Issue in Quantum Fisher Kernel | 56 |
| 4.2.6 | Numerical Demonstration | 57 |
| 4.2.7 | Expressivity and Performance | 61 |
| 4.2.8 | Conclusion & Outlook | 63 |
| 5 | Quantum Noise-Induced Reservoir Computing | 65 |
| 5.1 | Proof-of-Principle Demonstration | 65 |
| 5.1.1 | Introduction | 66 |
| 5.1.2 | Quantum Noise-Induced Reservoir Systems | 66 |
| 5.1.3 | Experimental Demonstration | 67 |
| 5.1.4 | Conclusion & Outlook | 79 |
| 5.2 | Information Processing Capability Induced by Quantum Noise | 79 |
| 5.2.1 | Introduction | 79 |
| 5.2.2 | Temporal Information Processing Capacity (TIPC) | 80 |
| 5.2.3 | TIPC Profile for QR Systems Simulated by Quantum Noise Models | 81 |
| 5.2.4 | Benchmark Tasks | 84 |
| 5.2.5 | TIPC Profile for QR Systems on Quantum Devices | 86 |
| 5.2.6 | Conclusion & Outlook | 90 |
| 6 | Conclusion and Outlook | 91 |
| 6.1 | Conclusion | 91 |
| 6.2 | Outlook | 92 |
| | Bibliography | 93 |
| A | Analytical Results for Vanishing Similarity Issue in Quantum Kernels | 109 |
| A.1 | Proof of Proposition 1 | 109 |
| A.1.1 | Case (1): Globally-Random Quantum Circuits | 109 |
| A.1.2 | Case (2): Alternating Layered Ansatzes | 110 |
| A.2 | Proof of Theorem 1 | 113 |
| A.2.1 | Case (1): Globally-Random Quantum Circuits | 113 |
| A.2.2 | Case (2): Alternating Layered Ansatzes | 117 |
| A.3 | Further Analytical Results | 123 |

Chapter 1

Introduction

Machine learning is a subfield of artificial intelligence (AI) technology where machines make inferences and predictions based on rules they learn from data [1,2]. In today's era of big data, it is essential to extract and utilize valuable information from an enormous amount of data that humans cannot handle. Machine learning can facilitate such data analysis and hence its impacts on our lives have grown tremendously. To date, a wide range of applications have been reported, including image processing [3–5], natural language processing [6,7], object detection [8], bioinformatics [9], and finance [10,11]. In addition, emerging generative AI models such as ChatGPT and related technologies have demonstrated the potential to revolutionize society by assisting widespread human activities, such as writing, debugging codes, and reducing the burdens of office work [12]. Thus, it is expected that demands for such machine learning technologies will keep increasing in the future.

Advances in machine learning technology are supported by the computational power of information processing devices. This indicates performance improvement in computers themselves is critical for the further development of machine learning. One of the next-generation computers that can boost information processing is a quantum computer, which makes use of quantum mechanical properties such as entanglement and interference for computing. Potentially, quantum computers can perform computations that are not efficiently executable by classical computers. The idea of quantum computing was first proposed by Richard Feynman, based on his view that quantum computers are more powerful tools to simulate quantum systems than conventional classical computers [13]. After this proposal, David Deutsch developed the theoretical formulation of quantum computers [14]. Subsequently, several quantum algorithms have been proposed that can outperform classical counterparts from the computational complexity perspective: for example, factoring a prime number [15], unstructured search [16], phase estimation [17], and solving linear systems of equations [18]. These quantum algorithms with theoretical guarantees have driven researchers to seek advantages in various fields, including machine learning [19], quantum chemistry [20–22], and finance [23].

Moreover, the development of quantum hardware has been remarkable these days. For instance, major companies such as IBM and Google have been developing superconducting quantum computers. Start-up companies also launched projects to build quantum computing devices. Currently available quantum information processors consist of a limited number of qubits (50 to hundreds of qubits) and cannot avoid the effects of noise; quantum computers at present are thus called NISQ (Noisy Intermediates-Scale Quantum) devices [24]. This means the quantum algorithms mentioned above are challenging to implement on these devices. In contrast, a recent study unveiled the potential of NISQ devices for advantages in sampling tasks [25]. Therefore, a number of researchers are pursuing the advantages of quantum computing not only in the long term but also in the NISQ era.

The interdisciplinary research area of quantum computing and machine learning is called quantum machine learning [19]. The main objectives of quantum machine learning can be broadly divided into two categories. One is to improve the computational speed of existing methods by leveraging the above-mentioned quantum algorithms. Thus far, the HHL algorithm [18], a quantum solver of linear systems of equations, has been utilized for regression models [26–28] and classifiers [29, 30]. Other primitive quantum algorithms have also been exploited to improve the computational speed. The other is to improve the performance of pattern recognition, where quantum computers are used to discover the underlying regularities in the data. The motivation behind this idea is to utilize the Hilbert space, whose dimension scales exponentially in the number of qubits, as a feature space for machine learning. In other words, the quality of the data is improved by mapping it onto the so-called quantum-enhanced feature space where data structure can be easily found. This field is called quantum-enhanced machine learning. Actually, it has been theoretically proven that some synthetic classification tasks cannot be solved efficiently by classical methods but by fine-tuned quantum-enhanced machine learning models [31–33]. In addition, 100 qubits available even in the NISQ devices result in a feature space of approximately 10^{30} -dimension, which might be hard for classical computers to access effectively. Therefore, quantum-enhanced machine learning has also been explored for its practical advantages even in the NISQ era.

This thesis discusses the design principles of quantum-enhanced machine learning models for practical applications. Although it has been suggested that quantum-enhanced feature space can improve performance for specific tasks, this field is at an early stage, and further investigations on how to construct powerful models are needed for real-world applications. Thus, the goals of this thesis are to analyze the performance of quantum-enhanced machine learning and provide guidelines for designing the models in practical situations. More specifically, we examine two types of quantum-enhanced machine learning: quantum kernel methods [34, 35] and quantum reservoir computing [36]. The main contributions of this thesis are summarized in the following.

Quantum kernel methods use a function called the quantum kernel to utilize the quantum-enhanced feature space for pattern recognition tasks such as classification. Due to the provable advantages of tailored quantum kernels for specific tasks [31–33], many researchers explore their utility in practical situations. On the other hand, there are caveats when quantum kernel-based learning models are used for actual machine learning tasks. For instance, users should choose the quantum feature map, which heavily depends on the performance. In addition, the commonly-used fidelity-based quantum kernel suffers from implementation and trainability problems as the quantum system size increases (the so-called vanishing similarity issue) [37–39]. Here, we address these two issues. For the first case, we propose a quantity to assess the classification performance of quantum feature maps, which can help screen suitable quantum feature maps among many candidates [40]. We also examine the effectiveness of the synthesis approach to construct a powerful quantum feature map. As for the second issue, we theoretically analyze the issue for the fidelity-based quantum kernel and then propose a new class of quantum kernels called quantum Fisher kernels as a circumventing approach to the vanishing similarity issue [38]. We analytically and numerically demonstrate that our proposal can circumvent the problem when shallow alternating layered ansatzes are used. We elaborate on these results in Chapter 4.

Another typical quantum-enhanced machine learning model is quantum reservoir computing, where quantum-enhanced feature space is exploited for temporal information processing tasks. The core of quantum reservoir computing lies in a quantum reservoir, an input-driven quantum system, which plays a role in the feature extractions of time-series data. However, there is room for investigation in designing better-performing and efficiently implementable quantum reservoirs. With a focus on the problem, we propose a new quantum reservoir computing framework that makes use of unavoidable quantum noise to enhance the power of temporal information pro-

cessing. We experimentally demonstrate that seemingly harmful quantum noise can be utilized to enrich the power of sequential data processing [41]. We also analyze the temporal information processing capabilities induced by quantum noise via a tool called temporal information processing [42]. Our analysis clarifies dissipation noise such as the amplitude damping noise can induce information processing capabilities. We detail these results in Chapter 5.

The rest of this thesis is organized as follows. Chapter 2 provides an introduction to quantum computing and a brief overview of quantum machine learning. Then, in Chapter 3, we review quantum-enhanced machine learning and explain quantum kernel methods and quantum reservoir computing in detail. This chapter will facilitate the understanding of the main topics discussed in the thesis. The main parts of this thesis are in Chapter 4 and Chapter 5. In Chapter 4, we present our results on quantum kernel methods. More precisely, we discuss analysis and synthesis methods for quantum feature maps in Sec. 4.1, and propose a new quantum kernel to mitigate the vanishing similarity issue in Sec. 4.2. Next, Chapter 5 develops our new framework of quantum reservoir computing, where quantum noise intrinsic in actual quantum hardware is exploited to enhance temporal information processing. We provide a proof-of-principle demonstration in Sec. 5.1 and a quantitative analysis of temporal information processing abilities in Sec. 5.2. Lastly, we conclude this thesis and present outlooks in Chapter 6.

Chapter 2

Quantum Computing

In this chapter, we explain the fundamentals of quantum computing. We first review building blocks of quantum computation, such as quantum bits, quantum gates, measurements, and quantum circuits. Then, we present the density operator formalism, followed by introducing the quantum operation formalism, a mathematical framework necessary to describe more general transformations of quantum systems. Subsequently, we discuss the current situation of quantum hardware. Finally, we briefly introduce machine learning techniques using quantum computers known as *quantum machine learning* to clarify the topics covered in this thesis. We remind readers of the literature [43] for more details of quantum computing and its background.

2.1 Basics of Quantum Computation

2.1.1 Dirac Notation for Quantum States

In quantum computing, information is processed using quantum bits represented by two-level quantum systems. Dirac notation is commonly used in quantum mechanics to describe such quantum systems, and we will adopt it throughout this thesis. Hence, this subsection briefly introduces the representation of quantum states in Dirac notation.

In quantum mechanics, quantum states are represented by column vectors in a Hilbert space, a complex vector space equipped with an inner product. For example, a quantum state labeled ψ in the d -dimensional complex vector space is represented as

$$|\psi\rangle = \begin{pmatrix} z_1 \\ \vdots \\ z_d \end{pmatrix} \in \mathbb{C}^d \quad (2.1)$$

with complex numbers $\{z_i \in \mathbb{C} | i = 1, \dots, d\}$. The representation of quantum states is called Dirac notation. In this notation, $|\cdot\rangle$ is called a ket vector that denotes a state of a certain quantum system, and the self-adjoint of the ket vector is called a bra vector represented as $\langle \cdot |$. We note that $A^\dagger = (A^*)^T$ is the conjugate transpose of A , where A^* and A^T denote the complex conjugate and transpose of A , respectively. Using this notation, the inner product between two quantum states $|v\rangle$ and $|w\rangle$ in d -dimensional complex vector space is expressed as follows;

$$\langle v|w\rangle = \begin{pmatrix} v_1^* & \dots & v_d^* \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_d \end{pmatrix} = \sum_i v_i^* w_i \in \mathbb{C}. \quad (2.2)$$

Similarly, the outer product is expressed as

$$|w\rangle\langle v| = \begin{pmatrix} w_1 \\ \vdots \\ w_d \end{pmatrix} \begin{pmatrix} v_1^* & \dots & v_d^* \end{pmatrix} = \begin{pmatrix} w_1 v_1^* & w_1 v_2^* & \dots & w_1 v_d^* \\ w_2 v_1^* & w_2 v_2^* & \dots & w_2 v_d^* \\ \vdots & \vdots & \ddots & \vdots \\ w_d v_1^* & w_d v_2^* & \dots & w_d v_d^* \end{pmatrix}. \quad (2.3)$$

2.1.2 Quantum Bits (Qubits)

Bits play an essential role in information processing in classical computers. A bit, short for binary digit, is the smallest unit of information and can have either one of two states, 0 or 1. Quantum computation has a similar concept corresponding to the bit. It is called a *quantum bit* or a *qubit*. Like classical bits, a qubit can represent two quantum states, $|0\rangle$ and $|1\rangle$; quantum states $|0\rangle$ and $|1\rangle$ reside in the Hilbert space $\mathcal{H} = \mathbb{C}^2$ and corresponds to 0 and 1 for the classical bit, respectively. These are also called computational basis states. Then, each state is expressed as follows:

$$|0\rangle \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad |1\rangle \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}. \quad (2.4)$$

Unlike classical bits, which can only have 0 or 1 states simultaneously, a single qubit can express a superposition of $|0\rangle$ and $|1\rangle$. That is, any single-qubit state $|\psi\rangle$ can be represented as

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}, \quad (2.5)$$

where $\alpha, \beta \in \mathbb{C}$ satisfies the normalization condition, i.e.,

$$|\alpha|^2 + |\beta|^2 = 1. \quad (2.6)$$

Eq. (2.5) can also be expressed as

$$|\psi\rangle = e^{i\gamma} \left(\cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle \right) \quad (2.7)$$

with $\theta \in [0, \pi]$, $\phi \in [0, 2\pi)$, and $\gamma \in \mathbb{R}$. We remark that $e^{i\gamma}$ is called the global phase that is meaningless from the viewpoint of measurement. Therefore, we can ignore the term and rewrite a single-qubit state as

$$|\psi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle. \quad (2.8)$$

From Eq. (2.8), it turns out that a single-qubit state can be visualized on the surface of the unit three-dimensional sphere as shown in Fig. 2.1. This sphere is called the *Bloch sphere*, and the three-dimensional vector \mathbf{r} that determines the quantum state on this sphere is called the *Bloch vector*. This Bloch vector $\mathbf{r} = (r_x, r_y, r_z)^T$ can be represented by expectation values of Pauli operators;

$$\mathbf{r} = \begin{pmatrix} r_x \\ r_y \\ r_z \end{pmatrix} = \begin{pmatrix} \langle \psi | \sigma_x | \psi \rangle \\ \langle \psi | \sigma_y | \psi \rangle \\ \langle \psi | \sigma_z | \psi \rangle \end{pmatrix} = \begin{pmatrix} \sin \theta \cos \phi \\ \sin \theta \sin \phi \\ \cos \theta \end{pmatrix}, \quad (2.9)$$

where Pauli operators are defined as

$$\sigma_x \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \quad (2.10)$$

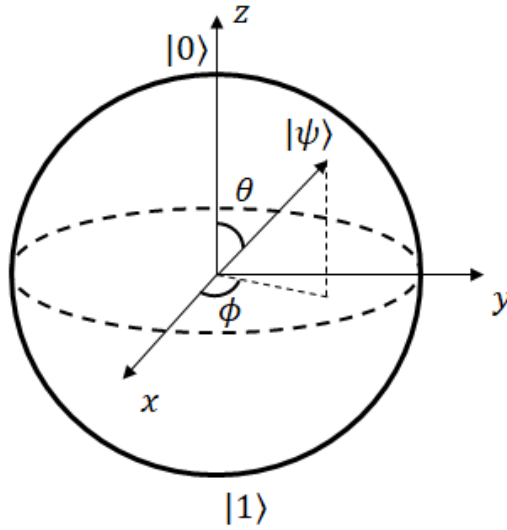


Figure 2.1: Bloch sphere representation of a single-qubit state. A pure quantum state can be plotted on the surface of the sphere. Here, the north and south poles correspond to $|0\rangle$ and $|1\rangle$, respectively. As shown in Eq. (2.8), an arbitrary single-qubit state can be expressed by determining the parameters θ and ϕ .

We remind the readers that expectation values of quantum states are explained in detail in Sec. 2.1.4.

As with classical computers that use bit-strings to perform calculations, multiple qubits are used in quantum computation. The states of multiple qubits are mathematically expressed using tensor products of qubits. Let us consider an n -qubit system. Then, a computational basis state of the n -qubit system can be expressed as

$$|b_1\rangle \otimes |b_2\rangle \otimes \dots \otimes |b_n\rangle \quad (2.11)$$

where $|b_i\rangle \in \{|0\rangle, |1\rangle\}$ represents the computational basis of the i -th qubit and \otimes denotes the tensor product operation. In general, a pure state composed of n qubits exists in the 2^n -dimensional Hilbert space and can be expressed as

$$\sum_{x \in \{0,1\}^n} \alpha_x |x\rangle. \quad (2.12)$$

Here, $x \in \{0,1\}^n$ denotes a bit sequence of length n and the complex-valued coefficients satisfy $\sum_{x \in \{0,1\}^n} |\alpha_x|^2 = 1$.

We lastly state an essential property of quantum mechanics, *entanglement*. Suppose we have a composite system $|\psi_{AB}\rangle$. If the system can be written as a tensor product of arbitrary pure states, i.e., $|\psi_{AB}\rangle = |\phi_A\rangle |\phi_B\rangle$, it is called a product state. On the other hand, if the state cannot be written in this way, it is called an entangled state. Entanglement is a property unique to quantum mechanics and plays a vital role in fields such as quantum computation and quantum teleportation.

2.1.3 Quantum Gates

In quantum computing, operations are performed by applying unitary operators called *quantum gates* to the qubits defined in the previous section. These quantum gates are conceptually equivalent to the logic gates in classical computation. In other words, as classical computers use logic gates such as NOT, AND, and XOR to perform logic operations, quantum computers use quantum gates for information processing. Mathematically, a quantum gate is defined as an operator $U : \mathcal{H} \rightarrow \mathcal{H}$ that acts on the Hilbert space \mathcal{H} . We notice that quantum gates must be unitary operators, which possess the following property;

$$U^\dagger U = U U^\dagger = I. \quad (2.13)$$

This means that the norm of quantum states is preserved under the operations. It also indicates that quantum gate operations are reversible.

In what follows, we first describe basic single-qubit gates commonly used in quantum computation. As mentioned above, due to the unitarity of the operations, a quantum gate can be regarded as an operation that transforms a unit vector to another unit vector in the Hilbert space. For ease of understanding, let us consider the Bloch sphere shown in Fig. 2.1. Then, we can think of a quantum gate operation as a rotation of a unit vector. Typical examples of single-qubit gates are the Pauli gates.

$$X \equiv \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y \equiv \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z \equiv \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad (2.14)$$

where X , Y , and Z correspond to the axes of rotation on the Bloch sphere. These Pauli gates transform the state $\alpha|0\rangle + \beta|1\rangle$ as follows;

$$\begin{aligned} X : \alpha|0\rangle + \beta|1\rangle &\rightarrow \alpha|1\rangle + \beta|0\rangle, \\ Y : \alpha|0\rangle + \beta|1\rangle &\rightarrow i\alpha|1\rangle - i\beta|0\rangle, \\ Z : \alpha|0\rangle + \beta|1\rangle &\rightarrow \alpha|0\rangle - \beta|1\rangle. \end{aligned}$$

Besides, typical examples are the Hadamard gate H , the phase gate S , and the T gate. Each of these gates is expressed as follows.

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad (2.15)$$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad (2.16)$$

$$T = \begin{pmatrix} 1 & 0 \\ 0 & \exp(i\pi/4) \end{pmatrix} = \exp(i\pi/8) \begin{pmatrix} \exp(-i\pi/8) & 0 \\ 0 & \exp(i\pi/8) \end{pmatrix}. \quad (2.17)$$

Moreover, there are gates that rotate quantum states around each axis of the Bloch sphere by arbitrary rotation angles. The gates can be defined by setting θ to an arbitrary rotation angle on each Pauli axis: that is,

$$Rx(\theta) \equiv e^{-i\theta X/2} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} X = \begin{pmatrix} \cos \frac{\theta}{2} & -i \sin \frac{\theta}{2} \\ -i \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}, \quad (2.18)$$

$$Ry(\theta) \equiv e^{-i\theta Y/2} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} Y = \begin{pmatrix} \cos \frac{\theta}{2} & -\sin \frac{\theta}{2} \\ \sin \frac{\theta}{2} & \cos \frac{\theta}{2} \end{pmatrix}, \quad (2.19)$$

$$Rz(\theta) \equiv e^{-i\theta Z/2} = \cos \frac{\theta}{2} I - i \sin \frac{\theta}{2} Z = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}. \quad (2.20)$$

Importantly, with these rotation gates, an arbitrary unitary operation on a single qubit can be expressed as

$$U = e^{i\alpha} Rz(\beta) Ry(\gamma) Rz(\delta) \quad (2.21)$$

with $\alpha, \beta, \gamma, \delta \in \mathbb{R}$.

Next, we introduce multiple-qubit gates, focusing on controlled gates. A controlled gate performs operations to target qubits conditionally on the state of control qubits. Such controlled operations are critical in logic operations. A typical example of controlled operations is a controlled-*NOT* gate (*CNOT*). This two-qubit gate acts on a control qubit and a target qubit. If the control qubit is $|1\rangle$, *X* gate is applied to the target qubit, while nothing is done if the control qubit is $|0\rangle$. More specifically, suppose we have a two-qubit state, and the first and second qubits are regarded as the control and target ones, respectively. Then, the gate operation is summarized as follows;

$$|00\rangle \rightarrow |00\rangle; \quad |01\rangle \rightarrow |01\rangle; \quad |10\rangle \rightarrow |11\rangle; \quad |11\rangle \rightarrow |10\rangle. \quad (2.22)$$

Also, a matrix representation of this *CNOT* gate is given by

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (2.23)$$

Another common two-qubit gate is the controlled-*Z* gate (*CZ*). This gate performs *Z* operation on the target qubit when the control qubit is $|1\rangle$, whereas it does nothing when the control qubit is $|0\rangle$. This operation is represented as

$$CZ = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \quad (2.24)$$

in matrix representation. Besides, the swap gate is also an important operation. A swap gate swaps the state of two qubits and is expressed in the following.

$$SWAP = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.25)$$

Lastly, we introduce a three-qubit gate called the Toffoli gate. The Toffoli gate only performs the *X* gate on the target qubit when two control qubits are $|11\rangle$. The logic operation of the Toffoli gate is expressed as $|b_{c1}\rangle |b_{c2}\rangle |b_t\rangle \rightarrow |b_{c1}\rangle |b_{c2}\rangle |b_t \oplus b_{c1}b_{c2}\rangle$, where first two states represent the control qubits (denoted as b_{c1} and b_{c2} , respectively) and the rest is the target qubit. Here,

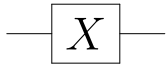
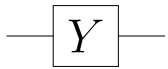
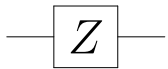
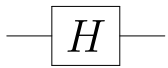
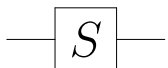
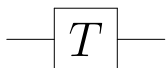
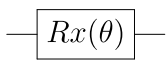
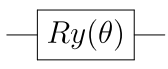
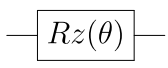
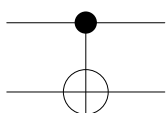
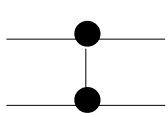
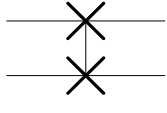
| | | | |
|-----------|---|---|--|
| X gate |  | ≡ | $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ |
| Y gate |  | ≡ | $\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ |
| Z gate |  | ≡ | $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ |
| H gate |  | ≡ | $\frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ |
| S gate |  | ≡ | $\begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ |
| T gate |  | ≡ | $\begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$ |
| Rx gate |  | ≡ | $\begin{pmatrix} \cos(\theta/2) & -i \sin(\theta/2) \\ -i \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$ |
| Ry gate |  | ≡ | $\begin{pmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{pmatrix}$ |
| Rz gate |  | ≡ | $\begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}$ |
| CNOT gate |  | ≡ | $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ |
| CZ gate |  | ≡ | $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}$ |
| SWAP gate |  | ≡ | $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ |

Figure 2.2: Lst of commonly-used single- and multiple-qubit gates.

\oplus denotes modulo two addition. Also, its matrix representation is

$$\text{Toffoli} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}. \quad (2.26)$$

Examples of these single and multiple qubit gates are shown in Fig. 2.2.

2.1.4 Measurement

We need to perform measurements to retrieve classical information from quantum states. In quantum mechanics, the probability of obtaining a certain measurement outcome is determined

probabilistically. Let us consider a quantum state $|\psi\rangle = \sum_i \alpha_i |i\rangle$. We note that $|i\rangle$ is a computational basis and $\alpha_i \in \mathbb{C}$. In this case, the probability of observing a certain basis $|j\rangle$ is given by

$$p(j) = |\langle j|\psi\rangle|^2 = |\alpha_j|^2. \quad (2.27)$$

In general, let $\{M_m\} : \mathbb{C}^d \rightarrow \mathbb{C}^d$ be a set of measurement operators, and suppose we measure a quantum state $|\psi\rangle \in \mathbb{C}^d$. Here, m denotes the measurement outcome corresponding to the operator M_m . In this case, the probability $p(m)$ of obtaining m for the quantum state $|\psi\rangle$ is defined as

$$p(m) = \langle \psi | M_m^\dagger M_m | \psi \rangle. \quad (2.28)$$

Also, the quantum state after measurement is represented as

$$\frac{M_m |\psi\rangle}{\sqrt{\langle \psi | M_m^\dagger M_m | \psi \rangle}}. \quad (2.29)$$

We remark that measurement operators satisfy the following equality:

$$\sum_m M_m^\dagger M_m = I. \quad (2.30)$$

This is because the sum of the probabilities of obtaining measurement outcomes is 1.

We can also measure physical quantities called *observables* to understand physical properties of quantum systems. The observable is mathematically represented by a Hermitian matrix A , which satisfies $A = A^\dagger$ and has real eigenvalues. A typical example of observables is a Pauli operator. Also, an arbitrary observable A for an n -qubit system can be rewritten via spectral decomposition; with Pauli strings $P_i \in \{I, X, Y, Z\}^{\otimes n}$, we can express the observable as

$$A = \sum_i a_i P_i. \quad (2.31)$$

Then, the expectation value of the observable on a quantum state $|\psi\rangle$ is written as follows;

$$\langle \psi | A | \psi \rangle = \langle \psi | \left(\sum_i a_i P_i \right) | \psi \rangle = \sum_i a_i \langle \psi | P_i | \psi \rangle. \quad (2.32)$$

In actual experiments, expectation values are computed by repeating the process of generating quantum states and measurements. More concretely, we execute a number of independent measurements followed by classical post-processing to obtain expectation values. This is because quantum states are collapsed to a certain state after measurement, as shown in Eq. (2.29). Conventionally, one round of measurement is called a *shot*. Of course, as the number of shots N_s increases, the estimation error of expectation values decreases in the scaling of $\mathcal{O}(1/\sqrt{N_s})$. Still, it is impossible to avoid statistical errors due to finite resources.

2.1.5 Quantum Circuit Models

Quantum computation involves the process of performing unitary operations on a prepared initial state and measurements. Quantum circuits are models used to represent these operations graphically. In the quantum circuit representation, there are wires corresponding to qubits. Then, quantum operations, i.e., quantum gates, are described on these wires, meaning that unitary gates act on the qubits chronologically from left to right. Finally, the measurement is denoted by a meter symbol. Fig. 2.2 shows the representation of gates used in quantum circuits.

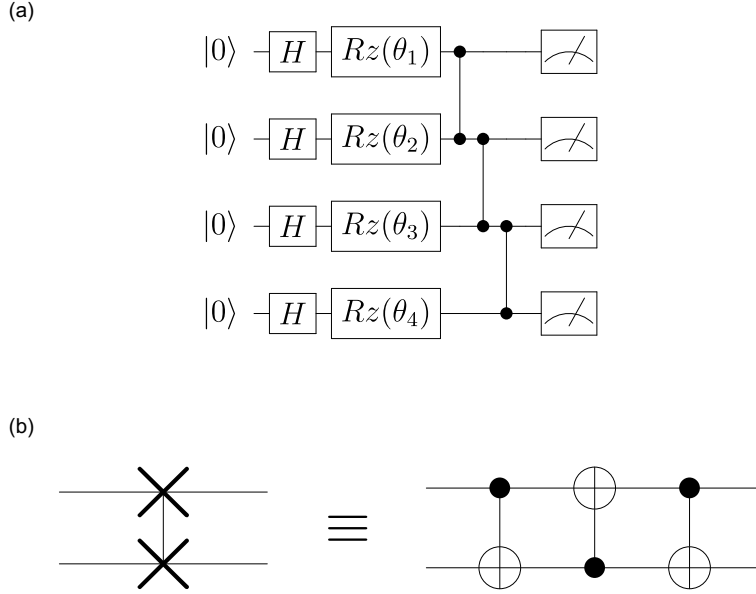


Figure 2.3: Examples of quantum circuits. Panel (a) shows a toy quantum circuit to explain the operational meaning of symbols and Panel (b) illustrates a quantum circuit that is equivalent to performing the swap operation.

In addition, Fig. 2.3 shows two concrete examples of quantum circuits. Fig. 2.3 (a) represents the following operations: (1) the Hadamard gate and R_z gates are applied to all four qubits, (2) the CZ gates are applied to the first and second, second and third, and third and fourth, in that order. Finally, all qubits were measured. Fig. 2.3 (b) shows a circuit equivalent to the $SWAP$ gate using $CNOT$ gates.

2.2 Density Matrices

Up to now, a state vector $|\psi\rangle$ has been used to represent quantum states. However, the density operator representation is more convenient than the state vector, not only because it can represent states that are mathematically equivalent to the state vectors but also because it can represent more general quantum systems. Specifically, suppose a quantum state $|\psi_i\rangle$ is sampled with probability $p_i \in [0, 1]$. In the density matrix formula, such a classical mixture of quantum states, $\{p_i, |\psi_i\rangle\}$, is expressed as follows;

$$\rho = \sum_i p_i |\psi_i\rangle \langle \psi_i| = \sum_i p_i \rho_i. \quad (2.33)$$

This state is called a mixed state. In contrast, a state represented by a state vector is called a pure state. By definition, we can immediately confirm that the density matrix can also represent pure states; that is, $\rho = |\psi\rangle \langle \psi|$.

The density operator is a non-negative operator, i.e., $\rho \geq 0$, and has the property that $\text{Tr}[\rho] = 1$, where $\text{Tr}[\cdot]$ denotes a trace operation. This can be easily shown by definition. Also, to determine whether a given quantum state is pure or mixed, we can use the *purity* defined by $\text{Tr}[\rho^2]$. Namely, $\text{Tr}[\rho^2] = 1$ for a pure state, and $\text{Tr}[\rho^2] < 1$ for a mixed state. A d -dimensional quantum state has the minimum purity of $1/d$ when I/d . This quantum state is called the maximally mixed state.

Density operators are also helpful in describing subsystems of quantum states. A composite system of quantum states is represented as a tensor product state, as in the case of state vectors. More concretely, a composite system consisting of n quantum systems is represented as $\rho_1 \otimes \rho_2 \otimes \dots \otimes \rho_n$ with the i -th quantum system ρ_i . On the other hand, an arbitrary pure state cannot always be described in such a manner because of entanglement. Consider a bipartite system $\rho^{S_1 S_2}$ consisting of two subsystems S_1 and S_2 . In this case, the subsystem S_1 is expressed as

$$\rho^{S_1} = \text{Tr}_{S_2}[\rho^{S_1 S_2}] \quad (2.34)$$

where $\text{Tr}_{S_2}[\cdot]$ denotes a partial trace operation with respect to the subsystem S_2 .

Finally, we summarize the mathematical representation of unitary operations and measurements using density operators. Unitary operations on a density operator ρ are expressed as

$$\rho \rightarrow U\rho U^\dagger. \quad (2.35)$$

Also, when the quantum state ρ is measured by a set of measurement operators $\{M_m\}$, the probability of obtaining the measurement outcome m is given by

$$p(m) = \text{tr}(M_m^\dagger M_m \rho). \quad (2.36)$$

The quantum state after the measurement reads

$$\frac{M_m \rho M_m^\dagger}{\text{tr}(M_m^\dagger M_m \rho)}. \quad (2.37)$$

2.3 Quantum Operations and Quantum Noise

Thus far, only unitary transformations and measurements have been explained as operations on quantum states. However, the unitary operator cannot describe the transformation of quantum states caused by quantum noise, for example. Therefore, we introduce a framework for describing more general quantum operations. Afterward, we show some concrete examples of quantum noise.

2.3.1 Quantum Operator Formalism

It is necessary that one can describe not only closed systems but also open systems to understand the transitions of quantum states mathematically. Quantum operator formalism is very helpful in describing various types of quantum state transitions. In the quantum operator formalism, the transformation of quantum states is expressed using density operators as follows.

$$\rho' = \mathcal{E}(\rho), \quad (2.38)$$

where ρ' represents the quantum state after the transition and $\mathcal{E}(\cdot)$ denotes quantum operations that act on a quantum state ρ .

Simple examples of quantum operations have already been shown: unitary time evolution and measurement. These are expressed as $\mathcal{E}(\rho) = U\rho U^\dagger$ and $\mathcal{E}(\rho) = M_i \rho M_i^\dagger$, respectively. There are three properties that should be satisfied for such quantum operations [43]:

- For any density operator ρ , the quantum operation $\mathcal{E}(\cdot)$ satisfies $0 \leq \text{Tr}[\mathcal{E}(\rho)] \leq 1$.
- The quantum operation $\mathcal{E}(\cdot)$ is a convex linear map of density operators, i.e.,

$$\mathcal{E}\left(\sum_i p_i \rho_i\right) = \sum_i p_i \mathcal{E}(\rho_i).$$

- The quantum operation $\mathcal{E}(\cdot)$ is completely positive. Namely, $\mathcal{E}(A)$ and $(I \otimes \mathcal{E})(A)$ are positive for any positive operator A .

In particular, quantum operations that satisfy $\text{Tr}[\mathcal{E}(\rho)] = 1$ for the first property are called completely positive trace-preserving (CPTP) maps. It is known that quantum operations describing quantum noise are always CPTP maps. The necessary and sufficient conditions to satisfy the above properties are as follows:

$$\mathcal{E}(\rho) = \sum_i E_i \rho E_i^\dagger, \quad (2.39)$$

where $\{E_i\}$ satisfies $\sum_i E_i E_i^\dagger \leq I$ and the equality holds when $\mathcal{E}(\cdot)$ is a CPTP map. This operator is also called the Kraus operator.

2.3.2 Quantum Noise

In the following, we show two examples of quantum noise that can actually occur in actual quantum hardware: the depolarizing noise and the amplitude damping noise.

First, the depolarizing noise is a quantum operation that keeps the quantum state ρ unchanged with probability $1 - p$ but replaces it with the completely mixed state I/d with probability p . We note that d denotes the dimension of the quantum state ρ . This is represented as

$$\mathcal{E}_{DEP}(\rho) = p \frac{I}{d} + (1 - p)\rho. \quad (2.40)$$

In the Kraus operator representation, the depolarizing noise channel can also be written as $\mathcal{E}_{DEP}(\rho) = \sum_i p(E_i) E_i \rho E_i^\dagger$, with the Pauli strings $E_i \in \{I, X, Y, Z\}^{\otimes n}$ and its probability $p(E_i)$.

Next, the (generalized) amplitude damping is the one that dissipates the energy in the quantum system to the external environment. Here, we consider a one-qubit system. In this case, the generalized amplitude damping can be expressed as

$$\mathcal{E}_{GAD}(\rho) = E_1 \rho E_1^\dagger + E_2 \rho E_2^\dagger + E_3 \rho E_3^\dagger + E_4 \rho E_4^\dagger, \quad (2.41)$$

where

$$E_1 = \sqrt{p_f} \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-p_e} \end{pmatrix}, \quad E_2 = \sqrt{p_f} \begin{pmatrix} 0 & \sqrt{p_e} \\ 0 & 0 \end{pmatrix}, \\ E_3 = \sqrt{1-p_f} \begin{pmatrix} \sqrt{1-p_e} & 0 \\ 0 & 1 \end{pmatrix}, \quad E_4 = \sqrt{1-p_f} \begin{pmatrix} 0 & 0 \\ \sqrt{p_e} & 0 \end{pmatrix}.$$

Here, p_f and p_e denote a probability for phase flip to occur and a probability of energy dissipation, respectively.

These noises are often used to analyze noisy quantum computation from a theoretical point of view. However, in real quantum devices, more complicated noise that the Markov model cannot describe can be present. In addition to the aforementioned noise, errors can occur in initial state preparation and measurement. There is also a noise called crosstalk, an unwanted coupling between qubits that could be difficult to model mathematically [44].

2.4 Near-Term and Long-Term Quantum Computers

Some quantum algorithms have been theoretically shown to outperform the corresponding conventional methods. These algorithms assume the use of large-scale quantum computers that are capable of error correction in the process of noisy quantum computation. Such quantum computers are called the *Fault tolerant quantum computers* (FTQCs). These days, major companies such as IBM and Google have been developing quantum hardware to realize such quantum computers. For example, IBM made its first 5-qubit device available via the cloud platform in 2016 and released a 433-qubit device in 2022. Moreover, various companies and research institutes are developing quantum devices using techniques such as superconducting qubits [45–48], photons [49, 50], ion traps [51, 52], silicon [53], and NMR [54]. While the advance of quantum hardware is rapid, it may be necessary to have millions of qubits to realize practical fault tolerant quantum computing. Thus, it would require time to get access to practical FTQCs.

Currently, small- to medium-scale devices of 50 to hundreds of qubits are available, where noise is unavoidable in quantum computation and qubit-connectivity is limited. Such quantum devices are called *noisy intermediate-scale quantum* (NISQ) computers [24]. While it is difficult for these NISQ devices to execute the quantum algorithms with the advantages, it is believed that NISQ computers still have the potential to show advantages for certain tasks. The most eye-catching example is the sampling task using a quantum computer, demonstrated by Google in 2019 [25]. Quantum supremacy has also been shown to exist even for quantum computation using noisy, shallow quantum circuits [55]. Therefore, the usefulness of NISQ devices has been investigated, with the aim of developing algorithms that can maximize the computational performance of current devices and techniques that can be adapted to large-scale FTQC algorithms. Furthermore, exploring advantages of NISQ have been facilitated by the error mitigation [56], a technique that reduces the noise level of output values obtained from quantum devices [57–62].

2.5 Application: Machine Learning

Examples of fields where quantum computers can enhance the performance of conventional methods include quantum chemistry and finance. Also, machine learning is another area where quantum advantage could be demonstrated; the emerging field is called *quantum machine learning* (QML). QML is an interdisciplinary field of quantum computing and machine learning, and thus, the term is used in broad and diverse contexts. Hence, in this section, we provide a brief introduction to QML algorithms to specify the scope of this thesis. Specifically, we explain QML methods in terms of settings, categories of methods, and types of quantum devices used.

First, approaches of QML can be divided into four types, depending on the type of data source and the type of algorithms. Fig. 2.4 illustrates the four types of settings [63]. Each of them is briefly described below.

- **CC setting:** This is the setting where classical datasets are processed using classical algorithms. This is exactly how traditional classical machine learning models work. Performing tasks on classical data with the so-called *dequantitized* algorithms [64–66] also falls within this framework.
- **QC setting:** This setting aims to perform tasks with quantum data using classical algorithms. One example is to generate quantum states via neural networks to understand the properties of quantum many-body systems; for instance, neural networks are utilized for quantum state tomography [67, 68]. In addition, deep learning and neural networks are

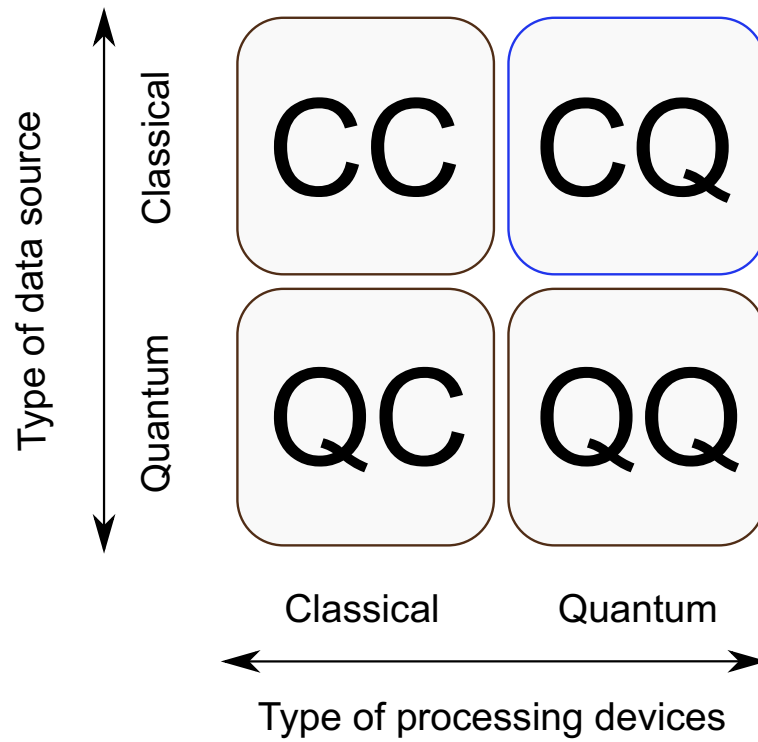


Figure 2.4: Four types of setting for QML. The settings can be categorized by types of data source (*classical data* or *quantum data*) and types of information processing devices (*classical computers* or *quantum computers*). For example, such categorization of QML can be found in Ref. [63]. CQ setting (colored in blue) is the main scope of this thesis.

used to detect the phase of matter of a given ground state [69, 70] or multipartite entanglement [71, 72]. From this perspective, the improved classical machine learning algorithm based on classical shadows [73, 74] can also be classified into this category.

- **CQ setting:** This case refers to a setting where quantum algorithms solve tasks on classical datasets. Quantum computers are used to solve tasks such as classification and regression using data generated from classical resources. Examples of the data are image data, text, and time-series data. Quantum computers are primarily used to analyze patterns in classical data and to speed up subroutines of conventional algorithms.
- **QQ setting:** In this case, both the data source and algorithms are quantum. For example, the so-called *quantum data* that is considered challenging to generate by classical means, such as ground states of quantum systems, can be used as inputs to quantum computers to solve tasks by quantum algorithms [75]. This setup is considered natural because quantum data is processed by *quantum-based* computers. Thus, this is the most promising candidate where quantum advantages can be found. Examples of tasks include quantum phase estimation and error correction.

Next, we explain the learning methods in QML. In classical machine learning, there are three categories of learning methods: supervised learning, unsupervised learning, and reinforcement learning. QML can also be categorized from the perspective of these learning paradigms. A brief characterization of these methods is given below. For more details, please refer to Ref. [2].

- **Supervised learning:** In the learning paradigm, a model is trained to reproduce a map of given input data and the corresponding desired output data. For example, given handwritten digit images and their labels (the actual digits) as training data, a model is trained to output the numbers corresponding to the images. The task is called a classification when output values are discrete or categorical variables. On the other hand, a task is called a regression problem when the outputs are continuous values.
- **Unsupervised learning:** The goal of unsupervised learning is to extract properties or features from given input data. Unlike supervised learning, desired outputs are not provided but only input values for unsupervised learning. A concrete example is clustering. In this method, groups of data are assigned based on the similarity between given data points. Another example is estimating the underlying probability distribution of given data.
- **Reinforcement learning:** This paradigm aims at learning appropriate actions of a model, called an agent, by maximizing rewards through trial and error. In contrast to supervised learning, the agent does not receive a desired output value corresponding to a reward. Instead, the agent undergoes a process where it is rewarded for good scores and penalized for bad scores according to its actions; as a result, optimal actions are acquired. The most famous example is learning the game Go [76].

Finally, we differentiate QML algorithms in terms of quantum devices used. As noted in the previous section, long-term quantum computers can demonstrate provable quantum supremacy, whereas their implementation is challenging at present. On the other hand, NISQ computers are currently available, and one can actually investigate the potential of NISQ algorithms now; however, noise cannot be avoided. In the following, we briefly summarize the features and examples of QML algorithms for each device.

- **QML algorithms with FTQC:** These algorithms mainly replace subroutines in existing classical methods with algorithms guaranteed to show quantum advantages. The primitive quantum algorithms used for the purpose include the HHL algorithm (a method for solving linear systems of equations) [18], the Grover algorithm [16], the quantum phase estimation algorithm [17], and the quantum amplitude estimation algorithm [77]. We note that the unified framework of these quantum algorithms is addressed in Ref. [78, 79]. For instance, quantum phase estimation and the HHL algorithm are used to speed up support vector machines based on kernel methods for classification tasks [29, 30]. Also, algorithms based on the HHL have been proposed for regression tasks [26–28]. For clustering, an unsupervised learning model, algorithms such as the Grover algorithm, the HHL, and quantum amplitude estimation have been proposed [80–82]. A quantum version of the principal component analysis (PCA) [83], a method used for dimensionality reduction, has also been presented using density matrix exponentiation and quantum phase estimation. A technique based on Grover’s algorithm has also been proposed for reinforcement learning [84, 85].
- **QML algorithms with NISQ devices:** These algorithms mainly perform machine learning tasks using shallow quantum circuits. The core idea is to take advantage of shallow quantum circuits, which have the potential to be more expressive than the corresponding classical machine learning models. A concrete example is the variational quantum algorithms (VQAs) [86]. VQAs learn tunable parameters in parameterized quantum circuits (PQCs) so that the output values obtained by measuring the PQCs are suitable for tasks such as classification and regression. The algorithms have also been utilized for regression tasks [87, 88], quantum classifiers [75, 89, 90], quantum generative models [91–93]. Another example is quantum kernel methods [34, 94]. As will be explained later, quantum kernel methods are supervised machine learning methods used primarily for classification tasks and have the potential of quantum advantages; it has been theoretically demonstrated that there exist datasets that are not efficiently learnable by classical models but by quantum kernels [31–33]. We note that what quantum kernel methods mainly do is to estimate a function called the quantum kernel, which can be done with the NISQ devices as well as the FTQC. Therefore, the practical advantages of the methods have been investigated. Quantum kernels can also be used for the kernel PCA, an unsupervised learning model.

In this thesis, we handle supervised QML models, quantum kernel methods and quantum reservoir computing, under CQ settings. We note that these QML algorithms are implementable on both long-term and NISQ devices. However, this thesis focuses on NISQ applications of the QML methods. We will detail these algorithms in the next section.

Chapter 3

Quantum-Enhanced Machine Learning

As mentioned in the previous chapter, there is hope that machine learning can be improved with the help of quantum computers. Specifically, the main focuses of interest are (1) to speed up the execution time of machine learning algorithms and (2) to improve the “quality of data” for pattern recognition tasks. For the first case, primitive quantum algorithms such as the HHL algorithm [18] are applied to conventional methods to achieve theoretically guaranteed speed-up. For the latter, the Hilbert space, which grows exponentially in dimension with the increase of the number of qubits, is utilized as a feature space in machine learning tasks to improve the pattern analysis [34, 87]. That is, it is believed that quantum computers are exploited to find features of data that conventional methods cannot discover, and hence, performance can be improved. We call this subfield *quantum-enhanced machine learning*. This thesis deals with two examples of quantum-enhanced machine learning: quantum kernel methods and quantum reservoir computing. Thus, in this chapter, we first explain the general concept of quantum-enhanced machine learning. In particular, we briefly review key ideas and settings of quantum-enhanced machine learning models. Then, we introduce details of quantum kernel methods and quantum reservoir computing.

3.1 Quantum-Enhanced Feature Space for Machine Learning

3.1.1 Quantum-Enhanced Feature Space

For ease of understanding, we here focus on classification tasks to explain the feature space. Given input data \mathbf{x} such as images and its corresponding label y , this task aims to train the model to output a label corresponding to the input data. Namely, the model seeks to find features inherent in the dataset through training to obtain the correct labels. However, it is generally challenging to discover intrinsic features in the original input space. We note that a space where d -dimensional input data points reside is called the input space. An approach to address the problem is to prepare a set of functions $\{\phi_k(\mathbf{x})\}$, and then construct a vector $\mathbf{x} \rightarrow \Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_M(\mathbf{x}))^T$ where $M > d$. Then, one would expect that suitable features can be found in such a nonlinearly-transformed high-dimensional space. The space spanned by the feature vectors is called a feature space. In addition, $\Phi(\cdot)$ is called a feature map. By constructing a good feature map, high performance is realizable in classification tasks.

Similarly, the quantum-enhanced feature space refers to a quantum space induced by quantum feature maps [34, 94]. We remind readers that the quantum space corresponds to the Hilbert

space. As stated, the Hilbert space increases exponentially in dimension as the number of qubits grows. This means that, even for 20 qubits, the space of size $2^{20} \approx 10^6$ can be utilized. Such a huge space is considered hard for classical means to access efficiently but could be accessible by quantum computers. Hence, using the Hilbert space for machine learning tasks could improve the performance of pattern recognition. Also, currently available quantum devices can utilize 50 to hundreds of qubits. Thus, this field has been actively investigated to find practical advantages in the NISQ era [24].

3.1.2 Quantum Feature Maps

It is necessary to define a suitable quantum feature map to fully utilize the quantum feature space. When dealing with classical data, the quantum feature mapping is performed by embedding the data into quantum states via quantum operations. To be more specific, a feature vector $|\Phi(\mathbf{x})\rangle$ in the quantum space is commonly constructed by applying the data-dependent unitary operator $U_\Phi(\mathbf{x})$ to an initial state $|\mathbf{0}\rangle$, i.e., $|\Phi(\mathbf{x})\rangle = U_\Phi(\mathbf{x})|\mathbf{0}\rangle$. This unitary operator $U_\Phi(\mathbf{x})$ is called a quantum feature map. Tailoring the quantum feature map well is critical because it determines the data structure in the quantum-enhanced feature space. We note that, when quantum data is handled in QML tasks, quantum states are used as input states in most cases; see e.g. Ref. [75, 95, 96]. In this case, quantum operations that transfer input quantum states to certain quantum states are regarded as quantum feature maps. However, since this thesis deals with only classical data, we will not dig into the details.

For a better understanding, we show some basic examples of quantum feature maps below. Note that we refer to Ref. [63].

- **Basis encoding:** This approach encodes the input x as a binary bit-string. For example, $x = 01011$ ($x = 11$) is represented as $|01011\rangle$ using 5 qubits. Thus, for the input bit-string of length n , the quantum feature map $U_\Phi(x)$ transforms the initial state as follows;

$$U_\Phi(x) : x \in \{0, 1\}^n \rightarrow |i\rangle, \quad i = \{0, 1\}^n \quad (3.1)$$

- **Amplitude encoding:** The $N(= 2^n)$ -dimensional normalized input vector $\mathbf{x} = (x_0, \dots, x_{N-1}) \in \mathbb{R}^N$ is associated with the amplitude of an n -qubit state $|\Phi(\mathbf{x})\rangle$. That is,

$$U_\Phi(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^N \rightarrow |\Phi(\mathbf{x})\rangle = \sum_{i=0}^{N-1} x_i |i\rangle. \quad (3.2)$$

- **Copies of a quantum state:** This approach constructs a state $|\psi_{\mathbf{x}}\rangle$ obtained by amplitude encoding with d replicas.

$$U_\Phi(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^N \rightarrow |\Phi(\mathbf{x})\rangle \otimes \dots \otimes |\Phi(\mathbf{x})\rangle. \quad (3.3)$$

- **Product encoding:** This approach encodes a input vector $\mathbf{x} = (x_0, \dots, x_{N-1}) \in \mathbb{R}^N$ into separate qubits using tensor products. As an example, the unitary operator corresponding to $|\Phi(x_i)\rangle = \cos x_i |0\rangle + \sin x_i |1\rangle$ for $i = 1, \dots, N$ is represented as

$$U_\Phi(\mathbf{x}) : \mathbf{x} \in \mathbb{R}^N \rightarrow \left(\begin{array}{c} \cos x_1 \\ \sin x_1 \end{array} \right) \otimes \dots \otimes \left(\begin{array}{c} \cos x_N \\ \sin x_N \end{array} \right) \in \mathbb{R}^{2^N}. \quad (3.4)$$

In practical situations, a quantum feature map is generated by constructing a data-dependent quantum circuit. Specifically, data is usually encoded into rotation angles of unitary gates in quantum circuits. Below are three typical examples of feature map circuits used for QML tasks.

- **Tensor product quantum circuits:** This quantum circuit is represented by the tensor product of single-qubit gates.

$$U_{\Phi}(\mathbf{x}) = \bigotimes_{k=1}^n V_k(\mathbf{x}) \quad (3.5)$$

with the number of qubits n . Concrete examples of V_k are Rx gate ($V_k(\mathbf{x}) = e^{-ix_k X_k/2}$) and $RyRz$ gate ($V_k(\mathbf{x}) = e^{-ix_k Z_k/2} e^{-ix_k Y_k/2}$), where the k -th element of the data x_k is injected into the rotation angle. We note that

$$\sigma_k = I \otimes \dots \otimes \underbrace{\sigma}_{k\text{-th qubit}} \otimes \dots \otimes I, \quad \sigma \in \{X, Y, Z\}. \quad (3.6)$$

- **IQP-based quantum circuits:** This quantum circuit is defined as follows;

$$U_{\Phi}(\mathbf{x}) = V_{\Phi}(\mathbf{x}) H^{\otimes n} V_{\Phi}(\mathbf{x}) H^{\otimes n} \quad (3.7)$$

with $H^{\otimes n}$ the Hadamard gates acting on all qubits in parallel. Here, $V_{\Phi}(\mathbf{x})$ is represented as

$$V_{\Phi}(\mathbf{x}) = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\mathbf{x}) \prod_{i \in S} Z_i\right), \quad (3.8)$$

where $\phi(\mathbf{x}) \in \mathbb{R}$ is an arbitrary function using data \mathbf{x} and $|S| \leq 2$. Here, ZZ gates are applied to only adjacent qubits. The IQP-based quantum circuit is proposed in Ref. [34] and has been suggested to have the potential to exhibit quantum supremacy. Originally, IQP (Instantaneous Quantum Polynomial) is a model proposed in Ref. [97], where it has been shown that the Polynomial Hierarchy collapses to its third level if its probability distribution can be efficiently sampled by a classical computer [97, 98]. Namely, it is conjectured that a classical computer cannot efficiently simulate the model. As this quantum circuit is based on the IQP circuit, the circuits' output is also considered intractable.

- **Hamiltonian evolution quantum circuits:** This quantum circuit is introduced in Ref. [99, 100] to study quantum many-body dynamics and enjoys practical usability. The quantum circuit plays a role in evolving the initial state in a data-dependent manner through the Hamiltonian $H(\mathbf{x})$. An example is a one-dimensional Heisenberg model. The quantum circuit is expressed as

$$U_{\Phi}(\mathbf{x}) = \prod_{j=1}^n \exp\left(-i \frac{t}{T} \phi(\mathbf{x}) (X_j X_{j+1} + Y_j Y_{j+1} + Z_j Z_{j+1})\right)^T, \quad (3.9)$$

where $\phi(\mathbf{x}) \in \mathbb{R}$ is an arbitrary function, T is the Trotter step, and t is arbitrary real number. As for the initial state, the tensor product of Haar-random states can be employed [37].

Moreover, quantum feature maps can comprise a quantum circuit with tunable parameters and a data-embedding circuit. More precisely, we can utilize quantum circuits in the form of

$$U_{\Phi}(\mathbf{x}, \boldsymbol{\theta}) = W(\boldsymbol{\theta}) V(\mathbf{x}). \quad (3.10)$$

Here, the unitary operator $W(\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$ is called a parametrized quantum circuit (PQC). A data-dependent unitary $V(\mathbf{x})$ is called an embedding quantum circuit. Parameters

are introduced to increase the model’s flexibility and make it easier to find a quantum feature space suitable for specific tasks. Quantum feature maps can also be constructed by repeating an embedding layer and a PQC layer;

$$\prod_{d=1}^L W_d(\boldsymbol{\theta})V_d(\mathbf{x}) = W_L(\boldsymbol{\theta}_L)V_L(\mathbf{x}) \dots W_2(\boldsymbol{\theta}_2)V_2(\mathbf{x})W_1(\boldsymbol{\theta}_1)V_1(\mathbf{x}). \quad (3.11)$$

This type of quantum circuit is called a data re-uploading quantum circuit [101]. It is known that this technique can improve the expressivity of quantum circuit models. Lastly, we remark that the quantum feature mapping uses various types of quantum circuits for W and V . Examples are hardware efficient ansatzes and the so-called alternating layered ansatzes [102], which are practically convenient to implement. In the context of QML, a quantum feature map, which can efficiently encode discrete variables into quantum states via a technique known for quantum information called quantum random access codes (QRAC) [103–105], has also been developed [106]. In the field of quantum chemistry, quantum circuits that take into account the properties of the physical system under investigation have also been proposed [107–109].

3.1.3 Models

Quantum-enhanced machine learning models can be broadly divided into *explicit models* and *implicit models*. Such categorization can be seen e.g., in Ref. [106, 110–112]. The main difference lies in how the features in the quantum-enhanced feature space are extracted to construct models. Specific features of each model are described below.

- **Explicit models:** The model takes expectation values of parametrized observables with respect to the data-embedded quantum state as the outputs. Mathematically, the output of the model is described by

$$f_{\boldsymbol{\theta}}(\mathbf{x}) = \text{Tr} \left[MU(\mathbf{x}, \boldsymbol{\theta})\rho_0 U^\dagger(\mathbf{x}, \boldsymbol{\theta}) \right] = \text{Tr} \left[\tilde{M}_{\boldsymbol{\theta}}\rho_{\mathbf{x}} \right], \quad (3.12)$$

where $U(\mathbf{x}, \boldsymbol{\theta}) = W(\boldsymbol{\theta})V(\mathbf{x})$, $\rho_{\mathbf{x}} = V(\mathbf{x})\rho_0 V^\dagger(\mathbf{x})$ and $\tilde{M}_{\boldsymbol{\theta}} = W^\dagger(\boldsymbol{\theta})MW(\boldsymbol{\theta})$. These models optimize the parameter $\boldsymbol{\theta}$ to find features in the quantum-enhanced space suitable for machine learning tasks. The model is often called a *quantum neural network* (QNN). The models also include the case when the data re-uploading quantum circuits are used for the quantum feature map, i.e., $U(\mathbf{x}, \boldsymbol{\theta}) = \prod_{d=1}^L W_d(\boldsymbol{\theta})V_d(\mathbf{x})$. In this case, the output is represented as $f_{\boldsymbol{\theta}}(\mathbf{x}) = \text{Tr} \tilde{M}_{\boldsymbol{\theta}}\rho_{\mathbf{x}, \boldsymbol{\theta}}$, where $\rho_{\mathbf{x}} = \tilde{V}(\mathbf{x}, \boldsymbol{\theta})\rho_0 \tilde{V}^\dagger(\mathbf{x}, \boldsymbol{\theta})$ with $\tilde{V}(\mathbf{x}, \boldsymbol{\theta}) = V_L(\mathbf{x})(\prod_{d=1}^L W_d(\boldsymbol{\theta})V_d(\mathbf{x}))$ and $\tilde{M}_{\boldsymbol{\theta}} = W_L^\dagger(\boldsymbol{\theta})MW_L(\boldsymbol{\theta})$.

- **Implicit models:** The model uses the inner product of features in the quantum-enhanced feature space to represent the output value. The inner product in the quantum feature space is represented by $k(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$ with the corresponding quantum feature vector $\Phi(\cdot)$. This function is called the quantum kernel [34, 94]. As will be explained in detail later, an example of quantum kernels is the fidelity-based quantum kernel represented by $k(\mathbf{x}, \mathbf{x}') = \text{Tr}[\rho_{\mathbf{x}}\rho_{\mathbf{x}'}]$ where $\rho_{\mathbf{x}} = V(\mathbf{x})\rho_0 V^\dagger(\mathbf{x})$. With the kernel, the output is expressed as

$$f_{\boldsymbol{\alpha}}(\mathbf{x}) = \sum_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) \quad (3.13)$$

These models optimize the parameter $\boldsymbol{\alpha}$ through training. This approach is called the *kernel methods*; we describe the details of the methods in the next section. Quantum

kernels can also be computed using quantum feature maps containing PQC layers $U(\mathbf{x}, \boldsymbol{\theta})$, such as data-reuploading quantum circuits. In this case, parameters in the quantum-enhanced feature space $\boldsymbol{\theta}$ are also optimized to find optimal quantum features suitable for the tasks. This learning strategy is called *quantum metric learning* [113].

As described above, quantum-enhanced machine learning models can be broadly classified into two categories in terms of how the models are constructed. However, these two models have the same aspect: the quantum-enhanced feature space is utilized to improve data quality. Furthermore, many supervised quantum machine learning models such as QNNs can be recast as kernel-based learning models, i.e., the implicit models. For the details of the proof for the mathematical equivalence, see Ref. [114]. This indicates the importance of the implicit models in the QML community.

3.2 Quantum Kernel Methods

Quantum kernel methods are popular QML techniques that leverage the quantum-enhanced feature space. The method is a quantum extension of kernel methods in classical machine learning. In kernel methods, functions called the *kernels* extract features inherent in input data by nonlinearly transforming the data onto a feature space. Then, the output values of kernels are used for machine learning tasks such as classification and regression. Especially, this is often used for classification tasks in combination with classifiers such as support vector machines (SVMs). In quantum kernel methods, similar to the classical case, quantum kernels are used for feature extraction. The point is that quantum computers are used to calculate quantum kernels, which are considered challenging for classical means to estimate efficiently. With quantum kernels, classical classifiers such as SVM are used for classification. As quantum kernels can be intractable for classical computers in specific situations, quantum kernel methods have the potential to outperform the classical counterparts [31–33]. We notice that quantum SVM implemented on the FTQCs [30] can also be used for the tasks, but this thesis focuses on classical classifiers with quantum kernels.

In this section, we provide the details of classical kernel methods and then introduce quantum kernel methods. We also briefly review SVMs used as classification algorithms.

3.2.1 Kernel Methods

Kernel methods are machine learning techniques used in pattern recognition, where the goal is to find structures and regularities in data. The core idea of the methods is to nonlinearly map the data points in the high-dimensional space so that the pattern can be easily recognized in that space called feature space. *Kernel functions* or *kernels* implicitly perform such nonlinear mapping and outputs the similarity of data in the feature space. The kernel is a real-valued function that takes two data points \mathbf{x} and \mathbf{x}' as inputs; given a pair of data \mathbf{x} and \mathbf{x}' , the kernel $k(\mathbf{x}, \mathbf{x}')$ outputs a scalar value which tells how close the data pair is. The necessary and sufficient conditions for valid kernels are as follows;

1. The function is symmetric.

$$k(\mathbf{x}, \mathbf{x}') = k(\mathbf{x}', \mathbf{x}). \quad (3.14)$$

2. A matrix G whose (i, j) element is the kernel given a pair of data \mathbf{x}_i and \mathbf{x}_j , i.e., $G_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ is called the Gram matrix or the kernel matrix. Then, the Gram matrix should be positive semidefinite. That is,

$$\sum_{i,j}^N c_i c_j G_{ij} \geq 0 \quad (3.15)$$

with N data points $\{\mathbf{x}_i\}_{i=1}^N$ and $c_i \in \mathbb{R}$ for $i = 1, \dots, N$.

Some examples of the commonly used kernel functions are the Gaussian kernel

$$k(\mathbf{x}, \mathbf{x}') = \exp(-\|\mathbf{x} - \mathbf{x}'\|^2/2\sigma^2), \quad (3.16)$$

the sigmoid kernel

$$k(\mathbf{x}, \mathbf{x}') = \tanh(\gamma \mathbf{x} \cdot \mathbf{x}' + r), \quad (3.17)$$

and the polynomial kernel

$$k(\mathbf{x}, \mathbf{x}') = (\gamma \mathbf{x} \cdot \mathbf{x}' + r)^M, \quad (3.18)$$

with $\sigma, \gamma, r, M \in \mathbb{R}$.

A kernel can be expressed as the inner product of a feature map $\Phi(\mathbf{x})$, i.e., $k(\mathbf{x}, \mathbf{x}') = \Phi^T(\mathbf{x})\Phi(\mathbf{x}')$. We take the polynomial kernel with degree $M = 2$ and $\gamma = 1$ as an example. In this case, the kernel can be rewritten as

$$\begin{aligned} (1 + \mathbf{x} \cdot \mathbf{x}')^2 &= (1 + x_1 x'_1 + x_2 x'_2)^2 \\ &= 1 + 2x_1 x'_1 + 2x_2 x'_2 + (x_1 x'_1)^2 + (x_2 x'_2)^2 + 2x_1 x'_1 x_2 x'_2 \\ &= \Phi^T(\mathbf{x})\Phi(\mathbf{x}'). \end{aligned} \quad (3.19)$$

Here we denote $\Phi(\mathbf{x})$ and $\Phi(\mathbf{x}')$ as

$$\Phi(\mathbf{x}) = \left(1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2\right)^T \quad (3.20)$$

and

$$\Phi(\mathbf{x}') = \left(1, \sqrt{2}x'_1, \sqrt{2}x'_2, x_1'^2, x_2'^2, \sqrt{2}x'_1 x'_2\right)^T, \quad (3.21)$$

respectively. We note that the Gaussian kernel can be expressed as the inner product of infinite-dimensional feature vectors. Like this, all valid kernel functions can be expressed as the inner product of feature vectors. We provide a mathematical explanation for it in the following.

In kernel theory, the Hilbert space \mathcal{H}_k associated to the kernel $k(\cdot, \cdot)$ on dataset \mathcal{X} is called the reproducing kernel Hilbert space (RKHS). More concretely, given the data points \mathbf{x}, \mathbf{x}' sampled from \mathcal{X} and the kernel $k(\mathbf{x}, \mathbf{x}')$ on the dataset \mathcal{X} , the space \mathcal{H}_k is called the RKHS associated to the kernel if the space possesses the reproducing property;

$$f(\mathbf{x}) = \langle f, \Phi_{\mathbf{x}} \rangle \quad (3.22)$$

for an arbitrary function $f \in \mathcal{H}_k$. Here, we denote the kernel as $\Phi_{\mathbf{x}} \equiv k(\cdot, \mathbf{x})$ to indicate that the kernel is a function of a variable given the data point \mathbf{x} . Also, the kernel $\Phi_{\mathbf{x}}$ is called the reproducing kernel. We recall that the Hilbert space is a vector space equipped with the inner product structure; the inner product is defined as $\langle f, g \rangle_{\mathcal{H}_k}$ for $f, g \in \mathcal{H}_k$, and the space is complete. Then, from Eq. (3.22), the kernel can be expressed in terms of reproducing kernels $\Phi_{\mathbf{x}}$ and $\Phi_{\mathbf{x}'}$:

$$k(\mathbf{x}, \mathbf{x}') = \langle \Phi_{\mathbf{x}}, \Phi_{\mathbf{x}'} \rangle. \quad (3.23)$$

Namely, the kernel is the inner product of the feature vectors $\Phi_{\mathbf{x}}$ and $\Phi_{\mathbf{x}'}$ in the RKHS. This means constructing the kernel can determine the inner product in the RKHS without explicitly considering the feature vectors. We note that the RKHS can be infinite-dimensional; the infinite-dimensional feature vectors can be exploited via specific kernels such as the Gaussian kernel. Also, the Moore-Aronszajn theorem shows that the RKHS associated with a kernel is uniquely defined [115].

Furthermore, kernel theory provides statements on the trainability of kernel-based learning models. More specifically, the representer theorem [116–118] shows that solutions of optimization tasks can be expressed in terms of kernels. Let $\mathcal{D}_s = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$ be a set of N data pairs sampled from $\mathcal{X} \times \mathcal{Y}$ for the input and output space, \mathcal{X} and \mathcal{Y} , respectively. Also, let \mathcal{H}_k be the RKHS associated to the kernel $k(\mathbf{x}, \mathbf{x}')$. Then, for an arbitrary loss L and a strictly monotonic increasing function Ψ , a minimizer of the regularized optimization problem

$$\min_{f \in \mathcal{H}_k} L((\mathbf{x}_1, y_1, f(\mathbf{x}_1)), (\mathbf{x}_2, y_2, f(\mathbf{x}_2)), \dots, (\mathbf{x}_N, y_N, f(\mathbf{x}_N))) + \Psi(\|f\|_{\mathcal{H}_k}^2) \quad (3.24)$$

can be expressed as

$$f_{opt}(\mathbf{x}) = \sum_{i=1}^N \alpha_i k(\mathbf{x}_i, \mathbf{x}) \quad (3.25)$$

with $\alpha_i \in \mathbb{R}$. This means that optimal solutions in the RKHS determined by the kernel can be represented as a weighted sum of kernel functions. This can be applied even if the RKHS is infinite-dimensional; this is important because it implies that optimization problems in infinite-dimensional space can be replaced by finite-dimensional optimization problems, that is, computationally feasible problems. Moreover, since kernels naturally appear in the dual representation of the linear model, the theorem can apply to many algorithms, such as ridge regression models, SVMs, and the kernel PCA. Thus, the theorem has a critical implication in machine learning.

Lastly, we state the importance to design kernels for specific tasks. Designing performant kernels is critical to show better performance, because the feature space is determined via the kernel. Thus, some previous works proposed kernels tailored to specific tasks. For example, some kernels are designed focusing on tasks with probabilistic generative models [2, 117]. A motivation for proposing kernels derived from probabilistic models is to construct discriminative models that take advantage of generative models. Basically, the discriminative approach directly models the target value for given data, while the generative approach aims to construct models that can output given data. The generative models are not performant compared to the discriminative models, but can naturally handle missing information and rare data points. Therefore, some kernels have been proposed to combine the advantages of both approaches. A typical example is the probability product kernel [119]

$$k(p, p') = \int_{\mathcal{X}} p(\mathbf{x})^c p'(\mathbf{x})^c d\mathbf{x} = \langle p^c, p'^c \rangle_{L_2} \quad (3.26)$$

with the probability models p and p' , and $c \in \mathbb{R}^+$. Specifically, when $c = 1/2$, the kernel is called the Bhattacharyya kernel [120] defined as

$$k(p, p') = \int_{\mathcal{X}} \sqrt{p(x)} \sqrt{p'(x)} dx. \quad (3.27)$$

This kernel is named after the coefficients appearing in the Bhattacharyya distance [121]. Also, Ref. [122] proposed the Fisher kernel, which is derived from probabilistic models using information geometric quantities. The Fisher kernel is defined as

$$k(\mathbf{x}, \mathbf{x}') = g(\mathbf{x}, \boldsymbol{\theta})^T I^{-1} g(\mathbf{x}', \boldsymbol{\theta}), \quad (3.28)$$

where $g(x, \boldsymbol{\theta}) = \nabla_{\boldsymbol{\theta}} \log p(\mathbf{x}|\boldsymbol{\theta})$ is the logarithmic derivative of the probabilistic model $p(\mathbf{x})$ called the *Fisher score* [123] and I is the Fisher information matrix expressed as

$$I = \mathbb{E}_{\mathbf{x}} [g(\mathbf{x}, \boldsymbol{\theta})g(\mathbf{x}, \boldsymbol{\theta})^T]. \quad (3.29)$$

Thanks to its expressivity realized by incorporating the data information, the Fisher kernel has been applied in several areas such as computer vision [124–128].

As shown above, it is important to construct kernels so that the information of the dataset can be incorporated into models. Besides, it is also possible to construct powerful kernels using different kernels. Below are some examples;

$$k_{new}(\mathbf{x}, \mathbf{x}') = ck_a(\mathbf{x}, \mathbf{x}'), \quad (3.30)$$

$$k_{new}(\mathbf{x}, \mathbf{x}') = \exp(k_a(\mathbf{x}, \mathbf{x}')), \quad (3.31)$$

$$k_{new}(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}, \mathbf{x}') + k_b(\mathbf{x}, \mathbf{x}'), \quad (3.32)$$

$$k_{new}(\mathbf{x}, \mathbf{x}') = k_a(\mathbf{x}, \mathbf{x}')k_b(\mathbf{x}, \mathbf{x}'), \quad (3.33)$$

where $k_a(\mathbf{x}, \mathbf{x}')$ and $k_b(\mathbf{x}, \mathbf{x}')$ are valid kernels and $c > 0$. These formula enables one to seek out new kernels that can show better performance.

3.2.2 Basics of Quantum Kernel Methods

Quantum kernel methods [34, 94] share the same concept with classical kernel methods. The difference comes down to the feature space utilized via kernels. In quantum kernel methods, quantum kernels are used to measure the similarity between data points in the quantum space, i.e., the Hilbert space. As the Hilbert space exponentially grows in dimension with the increase of the number of qubits, quantum kernels are considered challenging for classical means to estimate efficiently. Thus, the quantum kernels can potentially perform better than the conventional classical methods.

Originally, Ref. [34] proposed a quantum kernel defined as

$$k_Q(\mathbf{x}, \mathbf{x}') = \text{Tr}[\rho_{\mathbf{x}}\rho_{\mathbf{x}'}], \quad (3.34)$$

where $\rho_{\mathbf{x}} = U_{\Phi}(\mathbf{x})\rho_0U_{\Phi}^{\dagger}(\mathbf{x})$ is the density matrix representation of data-dependent quantum state, which is generated by data-dependent unitary $U(\mathbf{x})$ to an arbitrary pure initial state ρ_0 . The quantum kernel can be understood as the inner product of quantum states; more precisely, the quantum kernel can be rewritten as

$$k_Q(\mathbf{x}, \mathbf{x}') = |\langle \Phi(\mathbf{x})|\Phi(\mathbf{x}') \rangle|^2 = |\langle \mathbf{0}|U_{\Phi}^{\dagger}(\mathbf{x})U_{\Phi}(\mathbf{x}')|\mathbf{0} \rangle|^2 \quad (3.35)$$

with $\rho_0 = |\mathbf{0}\rangle\langle\mathbf{0}|$. In quantum information theory, the measure is called the *fidelity* for pure states, which tells the overlap between two quantum states, i.e., the closeness of quantum states [43]. Thus, we call the quantum kernel the fidelity-based quantum kernel.

There are two main approaches to computing the fidelity-based quantum kernels: the swap test [129] and the inversion test [34]. In the swap test, an ancilla qubit is prepared in addition to quantum states $|\Phi(\mathbf{x})\rangle$ and $|\Phi(\mathbf{x}')\rangle$, then Hadamard gates and a controlled-swap gate are applied. Lastly, measurements are performed on the ancilla qubit. As one can easily show, the fidelity corresponds to the expectation value of the Pauli Z operator on the ancilla qubit. Thus, we repeat the procedure a sufficient amount of times to get an approximate value of the fidelity. Fig. 3.1 (a) shows the quantum circuit representation of the swap test. As for the inversion test,

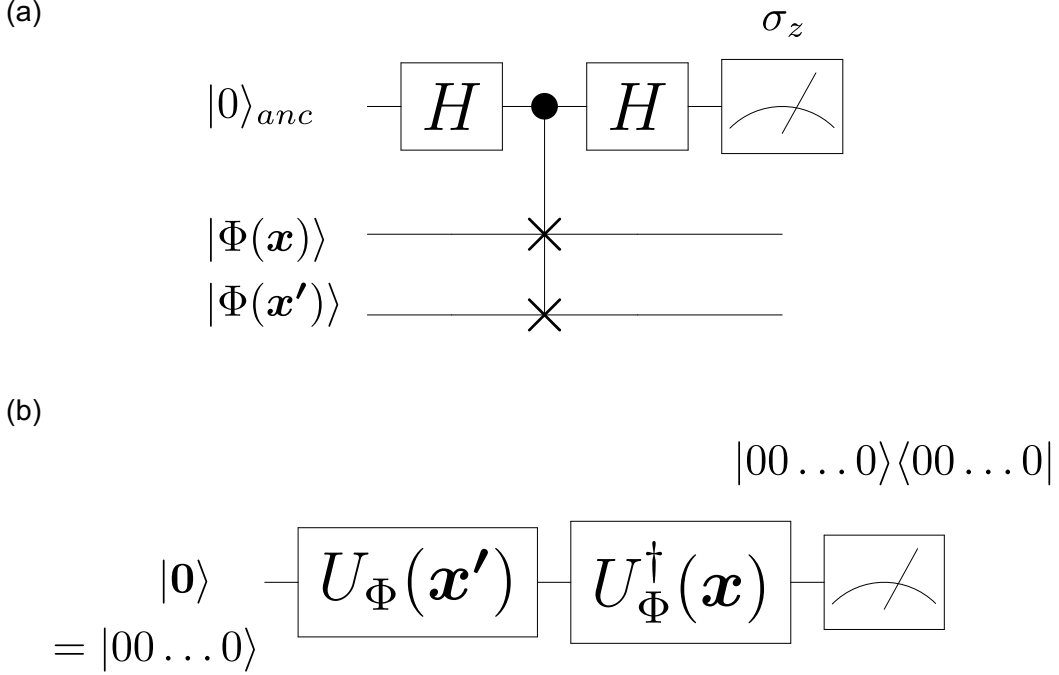


Figure 3.1: Two approaches to computing the fidelity-based quantum kernel. Quantum circuits shown in (a) and (b) depict the swap test and the inversion test, respectively.

the quantum kernel in Eq. (3.35) is straightforwardly computed; (1) prepare an initial state $|\mathbf{0}\rangle$ (without loss of generality, the all-zero state), (2) apply $U_{\Phi}(\mathbf{x}')$ and $U_{\Phi}^{\dagger}(\mathbf{x})$ in that order, and (3) measure the probability of obtaining the initial state $|\mathbf{0}\rangle$, i.e., the probability of the all-zero bit-string. Fig. 3.1 (b) depicts the quantum circuit of the operation. These approaches have the pros and cons. The inversion test requires n qubits for n -qubit quantum states, while the swap test needs $2n + 1$ qubits. On the other hand, the quantum circuit for the inversion test could be longer than that of the swap test, which can result in more noisy outcomes; instead the swap test requires many controlled-swap gates.

Thus far, it has been theoretically proven that there exist datasets that cannot be efficiently learned by classical models but by the quantum kernel [31–33]. Some previous works have shown that quantum kernels with tailored quantum circuits can outperform existing classical algorithms for specific problems. We provide binary classification tasks with theoretical guarantees on quantum advantages and elaborate on quantum circuits used to realize supremacy.

- **Discrete logarithmic problem-based dataset** [31]: For input data $x_i \in \mathbb{Z}_p^*$, the multiplicative group of integer modulo p , its label y_i is assigned according to

$$y_i = \begin{cases} +1 & \text{if } \log_g x_i \in \left[s, s + \frac{p-3}{2} \right] \\ -1 & \text{else.} \end{cases} \quad (3.36)$$

Here, p is a large prime number, g is a generator of $\mathbb{Z}_p^* = \{1, 2, \dots, p-1\}$ and s can be arbitrarily chosen from \mathbb{Z}_p^* . The task cannot be efficiently performed by any classical algorithms, i.e., in polynomial time in $n = \lceil \log_2 p \rceil$, because of the conjecture on the classical hardness of discrete logarithmic problems [130]. On the other hand, when we can

have the following quantum feature map $U_\Phi(x)$,

$$U_\Phi(x) : x \rightarrow |\Phi(x)\rangle = \frac{1}{\sqrt{2^k}} \sum_{i=0}^{2^k-1} |x \cdot g^i\rangle, \quad (3.37)$$

the SVM with the quantum kernel in Eq. (3.34) can perform well, because there exists a hyperplane $O_{w_s} = |w_s\rangle\langle w_s|$ that can produce $\text{Tr}[O_{w_s} |\Phi(x)\rangle\langle\Phi(x)|] \in 1/\text{poly}(n)$ for data points with label +1 and $\text{Tr}[O_{w_s} |\Phi(x)\rangle\langle\Phi(x)|] = 0$ for data labeled -1 [31]. The quantum feature map can be efficiently prepared on FTQCs using a subroutine algorithm in Shor's algorithm [15, 131]; the quantum circuit is implementable in Bounded-Error Quantum Polynomial-Time (BQP).

- **k -Forrelation-based dataset** [32]: Originally, the k -Forrelation is a promise problem introduced by Ref. [132]. Consider k Boolean functions $f_1, f_2, \dots, f_k : \{0, 1\}^n \rightarrow \{-1, +1\}$, and the quantity

$$\Psi_{f_1, \dots, f_k} = \frac{1}{2^{k+1}n/2} \sum_{x_1, x_2, \dots, x_k \in \{0, 1\}^n} f_1(x_1)(-1)^{x_1 \cdot x_2} f_2(x_2)(-1)^{x_2 \cdot x_3} \dots (-1)^{x_{k-1} \cdot x_k} f_k(x_k) \quad (3.38)$$

with $x \cdot y = \sum_i x_i y_i$. Then, the problem is to decide $|\Psi_{f_1, \dots, f_k}| \leq 1/100$ or $\Psi_{f_1, \dots, f_k} \geq 3/5$ for all f_1, f_2, \dots, f_k . The problem is known as PromiseBQP-complete, which includes BQP. The authors in Ref. [32] utilize the k -Forrelation problem to derive a binary classification problem, which also falls into PromiseBQP-complete. Here, given an input bit-string of length kn , where each Boolean function f_i is encoded into n -bits in specific manners, Π_+ is assigned to the input if $\Psi_{f_1, \dots, f_k} \geq 3/5$ and Π_- is assigned if $|\Psi_{f_1, \dots, f_k}| \leq 1/100$. Note that the inputs are promised to belong to Π_+ or Π_- . Based on results in Ref. [132] where the authors demonstrated a quantum algorithm that can solve the k -Forrelation problem efficiently, the following quantum feature map can capture the regularity of the k -Forrelation-based dataset;

$$U_\Phi(\mathbf{x}) = H^{\otimes n} V_{f_k} \dots H^{\otimes n} V_{f_2} H^{\otimes n} V_{f_1} H^{\otimes n} \quad (3.39)$$

where H is the Hadamard gate and the Boolean function dependent unitary V_{f_i} satisfies $V_{f_i} |x\rangle = f_i(x) |x\rangle$ for any $x \in \{0, 1\}^n$. More concretely, the probability of obtaining the all-zero bit-string by measuring the quantum state $|\Phi(\mathbf{x})\rangle = U_\Phi(\mathbf{x}) |0\rangle^{\otimes n}$ is the same as the quantity in Eq. (3.38). Thus, the SVM with the quantum kernel obtained using the quantum feature map can solve the problem. However, it is conjectured that existing classical algorithms cannot solve the problem efficiently [132].

In addition, a procedure to screen the intrinsic quantum advantages of the quantum kernel methods [37] has been recently proposed, resulting in explorations of real-world datasets [133, 134].

3.2.3 Support Vector Machines

Support vector machines (SVMs) are supervised machine learning models used for classification and regression tasks. Due to its high performance in pattern recognition tasks, SVMs in combination with kernel methods are often used in practical situations. In what follows, we explain the mathematical models of SVMs and then show the connection to kernel methods.

For ease of understanding, we consider binary classification tasks where the goal is to predict the class $C = \{+1, -1\}$ of new unseen data \mathbf{x}_{new} , given a training dataset composed of N pairs

of d -dimensional input data $\mathbf{x}_i \in \mathbb{R}^d$ and its label $y_i \in C$. The concept of SVMs is to acquire a linear model that can maximize the so-called *margin*. Suppose we have a linear classifier represented as

$$\tilde{y}(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b \quad (3.40)$$

where $\phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_M(\mathbf{x}))^T$ is a M -dimensional vector containing basis functions and (\mathbf{w}, b) are parameters to be trained. Then, the SVM aims to obtain the parameters (\mathbf{w}, b) so that the margin, the minimum distance between the decision hyperplane and any data points, is maximized.

In case all training data points are linearly separable, the distance between the hyperplane and a data point \mathbf{x}_i is given by

$$\frac{y_i \tilde{y}(\mathbf{x}_i)}{\|\mathbf{w}\|} = \frac{y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b)}{\|\mathbf{w}\|}. \quad (3.41)$$

As the objective of SVMs is to maximize the margin, the problem can be reduced to maximizing $1/\|\mathbf{w}\|$ under a constraint $y_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b) \geq 1$ for all $i \in \{1, \dots, N\}$. Equivalently, the problem is to obtain the minimum value of $\|\mathbf{w}\|^2$. Thus, the optimal solution to maximize the margin can be written as

$$\operatorname{argmin}_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2. \quad (3.42)$$

Also, the problem of minimizing a quadratic function under the constraint can be rewritten as a dual problem by introducing Lagrange multipliers $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_N)^T$ and the Karush-Kuhn-Tucker conditions (KKT conditions). More concretely the problem is to minimize the following cost function;

$$L(\boldsymbol{\alpha}) = - \sum_{i=1}^N \alpha_i + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j \phi^T(\mathbf{x}_i) \phi(\mathbf{x}_j) \quad (3.43)$$

under constraints

$$\alpha_i \geq 0, \quad (3.44)$$

$$\sum_i \alpha_i y_i = 0. \quad (3.45)$$

The solution to the problem is then represented as

$$\tilde{y}(\mathbf{x}_{new}) = \sum_{i=1}^N \alpha_i y_i \phi^T(\mathbf{x}_i) \phi(\mathbf{x}_{new}) + b. \quad (3.46)$$

We note that the KKT conditions,

$$\alpha_i \geq 0, \quad (3.47)$$

$$y_i \tilde{y}(\mathbf{x}_i) - 1 \geq 0, \quad (3.48)$$

$$\alpha_i \{y_i \tilde{y}(\mathbf{x}_i) - 1\} \geq 0 \quad (3.49)$$

show that all data points satisfy either $\alpha_i = 0$ or $y_i \tilde{y}(\mathbf{x}_i)$. This indicates that the data points that satisfy $\alpha_i = 0$ do not contribute to the prediction of the unknown data; we only have to keep the data points for which $\alpha_i \neq 0$. The data points that satisfy $\alpha_i \neq 0$ are called the *support vectors*. The property is crucial from the practical perspective because only support vectors are used and some data information can be discarded.

We can also mitigate the assumption that all training data is separable by introducing the slack variables $\{\xi_i\}_{i=1,\dots,N}$. In case misclassification is allowed, the constraint $y_i(\mathbf{w}^T\phi(\mathbf{x}_i)+b) \geq 1$ can be replaced with

$$y_i(\mathbf{w}^T\phi(\mathbf{x}_i) + b) \geq (1 - \xi_i), \quad \forall i = 1, \dots, N. \quad (3.50)$$

The relaxation technique is called the soft margin method in contrast to the original one (the hard margin method). In this case, the optimization problem that corresponds to Eq. (3.42) for the hard margin SVM can be written as

$$\frac{1}{2}\|\mathbf{w}\|^2 + C_h \sum_i \xi_i \quad (3.51)$$

with a parameter $C_h > 0$ that controls the trade-off between the model complexity and training errors. As for the dual problem, one minimizes Eq. (3.43) under

$$0 \leq \alpha_i \leq C_h, \quad (3.52)$$

$$\sum_i \alpha_i y_i = 0. \quad (3.53)$$

Lastly, we provide the connection of SVMs and kernel methods. In the dual problem representation, the cost function in Eq. (3.43) includes the inner product of feature vectors $\phi^T(\mathbf{x})\phi(\mathbf{x}')$. Actually, it is possible to regard the inner product as a kernel, i.e., $k(\mathbf{x}, \mathbf{x}') = \phi^T(\mathbf{x})\phi(\mathbf{x}')$. Thus, the cost function can be recast in terms of kernels as

$$L(\boldsymbol{\alpha}) = -\sum_{i=1}^N \alpha_i + \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j). \quad (3.54)$$

Also, the prediction can be rewritten as

$$\tilde{y}(\mathbf{x}_{new}) = \sum_{i=1}^N \alpha_i y_i k(\mathbf{x}_i, \mathbf{x}_{new}) + b. \quad (3.55)$$

This means that SVM algorithms can handle even infinite-dimensional feature vectors by introducing kernels; optimal hyperplanes in the large feature space can be obtained without explicitly computing the feature vectors. Replacing the inner product with the kernel is called the *kernel trick*. As the technique allows flexibility in solving practical tasks, the SVMs combined with the kernel methods can perform machine learning tasks such as classification with high accuracy.

3.3 Quantum Reservoir Computing

Quantum reservoir computing is a QML approach that utilizes the complex dynamics of quantum many-body systems for time-series data processing [36, 135]. By time-series data processing, we mean information processing of time-series data, such as speech recognition, stock price prediction, and robotic control. These days, such temporal information processing tasks have been performed using machine learning models such as neural networks [136, 137]. Reservoir computing is a kind of these neural network models and has been actively studied due to its advantage of fast and stable learning [138–141]. Quantum reservoir computing is based on such a reservoir computing framework. In contrast to the conventional methods, quantum reservoir computing utilizes the quantum-enhanced feature space to extract features of time-series data.

This field has also attracted increasing attention because this approach can be implemented on today’s noisy quantum computers and could have the potential to outperform conventional methods.

This section first describes the basics of reservoir computing. Then, we introduce physical reservoir computing derived from reservoir computing, where physical systems with complex dynamics are utilized as the *reservoir*, i.e., a black box that plays a role in extracting features of time-series data. We also review physical reservoir computing as quantum reservoir computing can be included in the learning paradigm. Finally, we present the details of quantum reservoir computing.

3.3.1 Framework of Reservoir Computing

Reservoir computing is a machine learning approach used for time-series data processing [138–140]. Originally, reservoir computing was derived from recurrent neural networks (RNNs) [137] and includes a class of machine learning algorithms such as echo state network (ESN) [142] and liquid state machine (LSM) [143]. The advantages of the reservoir computing framework over conventional RNNs are stable and fast learning [141]. Typically, RNNs use a learning method called backpropagation through time (BPTT), in which the error propagates backward in time [144, 145]. This technique is widely used to adjust the weight parameters in the neural network based on the gradient of the error. However, as the size of the neural network increases, this method has a problem of extremely long learning time. In addition, the vanishing gradient issue occurs; as a result, the convergence of the learning algorithm is poor. By contrast, reservoir computing fixes the randomly connected internal neural network during the training process, and only the output part is learned by a simple learning method. The fixed randomly connected network in the middle layer is called a *reservoir*. Since only the output weights are trained, reservoir computing can realize stable and fast learning compared to RNNs.

Reservoir computing models consist of three main parts: the input layer, the reservoir, and the readout layer. First, input data transformed through the input layer is mapped nonlinearly to a higher-dimensional feature space by the reservoir. Then, the readout layer outputs the weighted sum of signals obtained from the reservoir. We note that only the weights in the readout layer are adjusted in the learning process using a simple learning algorithm, such as linear regression, so that the output can approximate the target value. The core of the method is the reservoir that nonlinearly transforms input data to a higher dimensional feature space; the reservoir plays a role in extracting the features of temporal data. Thus, the performance of the task crucially depends on the reservoir. In this sense, the concept of reservoir computing is similar to kernel methods.

In what follows, we show the mathematical model of reservoir computing. In reservoir computing, the time evolution of the reservoir network state is given by

$$\mathbf{x}_t = f(\mathbf{W}_{in}\mathbf{u}_t + \mathbf{W}\mathbf{x}_{t-1}), \quad (3.56)$$

where t represents the timestep and $\mathbf{x}_t \in \mathbb{R}^{n_{res}}$ is a n_{res} -dimensional vector representing the reservoir state at timestep t . Also, $\mathbf{u}_t \in \mathbb{R}^{n_{in}}$ is a n_{in} -dimensional input vector at timestep t . $\mathbf{W}_{in} \in \mathbb{R}^{n_{res} \times n_{in}}$ and $\mathbf{W} \in \mathbb{R}^{n_{res} \times n_{res}}$ are fixed random weights for the input-reservoir connection and recurrent connection in the reservoir layer, respectively. The function $f(\cdot)$ represents an element-wise activation function. Examples of this function $f(\cdot)$ are sigmoid functions $f(x) = 1/(1 + \exp x)$ and hyperbolic tangents $f(x) = \tanh x$. Then, the output of reservoir computing models is expressed as

$$\bar{\mathbf{y}}_t = \mathbf{W}_{out}\mathbf{x}_t, \quad (3.57)$$

where $\mathbf{y}_t \in \mathbb{R}^{n_{out}}$ is a n_{out} -dimensional output vector of the reservoir model and $\mathbf{W}_{out} \in \mathbb{R}^{n_{out} \times n_{res}}$ is the tunable weight in the readout layer. As for learning, the weight in the readout layer, \mathbf{W}_{out} , is adjusted so that the mean square error between the output vector $\bar{\mathbf{y}}_t$ and the target vector \mathbf{y}_t is minimized for all t . The optimal weights can be computed using the Moore-Penrose pseudo inverse $\mathbf{X}^+ \equiv (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$;

$$\mathbf{W}_{out}^{opt} = \mathbf{X}^+ \mathbf{Y}. \quad (3.58)$$

Here, $\mathbf{X} \equiv [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_L]^T$ and $\mathbf{Y} \equiv [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_L]^T$ for total timestep L .

3.3.2 Physical Reservoir Computing

As explained in the previous subsection, the reservoir, a randomly connected internal network, plays a vital role in reservoir computing. This is because the reservoir extracts patterns inherent in time-series data through a nonlinear mapping of input data to a high-dimensional feature space. On the other hand, the reservoir does not necessarily be a neural network since it is not adjusted through training. In other words, a system with a high degree of freedom can fulfill the role of the reservoir. Based on this idea, it is possible to use complex and nonlinear physical systems as reservoirs for temporal data processing. The reservoir computing paradigm is called *physical reservoir computing* [146,147]. Compared to conventional reservoir computing, physical reservoir computing has advantages such as faster information processing and lower power consumption. Thus far, physical reservoir computing has been implemented using physical systems such as soft robotics [148–150], photonic systems, optoelectronic systems [151–153], and analog circuits [154]. More details are provided in Ref. [146].

3.3.3 Quantum Reservoir Computing Models

Quantum reservoir computing is a type of physical reservoir computing scheme that utilizes the dynamics of complex quantum many-body systems as reservoirs for time series data processing [36]. Below, we present the details of quantum reservoir computing models.

For simplicity, we assume that the input and target time-series data are one-dimensional. That is, the input and the corresponding output of length L are represented as $u = [u_1, u_2, \dots, u_L]^T$ and $\bar{y} = [\bar{y}_1, \bar{y}_2, \dots, \bar{y}_L]^T$, respectively. Here, we also assume that $u_l \in [0, 1], l = 1, \dots, L$. In this case, the time evolution of the quantum reservoir system is described as

$$\rho_t = \mathcal{T}_{u_t}(\rho_{t-1}) \quad (3.59)$$

where t represents timestep, ρ_t is the density operator representation of quantum reservoir state at timestep t and $\mathcal{T}_{u_t}(\cdot)$ is a map depending on u_t . As shown in Eq. (3.59), the transition of the quantum reservoir system is mathematically described by the input-dependent map $\mathcal{T}_{u_t}(\cdot)$. In Ref. [36], the paper that first proposed the quantum reservoir computing paradigm, the following map is used;

$$\mathcal{T}_{u_t}(\rho_{t-1}) = e^{-iH\tau} (\rho_{u_t} \otimes \text{Tr}_1[\rho_{t-1}]) e^{-iH\tau} \quad (3.60)$$

where τ is a hyperparameter, $\text{Tr}_1[\cdot]$ denotes a partial trace operation on a qubit system where the input is injected, and H is the user-defined and fixed Hamiltonian. A concrete example of the Hamiltonian is the transverse-field Ising model, i.e., $H = \sum_{i,j} J_{ij} X_i X_j + h Z_i$ with the parameters $J_{i,j}$ and h . Also, $\rho_{u_t} = |\psi_t\rangle \langle \psi_t|$ with $|\psi_t\rangle = \sqrt{u_t} |0\rangle + \sqrt{1-u_t} |1\rangle$ represents an

input-dependent quantum state. Then, the output of the quantum reservoir computing model \bar{y}_t is given by

$$\bar{y}_t = \mathbf{W}_{out} h(\rho_t) \quad (3.61)$$

with the tunable weight \mathbf{W}_{out} . Here, $h(\rho_t)$ represents a set of expectation values $\{\langle O_j \rangle\}$ for certain observables $\{O_j\}$. We note that $\langle O_j \rangle_t = \text{Tr}[O_j \rho_t]$. Commonly, Pauli Z basis for each qubit is used for the observable, i.e., $\{O_j\}_{j=1,\dots,n} = \{Z_j\}_{j=1,\dots,n}$. In the learning process, the weights \mathbf{W}_{out} are adjusted so that the mean squared error between the output \bar{y}_t and the target y_t is minimized for all t . Sec. 3.3.1 shows how to obtain the optimal weights.

To date, quantum reservoirs have been experimentally implemented using quantum systems such as superconducting quantum computers [41, 155], NMR [156], and atoms [157]. Also, theoretical and numerical studies have been conducted to investigate what kind of systems can achieve high performance [158–164]. In addition, quantum extreme learning, inspired by quantum reservoirs (and quantum kernel methods), is also being studied [165–167]. For a more detailed review of quantum reservoir computing, see Ref. [135].

Chapter 4

Quantum Kernel-Based Learning Models

In this chapter, we discuss the practicality of quantum kernel-based learning models. As we explained in Sec. 3.2, the quantum kernel method has the potential to outperform conventional methods due to its provable quantum advantages for certain learning tasks. However, caution needs to be taken when quantum kernels are used in practical situations.

First, it is non-trivial to construct a quantum feature map suitable for specific machine learning tasks. The performance of quantum kernel methods heavily depends on the choice of feature maps. In quantum kernel methods, however, one should specify the quantum feature map, i.e., a quantum circuit, to compute quantum kernels. Actually, tailoring feature maps would be critical to show practical advantages, as fine-tuned quantum kernels can realize the provable advantages. Therefore, it is imperative to give an insight into how to design quantum feature maps.

Second, the commonly-used fidelity-based quantum kernels suffer from the so-called *vanishing similarity issue*. As stated in detail later, vanishing similarity is a phenomenon where the expectation value and the variance of the fidelity-based quantum kernels vanish exponentially as the number of qubits increases. This hinders the efficient estimation of the quantum kernel on quantum hardware. In addition, learning machines based on the quantum kernel result in overfitting and thus show poor performance to new unseen data. This suggests the need to remedy the problem.

In the following, we address these two issues. As for the first case, we provide an approach to screen suitable quantum feature maps among many candidates. Also, we demonstrate that synthesizing quantum feature maps can lead to better performance. We give the details in Sec. 4.1. For the second case, we propose a new class of quantum kernels called the *quantum Fisher kernel* to mitigate the vanishing similarity issue. We then analytically and numerically demonstrate that our proposal can avoid the issue when the so-called *alternating layered ansatzes* are shallow. We elaborate on our proposal and the obtained results in Sec.4.2.

4.1 Analysis and Synthesis of Quantum Feature Maps¹

This section discusses how to analyze and utilize the quantum feature maps for specific classification tasks. We first develop a method to analyze the feature map for kernel-based quantum classifiers. More specifically, we introduce a quantity we call the *minimum accuracy*, which tells

¹Results shown in this section are based on the author's work [40].

the linear separability of data points in the quantum feature space induced by the quantum kernel. A proof-of-concept numerical study for four benchmark classification tasks demonstrates the method’s validity. Also, we examine the efficacy of a synthesis method where different quantum kernels are combined to construct a powerful quantum kernel.

4.1.1 Introduction

Quantum feature maps play a critical role in quantum kernel methods. As shown in Sec 3.2, the quantum kernel measures the similarity between a pair of data points in the Hilbert space defined via the quantum feature map. Thus, the performance of quantum kernel-based classifiers heavily relies on the choice of quantum feature maps. In fact, SVMs with quantum kernel methods can solve binary classification tasks that are not efficiently solvable by classical models when the quantum feature maps are properly tailored to specific tasks. Therefore, choosing appropriate quantum feature maps play a significant role in quantum kernel methods.

In classical kernel methods, choosing feature maps is equivalent to trying different types of kernels or the same kernel with different hyperparameters. This is because each kernel is associated with the corresponding feature space, as shown in Sec. 3.2. On the other hand, one should explicitly determine the quantum feature map, i.e., a data-dependent unitary operator (quantum circuit), in quantum kernel methods. An approach to preparing a suitable quantum feature map is to find the best feature map among many candidates by comparing their performance on training datasets. However, this is non-trivial and computationally demanding. Thus, developing a method that can efficiently screen better-performing quantum feature maps is imperative.

This section provides one such method based on a quantity called the *minimum accuracy*, which roughly estimates the linear separability of training data points in the quantum-enhanced feature space induced by the quantum feature map. The key idea of the quantity is the projection of quantum feature vectors onto one-dimensional space to facilitate calculation of the attainable accuracy for training datasets. Since the minimum accuracy is a possible solution for the optimization problem in the RKHS, it can serve as a lower bound of the training accuracy. Also, a quantity can be determined by the chosen feature map and the training dataset. Thus, the minimum accuracy can be used for screening quantum feature maps suitable for specific tasks. To verify our proposal, we work on four binary classification benchmark tasks using SVMs with two-qubit quantum kernels. More concretely, we demonstrate our idea using IQP-based quantum feature maps with five different encoding functions. Moreover, we study the effectiveness of another approach to seeking a suitable feature map: a synthesis method. This technique combines different quantum kernels to get a performant quantum kernel. We numerically check the validity of the method for the above-mentioned binary classification tasks using SVMs with several quantum kernels.

We note that we here consider SVMs with quantum kernels as the quantum kernel-based classifiers. However, these methods are applicable to any linear classical classifiers. Also, it would be possible to apply these techniques to regression tasks.

The rest of this section is organized as follows. Firstly, we detail the minimum accuracy in Sec. 4.1.2. Then, we explain the concept of the synthesis method in Sec. 4.1.3. Next, we present a proof-of-concept demonstration of these methods for several classification tasks using two-qubit fidelity-based quantum kernels in Sec. 4.1.4. Lastly, we give a conclusion and outlooks in Sec. 4.1.5.

4.1.2 A Method to Analyze Quantum Feature Maps

In what follows, we provide the details of the minimum accuracy.

First, we give the concept of the quantity. The minimum accuracy is a maximum classification accuracy attainable when candidates of separating hyperplanes in the quantum-enhanced feature space are restricted to the set of Pauli operators. As shown in Sec. 3.2, SVMs in combination with kernel methods aim to find an optimal hyperplane in the feature space induced by the kernel; similarly, quantum kernel-based SVMs try to obtain the best hyperplane in the quantum-enhanced feature space. Notably, the hyperplane in the quantum-enhanced feature space defined by the fidelity-based quantum kernel in Eq. (3.34) can be regarded as an observable. This can be easily shown by rewriting the optimal solution of kernel machines guaranteed by the representer theorem [116–118] as follows;

$$\begin{aligned}
 f_{opt}(\mathbf{x}) &= \sum_{i=1}^N \alpha_i^{opt} k_Q(\mathbf{x}_i, \mathbf{x}) \\
 &= \sum_{i=1}^N \alpha_i^{opt} \text{Tr}[\rho_{\mathbf{x}_i} \rho_{\mathbf{x}}] \\
 &= \text{Tr} \left[\left(\sum_{i=1}^N \alpha_i^{opt} \rho_{\mathbf{x}_i} \right) \rho_{\mathbf{x}} \right] \\
 &= \text{Tr} [O^{opt} \rho_{\mathbf{x}}].
 \end{aligned} \tag{4.1}$$

In other words, finding the optimal parameters $\{\alpha_i^{opt}\}$ is equivalent to finding the optimal observable, which can be represented as a linear combination of data-dependent quantum states, i.e., $O^{opt} = \sum_{i=1}^N \alpha_i^{opt} \rho_{\mathbf{x}_i}$. We note that such an argument is presented in Ref. [114] to show the equivalence between implicit models (i.e., quantum neural networks) and explicit models (i.e., quantum kernel-based models). Based on this idea, we introduce the minimum accuracy to roughly estimate the accuracy for the training dataset.

As mentioned, the key idea is to consider only the set of Pauli operators $\sigma_i \in \{I, X, Y, Z\}^{\otimes n}$ as observables for n -qubit quantum systems; then we obtain the maximum accuracy among the candidates. We note that the accuracy is defined as the ratio of the number of correct answers N_{true} and the total number of data points N , i.e., N_{true}/N . According to the representer theorem and Eq. (4.1), these observables are possible solutions in the RKHS associated with the fidelity-based quantum kernel. Thus, the accuracy obtained for the optimized classifiers on training datasets is guaranteed equal to or greater than the minimum accuracy. This means the optimized quantum kernel-based classifiers with the quantum feature map can perform well on the training dataset if the minimum accuracy is large. Also, due to the restriction on observables under investigation, we can calculate the quantity without actually performing optimization of classifiers.

The minimum accuracy can also be regarded as the accuracy attainable when the quantum feature vectors are projected onto an optimal Pauli basis. An arbitrary n -qubit density operator can be expanded by the set of Pauli operators as

$$\rho_{\mathbf{x}} = \sum_{\sigma_i \in \{I, X, Y, Z\}^{\otimes n}} a_{\sigma_i}(\mathbf{x}) \sigma_i. \tag{4.2}$$

Then, by substituting Eq. (4.2) into Eq. (3.34), we can obtain

$$k_Q(\mathbf{x}, \mathbf{x}') = 2^n \sum_{\sigma_i \in \{I, X, Y, Z\}^{\otimes n}} a_{\sigma_i}(\mathbf{x}) a_{\sigma_i}(\mathbf{x}'). \tag{4.3}$$

Here, we utilize the trace relation of Pauli matrices, i.e., $\text{tr}(\sigma_i \sigma_j) = 2^n \delta_{ij}$ with the Kronecker delta δ_{ij} . This means that the 4^n -dimensional vector $\mathbf{a}(\mathbf{x}) = [a_{I\dots I}(\mathbf{x}), \dots, a_{Z\dots Z}(\mathbf{x})]^T$ serves as a real-valued representation of the quantum feature map because the quantum kernel is expressed as the inner product of them. Namely, the input data \mathbf{x} is encoded into the 4^n -dimensional real-valued feature space via the quantum feature map. Hence, the minimum accuracy can be interpreted as the best accuracy obtained for one-dimensional space to which the real-valued quantum feature space $\mathbf{a}(\mathbf{x})$ is projected. We notice that each element of the real-valued vector $\mathbf{a}(\mathbf{x})$ can be obtained by measuring the expectation value of the corresponding Pauli operator and post-processing, i.e., $2^n a_{\sigma_i} = \text{Tr}[\sigma_i \rho_{\mathbf{x}}]$.

We show the procedure to compute the minimum accuracy below [40]. Let $\{(\mathbf{x}_k, y_k)\}_{k=1, \dots, N}$ be the training dataset composed of N pairs of input data \mathbf{x}_k and its label $y_k \in C = \{+1, -1\}$. Also, we denote N_+ (N_-) as the number of data points labeled with $+1$ (-1). That is, $N = N_+ + N_-$ is satisfied. Moreover, we consider the n -qubit fidelity-based quantum kernel in Eq. (3.34) and $\mathbf{a}(\mathbf{x})$ represents the real-valued representation of the quantum feature map $U_{\Phi}(\mathbf{x})$.

1. For a fixed Pauli operator $\sigma_i \in \{I, X, Y, Z\}^{\otimes n}$, compute the corresponding component of the real-valued quantum feature vectors, i.e., $\{a_{\sigma_i}(\mathbf{x}_k)\}_{k=1, \dots, N}$. The process can be regarded as the projection of the quantum feature vectors onto one-dimensional space defined by the Pauli operator, as depicted in Fig. 4.1 (b).
2. Sort $\{a_{\sigma_i}(\mathbf{x}_k)\}_{k=1, \dots, N}$ in ascending order and then choose the j -th hyperplane orthogonal to the axis of the Pauli σ_i ; the hyperplane is located between the j -th and $(j+1)$ -th data points for $j \in \{1, \dots, N-1\}$ as indicated by an arrow in Fig. 4.1 (b) and (c).
3. Calculate the accuracy for the j -th hyperplane

$$R_{\sigma_i}^j = \max\{N_+ - N_+^j + N_-^j, N_- - N_-^j + N_+^j\}/N, \quad (4.4)$$

where N_+^j (N_-^j) denotes the number of data points labeled with $+1$ (-1) in the sorted vector up to the j -th element. Recall that this quantity in Eq. (4.4) corresponds to the accuracy of the data for the j -th hyperplane when the quantum feature vector is projected onto the Pauli basis.

4. Calculate the accuracy for all j -th hyperplane with $j \in \{1, \dots, N-1\}$, and then take the maximum: $R_{\sigma_i} = \max_j R_{\sigma_i}^j$.
5. The minimum accuracy is computed as $R = \max_{\sigma_i \in \{I, X, Y, Z\}^{\otimes n}} R_{\sigma_i}$.

We lastly mention a downside of the quantity. As shown in the fifth process, we have to compute R_{σ_i} for all the Pauli operators. This would be computationally infeasible with the increase in the number of qubits since the number of Pauli operators scales exponentially in n . However, we can circumvent the issue by considering the subset of the Pauli operators, i.e., $\sigma_i \in \mathcal{P} \subset \{I, X, Y, Z\}^{\otimes n}$. The remedy of the issue will be further discussed later in Sec. 4.1.5.

4.1.3 Synthesized Quantum Feature Maps

Next, we discuss a synthesis method to construct performant quantum kernels. In classical kernel methods, some strategies exist to design effective kernels from valid kernels, as explained in Sec. 3.2.1. A straightforward approach is to combine different kernels. The scheme can compensate for the weakness of each kernel and the resultant kernel might have suitable characteristics for specific tasks.

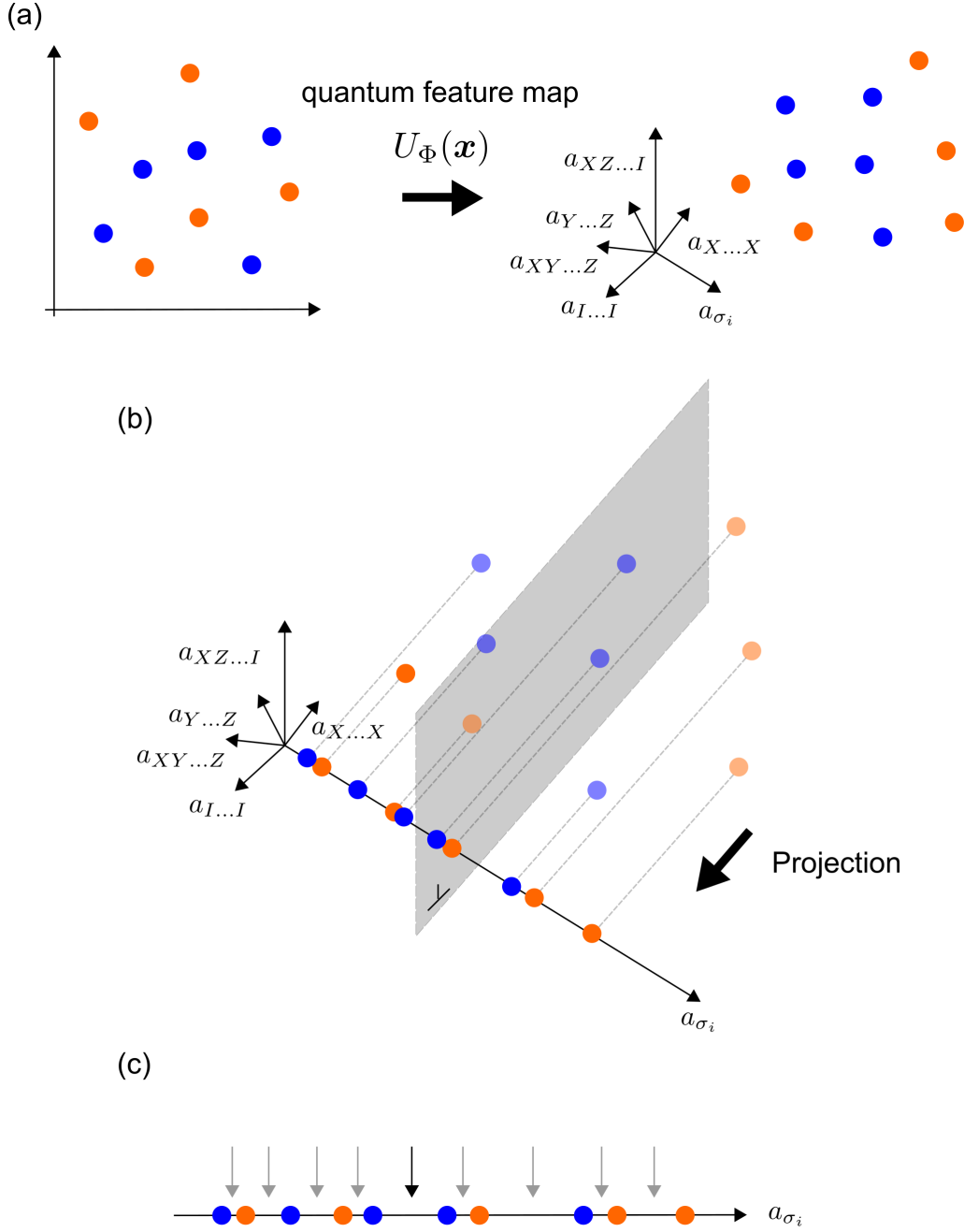


Figure 4.1: An example of calculating the minimum accuracy for ten pairs of data points $\{(\mathbf{x}_k, y_k)\}_{k=1, \dots, 10}$. The data points labeled with $+1$ and -1 are indicated in blue and orange, respectively. (a) The data is first encoded into the quantum-enhanced feature space via a quantum feature map $U_\Phi(\mathbf{x})$. Then, as shown in (b), the quantum feature vectors are projected onto the Pauli basis σ_i , where data points are classified by a chosen hyperplane. The panel (c) further illustrates a concrete example for computing $R_{\sigma_i}^j$ in Eq. (4.4). For the case $j = 5$, $N_+^5 = 3$ and $N_-^5 = 2$ (the number of blue and orange points on the left-hand side of the thick arrow). Thus, $R_{\sigma_i}^j = 0.6$. Also, by computing the quantity for all the hyperplanes, we get $R_{\sigma_i} = \max_j R_{\sigma_i}^j = 0.7$.

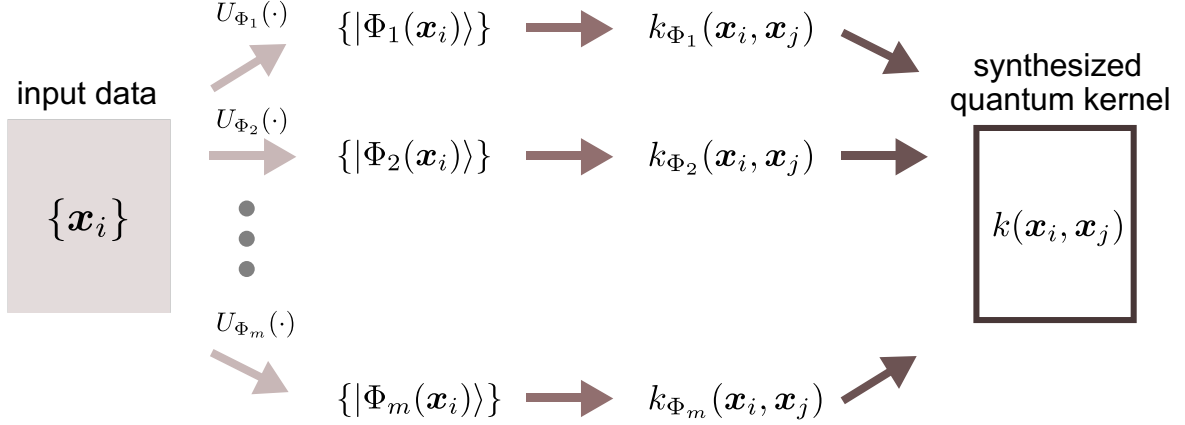


Figure 4.2: Schematic illustration of the synthesis method for quantum kernels. A concept behind this approach is to produce a performant quantum kernel from many (poor) quantum kernels.

We study the effectiveness of this idea in the quantum regime. In the NISQ era, estimating powerful quantum kernels on quantum hardware could be challenging due to the noise in devices, the limited number of qubits, and connectivity. As a result, only weak quantum kernel-based classifiers might be available. Also, due to the lack of design principles for quantum feature maps, it is non-trivial to construct a single quantum kernel that performs well. Moreover, the performance of the fidelity-based quantum kernel can be poor as the number of qubits increases [38, 39], which we discuss in detail in the next section. Thus, checking if the strategy works in the case of quantum kernels is critical. We note that the idea behind our motivation is similar to the so-called *ensemble learning*, where some weak classifiers are effectively combined to yield a powerful classifier [168]. Actually, some previous works have deeply investigated the scheme in the context of QML [35, 169].

Here, we focus on a typical method to synthesize kernels: a weighted sum of quantum kernels expressed as

$$k(\mathbf{x}_i, \mathbf{x}_j) = \sum_{l=1}^m \lambda_l k_{\Phi_l}(\mathbf{x}_i, \mathbf{x}_j), \quad (4.5)$$

where $\{\lambda_l\}$ are the weight parameters that satisfy $\sum_{l=1}^m \lambda_l = m$ and $k_{\Phi_l}(\mathbf{x}_i, \mathbf{x}_j)$ represents a quantum kernel obtained using the quantum feature map $U_{\Phi_l}(\mathbf{x})$. Fig. 4.2 depicts the synthesizing scheme of quantum feature maps. We note that the idea was briefly introduced in Ref. [170] before the publication of our work [40], but a concrete demonstration still needs to be presented. To demonstrate the approach's efficacy, we consider the following simple situation: equally weighted sum of two quantum kernels, i.e., $m = 2$ and $\lambda_1 = \lambda_2 = 1$. In this case, we can readily show that the resultant kernel can deal with a higher-dimensional feature space than the original ones:

$$\begin{aligned} k_{new}(\mathbf{x}_i, \mathbf{x}_j) &= k_{\Phi_1}(\mathbf{x}_i, \mathbf{x}_j) + k_{\Phi_2}(\mathbf{x}_i, \mathbf{x}_j) \\ &= \langle \Phi_1(\mathbf{x}_i), \Phi_1(\mathbf{x}_j) \rangle_{\mathcal{H}_{k_{\Phi_1}}} + \langle \Phi_2(\mathbf{x}_i), \Phi_2(\mathbf{x}_j) \rangle_{\mathcal{H}_{k_{\Phi_2}}} \\ &= \langle \Phi_1(\mathbf{x}_i) \oplus \Phi_2(\mathbf{x}_i), \Phi_1(\mathbf{x}_j) \oplus \Phi_2(\mathbf{x}_j) \rangle_{\mathcal{H}_{k_{new}}} \end{aligned} \quad (4.6)$$

where \mathcal{H}_k denotes the RKHS associated with the kernel k . This means that the data is encoded into the direct sum of two RKHS, and hence, the RKHS associated with the combined quantum kernel can be enlarged. Of course, we have many options to combine the kernels; see Ref. [2, 171]

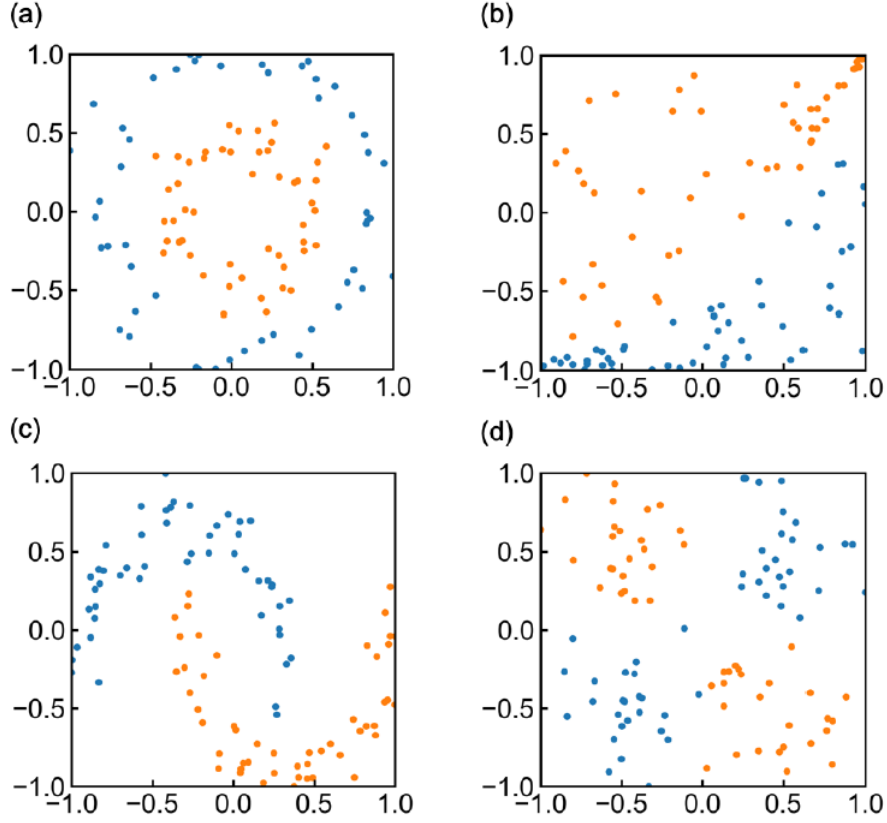


Figure 4.3: Datasets used in the numerical experiments: (a) *Circle*, (b) *Exp*, (c) *Moon* and (d) *Xor*. Reprinted figure from Ref. [40]. Copyright 2020 by Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama, and N. Yamamoto. [DOI:10.1007/s42484-020-00020-y].

for more details.

4.1.4 Numerical Demonstration

We perform numerical simulations to verify the effectiveness of the analysis and synthesis methods.

In the numerical experiments, we handle benchmark binary classification tasks with two-dimensional datasets shown in Fig. 4.3: *Circle*, *Exp*, *Moon* and *Xor*. Every dataset is composed of $N = 100$ pairs of input data and its label $\{(\mathbf{x}_k, y_k)\}_{k=1, \dots, 100}$, where $y_k \in \{-1, +1\}$ and each element of the data points lies in the range between -1 and 1 , i.e., $\mathbf{x}_k \in [-1, 1]^2$. Also, these datasets are balanced; that is, data points labeled with $+1$ and -1 are evenly distributed.

As for quantum classifiers, we focus on the framework used in Ref. [34]: (classical) SVMs with the two-qubit quantum kernel in Eq. (3.34). We note that we deal with the two-qubit quantum classifiers to set the number of qubits equal to the dimension of data points. Here, the IQP-based quantum circuits described in Sec. 4.1 are used as the quantum feature map $U_\Phi(\mathbf{x})$. For the two-qubit case, $U_\Phi(\mathbf{x})$ is expressed as

$$U_\Phi(\mathbf{x}) = V_\Phi(\mathbf{x})H^{\otimes 2}V_\Phi(\mathbf{x})H^{\otimes 2}, \quad (4.7)$$

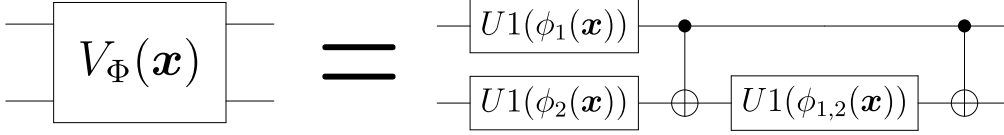


Figure 4.4: Quantum circuit representation of $V_{\Phi}(\mathbf{x})$ with the set of encoding functions $\Phi(\mathbf{x}) = \{\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \phi_{1,2}(\mathbf{x})\}$. Here $U1(\phi) = \text{diag}\{1, e^{-i\phi}\}$.

where H is the Hadamard gate and

$$V_{\Phi}(\mathbf{x}) = \exp(i\phi_1(\mathbf{x})ZI + i\phi_2(\mathbf{x})IZ + i\phi_{1,2}(\mathbf{x})ZZ). \quad (4.8)$$

We call $\Phi(\mathbf{x}) = \{\phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \phi_{1,2}(\mathbf{x})\}$ the set of *encoding functions*. The quantum circuit representation of the unitary gate $V_{\Phi}(\mathbf{x})$ is illustrated in Fig. 4.4. These encoding functions determine how the input data is embedded into the quantum state $|\Phi\mathbf{x}\rangle = U_{\Phi}(\mathbf{x})|0\rangle^{\otimes 2}$ and hence play a critical role in the feature mapping. However, users should choose the functions by themselves so that SVMs with the quantum kernel can perform well for the classification tasks. In this thesis, we prepare the following five sets of encoding functions [40];

$$\phi_1(\mathbf{x}) = x_1, \phi_2(\mathbf{x}) = x_2, \phi_{1,2}(\mathbf{x}) = \pi x_1 x_2, \quad (4.9)$$

$$\phi_1(\mathbf{x}) = x_1, \phi_2(\mathbf{x}) = x_2, \phi_{1,2}(\mathbf{x}) = \frac{\pi}{2}(1 - x_1)(1 - x_2), \quad (4.10)$$

$$\phi_1(\mathbf{x}) = x_1, \phi_2(\mathbf{x}) = x_2, \phi_{1,2}(\mathbf{x}) = \exp\left(\frac{|x_1 - x_2|^2}{8/\ln(\pi)}\right), \quad (4.11)$$

$$\phi_1(\mathbf{x}) = x_1, \phi_2(\mathbf{x}) = x_2, \phi_{1,2}(\mathbf{x}) = \frac{\pi}{3 \cos(x_1) \cos(x_2)}, \quad (4.12)$$

$$\phi_1(\mathbf{x}) = x_1, \phi_2(\mathbf{x}) = x_2, \phi_{1,2}(\mathbf{x}) = \pi \cos(x_1) \cos(x_2). \quad (4.13)$$

Among these sets, we fix $\phi_1(\mathbf{x}) = x_1$ and $\phi_2(\mathbf{x}) = x_2$ to focus on the effect of $\phi_{1,2}(\mathbf{x})$ on the classification performance. This is because the function $\phi_{1,2}(\mathbf{x})$ controls the entanglement of the two qubits, a unique property of quantum mechanics. The function $\phi_{1,2}(\mathbf{x})$ is chosen from a set of various nonlinear functions that satisfy $\max(\phi_{1,2}(\mathbf{x})) - \min(\phi_{1,2}(\mathbf{x})) \leq 2\pi$ for $x_1, x_2 \in [-1, 1]$. In particular, the functions in Eqs. (4.12) and (4.13) are empirically determined so that the resulting quantum classifiers can achieve high accuracy on the prepared datasets.

We perform five-fold cross validation to assess these quantum classifiers' performance. In this technique, one dataset is divided into five groups with the same number of examples, and then four groups are used for training and the rest for testing; this is repeated five times so that every group is assigned as the test dataset exactly once. The numerical simulations are performed using Qiskit [172] to estimate quantum kernels with 10,000 measurement shots. As for the SVM optimization, we use SVC provided in scikit-learn [173], a Python library for machine learning. Moreover, the hyperparameter in the SVM C_h is set to 10^{10} to realize the scenario where the minimum accuracy is computed, i.e., the hard-margin SVM.

Numerical Study on the Analysis Method

Here, we examine whether the minimum accuracy can help seek a set of suitable encoding functions among the candidates in Eqs. (4.9) to (4.13). As described above, the minimum accuracy is based on the real-valued quantum feature vectors $\mathbf{a}(\mathbf{x}) = [a_{I\dots I}(\mathbf{x}), \dots, a_{Z,\dots,Z}(\mathbf{x})]^T$.

Table 4.1: List of elements in the real-valued quantum feature vectors $\mathbf{a}(\mathbf{x}) = [a_{II}(\mathbf{x}), \dots, a_{ZZ}(\mathbf{x})]^T$ for the quantum feature map in Eq. (4.7) [40]. The encoding function $\phi_k(\mathbf{x})$ is denoted as ϕ_k for simplicity.

| Index σ_i | elements of the quantum feature vector a_{σ_i} |
|------------------|---|
| <i>II</i> | $1/4$ |
| <i>XI</i> | $\{\sin \phi_1(\sin \phi_2 \sin \phi_{1,2}^2 + \sin \phi_1 \cos \phi_{1,2}^2 + \cos \phi_2 \cos \phi_1 \sin \phi_{1,2})\}/4$ |
| <i>YI</i> | $\{-\sin \phi_2 \cos \phi_1 \sin \phi_{1,2}^2 - \sin \phi_1 \cos \phi_1 \cos \phi_{1,2}^2 + \cos \phi_2 \sin \phi_1^2 \sin \phi_{1,2}\}/4$ |
| <i>ZI</i> | $\cos \phi_1 \cos \phi_{1,2}/4$ |
| <i>IX</i> | $\{\sin \phi_2(\sin \phi_1 \sin \phi_{1,2}^2 + \sin \phi_2 \cos \phi_{1,2}^2 + \cos \phi_1 \cos \phi_2 \sin \phi_{1,2})\}/4$ |
| <i>XX</i> | $\{\sin \phi_1^2 \sin \phi_2^2 + \sin \phi_{1,2} \cos \phi_1 \cos \phi_2(\sin \phi_1 + \sin \phi_2)\}/4$ |
| <i>YX</i> | $\{-\sin \phi_2^2 \sin \phi_1 \cos \phi_1 + \sin \phi_{1,2} \cos \phi_2(\sin \phi_1 \sin \phi_2 - \cos \phi_1^2)\}/4$ |
| <i>ZX</i> | $\{\cos \phi_{1,2}(-\sin \phi_1 \cos \phi_2 \sin \phi_{1,2} + \cos \phi_1 \sin \phi_2^2 + \sin \phi_2 \cos \phi_2 \sin \phi_{1,2})\}/4$ |
| <i>IY</i> | $\{-\sin \phi_1 \cos \phi_2 \sin \phi_{1,2}^2 - \sin \phi_2 \cos \phi_2 \cos \phi_{1,2}^2 + \cos \phi_1 \sin \phi_2^2 \sin \phi_{1,2}\}/4$ |
| <i>XY</i> | $\{-\sin \phi_1^2 \sin \phi_2 \cos \phi_2 + \sin \phi_{1,2} \cos \phi_1(\sin \phi_1 \sin \phi_2 - \cos \phi_2^2)\}/4$ |
| <i>YY</i> | $\{\sin \phi_1 \cos \phi_1 \sin \phi_2 \cos \phi_2 - \sin \phi_{1,2}(\cos \phi_2^2 \sin \phi_1 + \sin \phi_2 \cos \phi_1^2)\}/4$ |
| <i>ZY</i> | $\{\sin \phi_2(-\sin \phi_1 \sin \phi_{1,2} \cos \phi_{1,2} - \cos \phi_2 \cos \phi_1 \cos \phi_{1,2} + \sin \phi_2 \cos \phi_{1,2} \sin \phi_{1,2})\}/4$ |
| <i>IZ</i> | $\cos \phi_2 \cos \phi_{1,2}/4$ |
| <i>XZ</i> | $\{\cos \phi_{1,2}(-\sin \phi_2 \cos \phi_1 \sin \phi_{1,2} + \cos \phi_2 \sin \phi_1^2 + \sin \phi_1 \cos \phi_1 \sin \phi_{1,2})\}/4$ |
| <i>YZ</i> | $\{\sin \phi_1(-\sin \phi_2 \sin \phi_{1,2} \cos \phi_{1,2} - \cos \phi_1 \cos \phi_2 \cos \phi_{1,2} + \sin \phi_1 \cos \phi_{1,2} \sin \phi_{1,2})\}/4$ |
| <i>ZZ</i> | $\cos \phi_1 \cos \phi_2/4$ |

In this setting where the quantum feature map in Eq. (4.7) is used, the 4²-dimensional vector $\mathbf{a}(\mathbf{x})$ can be explicitly calculated, as shown in Table 4.1. We remind the readers that quantum feature vectors can also be obtained by estimating expectation values of the Pauli operators with respect to the quantum states generated from the quantum feature maps.

First, we show the classification accuracy achieved by these classifiers for the four datasets mentioned earlier. The accuracy of quantum classifiers with the encoding functions in Eqs. (4.9) to (4.13) is summarized in Table 4.2. Overall, the quantum classifier with the set of functions in Eq. (4.12) shows good performance; the accuracy is larger than 0.95 for training and 0.88 for testing. On the other hand, the encoding functions in Eq. (4.9) do not always result in high performance. The accuracy of the functions for training data of *Circle* and *Xor* is 1.00, while the one for *Moon* is 0.85. This indicates that the choice of the encoding functions affects the quantum feature maps and thus the classification performance of the quantum kernel.

Next, we check what kind of regularities can be found in each element of the real-valued quantum feature vectors and compare them with the actual classification performance. In Figs. 4.6 to 4.10, we show the color map of each element $a_{\sigma_i}(\mathbf{x})$ on the two-dimensional input space $\mathbf{x} \in [-1, 1]^2$ for the encoding functions in Eqs. (4.9) to (4.13). Although the color maps are not generated based on specific datasets, some patterns can be seen in the two-dimensional space, affecting the classifiers' performance. For instance, every *ZZ* element $a_{ZZ}(\mathbf{x})$ in Figs. 4.6 to 4.10 witness the circle-like pattern, implying *Circle* dataset can be classified in the one-dimensional space projected on $a_{ZZ}(\mathbf{x})$. The observation is consistent with the fact that the *Circle* dataset is separable with high accuracy by these classifiers, as shown in Table 4.2. Similarly, Fig. 4.6 shows that the *YX* element of the vector $a_{YX}(\mathbf{x})$ for the encoding function in Eq. (4.9) can find a pattern like *Xor* dataset. As a result, the classifier constructed with Eq. (4.9) achieves the best training accuracy of 1.00 for the *Xor* dataset. Also, the encoding function in Eq. (4.12) leads to a high accuracy of 0.98 for *Exp* dataset, as the similar configuration can be found in the *YI* element in Fig. 4.9. These results indicate that the projected one-dimensional space can

Table 4.2: Classification accuracy of SVMs with fidelity-based quantum kernels using different encoding functions.

(a) Training accuracy

| encoding functions | <i>Circle</i> | <i>Exp</i> | <i>Moon</i> | <i>Xor</i> |
|--------------------|---------------|------------|-------------|------------|
| Eq. (4.9) | 1.00 | 0.91 | 0.85 | 1.00 |
| Eq. (4.10) | 1.00 | 0.93 | 0.96 | 0.97 |
| Eq. (4.11) | 1.00 | 0.97 | 0.91 | 0.93 |
| Eq. (4.12) | 1.00 | 0.98 | 1.00 | 0.95 |
| Eq. (4.13) | 1.00 | 0.94 | 0.98 | 0.93 |

(b) Test accuracy

| encoding functions | <i>Circle</i> | <i>Exp</i> | <i>Moon</i> | <i>Xor</i> |
|--------------------|---------------|------------|-------------|------------|
| Eq. (4.9) | 0.97 | 0.83 | 0.85 | 0.99 |
| Eq. (4.10) | 0.96 | 0.89 | 0.87 | 0.96 |
| Eq. (4.11) | 1.00 | 0.92 | 0.86 | 0.91 |
| Eq. (4.12) | 1.00 | 0.88 | 0.92 | 0.89 |
| Eq. (4.13) | 1.00 | 0.92 | 0.87 | 0.88 |

Table 4.3: Minimum accuracy obtained for each set of encoding functions.

| encoding functions | <i>Circle</i> | <i>Exp</i> | <i>Moon</i> | <i>Xor</i> |
|--------------------|---------------|------------|-------------|------------|
| Eq. (4.9) | 0.99 | 0.77 | 0.83 | 0.99 |
| Eq. (4.10) | 0.99 | 0.76 | 0.80 | 0.91 |
| Eq. (4.11) | 0.99 | 0.86 | 0.89 | 0.85 |
| Eq. (4.12) | 0.99 | 0.88 | 0.89 | 0.84 |
| Eq. (4.13) | 0.99 | 0.81 | 0.85 | 0.78 |

be utilized in some cases to see if the data regularity can be found in the quantum-enhanced feature space.

We are in a good position to compare the classification accuracy with our proposal, the minimum accuracy. Table 4.3 demonstrates the minimum accuracy for the encoding functions in Eqs. (4.9) to (4.13). As shown in Sec. 4.1.2, the minimum accuracy is attainable for any optimized (linear) classifiers and thus serves as a lower bound. The comparison demonstrates that the minimum accuracy actually gives a worst-case accuracy, indicating the potential of the quantity as a guide to choosing a suitable quantum feature map. Moreover, we compare the tendency of the minimum accuracy and the exact training accuracy. Fig. 4.5 summarizes the comparison where the blue and red bars denote the accuracy for training and the minimum accuracy, respectively. Interestingly, encoding functions chosen based on the minimum accuracy can produce the best accuracy. Because of its definition, the highest value for the minimum accuracy does not necessarily mean the best classifier among the candidates. Nevertheless, we can see a broad tendency that high minimum accuracy leads to high classification performance for training. This result positively implies that the minimum accuracy can potentially be used to select a better-performing feature map.

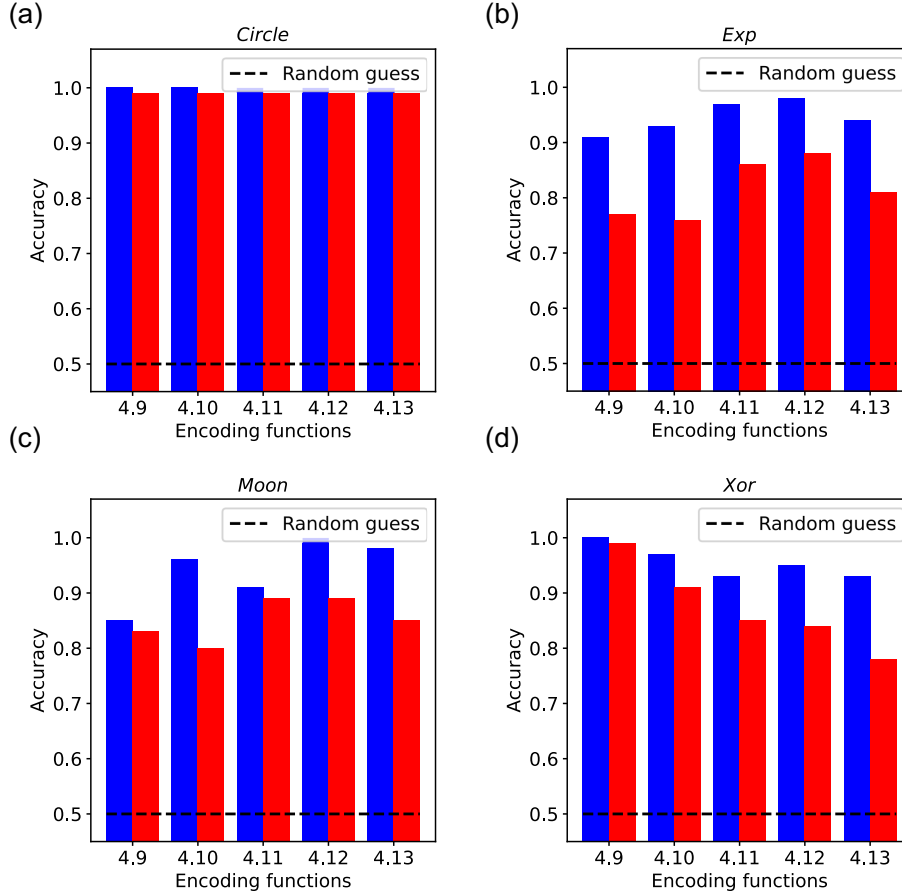


Figure 4.5: Comparison between the accuracy for training and the minimum accuracy.

Numerical Study on the Synthesis Method

We here study how combined kernels can improve the classification performance. More specifically, we focus on *Moon* dataset and *Exp* dataset. According to Table 4.2, the quantum classifier with the encoding function in Eq. (4.9) showed the worst accuracy of 0.85 on the *Moon* dataset. However, combining such a weak kernel with another quantum kernel using encoding functions in Eqs. (4.10) to (4.10) based on Eq. (4.5) can improve the classification performance. Table 4.4 (a) shows that every combined quantum kernel leads to higher accuracy than the original single quantum kernels. Notably, even when the weakest quantum kernels constructed using Eq. (4.9) and Eq. (4.11) are combined, the resultant kernel achieves the accuracy of 0.94 (the training accuracy is 0.85 and 0.91 for each quantum kernel, respectively). This indicates that even weak quantum classifiers can be combined to enhance the performance. Similarly, as shown in Table 4.4 (b), we can find that the quantum kernel composed of the kernel of Eq. (4.9) and the other shows better classification performance.

4.1.5 Conclusion & Outlook

This section discusses how to analyze and synthesize quantum feature maps to obtain powerful quantum classifiers. As for the analysis method, we propose a quantity we call the minimum accuracy, which can serve as a lower bound of the actual accuracy for training datasets. Numer-

Table 4.4: Accuracy of classifiers with combined quantum kernels.

| (a) Classification accuracy on <i>Moon</i> dataset | | | | |
|--|----------------|----------------|----------------|----------------|
| combination of encoding | (4.9) + (4.10) | (4.9) + (4.11) | (4.9) + (4.12) | (4.9) + (4.13) |
| Training | 1.00 | 0.94 | 1.00 | 1.00 |
| Testing | 0.95 | 0.90 | 0.98 | 0.96 |

| (b) Classification accuracy on <i>Exp</i> dataset | | | | |
|---|----------------|----------------|----------------|----------------|
| combination of encoding | (4.9) + (4.10) | (4.9) + (4.11) | (4.9) + (4.12) | (4.9) + (4.13) |
| Training | 0.96 | 0.93 | 0.95 | 0.95 |
| Testing | 0.92 | 0.90 | 0.88 | 0.92 |

ical experiments using some benchmarking binary classification tasks demonstrate the potential of the minimum accuracy as a quantity to effectively screen a suitable quantum feature map. We also examine the effectiveness of the synthesis method where (weak) kernels are combined to construct a performant kernel in the quantum regime. We numerically check that the synthesized quantum kernels can result in better classification performance. These results would pave the way to design a better-performing quantum feature map.

Although the minimum accuracy can give some insights into the design of quantum feature maps, its computational cost increases exponentially as the number of qubits increases. This is because we consider the whole Pauli operators as candidates of the separating hyperplane in the quantum-enhanced feature space. A possible remedy to this problem is to consider the subset of the Pauli operators; $\hat{R} = \max_{\sigma_i \in \mathcal{P}} R_{\sigma_i}$ with $\mathcal{P} \subset \{I, X, Y, Z\}^{\otimes n}$ is computed instead of $R = \max_{\sigma_i \in \{I, X, Y, Z\}^{\otimes n}} R_{\sigma_i}$ in the fifth procedure shown in Sec. 4.1.2. The subset can be chosen arbitrarily. One approach is randomly sampling a linear number of the Pauli operators. Another is to restrict the locality of the Pauli operators; we only consider the Pauli operators containing at most k non-identity operators (i.e., $\{X, Y, Z\}$) for $k < n$. While \hat{R} could be smaller than R in some cases, it can still serve as a quantity to assess the worst-case accuracy for training because of the equivalence between quantum kernel-based learning models and quantum neural networks shown in Eq. (4.1). We also note that the concept of the minimum accuracy is similar to the *projected quantum kernel* [37], which aims to avoid a detrimental issue in quantum kernel methods. As discussed in detail in the next section, the fidelity-based quantum kernel suffers from implementation infeasibility and the trainability problem with the increase in the number of qubits. Projected quantum kernels are proposed to mitigate the issue by projecting quantum feature vectors onto a classically tractable space. As projected quantum kernels can still perform well, our minimum accuracy is also helpful in understanding the power of quantum feature maps in specific machine learning tasks.

An open question on the synthesis method is how one combines quantum kernels. This thesis focuses on the weighted sum of the quantum kernels in Eq. (4.5). While we set the parameters to $\lambda_1 = \lambda_2 = 1$, one can try different hyperparameters to attain better performance. Actually, Table 4.4 (b) shows that the classification performance is not so improved by combining two quantum kernels compared to the single kernel. Thus, it would be interesting to investigate a general and systematic approach to synthesizing quantum feature maps. Moreover, there are many methods to synthesize kernels; some examples are shown in Sec. 3.2.1. Exploring the effect of these synthesizing tools also helps to design good quantum kernels.

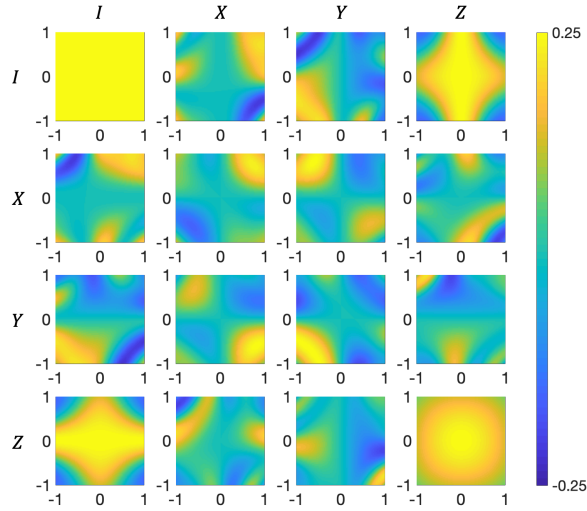


Figure 4.6: Color map representation of the real-valued quantum feature map with the encoding function in Eq. (4.9) on the two-dimensional input space. Each color map represents a element of the vector $a_{\sigma_i}(\mathbf{x})$ in $\sigma_i \in \{I, X, Y, Z\}^{\otimes 2}$. Reprinted figure from Ref. [40]. Copyright 2020 by Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama, and N. Yamamoto. [DOI:10.1007/s42484-020-00020-y].

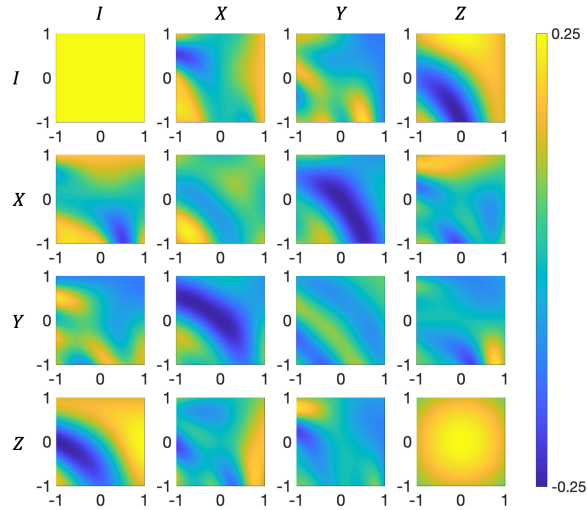


Figure 4.7: Color map representation of the real-valued quantum feature map with the encoding function in Eq. (4.10) on the two-dimensional input space. Each color map represents a element of the vector $a_{\sigma_i}(\mathbf{x})$ in $\sigma_i \in \{I, X, Y, Z\}^{\otimes 2}$. Reprinted figure from Ref. [40]. Copyright 2020 by Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama, and N. Yamamoto. [DOI:10.1007/s42484-020-00020-y].

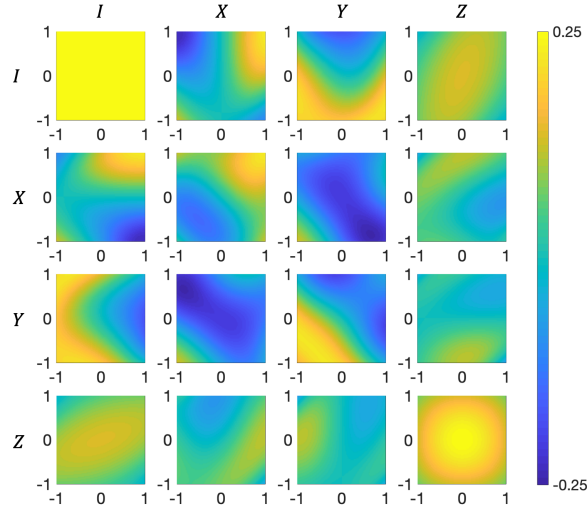


Figure 4.8: Color map representation of the real-valued quantum feature map with the encoding function in Eq. (4.11) on the two-dimensional input space. Each color map represents a element of the vector $a_{\sigma_i}(\mathbf{x})$ in $\sigma_i \in \{I, X, Y, Z\}^{\otimes 2}$. Reprinted figure from Ref. [40]. Copyright 2020 by Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama, and N. Yamamoto. [DOI:10.1007/s42484-020-00020-y].

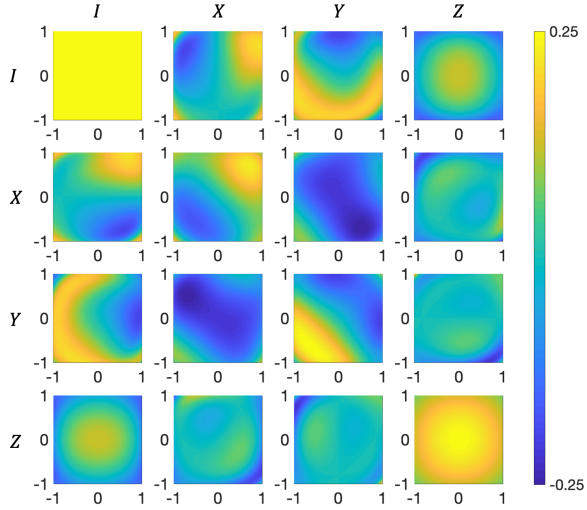


Figure 4.9: Color map representation of the real-valued quantum feature map with the encoding function in Eq. (4.12) on the two-dimensional input space. Each color map represents a element of the vector $a_{\sigma_i}(\mathbf{x})$ in $\sigma_i \in \{I, X, Y, Z\}^{\otimes 2}$. Reprinted figure from Ref. [40]. Copyright 2020 by Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama, and N. Yamamoto. [DOI:10.1007/s42484-020-00020-y].

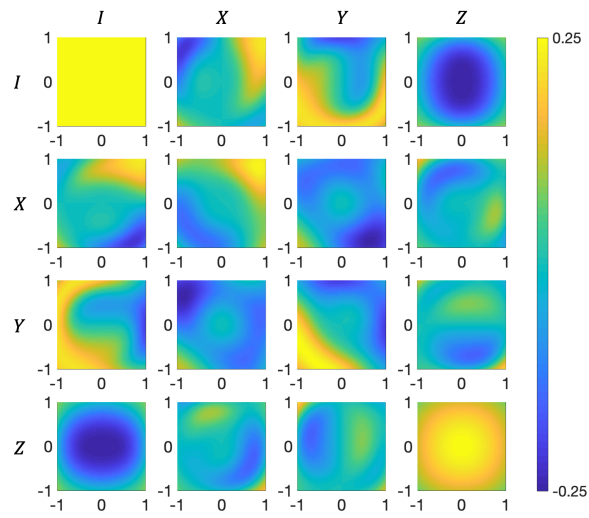


Figure 4.10: Color map representation of the real-valued quantum feature map with the encoding function in Eq. (4.13) on the two-dimensional input space. Each color map represents a element of the vector $a_{\sigma_i}(\mathbf{x})$ in $\sigma_i \in \{I, X, Y, Z\}^{\otimes 2}$. Reprinted figure from Ref. [40]. Copyright 2020 by Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama, and N. Yamamoto. [DOI:10.1007/s42484-020-00020-y].

4.2 A Remedy to the Vanishing Similarity Issue: Quantum Fisher Kernel²

This section addresses another practical issue in quantum kernel methods: the so-called *vanishing similarity issue*. Quantum kernel methods hold a crucial position in QML due to the provable quantum advantages. On the other hand, the commonly-used fidelity-based quantum kernel suffers from the vanishing similarity issue (or exponential concentration problem) [38,39], where the exponential decay of the expectation value and the variance of the quantum kernel result in infeasible implementation and poor trainability. We make two contributions to the issue. First, we show from both analytical and numerical perspectives that the fidelity-based quantum kernel cannot avoid the vanishing similarity issue regardless of types of quantum circuits. The second and most significant result is to propose a new type of quantum kernel termed the *quantum Fisher kernel* (QFK). We analytically and numerically demonstrate that QFKs can circumvent the issue when shallow alternating layered ansatzes [102] are used. We further perform numerical simulations to see the expressivity and the performance of QFK.

4.2.1 Introduction

Previous works have theoretically shown datasets that are not efficiently learnable by classical models but by the fidelity-based quantum kernels [31–33]. This motivates a number of researchers to pursue practical advantages of the methods. In general, it is conjectured that quantum advantages are realized when a large number of qubits are used. However, the fidelity-based quantum kernel suffers from practical issues with the increase in the qubit numbers; for instance, a significantly large number of measurement shots are needed to precisely estimate the quantum kernel on quantum devices, and the performance to an unseen new data (i.e., the generalization performance) is poor. These issues arise because expectation value and variance of the quantum kernel decay exponentially in the number of qubits. The problem is called the vanishing similarity issue, which we elaborate on later in Sec. 4.2.3. Resolving the issue is imperative for the practical use of quantum kernel methods to acquire advantages in real-world applications.

This section discusses vanishing similarity in the fidelity-based quantum kernel and a new type of quantum kernel that we propose as a circumventing approach to the issue, the quantum Fisher kernel (QFK). In our analysis, we consider two types of quantum feature maps (i.e., quantum circuits): (1) globally-random quantum circuits and (2) alternating layered ansatzes (ALAs). Then, we assume that the quantum circuits are independent and form 2-designs [97, 98, 174–176]. With the assumption, we analytically show that the vanishing similarity issue is not avoidable for the fidelity-based quantum kernel regardless of the types of quantum circuits. On the other hand, we demonstrate that QFKs can mitigate the issue when shallow ALAs are used. Numerical simulation also supports these analytical results.

Moreover, Fourier analysis is numerically performed to show that QFKs and the fidelity-based quantum kernel have comparable expressivity. We then demonstrate an example of classification tasks where the proposed QFK performs well, whereas the performance of the fidelity-based quantum kernel deteriorates due to the vanishing similarity issue. These results indicate the effectiveness of our QFK for machine learning tasks when large quantum systems are involved.

Lastly, we mention some related work on the vanishing similarity issue. A concept identical to the vanishing similarity issue was first addressed in Ref. [37]. Following this work, some attempts have been made to analytically elucidate the issue [39, 177–179]. Especially, Ref [39] clarifies

²Results shown in this section are based on the author’s work [38].

four sources of the issue: expressivity, global measurement, the entanglement of quantum states, and quantum noise. Yet, the analysis in this work does not take into account types of quantum circuits as our work [38]. Also, another type of quantum kernel, the projected quantum kernel, has been investigated from the vanishing similarity perspectives [39, 179].

The structure of this section is given as follows. In Sec. 4.2.2, we provide techniques used in our analysis: integration formulas of Haar random unitary. Then, after presenting the definition of the vanishing similarity issue and settings, we detail analytical results on vanishing similarity in the fidelity-based quantum kernel in Sec. 4.2.3. Next, Sec. 4.2.4 explains our proposed QFK, followed by an analytical investigation in Sec. 4.2.5. Subsequently, we show numerical experiments to support our analytical results in Sec. 4.2.6 and the performance of QFK in comparison with the fidelity-based quantum kernel in Sec. 4.2.7. Lastly, we conclude our work in Sec. 4.2.8.

4.2.2 Preliminary

Our analysis utilizes integration formulas of Haar random unitary to derive expectation values and variance of quantum kernels analytically. We thus provide the techniques before going into the details of our analytical results.

For ease of analysis, we assume that quantum circuits form t -designs. The t -design is an ensemble of unitary operators whose statistical property agrees with that of the unitary sampled from the unitary group with respect to the Haar measure up to the t -th moment [97, 98, 174]. An important property of Haar random unitary is left- and right-invariance; for any function $g(V)$ and arbitrary unitary operator U , the Haar random unitary V satisfies

$$\int d\mu_{Haar}(V)g(V) = \int d\mu_{Haar}(V)g(UV) = \int d\mu_{Haar}(V)g(VU), \quad (4.14)$$

where $d\mu_{Haar}(V)$ represents the Haar measure. If an ensemble of unitary $\{p_i, V_i\}$ (i.e., V_i is sampled with probability p_i) is a t -design, the same result can be obtained up to a polynomial function of at most degree t .

In addition, when a quantum circuit V forms a 1-design, we can have the following expression;

$$\int d\mu(V)V_{i,j}V_{l,k}^* = \frac{\delta_{i,l}\delta_{j,k}}{d}, \quad (4.15)$$

where d is the dimension of the unitary V and $\delta_{i,j}$ represents the Kronecker delta. Also, we can derive the following equality for the 2-design case;

$$\begin{aligned} \int d\mu(V)V_{i_1,j_1}V_{l_1,k_1}^*V_{i_2,j_2}V_{l_2,k_2}^* &= \frac{\delta_{i_1,l_1}\delta_{i_2,l_2}\delta_{j_1,k_1}\delta_{j_2,k_2} + \delta_{i_1,l_2}\delta_{i_2,l_1}\delta_{j_1,k_2}\delta_{j_2,k_1}}{d^2 - 1} \\ &\quad - \frac{\delta_{i_1,l_1}\delta_{i_2,l_2}\delta_{j_1,k_2}\delta_{j_2,k_1} + \delta_{i_1,l_2}\delta_{i_2,l_1}\delta_{j_1,k_1}\delta_{j_2,k_2}}{d(d^2 - 1)}. \end{aligned} \quad (4.16)$$

Moreover, we work on the integration of some functions over the local unitary operators. Thus, we also provide five Lemmas that are helpful in our calculation and were derived in Supplementary Information of Ref. [102];

Lemma 1. *Let a unitary operator V acting on the d -dimensional Hilbert space \mathcal{H}_v be a t -design with $t \geq 1$. Then, for arbitrary operators $A, B : \mathcal{H}_v \rightarrow \mathcal{H}_v$, we have*

$$\int d\mu(V)\text{Tr}[VAV^\dagger B] = \frac{\text{Tr}[A]\text{Tr}[B]}{d}. \quad (4.17)$$

Lemma 2. Let a unitary operator V acting on the d -dimensional Hilbert space \mathcal{H}_v be a t -design with $t \geq 2$. Then, for arbitrary operators $A, B, C, D : \mathcal{H}_v \rightarrow \mathcal{H}_v$, we have

$$\begin{aligned} \int d\mu(V) \text{Tr} [VAV^\dagger BVCV^\dagger D] &= \frac{1}{d^2 - 1} (\text{Tr} [A] \text{Tr} [C] \text{Tr} [BD] + \text{Tr} [AC] \text{Tr} [B] \text{Tr} [D]) \\ &\quad - \frac{1}{d(d^2 - 1)} (\text{Tr} [A] \text{Tr} [B] \text{Tr} [C] \text{Tr} [D] + \text{Tr} [AC] \text{Tr} [BD]). \end{aligned} \quad (4.18)$$

Lemma 3. Let a unitary operator V on the d -dimensional Hilbert space \mathcal{H}_v be a t -design with $t \geq 2$. Then, for arbitrary operators $A, B, C, D : \mathcal{H}_v \rightarrow \mathcal{H}_v$, we have

$$\begin{aligned} \int d\mu(V) \text{Tr} [VAV^\dagger B] \text{Tr} [VCV^\dagger D] &= \frac{1}{d^2 - 1} (\text{Tr} [A] \text{Tr} [B] \text{Tr} [C] \text{Tr} [D] + \text{Tr} [AC] \text{Tr} [BD]) \\ &\quad - \frac{1}{d(d^2 - 1)} (\text{Tr} [A] \text{Tr} [C] \text{Tr} [BD] + \text{Tr} [AC] \text{Tr} [B] \text{Tr} [D]). \end{aligned} \quad (4.19)$$

Lemma 4. Let a unitary operator V acting on the d_v -dimensional Hilbert space \mathcal{H}_v be a t -design with $t \geq 2$. In addition, suppose $\mathcal{H} = \mathcal{H}_{\bar{v}} \otimes \mathcal{H}_v$ be $d_v d_{\bar{v}}$ -dimensional. Then, for arbitrary operators $A, B : \mathcal{H} \rightarrow \mathcal{H}$, we have

$$\int d\mu(V) (\mathbb{I}_{\bar{v}} \otimes V) A (\mathbb{I}_{\bar{v}} \otimes V^\dagger) B = \frac{\text{Tr}_v [A] \otimes \mathbb{I}_v}{d_v} B, \quad (4.20)$$

and

$$\int d\mu(V) \text{Tr} [(\mathbb{I}_{\bar{v}} \otimes V) A (\mathbb{I}_{\bar{v}} \otimes V^\dagger) B] = \frac{1}{d_v} \text{Tr} [\text{Tr}_v [A] \text{Tr}_v [B]]. \quad (4.21)$$

Here, $\mathbb{I}_v (\mathbb{I}_{\bar{v}})$ represents the identity matrix acting on the Hilbert space $\mathcal{H}_v (\mathcal{H}_{\bar{v}})$ and the partial trace over $\mathcal{H}_v (\mathcal{H}_{\bar{v}})$ is denoted as $\text{Tr}_v (\text{Tr}_{\bar{v}})$. Also, \bar{A} denotes the complement of A .

Lemma 5. Let V be a unitary operator acting on the d_v -dimensional Hilbert space \mathcal{H}_v . In addition, suppose $\mathcal{H} = \mathcal{H}_{\bar{v}} \otimes \mathcal{H}_v$ be $d_v d_{\bar{v}}$ -dimensional with $d_v = 2^m$ and $d_{\bar{v}} = 2^{n-m}$. Then, for arbitrary operators $A, B : \mathcal{H} \rightarrow \mathcal{H}$, we have

$$\text{Tr} [(\mathbb{I}_{\bar{v}} \otimes V) A (\mathbb{I}_{\bar{v}} \otimes V^\dagger) B] = \sum_{\mathbf{p}, \mathbf{q}} \text{Tr} [VA_{\mathbf{qp}}, V^\dagger B_{\mathbf{pq}}], \quad (4.22)$$

where

$$A_{\mathbf{qp}} = \text{Tr}_{\bar{w}} [(|\mathbf{p}\rangle \langle \mathbf{q}| \otimes \mathbb{I}_w) A], \quad B_{\mathbf{pq}} = \text{Tr}_{\bar{w}} [(|\mathbf{q}\rangle \langle \mathbf{p}| \otimes \mathbb{I}_w) B]. \quad (4.23)$$

Here \mathbf{q} and \mathbf{p} represent bit-strings of length $n - m$.

We note that these Lemmas are applicable in case V is replaced with $U_l V U_r$ where U_l and U_r are arbitrary unitary operators and V is the Haar random unitary; this can be easily checked by using left- and right- invariant property.

4.2.3 Vanishing Similarity Issue in Fidelity-Based Quantum Kernel

Next, we show the vanishing similarity issue in the fidelity-based quantum kernel. This work focuses on the parameterized fidelity-based quantum kernel instead of Eq. (3.34):

$$k_Q(\mathbf{x}, \mathbf{x}') = \text{Tr} [\rho_{\mathbf{x}, \theta} \rho_{\mathbf{x}', \theta}], \quad (4.24)$$

where $\rho_{\mathbf{x},\boldsymbol{\theta}} = U(\mathbf{x},\boldsymbol{\theta})\rho U^\dagger(\mathbf{x},\boldsymbol{\theta})$ is the data-dependent quantum state generated by applying $U(\mathbf{x},\boldsymbol{\theta})$ to initial state ρ_0 . We consider the data- and parameter-dependent quantum circuits $U(\mathbf{x},\boldsymbol{\theta})$ because tunable parameters $\boldsymbol{\theta}$ are introduced in practical situations so that powerful quantum feature maps are engineered. We remark that $U(\mathbf{x},\boldsymbol{\theta})$ can be regarded as the general expression of $U(\mathbf{x})$; thus, our analytical results provided later can be applied to the case in Eq. (3.34).

In the following, we first present the details of the vanishing similarity issue. Then, we analytically address the issue in the fidelity-based quantum kernel after providing the setting in our analysis.

Vanishing Similarity Issue

We explain the vanishing similarity issue in detail. The provable advantages of quantum kernel methods lie in cases where a large number of qubits are used. However, the fidelity-based quantum kernel outputs significantly small values that concentrate exponentially onto a fixed value as the size of qubit systems increases. We recall that a kernel function estimates the similarity between a pair of data points in the feature space. Hence, this indicates that the similarity vanishes exponentially in the quantum-enhanced feature space with the increase of the number of qubits; also, differentiating features of data via the quantum kernels becomes challenging.

More concretely, the vanishing similarity issue is defined as

$$\text{Var}_{\{(\mathbf{x},\mathbf{x}')\}} [k_Q(\mathbf{x},\mathbf{x}')] = b, \quad b \in \mathcal{O}(1/c^n) \quad (4.25)$$

with the number of qubits n and $c > 1$. Eq (4.25) states that the variance of the quantum kernel in Eq. (4.24) taken over a pair of data points (\mathbf{x},\mathbf{x}') drawn from certain distribution is upper bounded by an exponentially small value. The variance can also be taken over a pair of data-dependent unitary operators $(U(\mathbf{x},\boldsymbol{\theta}), U(\mathbf{x}',\boldsymbol{\theta}))$, because quantum kernels are dependent on data \mathbf{x} only via the quantum feature map by definition. That is, Eq. (4.25) can be restated as

$$\text{Var}_{\{(U(\mathbf{x},\boldsymbol{\theta}), U(\mathbf{x}',\boldsymbol{\theta}))\}} [k_Q(\mathbf{x},\mathbf{x}')] = b, \quad b \in \mathcal{O}(1/c^n). \quad (4.26)$$

Note that we consider a pair of unitary operators to compute expectation values and variance throughout this thesis and hence will not explicitly express $\{(U(\mathbf{x},\boldsymbol{\theta}), U(\mathbf{x}',\boldsymbol{\theta}))\}$.

The issue is detrimental for two reasons. Firstly, measurement must be repeated a significant amount of times to obtain the precise value of the fidelity-based quantum kernel. In realistic situations, the fidelity-based quantum kernels are estimated on quantum hardware by the swap test or the inversion test, as shown in Sec. 3.2. This means the precision of the estimated quantum kernel is determined by the number of measurement shots N_s . However, vanishing similarity in Eq. (4.25) states that the difference between two quantum kernels is crucially small, and thus, an exponential number of measurements is required. Secondly, learning models based on the quantum kernel fail to predict the target value of a given unseen data point. This is because the Gram matrix, a matrix whose (i, j) entry is given by

$$G_{ij} = k_Q(\mathbf{x}_i, \mathbf{x}_j), \quad (4.27)$$

gets close to the identity matrix and thus the models easily cause overfitting. Suppose the SVM algorithm with kernel methods, for instance. If the identity matrix is used as the Gram matrix, the convex optimization problem shown in Eq. (3.54) becomes trivial; as a result, the generalization performance of the optimized model is poor.

The vanishing similarity issue is analogous to the barren plateau problem in variational quantum algorithms. A barren plateau addressed in Ref. [180] represents a status of the cost function landscape of variational quantum algorithms, where the magnitude of gradients vanishes exponentially as the number of qubits increases. This is problematic as the phenomenon makes the algorithms untrainable. Thus far, some works theoretically analyzed the barren plateaus to understand trainable situations and how to avoid the issue [102,180–185]. For example, Ref. [102] demonstrates that using local cost functions and ALAs can alleviate the vanishing gradients problem. Actually, this work motivates us to work on ALAs for analyzing vanishing similarity in quantum kernel methods. We lastly note that the implicit models in quantum-enhanced machine learning (see Sec. 3.1.3) can be categorized into variational quantum algorithms. Thus, these issues can be interpreted in a unified framework; implicit and explicit models suffer from the vanishing similarity issue and the barren plateau problems, respectively.

Setting of the Analysis

We analytically demonstrate the issue in the fidelity-based quantum kernel in Eq. (4.24). Here, we detail the setup of our analysis.

We consider two types of quantum circuits: globally-random quantum circuits and ALAs composed of m -qubit local unitary blocks, depicted in Fig. 4.11 (a) and (b), respectively. The globally-random quantum circuits are denoted as $U(\mathbf{x}, \boldsymbol{\theta})$. Also, we express the ALA as

$$\begin{aligned} U(\mathbf{x}, \boldsymbol{\theta}) &= \prod_{d=1}^L V_d(\mathbf{x}, \boldsymbol{\theta}) \\ &= \prod_{d=1}^L \left(\prod_{k=1}^{\kappa} W_{k,d}(\mathbf{x}, \boldsymbol{\theta}_{k,d}) \right), \end{aligned} \quad (4.28)$$

where L denotes the total number of layers and κ represents the number of local unitary blocks in a layer satisfying $n = m\kappa$ with the total number of qubits. Here, both a unitary block in one layer and the one in the adjacent layer act on at most $m/2$ qubits in common; for instance, $S_{(k,1)}$ and $S_{(k,2)}$ have $m/2$ -qubit subspace in common, where $S_{(k,d)}$ is the subspace of qubits which the unitary block $W_{k,d}(\mathbf{x}, \boldsymbol{\theta}_{k,d})$ acts on. The detail is illustrated in Fig. 4.11 (c). Note that local unitary blocks can be arbitrary; that is, data-dependent gates, parameter-dependent gates, and data- and parameter-independent gates can constitute local unitary blocks. However, we assume that the parameterized gates are expressed in terms of single-qubit rotation gates, i.e., $R_{\sigma}(\theta) = \exp(-i\theta\sigma/2)$ with a Pauli operator $\sigma \in \{X, Y, Z\}$.

Then, we assume that the globally-random quantum circuits and local unitary blocks in the ALAs are independent and form 2-designs. As shown in Sec. 4.2.2, the 2-design is an ensemble of unitary operators with the same statistical property as the Haar random unitary up to the second moment. Roughly speaking, the assumption indicates that the quantum circuits are so expressive that the ensemble of Haar random states can be explored. Previous works assume the 2-design to analyze barren plateaus [102, 180, 186–188] and vanishing similarity [39, 177]. However, we note that the assumption might not hold in practice.

Analytical Results

With the setting mentioned above, we analytically compute the expectation value and the variance of the fidelity-based quantum kernel in Eq. (4.24). See Appendix A.1 for the proof.

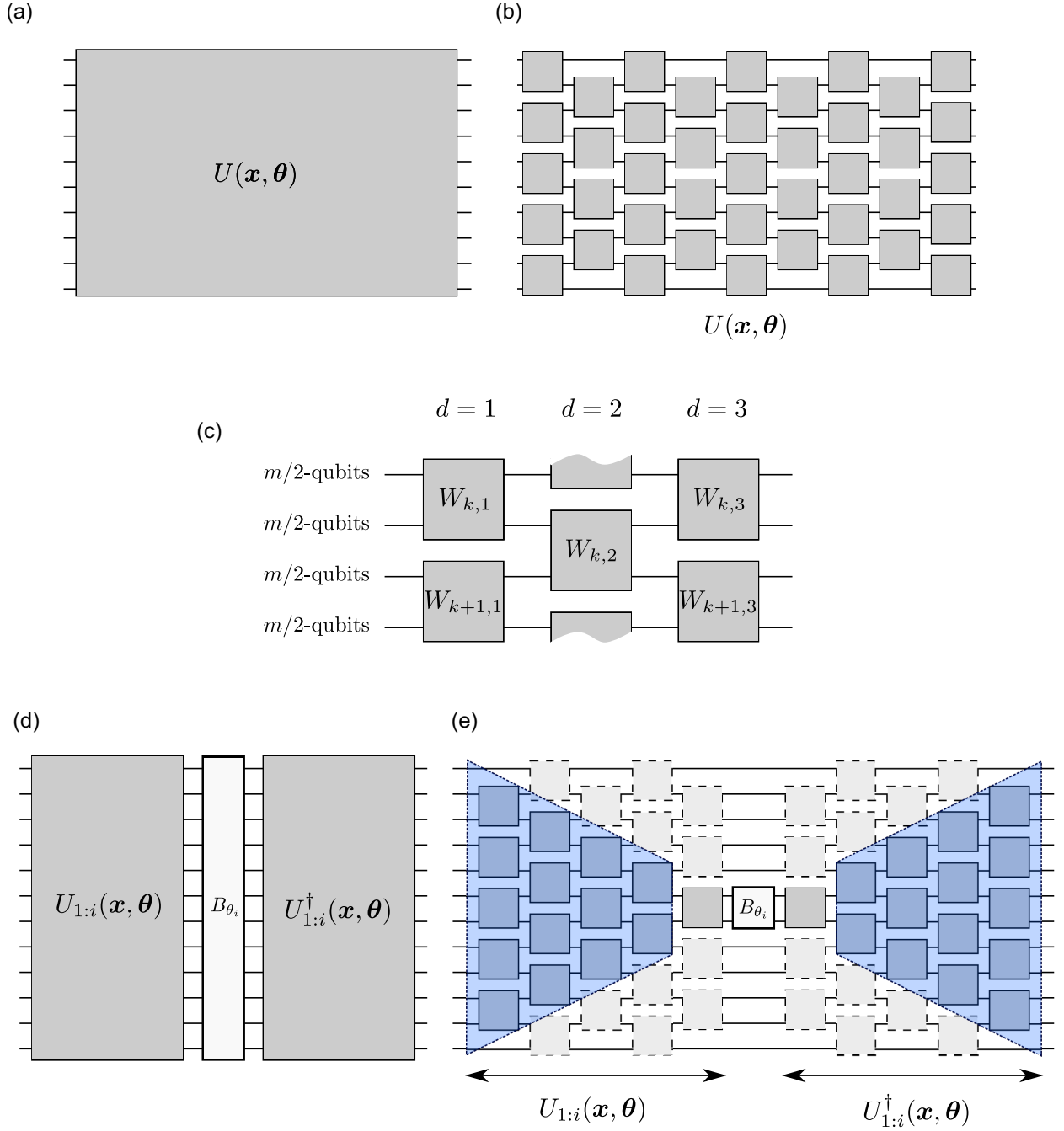


Figure 4.11: Quantum circuits used in our analysis. The globally-random quantum circuit and the ALA are shown in (a) and (b), respectively. Panel (c) illustrates the detail of the ALA with a focus on k and $k + 1$ unitary blocks in the first three layers. For simplicity, we denote $W_{k,d}(\mathbf{x}, \boldsymbol{\theta}_{k,d}) \equiv W_{k,d}$. Panels (d) and (e) represent $\tilde{B}_{\mathbf{x}, \theta_i} = U_{1:i}^\dagger(\mathbf{x}, \boldsymbol{\theta}) B_{\theta_i} U_{1:i}(\mathbf{x}, \boldsymbol{\theta})$ for the globally-random quantum circuit and the ALA, respectively. In panel (e), the thick gray unitary block adjacent to B_{θ_i} is represented by $\tilde{W}_{k,d}(\mathbf{x}, \theta_i)$ and the shaded region represent $V_r(\mathbf{x}, \boldsymbol{\theta})$.

Proposition 1. *Let the expectation value and the variance of the n -qubit fidelity-based quantum kernel defined in Eq. (4.24) be $\langle k_Q \rangle$ and $\text{Var}[k_Q]$, respectively. Also, let the initial state ρ_0 be an arbitrary pure state.*

- (1) *When globally-random quantum circuits $U(\mathbf{x}, \boldsymbol{\theta})$ and $U(\mathbf{x}', \boldsymbol{\theta})$ are independent, and at least either $U(\mathbf{x}, \boldsymbol{\theta})$ or $U(\mathbf{x}', \boldsymbol{\theta})$ is a t -design with $t \geq 2$, the expectation value and the variance are given by*

$$\langle k_Q \rangle = \frac{1}{2^n}, \quad (4.29)$$

$$\text{Var}[k_Q] = \frac{2^n - 1}{2^{2n}(2^n + 1)} \approx \frac{1}{2^{2n}}. \quad (4.30)$$

- (2) *Let $U(\mathbf{x}, \boldsymbol{\theta})$ and $U(\mathbf{x}', \boldsymbol{\theta})$ be the ALAs in Eq. (4.28), and let m -qubit local unitary blocks in either $U(\mathbf{x}, \boldsymbol{\theta})$ or $U(\mathbf{x}', \boldsymbol{\theta})$ be independent and t -designs with $t \geq 2$. Then, the expectation value and the upper bound of the variance are given by*

$$\langle k_Q \rangle = \frac{1}{2^n}, \quad (4.31)$$

$$\text{Var}[k_Q] \leq \frac{2^\kappa}{(2^{2m} - 1)^\kappa} - \frac{1}{2^{2n}} \approx \frac{1}{2^{n(2 - \frac{1}{m})}}. \quad (4.32)$$

The implication of Proposition 1 is that vanishing similarity is not avoidable for the fidelity-based quantum kernel regardless of the circuit type. This is because (the upper bound of) the variance decays exponentially in the number of qubits n . In other words, devising circuit structures cannot mitigate the problem if the (global) fidelity is used as the metric. This also means that the tensor-product quantum circuit, which is tractable by classical computers, also leads to vanishing similarity. We note that a similar result is demonstrated in Ref. [39] for case (1). Also, the result for case (2) could be related to Eq. (170) in Ref. [102]. Nevertheless, our work [38] is the first to elucidate the presence of vanishing similarity in the fidelity-based quantum kernel for ALAs.

Let us remark that we can obtain the same results in some cases, even when $U(\mathbf{x}, \boldsymbol{\theta})$ and $U(\mathbf{x}', \boldsymbol{\theta})$ are correlated. This can be checked by utilizing the left- and right-invariant property of the Haar random measure in Eq. (4.14).

4.2.4 Quantum Fisher Kernel

As discussed, the fidelity-based quantum kernels would suffer from the vanishing similarity issue, which suggests the need to design quantum kernels instead of fidelity-based ones. In this thesis, we propose a new quantum kernel called the quantum Fisher kernel (QFK) that can mitigate the issue. The idea behind our proposal is to incorporate data structures into learning models. Recent works have demonstrated the importance of building models that encompass the information on datasets [31, 189]. In addition, a remedy of barren plateaus, an analogy of the vanishing similarity issue for explicit models, is to take into account the structure of quantum feature map $U(\mathbf{x}, \boldsymbol{\theta})$ [102]: the structure of parameterized quantum circuits and cost function designs. Notably, the classical Fisher kernel is constructed using the information geometric quantity (i.e., the logarithmic derivatives of generative models), with the motivation of taking advantage of data sources for kernel designs. Actually, the classical Fisher kernel has been applied in some fields such as computer vision, due to its expressivity [124–126, 128]. Thus, based on the design principle of the classical Fisher kernel, we propose QFKs.

QFKs are defined as follows;

$$\begin{aligned} k_{QF}^\gamma(\mathbf{x}, \mathbf{x}') &\equiv \left\langle \mathbf{L}_{\mathbf{x}, \boldsymbol{\theta}}^\gamma, \mathbf{L}_{\mathbf{x}', \boldsymbol{\theta}}^\gamma \right\rangle_{\mathcal{F}_\gamma^{-1}} \\ &= \sum_{i,j} \mathcal{F}_{\gamma, i, j}^{-1} \left(L_{\mathbf{x}, \theta_i}^\gamma, L_{\mathbf{x}', \theta_j}^\gamma \right)_\rho \end{aligned} \quad (4.33)$$

where $(A, A')_\rho = \frac{1}{2} \text{Tr}[\rho(A'A^\dagger + A^\dagger A')]$ with certain quantum state ρ is the pre-inner product for operators [190] and F_γ is the quantum Fisher information matrix. Here, $\mathbf{L}_{\mathbf{x}, \boldsymbol{\theta}}^\gamma = [L_{\mathbf{x}, \theta_1}^\gamma, L_{\mathbf{x}, \theta_2}^\gamma, \dots]^T$ is the vector containing the quantum version of logarithmic derivatives (i.e., the Fisher score). While there are multiple definitions of the ‘‘quantum’’ Fisher score [191], we here focus on the symmetric logarithmic derivative (SLD) [192, 193] and the anti-symmetric logarithmic derivative (ALD) [190]. The SLD $L_{\mathbf{x}, \theta_l}^S$ and the ALD $L_{\mathbf{x}, \theta_l}^A$ regarding the l -th parameter θ_l for the quantum state $\rho_{\mathbf{x}, \boldsymbol{\theta}} = U(\mathbf{x}, \boldsymbol{\theta})\rho_0 U^\dagger(\mathbf{x}, \boldsymbol{\theta})$ are defined as solutions of the following equations, respectively;

$$\partial_{\theta_l} \rho_{\mathbf{x}, \boldsymbol{\theta}} = \frac{1}{2} \left(\rho_{\mathbf{x}, \boldsymbol{\theta}} L_{\mathbf{x}, \theta_l}^S + L_{\mathbf{x}, \theta_l}^S \rho_{\mathbf{x}, \boldsymbol{\theta}} \right), \quad (4.34)$$

$$\partial_{\theta_l} \rho_{\mathbf{x}, \boldsymbol{\theta}} = \frac{1}{2} \left(\rho_{\mathbf{x}, \boldsymbol{\theta}} L_{\mathbf{x}, \theta_l}^A - L_{\mathbf{x}, \theta_l}^A \rho_{\mathbf{x}, \boldsymbol{\theta}} \right). \quad (4.35)$$

Here we denote $\partial_{\theta_l} \equiv \partial/\partial\theta_l$ for simplicity. When the initial state ρ_0 is pure, a solution of the SLD equation can be expressed as

$$L_{\mathbf{x}, \theta_l}^S = 2\partial_{\theta_l} \rho_{\mathbf{x}, \boldsymbol{\theta}}. \quad (4.36)$$

Also, the ALD equation can be solved as follows;

$$L_{\mathbf{x}, \theta_l}^A = i(B_{\mathbf{x}, \theta_l} - \text{Tr}[\rho_{\mathbf{x}, \boldsymbol{\theta}} B_{\mathbf{x}, \theta_l}]), \quad (4.37)$$

with $B_{\mathbf{x}, \theta_l} = 2i(\partial_{\theta_l} U(\mathbf{x}, \boldsymbol{\theta}))U^\dagger(\mathbf{x}, \boldsymbol{\theta})$. We note that these equations are not uniquely determined. Also, we introduce γ to differentiate the ALD and SLD, i.e., $\gamma \in \{A, S\}$. By introducing Eq. (4.36) or Eq. (4.37) into Eq. (4.33), we can rewrite the QFK as

$$k_{QF}(\mathbf{x}, \mathbf{x}') = \frac{1}{2} \sum_i \mathcal{F}_{i, j}^{-1} \text{Tr} \left[\rho_0 \left\{ \tilde{B}_{\mathbf{x}, \theta_i} - \text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}], \tilde{B}_{\mathbf{x}', \theta_j} - \text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}', \theta_j}] \right\} \right] \quad (4.38)$$

with the anti-commutator $\{\cdot, \cdot\}$ and $\tilde{B}_{\mathbf{x}, \theta_i} = U_{1:i}^\dagger(\mathbf{x}, \boldsymbol{\theta}) B_{\theta_i} U_{1:i}(\mathbf{x}, \boldsymbol{\theta})$ using $U_{i:j}(\mathbf{x}, \boldsymbol{\theta})$. Here, $U_{i:j}(\mathbf{x}, \boldsymbol{\theta})$ denotes a sequence of unitary gates from $U_i(\mathbf{x}, \theta_i)$ to $U_j(\mathbf{x}, \theta_j)$, for the unitary operator $U(\mathbf{x}, \boldsymbol{\theta}) = U_D(\mathbf{x}, \theta_D) \cdots U_2(\mathbf{x}, \theta_2) U_1(\mathbf{x}, \theta_1)$. We omit the index γ because the QFK for both cases results in Eq. (4.38).

Throughout this thesis, we set the quantum Fisher information matrix in the QFK of Eq. (4.38) to the identity matrix, i.e., $\mathcal{F} = I$, as in the classical case. Ref. [122] demonstrates that the Fisher information in classical Fisher kernel is less significant in performance; previous works have also practically used the Fisher kernel with identity matrix [2, 122, 126, 127, 194] or the diagonal matrix [128, 195] due to the computational efficiency. Similarly, the quantum Fisher score could be less important, and thus we rather focus on the quantum Fisher score. Also, we do not take into account the terms $\text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}]$ and $\text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}', \theta_j}]$ in Eq. (4.38) to simplify the discussion on the vanishing similarity issue; the remaining terms are dominant in the variance scaling and the same variance scaling can be obtained even when the terms are included as shown in Appendix A.2.

4.2.5 Vanishing Similarity Issue in Quantum Fisher Kernel

We then analytically demonstrate the vanishing similarity issue in the QFK. We consider the i -th term of the QFK in Eq. (4.38), i.e.,

$$k_{QF}^{(i)} \equiv \text{Tr}[\rho_0\{\tilde{B}_{\mathbf{x},\theta_i}, \tilde{B}_{\mathbf{x}',\theta_i}\}]/2. \quad (4.39)$$

The variance of Eq. (4.38) differs from that of the i -th term. Yet, we focus on its variance to understand how deep quantum circuits can be exploited. We note that the implication of Theorem 1 shown below is consistent for the case of Eq. (4.38) as well.

Setting of the Analysis

As in the case for the fidelity-based quantum kernel shown in Sec. 4.2.3, we consider (1) globally-random quantum circuits and (2) ALAs with m -qubit local unitary blocks. Due to the form of QFK, we assume that $U_{1:i}(\mathbf{x}, \boldsymbol{\theta})$ in the globally-random circuit is a 2-design for arbitrary i . Also, suppose the k -th unitary block in the d -th layer, $W_{k,d}(\mathbf{x}, \boldsymbol{\theta}_{k,d})$, contains the i -th parameter. In addition, $U_{1:i}(\mathbf{x}, \boldsymbol{\theta})$ can be decomposed as $\tilde{W}_{k,d}(\mathbf{x}, \theta_i)V_r(\mathbf{x}, \boldsymbol{\theta})$, where $\tilde{W}_{k,d}(\mathbf{x}, \theta_i)$ denotes a sequence of gates that includes the first gate in $W_{k,d}(\mathbf{x}, \boldsymbol{\theta}_{k,d})$ through the gate with i -th parameter, and $V_r(\mathbf{x}, \boldsymbol{\theta})$ represents unitary blocks in the light-cone of $W_{k,d}(\mathbf{x}, \boldsymbol{\theta}_{k,d})$. Then, we assume $\tilde{W}_{k,d}(\mathbf{x}, \theta_i)$ and all unitary blocks in the ALAs are 2-designs for any k and d . For ease of understanding, we show the quantum circuit representation of $\tilde{B}_{\mathbf{x},\theta_i}$ in Fig. 4.11 (d) and (e).

Analytical Results

We derive the expectation value and the variance of the i -th term of QFK in Eq. (4.38) for the above setting.

Theorem 1. *Let the expectation value and the variance of i -th term for the n -qubit QFK in Eq. (4.39) be $\langle k_{QF}^{(i)} \rangle$ and $\text{Var}[k_{QF}^{(i)}]$, respectively. Also, let the initial state ρ_0 be pure.*

- (1) *When globally-random quantum circuits $U(\mathbf{x}, \boldsymbol{\theta})$ and $U(\mathbf{x}', \boldsymbol{\theta})$ are independent, and both $U_{1:i}(\mathbf{x}, \boldsymbol{\theta})$ and $U_{1:i}(\mathbf{x}', \boldsymbol{\theta})$ are t -designs with $t \geq 2$, then we have*

$$\langle k_{QF}^{(i)} \rangle = 0, \quad (4.40)$$

$$\text{Var} \left[k_{QF}^{(i)} \right] = \frac{2^n}{2(2^{2n} - 1)} \left(1 + \frac{2^n - 2}{2^n(2^n + 1)} \right) \approx \frac{1}{2^{n+1}}. \quad (4.41)$$

- (2) *Let $U(\mathbf{x}, \boldsymbol{\theta})$ and $U(\mathbf{x}', \boldsymbol{\theta})$ be the ALAs. Also, $\tilde{W}_{k,d}(\mathbf{x}, \theta_i)$, $\tilde{W}_{k,d}(\mathbf{x}', \theta_i)$ and unitary blocks in $V_r(\mathbf{x}, \boldsymbol{\theta})$ and $V_r(\mathbf{x}', \boldsymbol{\theta})$ are independent and t -designs with $t \geq 2$. Then, the expectation value is given by*

$$\langle k_{QF}^{(i)} \rangle = 0. \quad (4.42)$$

Additionally, we assume the initial state ρ_0 is represented as the tensor product of arbitrary single-qubit pure states $\{\rho_{0,i}\}_{i=1}^n$, i.e., $\rho_0 = \rho_{0,1} \otimes \rho_{0,2} \otimes \dots \otimes \rho_{0,n}$. Then, the lower bound of the variance is given by

$$\text{Var} \left[k_{QF}^{(i)} \right] \geq \frac{2^{2md} (2^{md} - 1)}{2(2^{2m} - 1)^2 (2^m + 1)^{4(d-1)}}. \quad (4.43)$$

We remark that the assumption on the initial state being a tensor product state for the variance calculation in case (2) is moderate from the practical perspective. This is because the tensor product state is a common choice for the initial state preparation. We can also derive the lower bound for a larger class of initial states, shown in Appendix A.2.

Theorem 1 implies that the QFK can preserve the variance compared to fidelity-based quantum kernels. In case (1), the variance scaling for the QFK is quadratically better than the fidelity-based ones, while QFK's variance also exponentially decreases in the number of qubits. Remarkably, in case (2), the lower bound of the variance for ALAs depends on the size of local unitary blocks m , circuit depth d of the local unitary block $W_{k,d}(\mathbf{x}, \boldsymbol{\theta}_{k,d})$. Namely, the i -th term of the QFK in a shallow region of ALAs can avoid the vanishing similarity issue. This indicates that the QFK has the potential to utilize quantum circuits whose depth is possibly $\mathcal{O}(\text{poly log}(n))$. We note that this comes from results in Ref. [102], stating the transition point between exponential and polynomial decay would lie in the region of depth $d \in \mathcal{O}(\text{poly log}(n))$. We remind that the implication could be consistent with the case for the QFK in Eq. (4.38); see Appendix A.3 for the details.

We lastly mention that QFK in the form of Eq. (4.38) might not exploit quantum circuits whose depth lies in the region $d \in \mathcal{O}(\text{poly log}(n))$, because the ones in $\mathcal{O}(\text{log}(n))$ will contribute to the QFK significantly. However, we can alleviate the problem by considering the following weighted-sum representation of the QFK;

$$k_{wQF}(\mathbf{x}, \mathbf{x}') = \frac{1}{2} \sum_i w_i \text{Tr}[\rho_0 \{ \tilde{B}_{\mathbf{x}, \theta_i}, \tilde{B}_{\mathbf{x}', \theta_i} \}] \quad (4.44)$$

with properly chosen weights $\{w_i\}$.

4.2.6 Numerical Demonstration

We perform numerical simulations to support our analytical results in Sec. 4.2.3 and 4.2.5. We here numerically compute the variance of the fidelity-based quantum kernel and the QFK for three types of quantum circuits: tensor product quantum circuits, ALAs with two-qubit local unitary blocks, and hardware efficient ansatzes (HEAs), as shown in Fig. 4.12 (a), (b) and (c), respectively. More concretely, we employ the data re-uploading technique [101] to construct these quantum circuits, i.e., $U(\mathbf{x}, \boldsymbol{\theta}) = \prod_{d=1}^L V(\boldsymbol{\theta}_d) V(\mathbf{x})$ where each parameterized quantum circuit layer $V(\boldsymbol{\theta}_d)$ is one of these quantum circuits and the input-embedding circuit $V(\mathbf{x})$ is the tensor product quantum circuit for all cases. As for the input, we randomly generate five sets of 100 data points $\{\mathbf{x}_i\}_{i=1}^{100}$, where we set the dimension of data points equal to the number of qubits and each element ranges from $-\pi$ to π . In input-embedding circuits, each element of data points is injected into a corresponding qubit, i.e., $\alpha_i = \alpha_{n+i} = x_i$ in Fig. 4.12 (a). Also, five sets of parameters $\boldsymbol{\theta}$ are randomly generated from the same range; we note $\boldsymbol{\theta} = \boldsymbol{\alpha}$ in Fig. 4.12. We then calculate the variance of quantum kernels for different pairs of data points for all combinations of input datasets and sets of parameters. We consider the setup to realize the 2-design assumption, while these quantum circuits might not hold the property. We also focus on the normalized QFK,

$$\tilde{k}_{QF}(\mathbf{x}, \mathbf{x}') = \frac{1}{2p} \sum_i \text{Tr} \left[\rho_0 \left\{ \tilde{B}_{\mathbf{x}, \theta_i}, \tilde{B}_{\mathbf{x}', \theta_j} \right\} \right], \quad (4.45)$$

with the number of parameters p , to set the trace of the Gram matrix equal to the number of data points. All the simulation in this section is performed by Cirq [196].

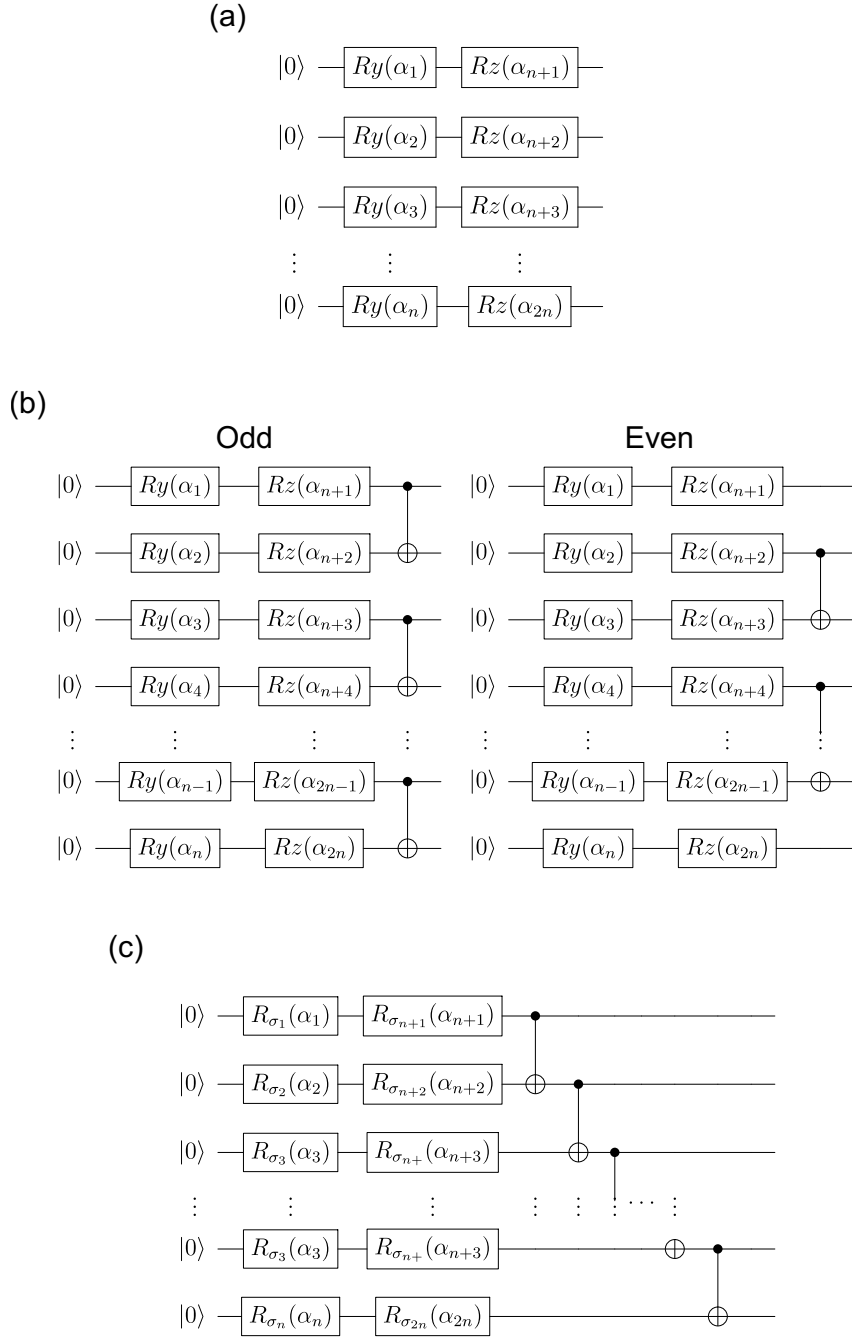


Figure 4.12: Quantum circuits used in numerical simulations in Sec. 4.2. (a) Tensor product quantum circuit, (b) ALA with two-qubit local unitary blocks, and (c) HEA. As for the ALA, alternating layers are realized by preparing different entanglers for even and odd layers. We also use randomly chosen Pauli operators $\sigma_i \in \{X_i, Y_i, Z_i\}$ for the HEA in (c).

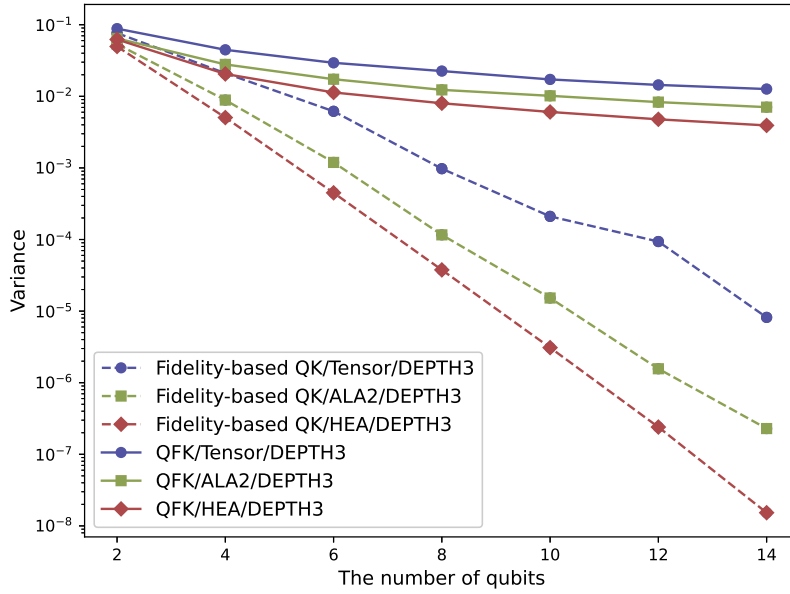


Figure 4.13: Variance of fidelity-based quantum kernels and QFKs against the number of qubits, $n \in \{2, 4, 6, 8, 10, 12, 14\}$. We use three types of quantum circuits: tensor product quantum circuits, ALAs with two-qubit local unitary blocks, and HEAs.

Fig. 4.13 shows a semi-log plot of the variance of these quantum kernels for three types of quantum circuits. We find that the fidelity-based quantum kernel witnesses exponential decay of its variance regardless of types of quantum circuits, as demonstrated in Proposition 1. On the other hand, the variance of the QFK does not vanish exponentially for all cases. The gradual decay of the variance for QFK with HEAs seems to contradict Theorem 1 (1). However, the assumption that quantum circuits form 2-designs is not satisfied. In addition, the i -th term in shallow regions would contribute to the non-vanishing variance. We also note that the variance of QFK for ALAs decreases as the number of qubits increases because a normalization factor p scales linearly in the number of parameters, which scales linearly in the number of qubits.

Furthermore, we numerically check the variance of the i -th term of QFK in different layers. Fig. 4.14 shows the variance in different layers against the number of qubits for the three quantum circuits mentioned above. We focus on the term $k_{QF}^{(i)}$ whose parameter θ_i is the angle of the rotation Z gate that acts on the $\lceil n/2 \rceil$ -th qubit (in the middle of the width) in each layer. As for the tensor product quantum circuits and ALAs, the variance remains unchanged in the number of qubits. The variance for deeper ALAs decreases in the region of the small qubit numbers but then levels off. This is mainly because the number of the unitary blocks in the light-cone is saturated for the case of a large number of qubits; violating the 2-design assumption is also attributed to the tendency of the variance. On the other hand, the variance for HEAs declines more quickly than that of other cases. Such depth dependence would be because the HEAs become more expressive to satisfy the property of the 2-design as the depth increases. Again, the violation of the 2-design assumption on $\tilde{W}_{k,d}(\mathbf{x}, \theta_i)$ contributes to the tendency of the variance. We note that the variance of the terms increases more slowly than the fidelity-based quantum kernels.

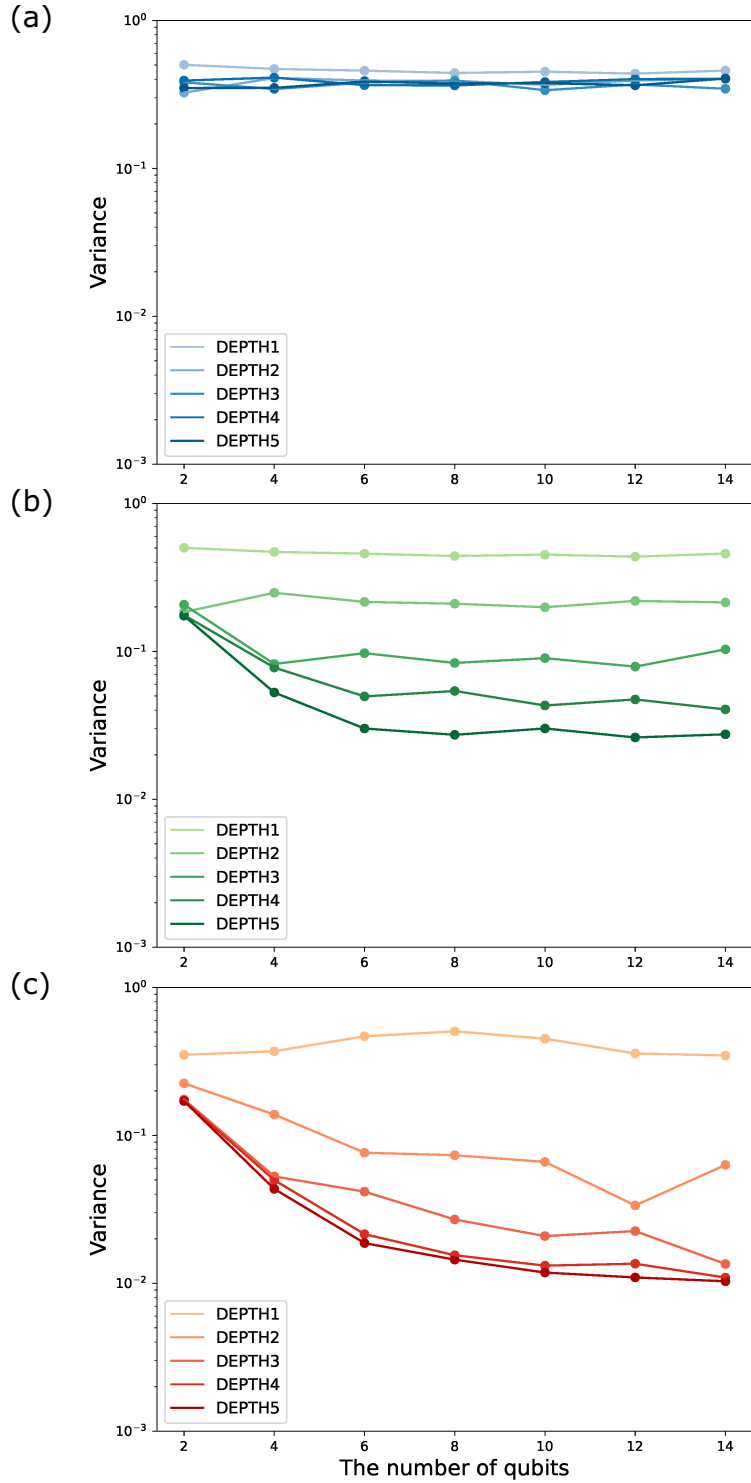


Figure 4.14: Variance of the i -th term of the QFK, $k_{QF}^{(i)}$, in different layers $d \in \{1, 2, 3, 4, 5\}$ against the number of qubits $n \in \{2, 4, 6, 8, 10, 12, 14\}$. We use (a) tensor product quantum circuits, (b) ALAs, and (c) HEAs. Note that the parameter θ_i is the angle of the rotation Z gate that acts on the $\lceil n/2 \rceil$ -th qubit in each layer.

4.2.7 Expressivity and Performance

QFKs possess an important property of avoiding the vanishing similarity issue. This is a necessary condition for the practicality of quantum kernel-based learning models. However, the property is insufficient to state that QFKs are powerful enough to perform machine learning tasks well. Hence, we further examine the expressivity of QFKs and then investigate the performance in specific classification tasks. Precisely, we exploit Fourier analysis to elucidate how expressive QFKs are. Also, we demonstrate a binary classification task where QFK can show high performance whereas fidelity-based quantum kernels perform poorly.

Expressivity via Fourier Analysis

The expressivity of both explicit and implicit models can be quantitatively examined via Fourier analysis [114, 197]. Ref. [114] demonstrates that quantum kernels can be expressed as an inner product of two Fourier series:

$$k_Q(\mathbf{x}, \mathbf{x}') = \sum_{\omega, \omega' \in \Omega} e^{i\omega\mathbf{x}} e^{i\omega'\mathbf{x}'} c_{\omega, \omega'} \quad (4.46)$$

with the Fourier coefficient $c_{\omega, \omega'}$ satisfying $c_{\omega, \omega'}^* = c_{-\omega, -\omega'}$ and $\Omega \in \mathbb{R}^d$. Here, d represents the dimension of data points. Therefore, we measure the expressivity of quantum kernels by numerically computing the magnitude of coefficients $\{c_{\omega, \omega'}\}$ over the effective frequency set. That is, the more non-zero Fourier coefficients the model has, the higher its expressivity is. However, computational costs to perform Fourier decomposition increases exponentially as the data dimension grows. We hence focus on one-dimensional data points, $d = 1$, and the truncated frequency set $\Omega = \{-12, -11, \dots, 10, 11, 12\}$. We use “curve_fit” provided by Scipy [198] to numerically compute Fourier coefficients; coefficients are obtained so that the quantum kernel fits to its Fourier representation.

Fig. 4.15 (a) shows the amplitudes of all Fourier coefficients $\{c_{\omega, \omega'}\}$ for fidelity-based quantum kernels and the normalized QFK for ALAs with the number of qubits $n = \{1, 2, 3\}$ and circuit depth $L = \{2, 3, 4\}$. We note that the index of Fourier coefficients (i.e., x-axis) in Fig. 4.15 (a) is aligned in the order shown in Fig. 4.15 (b). We find that QFKs have almost the same Fourier coefficients as the fidelity-based quantum kernel, indicating that QFKs are comparable to fidelity-based quantum kernels from the perspective of expressivity.

Performance Comparison

We further demonstrate a binary classification task where the performance of fidelity-based quantum kernels and QFKs differ due to the presence of vanishing similarity. We here consider one-dimensional synthetic datasets $\{(x_i, y_i)\}$ consisting of a one-dimensional input $x_i \in [-\pi, \pi)$ and its label $y_i \in \{+1, -1\}$, which is determined according to

$$y_i = \text{sign}(\sin(wx_i + b)) \quad (4.47)$$

with the dataset hyperparameters $w, b \in \mathbb{R}$. Intuitively, w and b determine the frequency and phase of the dataset, respectively; when w is large, high-order frequency components in terms of Fourier analysis are required to solve the task. Fig. 4.16 (a) shows examples of the dataset for $(w, b) = (2, 0.3), (4, 0.3)$ where binary labels are represented in blue and orange. We here focus on the dataset for $(w, b) = (2, 0.3)$ to examine the performance of SVMs with these quantum kernels for the different numbers of qubits. We use four-layer data re-uploading quantum feature maps

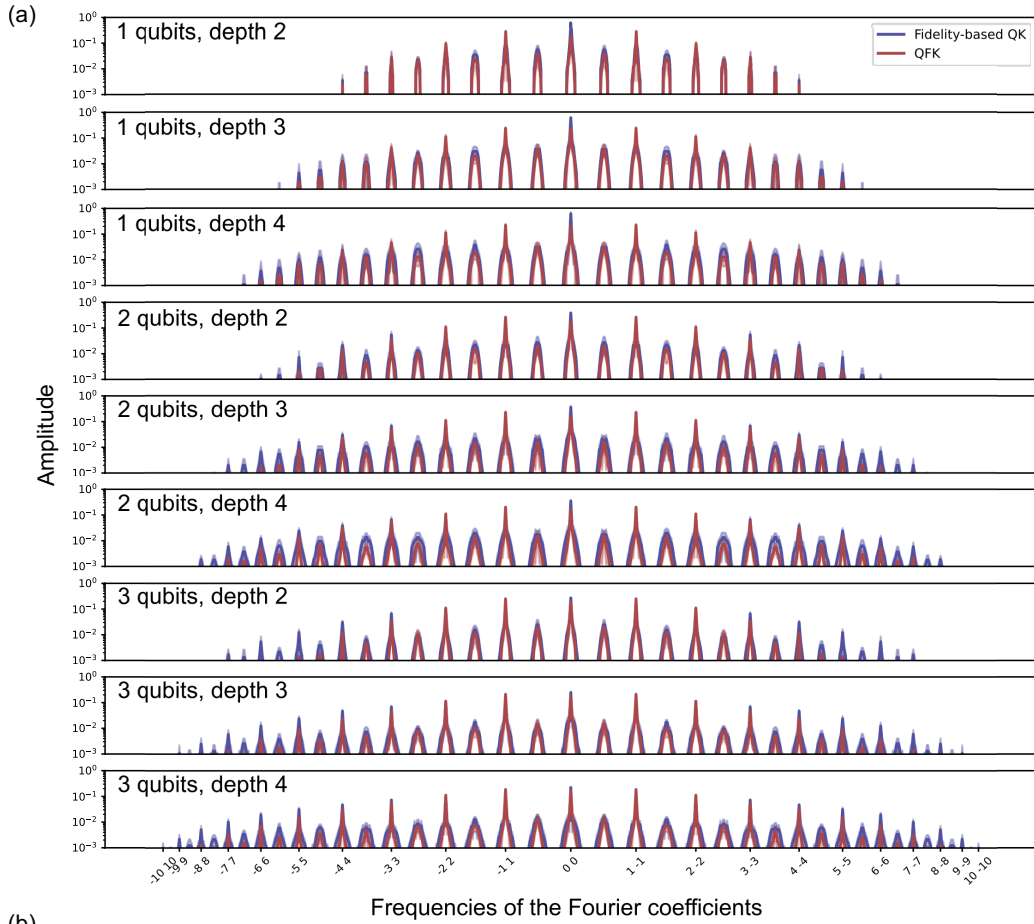


Figure 4.15: Amplitudes of Fourier coefficients of fidelity-based quantum kernels and QFKs. Panel (a) shows amplitudes of all Fourier coefficients (i.e., 625 coefficients in total) for fidelity-based quantum kernels (blue) and QFKs (red). Here, we used ALAs with the different number of qubits and circuit depth. Panel (b) illustrates how coefficients on the x-axis in (a) are aligned by taking the case for $\tilde{\Omega} = \{-2, -1, 0, 1, 2\}$.

where ALAs and tensor product quantum circuits are used as parameterized quantum circuit layers and input-embedding layers, respectively. Here, we rescale the data depending on the position of the qubit the single-qubit gate acts on; $\alpha_k = \alpha_{n+k} = kx_i$ for the tensor product quantum circuit in Fig. 4.12. We note that employing the rescaling technique can enhance the expressivity of quantum circuits and thus is a natural choice in case we have no idea about the difficulty of tasks we deal with.

Fig. 4.16 (b) shows the accuracy of fidelity-based quantum kernels and QFKs on the synthetic dataset against the number of qubits. QFKs perform consistently well regardless of the used qubit numbers; this is convincing because the classification task is trivial. On the other hand, the performance of fidelity-based quantum kernels deteriorates as the number of qubits increases because of the vanishing similarity issue. As shown in Fig. 4.16 (c), the Gram matrix of the fidelity-based quantum kernels for a large number of qubits is close to the identity matrix and thus the generalization performance worsens. However, off-diagonal elements of the Gram matrices for QFKs are not vanishing. These results imply the potential of QFKs to show better performance than fidelity-based quantum kernels for large quantum systems.

4.2.8 Conclusion & Outlook

This section addresses a serious issue called the vanishing similarity issue in the commonly-used fidelity-based quantum kernels and proposes QFKs as a quantum extension of the classical Fisher kernel. From analytical and numerical perspectives, we elucidate that QFKs can be free from vanishing similarity, whereas fidelity-based ones cannot. Fourier analysis is also numerically performed to clarify that the expressivity of QFKs is comparable to fidelity-based quantum kernels that can provably outperform classical counterparts for specific machine learning tasks. Moreover, we demonstrate a situation where QFK can perform better than the fidelity-based quantum kernel because of the absence of vanishing similarity. These results indicate that QFKs are promising candidates to show quantum advantages for practical use.

An open question is whether the setup in our analysis is realistic; that is, the 2-design assumptions might be challenging to realize in actual experimental settings. Indeed, some works empirically demonstrate that limiting the rotation angles for input-embedding layers can alleviate the issue at the expense of the model’s expressivity [177, 178, 199]. The implication of our analytical results is critical to give insight into the design principles of quantum kernel-based learning models in general. However, it is also essential to check if the issue is unavoidable in more realistic situations. An exciting path is to investigate quantum kernel methods from the perspective of geometric quantum machine learning [200–204]; inductive bias such as symmetry and permutation invariance is reflected on building QML models.

Moreover, further investigation is needed to demonstrate a practical advantage of QFKs. While we show the expressivity of QFKs via Fourier analysis, it is not thoroughly examined because of the computational difficulty in Fourier decomposition. Thus, it would be critical to see the performance of QFKs for actual machine learning tasks. Also, due to the unique structure in QFKs, i.e., UBU^\dagger , it would be interesting to explore the performance in tasks involving quantum dynamics; the structure can be seen in measures to investigate quantum chaos and quantum information scrambling, such as the Loschmidt echo [205, 206] and out-of-time-ordered correlator functions [207, 208].

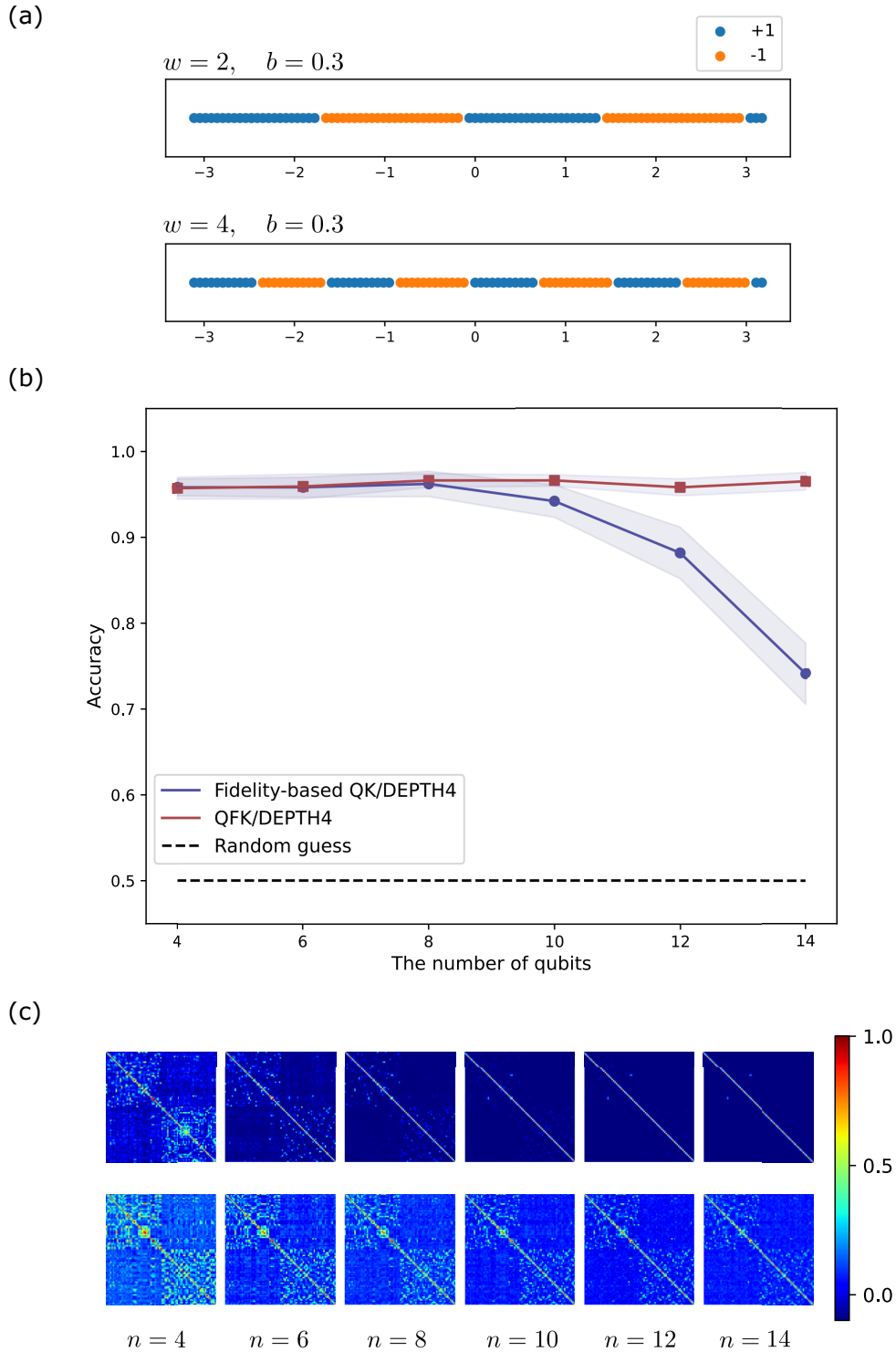


Figure 4.16: Classification performance of Fidelity-based quantum kernels and QFKs on synthetic datasets. (a) Examples of synthetic datasets for $(w, b) = (2, 0.3)$ and $(2, 0.3)$. (b) Accuracy of fidelity-based quantum kernels (blue) and QFKs (red) against the number of qubits $n \in \{2, 4, 6, 8, 10, 12, 14\}$. (c) Gram matrices obtained for the classification tasks: top figures for fidelity-based quantum kernels and bottom ones for QFKs.

Chapter 5

Quantum Noise-Induced Reservoir Computing

This chapter examines quantum reservoir computing (QRC) from a practical perspective. As shown in Sec. 3.3, QRC exploits complex quantum dynamical systems to enhance the performance of temporal information processing tasks. The key of this QRC framework is the *quantum reservoir* (QR), which is the input-driven quantum system playing a role in extracting features of time-series data; to do so, QRs nonlinearly map input sequences to a high-dimensional space, i.e., the quantum-enhanced feature space. To date, several platforms, such as disordered quantum spins [36, 158], fermionic or bosonic networks [160, 209], and harmonic oscillators [210, 211], have been proposed as candidates for QRs. On the other hand, further investigation is needed to design performant and efficiently implementable QR systems for practical use.

In this thesis, we propose a new QRC framework that utilizes quantum noise as a computational resource for temporal information processing. In the NISQ era, significant effort has been made to reduce the noise to fully exploit the power of quantum computing. In contrast to the common thought that quantum noise is detrimental, our strategy is to make use of such unavoidable quantum noise to enrich the complexity of quantum dynamical systems for temporal tasks. We elaborate on our scheme and its experimental demonstration on gate-based superconducting quantum processors in Sec. 5.1. Moreover, we quantitatively characterize the information processing capability induced by quantum noise via a tool called temporal information processing capacity [212]. Numerical simulations identify the type of noise that can induce the capability. Also, we use the tool to analyze the information processing capacity of QR systems on quantum hardware. We summarize the quantitative analysis of quantum noise-induced reservoir systems in Sec. 5.2.

5.1 Proof-of-Principle Demonstration¹

This section proposes a new paradigm of QRC that utilizes quantum noise on actual superconducting quantum processors to harness the performance of temporal information processing tasks. We examine the performance of our scheme on a benchmark time-series regression task and a practical classification task to identify the object from its sequential sensor data, showing our proposal outperforms classical linear models. These results suggest the potential of quantum noise as a computational resource to perform temporal information processing.

¹Results shown in this section are based on the author's work [41].

5.1.1 Introduction

An open question of QRC is how to design QR systems that can be implemented efficiently and perform well for specific sequential data processing. In conventional reservoir computing, guidelines for constructing reservoirs have been explored. An example is to set the spectral radius of weights in reservoir layers to less than one because the condition ensures the echo state property (ESP), a prerequisite of the reservoirs to forget its initial state asymptotically [141, 213, 214]. However, physical reservoir computing, which encompasses the concept of QRC, has difficulty in constructing performant physical reservoir systems that satisfy the ESP. This suggests the need to investigate design principles of physical reservoirs thoroughly. Thus, QRC has been theoretically and numerically studied to elucidate what types of systems are amenable to implementation [36, 155–157] and how well the QRs can perform [158–164].

This section proposes a new framework of QRC that exploits the quantum noise ubiquitous in quantum hardware to enhance the power of temporal information processing. One would like to reduce quantum noise as much as possible because it hampers the power of quantum computing. On the other hand, some literatures have demonstrated the potential of quantum noise to work positively in some specific situations. For instance, several types of quantum noise are used to induce universal quantum computation [215], to enhance the robustness of quantum classifiers [216], and to prepare high-fidelity thermal states [217]. This motivates us to propose a new framework; hardware-specific quantum noise is utilized to enrich the complexity of quantum dynamical systems for temporal processing.

We implement our scheme on IBM superconducting quantum processors to show its performance experimentally. Specifically, we work on two temporal tasks: emulation of the Nonlinear Auto-Regressive Moving Average dynamics (NARMA task) and classification of different objects using sensor signals gained by grabbing them (object classification task). Then, the QR systems realized on “ibmq_16_melbourne” and “ibmq_toronto” perform better than classical linear models for the NARMA and object classification tasks, respectively. These results indicate the potential of quantum noise-induced systems as candidates for implementation-friendly QR systems.

The rest of this section is given as follows. First, we provide a quantum noise-induced reservoir computing framework in Sec. 5.1.2. Then, we experimentally demonstrate the performance of our scheme on real quantum processors in Sec. 5.1.3. Lastly, Sec. 5.1.4 concludes this section.

5.1.2 Quantum Noise-Induced Reservoir Systems

In the following, we present our QRC scheme that exploits unavoidable quantum noise on actual quantum hardware.

As shown in Sec. 3.3, the time evolution of the QR system is described by an input-dependent CPTP map, i.e., $\mathcal{T}_{u_t}(\cdot)$ in Eq. (3.59). The CPTP map plays a crucial role in the feature extraction of time-series data and should be designed to achieve high performance. For example, a map employed in Ref. [36] simultaneously drives the reservoir system and an input-dependent qubit system by an input-independent unitary operator, as shown in Eq. (3.60). Ref. [155] also implemented the same map with an additional mechanism to forget its initial state. In contrast to these maps, we propose a CPTP map that explicitly exploits the dissipative nature of quantum systems realized on quantum hardware. More concretely, the dynamics of our QR system is represented as follows;

$$\begin{aligned} \rho_t &= \mathcal{T}_{u_t}(\rho_{t-1}) \\ &= \mathcal{E}_{qn} \left(U(u_t) \rho_{t-1} U^\dagger(u_t) \right), \end{aligned} \tag{5.1}$$

where $U(u_t)$ is a unitary operator dependent on input u_t and $\mathcal{E}_{qn}(\cdot)$ denotes an un-modeled CPTP map corresponding to quantum noise (i.e., the dynamical behavior) in quantum hardware. Eq. (5.1) means that our scheme drives the QR system by quantum noise intrinsic in quantum hardware as well as an input-dependent unitary operator. This formulation allows us to naturally exploit the unwanted quantum noise as a computational resource for QRC.

Notably, some types of quantum noise that can occur in actual quantum devices possess the echo state property (ESP), an indispensable property for reservoirs. The ESP is defined as follows; given a sequence of input $\mathbf{u}_l = [u_1, u_2, \dots, u_l]^T$, reservoir output vectors $\hat{h}(\rho_0, \mathbf{u}_l)$ and $\hat{h}(\rho'_0, \mathbf{u}_l)$ whose initial states are arbitrary quantum states, ρ_0 and ρ'_0 , respectively, hold

$$\lim_{l \rightarrow \infty} \|\hat{h}(\rho_0, \mathbf{u}_l) - \hat{h}(\rho'_0, \mathbf{u}_l)\|_2 = 0. \quad (5.2)$$

The property is essential for reservoir systems to ensure the reproducibility of reservoir computing models. As an extension of Eq. (5.2) for density operators [161, 162], we can also formulate the ESP as

$$\lim_{l \rightarrow \infty} \|\rho_l - \rho'_l\|_2 = 0. \quad (5.3)$$

Note that Eq (5.3) would be a sufficient condition of Eq (5.2). Then, Eq (5.3) holds for some types of quantum noise. An example is the depolarizing noise errors defined as

$$\mathcal{E}_{DEP}(\rho) = p \frac{I}{d} + (1 - p)\rho \quad (5.4)$$

where ρ is the density operator representation of a d -dimensional quantum state and p is the probability of swapping the original system with the completely mixed state. We can easily show that the QR systems where the depolarizing error successively occurs after time evolution, i.e., $\mathcal{E}_{qn}(\cdot) = \mathcal{E}_{dep}(\cdot)$ in Eq. (5.1), can satisfy the ESP; any QR systems under the noise asymptotically converge to the completely mixed state. Also, the amplitude damping noise \mathcal{E}_{AD} for a single-qubit can satisfy the ESP. Recall that the amplitude damping is expressed as

$$\mathcal{E}_{AD}(\rho) = E_1 \rho E_1^\dagger + E_2 \rho E_2^\dagger, \quad (5.5)$$

where

$$E_1 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad E_2 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}.$$

Here, γ denotes a probability of energy dissipation. In this case, we can show that $\|\rho_l - \rho'_l\|_2 \leq (1 - \gamma)^{l/2} \|\rho_0 - \rho'_0\|_2$; namely, the norm asymptotically get close to zero. Note that the ESP does not hold for errors represented by the unitary operation, such as over-rotation of single-qubits and unexpected entangling gate operations, because the norm is invariant under unitary transformation.

Moreover, Ref. [218] reported the actual quantum processors provided by IBM can produce complex noise, such as non-Markovian noise; this indicates that non-trivial natural noise could be exploited for the QRC framework, by actually implementing the scheme on real quantum devices.

5.1.3 Experimental Demonstration

We present our scheme using IBM superconducting quantum processors. In the following, after we detail the setup of our QR system, we show its performance on two time-series data tasks: the NARMA task and the object classification task.

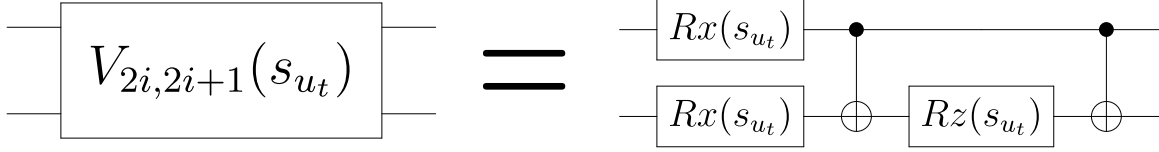


Figure 5.1: Quantum circuit representation of the local two-qubit unitary operator in Eq. (5.7).

Setup

In our proof-of-principle study, we consider n -qubit QR systems consisting of m two-qubit subsystems, i.e., $n = 2m$. Specifically, we consider the following input-dependent unitary operators in Eq. (5.1);

$$U(u_t) = V_{0,1}(u_t) \otimes V_{2,3}(u_t) \otimes \cdots \otimes V_{n-2,n-1}(u_t), \quad (5.6)$$

where $V_{2i,2i+1}(u_t)$ is the unitary operator acting on i -th subsystem for $i \in \{0, \dots, m-1\}$. Here, we assume the unitary operators $\{V_{2i,2i+1}(u_t)\}$ are identical for any i and expressed as

$$V_{2i,2i+1}(u_t) = CNOT_{2i,2i+1}Rz_{2i+1}(s_{u_t})CNOT_{2i,2i+1}Rx_{2i+1}(s_{u_t})Rx_{2i}(s_{u_t}), \quad (5.7)$$

where $S_{u_t} = au_t$ is the input scaled by the factor a . The quantum circuit representation of the unitary operator is illustrated in Fig. 5.1. We chose this type of hardware-efficient ansatz as the two-qubit unitary operator to check if the quantum noise could enrich the complexity of the dynamical system. We consider such limited types of gates to focus on the performance improvement by the intrinsic quantum noise. Actually, the two-qubit unitary operator in the noiseless situation cannot carry the information of input sequences when the initial state is $\rho_0 = |+\otimes^n\rangle\langle+\otimes^n|$ with $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ and the observables are a set of single-qubit Pauli Z , i.e., $O_k = Z_k$. In other words, the QR output vector results in the zero vector, $h(\rho_t) = [\text{Tr}[Z_0\rho_t], \dots, \text{Tr}[Z_{n-1}\rho_t]]^T = \mathbf{0}$ for any t . Moreover, as we assume the tensor product of the local unitary blocks for the input-dependent unitary operator, the whole QR system is also trivial. However, quantum noise on quantum hardware could cause significant effects such as interaction with the neighboring subsystems and environment (e.g., crosstalk [44, 219]).

Lastly, throughout this section, the initial state is $\rho_0 = |+\otimes^n\rangle\langle+\otimes^n|$ and the measurement observables are single-qubit Pauli Z operators, i.e., the QR output vector is represented as

$$h(\rho_t) = [\text{Tr}[Z_0\rho_t], \dots, \text{Tr}[Z_{n-1}\rho_t]]^T. \quad (5.8)$$

To obtain the expectation values at each timestep, we successively apply the input-dependent unitary operator from the beginning,

$$\rho_t = \mathcal{T}_{u_t} \circ \mathcal{T}_{u_{t-1}} \circ \cdots \circ \mathcal{T}_{u_1}(\rho_0) \quad (5.9)$$

and then measure the resultant quantum states on the computational basis $N_s = 8,192$ times. The procedure is described in Fig. 5.2. Also, we use “ibmq_16_melbourne” (Melbourne device) and “ibmq_toronto” (Toronto device) to perform temporal information processing tasks; the configuration of these devices is shown in Fig. 5.3, respectively. We chose these quantum processors with different configurations because the noise effect would differ due to the qubit-connectivity. Moreover, the optimization option of the transpiler for reducing noise in Qiskit [172] is set to zero to see the effect of quantum noise.

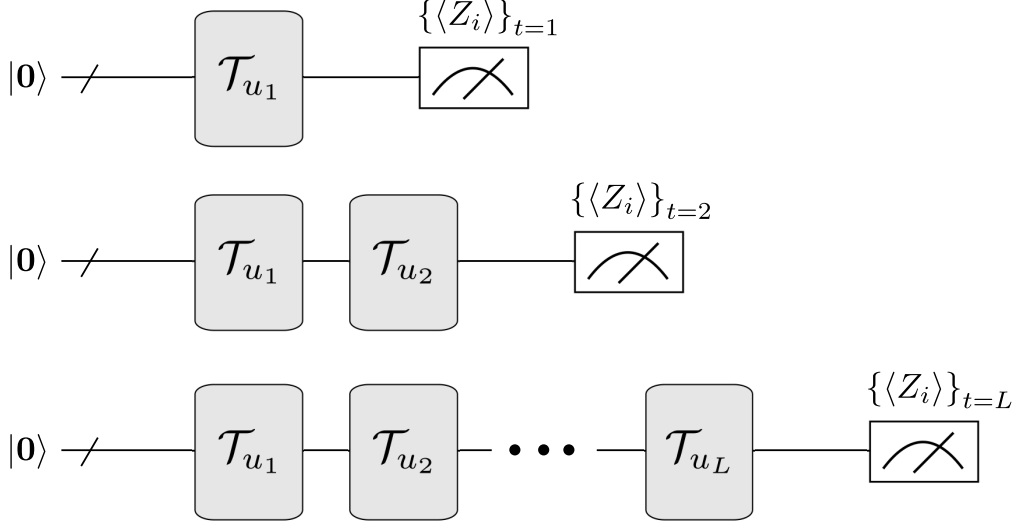


Figure 5.2: Circuit diagram of experimental demonstration of our scheme. Here, $|0\rangle$ stands for the initial state and \mathcal{T}_{u_i} is the CPTP map in Eq. (5.1). We repeatedly apply the CPTP map from the beginning to obtain the components of the QR output vector at each timestep.

NARMA Task

We first work on the NARMA task, a benchmark task used to evaluate the capability of dynamical models from perspectives of the nonlinearity and dependence on past output [220, 221]. The task aims to emulate the dynamics generating the NARMA output sequence $\{y_t\}_{t=1}^L$. An example studied in [36, 222] is described as

$$y_{t+1} = 0.4y_t + 0.4y_t y_{t-1} + 0.6u_t^3 + 0.1, \quad (5.10)$$

with the input sequence u_t . Another NARMA dynamics studied in [36, 162, 222] is expressed as

$$y_{t+1} = \alpha y_t + \beta y_t \left(\sum_{j=0}^{n_d-1} y_{t-j} \right) + \gamma u_{t-n_d+1} u_t + \delta, \quad (5.11)$$

where $(\alpha, \beta, \gamma, \delta) = (0.3, 0.05, 1.5, 0.1)$ and n_d is the order that determines the degree of the nonlinearity. In our experiments, we work on the following three NARMA dynamics; NARMA in Eq. (5.10) (we call it NARMA2), and NARMAs in Eq. (5.11) with $n_d = 5$ and $n_d = 10$ (NARMA5 and NARMA10, respectively). We notice that the number in the task name (e.g., 2 in “NARMA2”) indicates the order of nonlinearity.

The input sequence we handle for all the NARMA tasks is represented as follows;

$$u_t = 0.1 \left(\sin \left(\frac{2\pi\bar{\alpha}t}{T} \right) \sin \left(\frac{2\pi\bar{\beta}t}{T} \right) \sin \left(\frac{2\pi\bar{\gamma}t}{T} \right) + 1 \right), \quad (5.12)$$

where $(\bar{\alpha}, \bar{\beta}, \bar{\gamma}, T) = (2.11, 3.73, 4.11, 100)$. Note that the setting is used in, e.g., Ref. [36]. Here, the length of the input sequence is $L = 100$, where the first 10 timesteps are used for *washout*, the following 70 timesteps are used for training, and the remaining 20 timesteps are used for testing. The washout phase is necessary for the QR system to forget its initial state ρ_0 . Fig. 5.4 shows the inputs and the target output sequences for each NARMA task.

In the experiments, we used the Melbourne and Toronto devices to check the difference in the performance due to the hardware-specific noise. We implement the QR system in Eq. (5.6) with

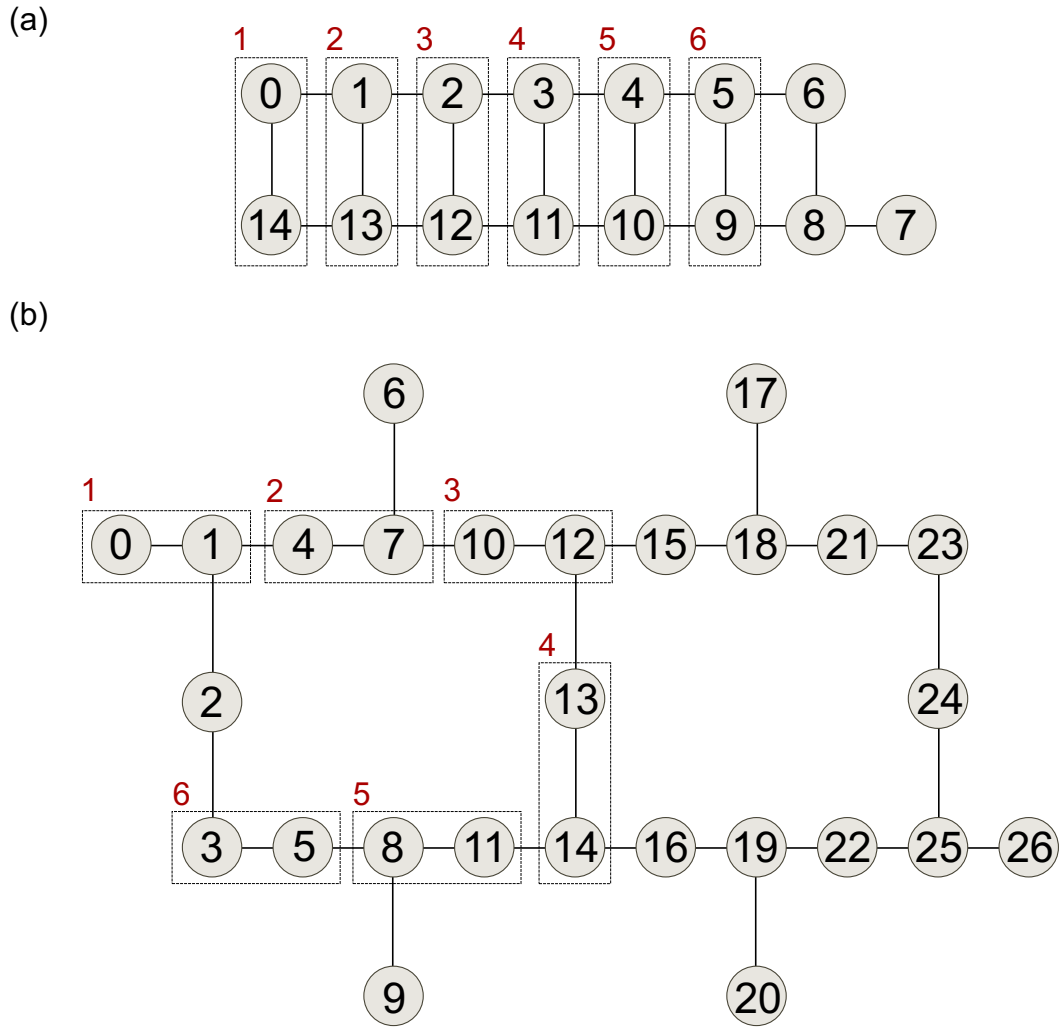


Figure 5.3: Configuration of quantum hardware used in the experiments: (a) Melbourne device and (b) Toronto device. Nodes and edges are used to indicate qubits and physical connectivity, respectively. The number shown in each node (black) denotes the label of the corresponding qubit. Dashed boxes show the subsystems constituting the whole QR system (each subsystem is labeled by the number in red).

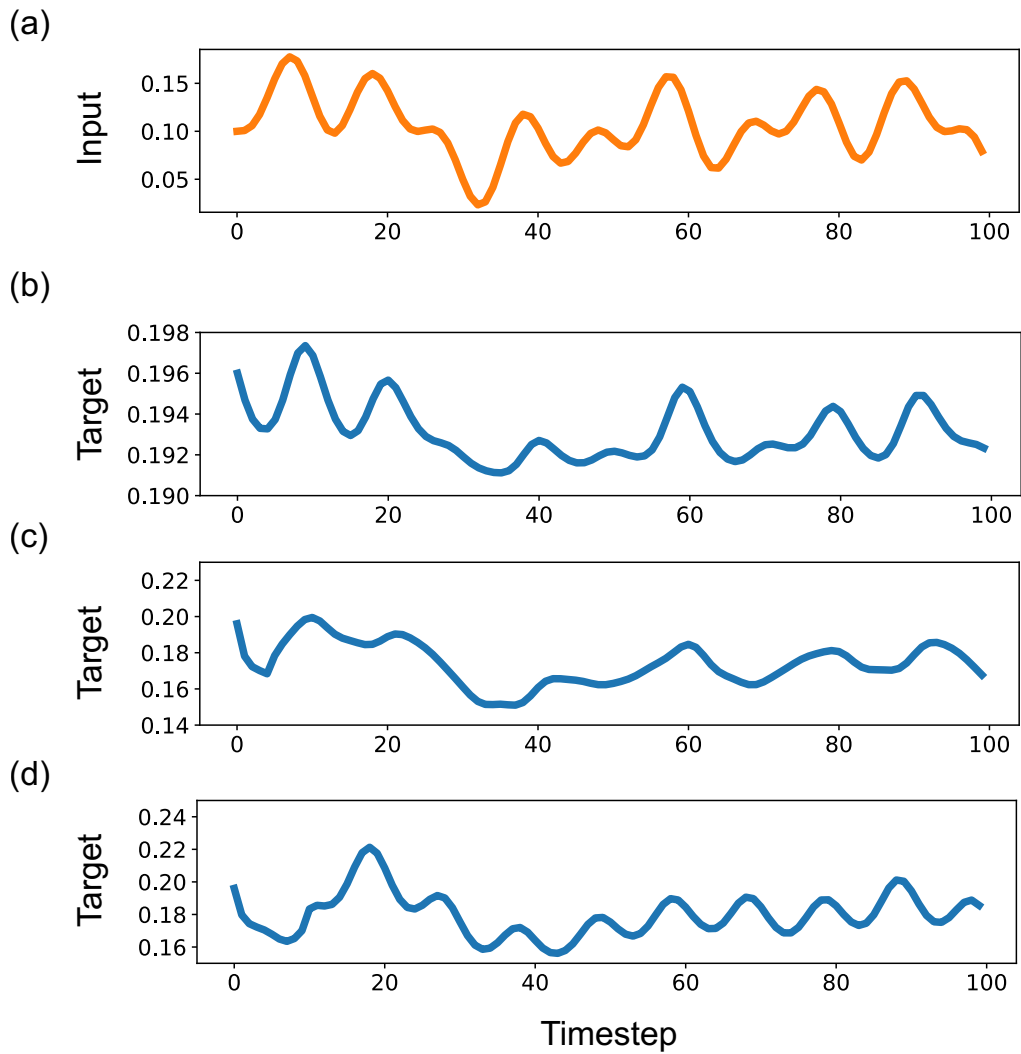


Figure 5.4: Time-series data used for the NARMA tasks. (a) Input sequence, (b) NARMA2, (c) NARMA5, and (d) NARMA10.

Table 5.1: List of NMSEs for (a) NARMA2, (b) NARMA5, and (c) NARMA10. For comparison, NMSEs of the classical linear regression model (denoted as LR) are shown. Here, bold scripts indicate the best NMSEs for NARMA tasks.

(a) NARMA2

| | QR systems | | | | | | Classical model |
|------|----------------------|----------------------|--|----------------------|----------------------|----------------------|----------------------|
| | Melbourne device | | | Toronto device | | | LR |
| | $m = 4$ | $m = 5$ | $m = 6$ | $m = 4$ | $m = 5$ | $m = 6$ | |
| Mean | 1.3×10^{-5} | 1.3×10^{-5} | 8.9×10^{-6} | 2.9×10^{-5} | 2.5×10^{-5} | 2.2×10^{-5} | 1.8×10^{-5} |
| Std | 6.3×10^{-5} | 2.8×10^{-6} | 2.8×10^{-6} | 6.7×10^{-6} | 1.3×10^{-5} | 4.1×10^{-6} | — |

(b) NARMA5

| | QR systems | | | | | | Classical model |
|------|--|--|--|----------------------|----------------------|----------------------|----------------------|
| | Melbourne device | | | Toronto device | | | LR |
| | $m = 4$ | $m = 5$ | $m = 6$ | $m = 4$ | $m = 5$ | $m = 6$ | |
| Mean | 1.3×10^{-3} | 1.3×10^{-3} | 1.3×10^{-3} | 2.7×10^{-3} | 2.2×10^{-3} | 1.9×10^{-3} | 2.6×10^{-3} |
| Std | 6.7×10^{-4} | 4.0×10^{-4} | 5.3×10^{-4} | 9.6×10^{-4} | 2.1×10^{-4} | 3.7×10^{-4} | — |

(c) NARMA10

| | QR systems | | | | | | Classical model |
|------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------|--|
| | Melbourne device | | | Toronto device | | | LR |
| | $m = 4$ | $m = 5$ | $m = 6$ | $m = 4$ | $m = 5$ | $m = 6$ | |
| Mean | 1.9×10^{-3} | 2.1×10^{-3} | 2.0×10^{-3} | 3.6×10^{-3} | 3.1×10^{-3} | 2.3×10^{-3} | 9.7×10^{-4} |
| Std | 4.8×10^{-4} | 6.0×10^{-4} | 3.5×10^{-4} | 8.5×10^{-4} | 1.2×10^{-3} | 5.3×10^{-4} | — |

the scaling factor in local unitary in Eq. (5.7) as $a = 2$. As for the physical configuration, we assign the subsystems as shown in Fig. 5.3, where each subsystem is indicated by a dashed black box with its label (the number colored in red). Then, we examine the performance of the QR systems with $n = 8, 10, 12$ (correspondingly, $m = 4, 5, 6$ subsystems). Note that the subsystems labeled 1 through m are used to implement the QR system of size $2m$ -qubit throughout this section. Also, the experiments had been performed during the period between Aug. 16th and Nov. 2nd in 2020.

Table. 5.1 summarizes the performance of our QR systems, where the results for QRs are the averaged performance over ten trials. Here, we use the metric, the normalized mean squared errors (NMSE) between the output in Eq. (5.8) and the target, defined as

$$NMSE = \frac{\sum_{t=t_l}^{t_e} (\bar{y}_t - y_t)^2}{\sum_{t=t_l}^{t_e} y_t^2}. \quad (5.13)$$

Here, t_l and t_e represent the start and the end of the timestep for the test phase, i.e., $t_l = 81$ and $t_e = 100$ in this case, respectively. Figs. 5.5 to 5.7 also show the result of the QRs for each NARMA task.

In the following, we show the performance of our QR systems from the following perspectives: (1) the dependence on quantum devices, (2) the dependence on the system size, and (3) the comparison with classical models.

(1) Dependence on quantum devices: First, Table. 5.1 shows that the performance heavily relies on the quantum hardware. The Melbourne device outperforms the Toronto device

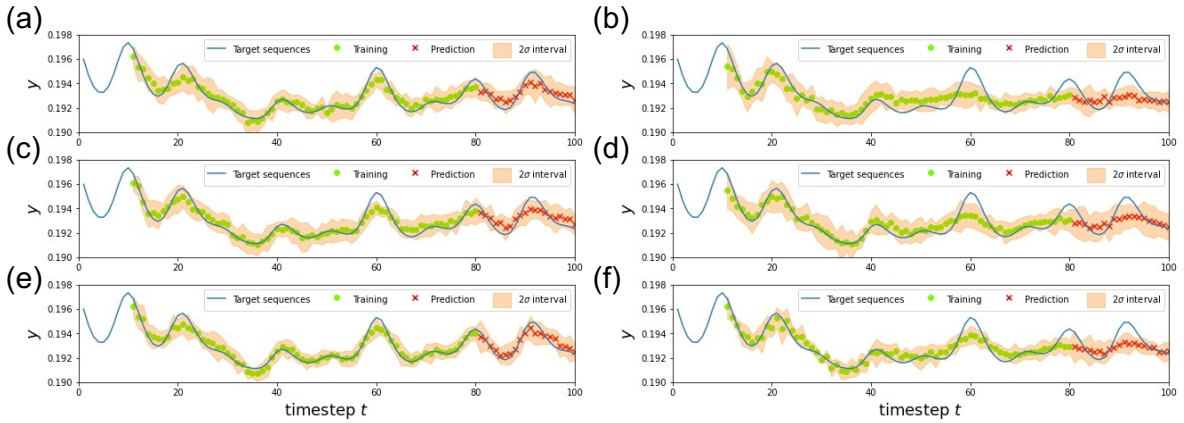


Figure 5.5: Visualization of the obtained results for NARMA2 using different QR systems. Panels (a), (c), and (e) show the results for 4, 5, and 6 subsystems on the Melbourne device, respectively. Similarly, panels (b), (d), and (f) are results for $m = 4, 5,$ and 6 using the Toronto device. Here, the blue line represents the target, and green circles and red crosses are the predictions in the training and testing phase, respectively. The orange regions indicate 2σ intervals. Figures reproduced from Ref. [41] by Y. Suzuki, Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto. Creative Commons Attribution 4.0 International license [DOI:10.1038/s41598-022-05061-w].

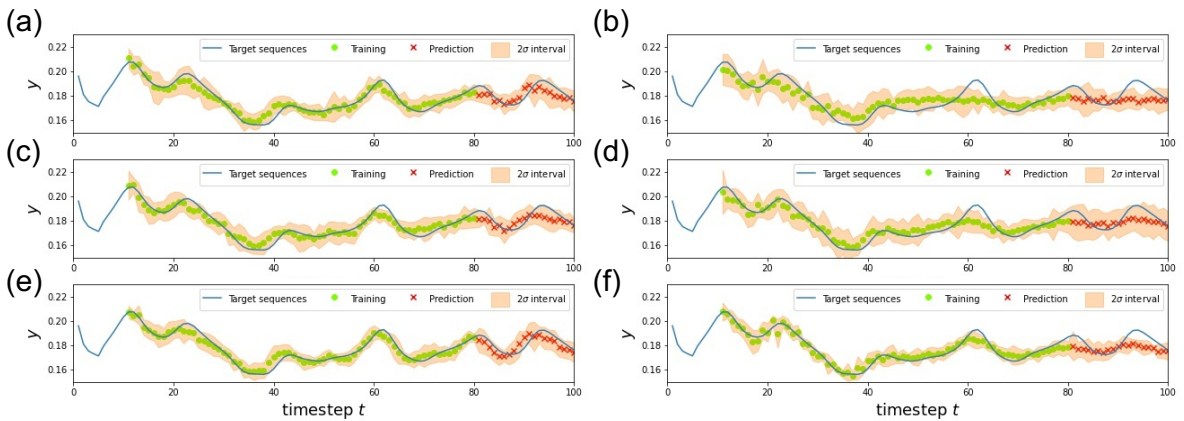


Figure 5.6: Visualization of the obtained results for NARMA5 using different QR systems. Panels (a), (c), and (e) show the results for 4, 5, and 6 subsystems on the Melbourne device, respectively. Similarly, panels (b), (d), and (f) are results for $m = 4, 5,$ and 6 using the Toronto device. Here, the blue line represents the target, and green circles and red crosses are the predictions in the training and testing phase, respectively. The orange regions indicate 2σ intervals. Figures reproduced from Ref. [41] by Y. Suzuki, Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto. Creative Commons Attribution 4.0 International license [DOI:10.1038/s41598-022-05061-w].

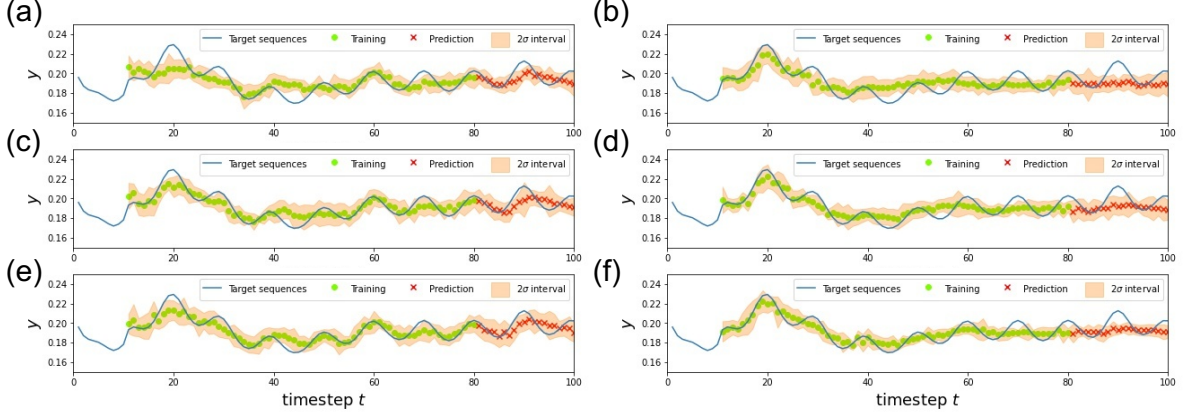


Figure 5.7: Visualization of the obtained results for NARMA10 using different QR systems. Panels (a), (c), and (e) show the results for 4, 5, and 6 subsystems on the Melbourne device, respectively. Similarly, panels (b), (d), and (f) are results for $m = 4, 5,$ and 6 using the Toronto device. Here, the blue line represents the target, and green circles and red crosses are the predictions in the training and testing phase, respectively. The orange regions indicate 2σ intervals. Figures reproduced from Ref. [41] by Y. Suzuki, Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto. Creative Commons Attribution 4.0 International license [DOI:10.1038/s41598-022-05061-w].

for all tasks even though mathematically identical operations are performed for both devices. We can clearly witness the tendency in Figs. 5.5 to 5.7; the Melbourne device can reproduce the trajectory better than the Toronto. This would be attributed to the noise level induced by the topology of quantum hardware; the noise in a device with the dense square lattice structure like the Melbourne device is more severe than the one with the sparse hexagonal structure like the Toronto due to the errors, such as the frequency-collision [223].

(2) Dependence on the system size: Next, we check the dependence of the performance on the system size. Ref. [36] numerically shows that increasing the system size of the QR can enhance performance. This motivates us to examine if the performance of our scheme can be improved as the system size is enlarged. As for the Toronto device, we can observe that the larger QR systems perform better for all NARMA tasks. The performance of the Melbourne device is also improved for the NARMA2 task, whereas the tendency is obscure for the NARMA5 and NARMA10. This result implies that the device architecture also affects how well the performance is improved with the increase in the number of subsystems.

(3) Performance comparison with classical models: Lastly, we compare the performance of our QR systems with classical learning models: a simple linear regression (LR) model and the standard ESN. We begin with the LR model that predicts the target by the output $\bar{y}_{t+1} = wu_t + b_0$ with optimized parameters w and b_0 . The NMSEs of the LR model for NARMA tasks are listed in Table 5.1. We can see that the QR systems on the Melbourne device outperform the LR model for NARMA2 and NARMA5, while the performance of the LR model is better for NARMA10. This indicates that the noise in the Melbourne device can enhance the performance, especially for these tasks. On the other hand, the Toronto device cannot outperform the LR model except for the 6 subsystems on the NARMA5 task, suggesting the types of hardware-specific quantum noise significantly affect the performance. Moreover, we employ the standard ESN model represented as

$$\bar{y}_t = W_{out}^T g(W^T \mathbf{x}_{t-1} + W_{in}^T u_t), \quad (5.14)$$

Table 5.2: List of the global average of NMSEs and the global minimum of NMSEs of the ESNs for NARMA tasks. The number of internal nodes of ESN are $N_{ESN} = 2, 5, 10, 20, 50$, which are denoted in the parenthesis. As for the global minimum of NMSE, the optimal spectral radius of W denoted as $\rho(W)$ is also shown.

| Task | Model | the global average of NMSE | | the global minimum of NMSE | | |
|---------|----------|----------------------------|----------------------|----------------------------|----------------------|-----------|
| | | Mean | Std | Mean | Std | $\rho(W)$ |
| NARMA2 | ESN (2) | 1.3×10^{-5} | 1.3×10^{-5} | 8.9×10^{-6} | 1.1×10^{-5} | 0.01 |
| | ESN (5) | 3.5×10^{-6} | 7.6×10^{-6} | 1.4×10^{-6} | 4.5×10^{-6} | 0.01 |
| | ESN (10) | 7.6×10^{-7} | 2.0×10^{-6} | 1.5×10^{-7} | 7.6×10^{-8} | 0.01 |
| | ESN (20) | 1.7×10^{-7} | 6.0×10^{-7} | 2.4×10^{-8} | 1.8×10^{-8} | 0.14 |
| | ESN (50) | 1.9×10^{-7} | 1.0×10^{-6} | 2.1×10^{-9} | 1.9×10^{-9} | 0.47 |
| NARMA5 | ESN (2) | 1.8×10^{-3} | 8.5×10^{-4} | 1.5×10^{-3} | 2.5×10^{-4} | 0.01 |
| | ESN (5) | 4.9×10^{-4} | 1.1×10^{-3} | 2.1×10^{-4} | 4.3×10^{-4} | 0.23 |
| | ESN (10) | 1.1×10^{-4} | 2.9×10^{-4} | 2.7×10^{-5} | 1.7×10^{-5} | 0.18 |
| | ESN (20) | 1.9×10^{-5} | 9.7×10^{-5} | 4.3×10^{-6} | 1.4×10^{-6} | 0.13 |
| | ESN (50) | 1.2×10^{-5} | 2.5×10^{-5} | 2.4×10^{-6} | 2.3×10^{-6} | 0.05 |
| NARMA10 | ESN (2) | 1.3×10^{-3} | 7.2×10^{-4} | 1.2×10^{-3} | 6.4×10^{-4} | 0.01 |
| | ESN (5) | 7.7×10^{-4} | 5.9×10^{-4} | 5.7×10^{-4} | 3.4×10^{-4} | 0.23 |
| | ESN (10) | 4.2×10^{-4} | 4.1×10^{-4} | 2.6×10^{-4} | 2.2×10^{-4} | 0.50 |
| | ESN (20) | 2.6×10^{-4} | 2.5×10^{-4} | 1.8×10^{-4} | 9.0×10^{-5} | 0.64 |
| | ESN (50) | 1.0×10^{-4} | 2.0×10^{-4} | 4.9×10^{-5} | 3.8×10^{-5} | 0.68 |

where W_{out} is the optimized weight and $g(\cdot)$ is the element-wise hyperbolic tangent function. Also, W and W_{in} are randomly initialized weight matrices. The performance of the ESN models depends on the internal nodes N_{ESN} (i.e., the state vector $\mathbf{x}_t \in \mathbb{R}^{N_{ESN}}$), and the spectral radius of W . Thus, we investigate the performance for $N_{ESN} = 2, 5, 10, 20, 50$ and the spectral radius ranging from 0.01 to 1 in increments of 0.01 over 100 trials with different W_{in} and W . The performance is shown in Table. 5.2, where we calculate the global average of NMSE and the global minimum of NMSE introduced in Ref. [150]. Roughly speaking, the former means the expected performance, and the latter represents the optimal performance. Then, Table. 5.1 together with Table. 5.2 demonstrates that the Melbourne device is comparable to the ESN with several nodes, and the Toronto device is worse than the ESN with only a few nodes. However, this result is not so surprising because our QR system is over-simplified in this study to see the contribution of quantum hardware noise. For example, we could improve the performance by changing the set of gates, the data-encoding scheme, and the types of quantum hardware architecture. In this sense, our QR system could perform well by fully tuning these hyperparameters.

Object Classification Task

Next, we work on a practical time-series information processing task: the object classification task. The goal of this task is to identify the objects from the sequence of sensor data obtained by grabbing them with a robotic hand.

In the experiments, we deal with three objects shown in Fig. 5.8 (a): a cube made of ABS LEGO blocks (object A), a polylactic acid (PLA) cube, and a sphere created using a 3D printer (objects B and C, respectively). We collect the sensor data of these objects by grabbing them using the triboelectric nanogenerator (TENG) sensor in Fig. 5.8 (b) and the grabbing robot in Fig. 5.8 (c). The TENG sensor detects pressure using an electronegative silicone bubble-shaped

dome and an electropositive nylon layer as the active materials. The robot grasped these objects 25 times, and the pressure on the TENG sensor was recorded, as illustrated in Fig. 5.8 (d). In the classification task, we used 20 cycles of sensor data of 90 timesteps and pre-processed as follows; $u_t = u'_{t+1} - u'_t$, where u_t and u'_t denote the pre-processed and the raw data at time t , respectively. We notice that the computational cost for the pre-processing is negligibly small.

As for the reservoir-based classifiers, we adopt the learning method employed in Ref. [153, 224]. Here, the output of the reservoir model predicts the one-hot vector representation of the label that corresponds to the input sequence at every timestep. More precisely, the linear regression technique is employed to train the readout weight $W_{out} \in \mathbb{R}^{N+1} \times \mathbb{R}^K$, where N is the number of observed signals from the reservoir state and K is the number of classes. Recall that we also take into account the bias term for W_{out} ; namely, the QR output vector includes a bias term, i.e., $\tilde{h}(\rho_t) = (h^T(\rho_t), 1)^T$. The optimal weights can be obtained by simply solving the following equation;

$$[\mathbf{Y}_1, \dots, \mathbf{Y}_{N_{train}}] = W_{out}^T [\tilde{\mathbf{X}}_1, \dots, \tilde{\mathbf{X}}_{N_{train}}], \quad (5.15)$$

where $\tilde{\mathbf{X}}_i = [\tilde{h}(\rho_{t_s}^i), \dots, \tilde{h}(\rho_{t_e}^i)]$ and $\mathbf{Y}_i = [y^i, \dots, y^i]$. Here, ρ_t^i is the reservoir state at timestep t given input sequence \mathbf{u}_i , and N_{train} is the total number of the training data. In addition, y^i is the one-hot vector representation of the target output for the i -th training data: for instance, the target values can take $[0, 1]^T$ or $[1, 0]^T$ for binary classification tasks. Then, the optimized parameter W_{out}^{opt} is used to predict the label of the unseen testing data \mathbf{u}_{new} as follows;

$$t_{new} = \operatorname{argmax} \left(\operatorname{mean}_t \left(W_{out}^{optT} \tilde{\mathbf{X}}_{new} \right) \right), \quad (5.16)$$

with $\tilde{\mathbf{X}}_{new} = [\tilde{h}(\rho_{t_s}^{new}), \dots, \tilde{h}(\rho_{t_e}^{new})]$.

In the experiments, we focus on the following situations: three binary classification tasks (i.e., A vs. B, A vs. C, and B vs. C) and a three-class classification task (i.e., A vs. B vs. C). We performed ten-fold cross validation for all tasks to assess the performance via the averaged accuracy. We here consider the QR system composed of 4 subsystems (labeled one to four) on the Toronto device in Fig. 5.3, where we set $a = \pi$ for the unitary operators in Eq. 5.7. In addition, we discard the first 40 timesteps for washout, and the remaining 49 timesteps are used for the learning, i.e., $t_s = 41$, $t_e = 89$. We conducted the experiments from Feb. 22nd to Feb. 23rd in 2021.

Table 5.3 summarizes the classification accuracy of the QR systems and a simple linear classification model. The linear classical classifier predicts the class by the output $\bar{y}_t = W_{out}^T u_t + b$ for timestep t . We find that our scheme performs better than the classical linear model for the classification of A and C, and the three-class classification, whereas the linear model is superior for the task with objects A and B. Remarkably, the accuracy of the QR system for the three-class identification is better than the linear model by 0.3, indicating the QR system's potential to accurately classify different objects from the sensor data. Fig. 5.9 shows the confusion matrices, which summarize the correct and incorrect classification results for the three-class classification: i.e., the diagonal elements correspond to the accurate prediction, and off-diagonal elements indicate misclassification. The matrix clarifies that the linear models cannot recognize object C when trained with objects A and B, while our QR system can. These experiments demonstrate our scheme's potential to perform temporal data classification tasks well.

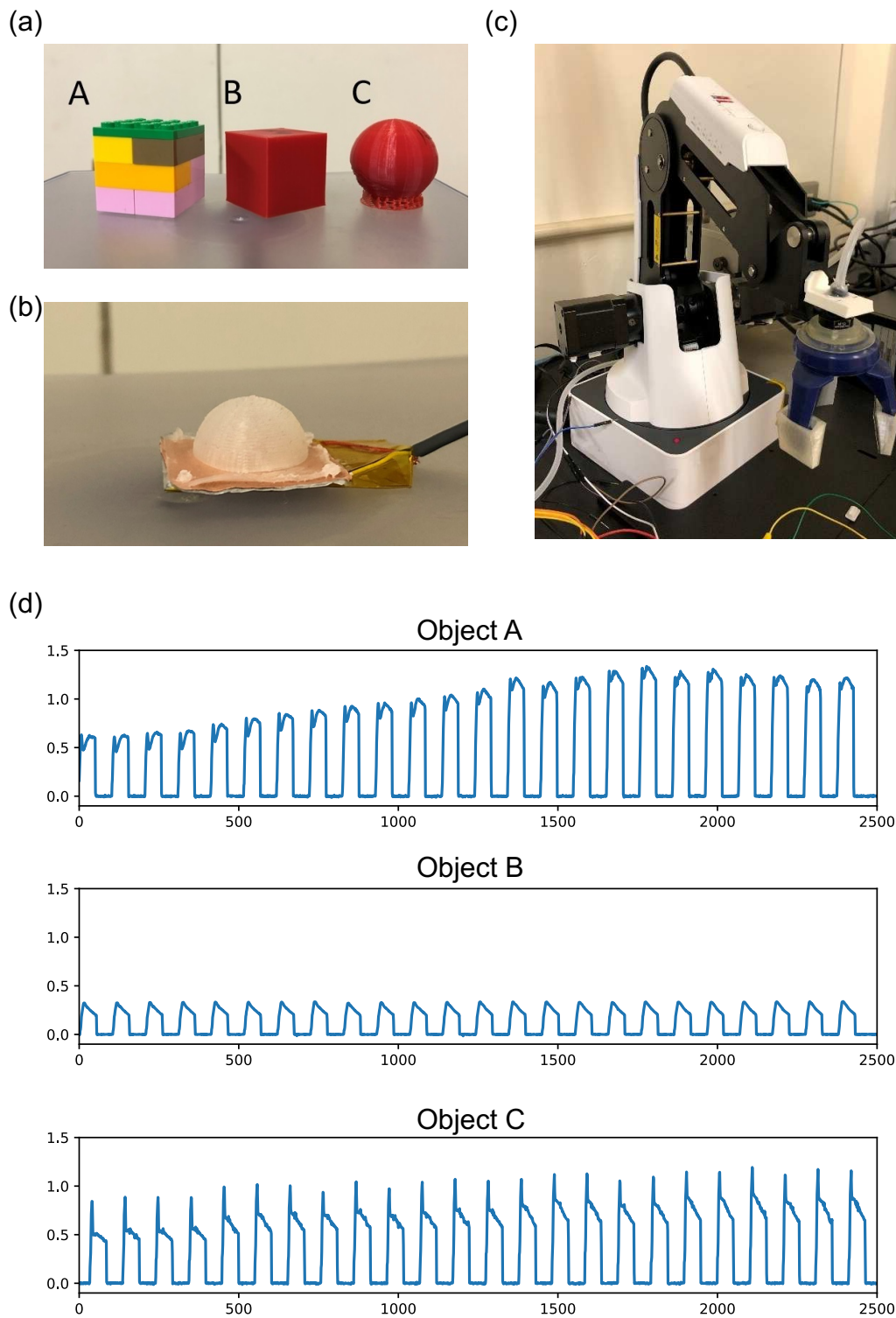


Figure 5.8: Sensor data obtained from objects and an instrument used to collect time-series data. (a) Objects used for the classification tasks, (b) the TENG sensor, (c) the grabbing robot, and (d) sequences obtained by grasping these objects. Figures reproduced from Ref. [41] by Y. Suzuki, Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto. Creative Commons Attribution 4.0 International license [DOI:10.1038/s41598-022-05061-w].

Table 5.3: Classification accuracy of QR systems and a simple linear regression classifier for object classification tasks.

| (a) Our QR systems | | | | |
|--------------------|---------|-------------|-------------|---------------|
| | A vs. B | A vs. C | B vs. C | A vs. B vs. C |
| Mean | 0.90 | 1.00 | 1.00 | 0.95 |
| Std | 0.20 | 0.00 | 0.00 | 0.11 |

| (b) Linear classifiers | | | | |
|------------------------|-------------|---------|-------------|---------------|
| | A vs. B | A vs. C | B vs. C | A vs. B vs. C |
| Mean | 1.00 | 0.90 | 1.00 | 0.67 |
| Std | 0.00 | 0.20 | 0.00 | 0.00 |

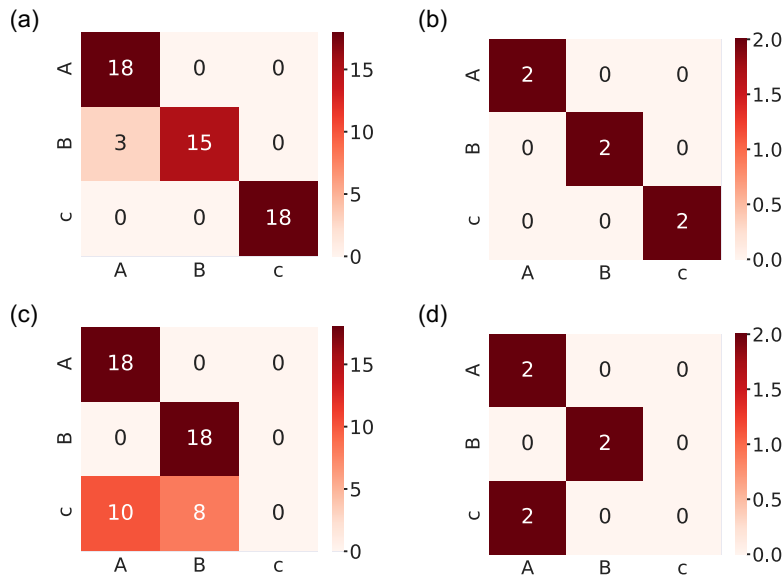


Figure 5.9: Confusion matrices of QR systems and classical linear classifiers. The results of our QR systems for training and testing are shown in panels (a) and (b), respectively. Similarly, the matrices in (c) and (d) respectively show the results of the linear classifiers for training and testing. Figures reproduced from Ref. [41] by Y. Suzuki, Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto. Creative Commons Attribution 4.0 International license [DOI:10.1038/s41598-022-05061-w].

5.1.4 Conclusion & Outlook

This section proposes a new paradigm of QRC that positively utilizes unavoidable quantum noise on quantum hardware. Our scheme realized on IBM superconducting quantum processors demonstrates that hardware-specific quantum noise can enhance the complexity of the dynamical system; as a result, the quantum noise-induced reservoir systems can outperform the linear classical models for the NARMA tasks and the object classification tasks. This experimental study suggests that quantum noise is potentially useful for temporal processing tasks. We note that this scheme is applicable even in the NISQ era.

This scheme requires further investigation for practical use. First, elucidating the underlying mechanism of quantum noise on quantum hardware is critical for designing performant QR systems. More specifically, our primary objective will be to quantitatively analyze what kind of quantum noise can enhance the performance. Quantum process tomography [225–227] can be a valuable tool to explore the effect of quantum noise from the perspective of quantum operations. Recent work also proposed methods to probe complex noise such as crosstalk [44, 219]. These approaches will help to identify a suitable configuration of quantum reservoir systems and an appropriate set of unitary gates that constitute input-dependent unitary operators. In the next section, we elaborate on the link between the performance and types of quantum noise from the information processing capacity perspective.

Moreover, improving the processing speed is imperative. In our scheme, we have to execute quantum circuits iteratively to obtain the expectation values at each timestep, as shown in Fig. 5.2. The process requires $N_s L$ quantum circuits for the time-series data of length L and the number of measurement shots to obtain expectation values N_s , which is time-consuming and would prevent our scheme from practical applications. We had no choice but to employ this procedure because of the limited operations on the quantum devices at that time. However, thanks to the advances in quantum hardware, a mid-circuit measurement is now available. The technique allows us to keep running circuits even if the measurements are performed in the middle; we have only to execute quantum circuits N_s times to acquire the same outputs from the QR systems. An exciting direction would be to explore the number of measurement shots to achieve performance comparable to the case for a sufficient amount of shots.

5.2 Information Processing Capability Induced by Quantum Noise²

We build on the work in the previous section [41] and quantitatively analyze the information processing capabilities induced by quantum noise via a tool called *temporal information processing capacity*. We demonstrate that QR systems driven by specific quantum noise models can induce information processing ability. We also verify our views by examining the QR systems on actual quantum hardware and obtain similar characteristics. These results also support our idea that quantum noise can positively enhance the power of temporal information processing.

5.2.1 Introduction

The previous section 5.1 provides a QRC framework that positively utilizes quantum noise to enrich the power of time-series data processing. Despite the exciting experimental demonstration

²Results shown in this section are based on the author’s work [42]. Note that the first authorship is shared with Dr. Tomoyuki Kubota and the author. The author and T.K. mainly contribute to the implementation of our framework on actual quantum devices and the numerical analysis of its capability via temporal information processing capacity, respectively.

of the scheme on actual superconducting quantum processors, the underlying mechanism to induce the capability is still unclear. For practical applications, precise knowledge about what kind of quantum noise can cause such positive effects is essential to design performant QR systems.

This section explores the research question by using a powerful tool called *temporal information processing capacity* (TIPC) [212]. The TIPC assesses the ability of dynamical systems to reproduce polynomial functions of the input sequence and the internal state history. Thus, the TIPC profile can quantitatively clarify the memory effects and information processing mechanism induced by quantum noise. We construct several quantum noise models and numerically investigate the effect of quantum noise through the lens of the TIPC. Numerical simulations then show that amplitude damping can induce temporal processing capabilities. We also apply the technique to analyze the profile of QR systems on real quantum devices. While quantum noise on quantum hardware is non-trivial and hence could differ from the simulated models, we observe similar characteristics of memory profiles. Notably, we also find the correlation between the capacity and error rates of *CNOT* gates, implying that quantum processors with higher noise levels could better carry temporal information.

The structure of this section is organized as follows. We first detail the TIPC in Sec. 5.2.2, and demonstrate the TIPC profiles for the QR systems driven by simulated quantum noise models in Sec. 5.2.3. We then perform benchmark tasks to verify the profiles obtained for the numerical quantum noise models in Sec. 5.2.4. Subsequently, we examine the QR systems on IBM quantum hardware via the TIPC in Sec. 5.2.5. Lastly, we conclude this section in Sec. 5.2.6.

5.2.2 Temporal Information Processing Capacity (TIPC)

We provide the details of the TIPC, which is the main tool for analyzing the capability of QR systems in this section. The TIPC evaluates the capability of dynamical systems to reconstruct polynomial functions of input history and previous internal states. We here elaborate on its definition.

In general, the d_r -dimensional reservoir state \mathbf{x}_t at timestep t can be represented as a function of input history $\{u_{t-l}\}_{l=1}^t$ and the reservoir states at previous timesteps $\{\mathbf{x}_{t-l}\}_{l=1}^t$. Note that the reservoir state corresponds to the QR output vector $h(\rho_t)$ in our scheme. With the orthonormal basis function dependent on input history and time, $z_{k,t} \equiv z_k(t, u_t, u_{t-1}, \dots)$, the linearly-independent normalized reservoir state $\tilde{\mathbf{x}}_t$ can be expanded as

$$\tilde{\mathbf{x}}_t = \sum_k \gamma_k z_{k,t}. \quad (5.17)$$

Here, the normalized reservoir state $\tilde{\mathbf{x}}_t$ can be obtained by performing singular value decomposition of $X \equiv [\mathbf{x}_0, \dots, \mathbf{x}_{L-1}]^T \in \mathbb{R}^{L \times d_r}$ for total timestep length L ; the left singular vector of $X = P_l \Sigma P_r^T$ corresponds to the normalized state vector, i.e., $P_l = [\tilde{\mathbf{x}}_0, \dots, \tilde{\mathbf{x}}_{L-1}]^T$ ($P_l \in \mathbb{R}^{L \times r}$, $P_r \in \mathbb{R}^{r \times d_r}$ and $\Sigma \in \mathbb{R}^{r \times r}$ with the rank of matrix $1 \leq r \leq \min\{N, d_r\}$). Then, the k -th term of TIPC C_k is defined as the squared norm of the coefficient vector γ_k [212]:

$$C_k = \|\gamma_k\|^2. \quad (5.18)$$

The orthonormal basis function can be represented as

$$z_{k,t} = u_{t-1}^{n_1^{(k)}} u_{t-2}^{n_2^{(k)}} \cdots \hat{x}_{1,t-1}^{m_{1,1}^{(k)}} \hat{x}_{1,t-2}^{m_{1,2}^{(k)}} \cdots \hat{x}_{N,t-1}^{m_{N,1}^{(k)}} \hat{x}_{N,t-2}^{m_{N,2}^{(k)}} \cdots, \quad (5.19)$$

where we define the orders of inputs and the orders of reservoir states as $N_j = \sum_t n_t^{(j)}$ and $M_j = \sum_{k=1}^r \sum_t m_{k,t}^{(j)}$, respectively. From Eq. (5.19), the non-zero value of C_k suggests that the

reservoir state possesses the k -th basis function composed of the corresponding input and internal state. Thus, one can use this value to investigate dynamical systems' information processing ability. Also, the total capacity $C_{tot} = \sum_k C_k$ is equal to r by definition, indicating this quantity can comprehensively describe temporal information processing.

We further define d -th-order TIPC decomposition for time-invariant terms $C_{tot,d}^{TIV}$ and time-variant terms $C_{tot,d}^{TV}$ as

$$C_{tot,d}^{TIV} = \sum_{\{j|N_j=d, M_j=0\}} C_j, \quad (5.20)$$

$$C_{tot,d}^{TV} = \sum_{\{j|N_j=d, M_j>0\}} C_j, \quad (5.21)$$

respectively. The target output can be represented by a function of finite input history, and thus, only $C_{tot,d}^{TIV}$ can be used for temporal information processing tasks. Namely, $C_{tot,d}^{TV}$ measures information processing ability that is not reproducible, and may not be used for temporal tasks. However, we also introduce the metric to understand how the temporal information is processed in the QR systems. We note that the TIPC is a general tool for evaluating information processing capabilities and is thus applicable to dynamical systems.

From the perspective of numerical computation of TIPC, we perform Gram-Schmidt orthogonalization to obtain the orthonormalized bases $\zeta^{(k)} = [\zeta_{k,1}, \dots, \zeta_{k,L}]^T$ with $\|\zeta^{(k)}\| = 1$, which corresponds to $z^{(k)} \equiv [z_{k,1}, \dots, z_{k,L}]$. In this study, we employ the Volterra-Wiener-Korenberg series [228] and as the orthonormal polynomial expansion. Then, the k -th term of TIPC can be computed as follows:

$$C_k = C(X, \zeta^{(k)}) = 1 - \frac{\min_{\mathbf{w}} \sum_{t=1}^T (\zeta^{(k)} - \mathbf{w}^\top \mathbf{x}_t)^2}{\sum_{t=1}^T \zeta^{(k)2}}. \quad (5.22)$$

Also, in case the length of time-series data is finite, the numerical error of TIPC follows χ^2 distribution with r degrees of freedom [229]. Thus, in the following numerical calculation, we set the statistically significant level $p\%$ and then determine the threshold as $C_{th} = \sigma C_T$ with a scaling factor σ and the top p value C_T . We set $p = 10^{-4}$ and $\sigma = 2, 3$ for the simulated QRs in Sec. 5.2.3 and $p = 5 \times 10^{-2}$ and $\sigma = 1$ for the QRs implemented on real quantum machines in Sec. 5.2.5. With this threshold C_{th} , we truncated the capacity C : if the obtained value is smaller than the C_{th} , we ignore the capacity, i.e., $C = 0$.

5.2.3 TIPC Profile for QR Systems Simulated by Quantum Noise Models

We numerically investigate the TIPC profiles of QR systems under some quantum noise. Here, we consider the same QR systems in Sec. 5.1; the dynamics of the QR system is given by Eq. (5.1), where the input-dependent unitary operator is provided in Eq. (5.6) with local unitary blocks of Eq. (5.7). We also chose the observables $\{O_i\} = \{Z_i\}_{i=0}^{n-1}$, and the initial state $\rho_0 = |+\otimes^n\rangle\langle +\otimes^n| = H^{\otimes n}|\mathbf{0}\rangle\langle \mathbf{0}|H^{\otimes n}$ with the Hadamard gate H and $|\mathbf{0}\rangle = |0\rangle^{\otimes n}$. Recall that the QR system cannot carry the information under the noiseless situation because the QR output vector is the zero vector for any timestep t , i.e., $h(\rho_t) = [\text{Tr}[Z_0\rho_t], \dots, \text{Tr}[Z_{n-1}\rho_t]]^T = \mathbf{0}$.

In Sec. 5.1, we utilize a device-dependent CPTP map $\mathcal{E}_{qn}(\cdot)$ to drive the QR systems in an input-dependent manner, that is,

$$\rho_t = \mathcal{E}_{qn}\left(U(u_t)\rho_{t-1}U^\dagger(u_t)\right). \quad (5.23)$$

However, building exact noise models corresponding to quantum hardware is challenging due to the limited access to quantum processors, which prevents us from performing thorough analysis. Therefore, we replace the device-dependent map $\mathcal{E}_{qn}(\cdot)$ with the well-known noise models. More precisely, by introducing non-unitary noise channel $\mathcal{E}_d(\cdot)$ and unitary noise $\mathcal{N}(\cdot)$, we express the dynamics as follows;

$$\rho_t = \mathcal{E}_d \left(\mathcal{N} (U(u_t)) \rho_{t-1} \mathcal{N} (U(u_t))^\dagger \right). \quad (5.24)$$

Below is a list of quantum noise we consider:

Non-unitary noise $\mathcal{E}_d(\cdot)$

- **Bit-flip error:** This noise causes a single-qubit state to flip from $|0\rangle$ to $|1\rangle$ or vice versa. That is equivalent to Pauli X error occurs with probability p . The Kraus operator representation is $\mathcal{E}_d(\rho) = K_1\rho K_1^\dagger + K_2\rho K_2^\dagger$ with

$$K_1 = \sqrt{1-p}I, \quad K_2 = \sqrt{p}X.$$

- **Phase-flip error:** The noise flips the phase of a single-qubit state, which corresponds to Pauli Z error with probability p . The Kraus representation is as follows: $\mathcal{E}_d(\rho) = K_1\rho K_1^\dagger + K_2\rho K_2^\dagger$ with

$$K_1 = \sqrt{1-p}I, \quad K_2 = \sqrt{p}Z.$$

- **Depolarization:** This noise causes all types of Pauli errors with equal probability p . For a single-qubit case, the noise in the Kraus representation is as follows: $\mathcal{E}_d(\rho) = K_1\rho K_1^\dagger + K_2\rho K_2^\dagger + K_3\rho K_3^\dagger + K_4\rho K_4^\dagger$ with

$$K_1 = \sqrt{1-p}I, \quad K_2 = \sqrt{\frac{p}{3}}X, \quad K_3 = \sqrt{\frac{p}{3}}Y, \quad K_4 = \sqrt{\frac{p}{3}}Z.$$

- **Amplitude damping:** The noise corresponds to the energy dissipation to the environment. For a single-qubit case, its Kraus operators are as follows:

$$K_1 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad K_2 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix},$$

with the damping rate γ .

- **Phase damping:** The noise corresponds to the loss of the quantum phase. With the damping rate γ , its Kraus operators for a single qubit are as follows:

$$K_1 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad K_2 = \begin{pmatrix} 0 & 0 \\ 0 & \sqrt{\gamma} \end{pmatrix}.$$

Unitary noise $\mathcal{N}(\cdot)$

- **Single-qubit overrotation:** The noise causes the overrotation of single-qubit gates. Namely, this noise transforms the single rotation gate $R_\sigma(\theta)$ with $\sigma \in \{X, Y, Z\}$ to

$$\mathcal{N} (R_\sigma(\theta)) = R_\sigma(\theta(1 + \epsilon))$$

where $\epsilon \sim \text{Uniform}(0, c)$ with a constant c .

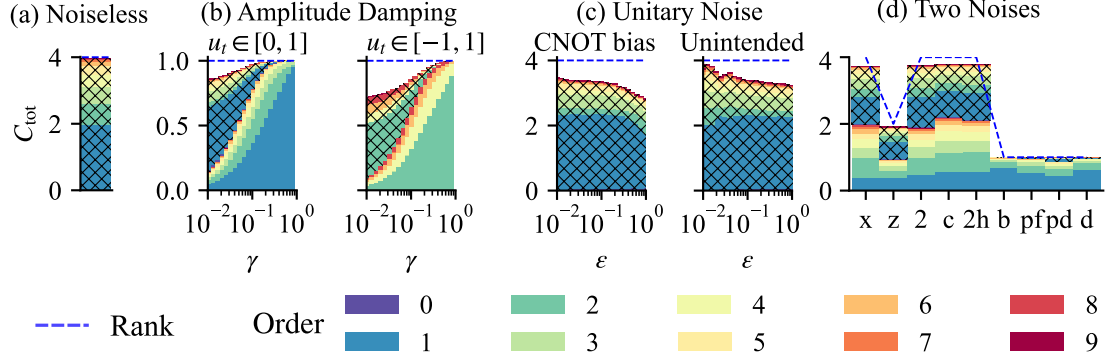


Figure 5.10: TIPC Profiles of 4-qubit QR models driven by unitary operators in Eq. (5.6) with (a) noiseless quantum circuits, (b) amplitude damping given the varying damping rate γ [left: $u_t \in [0, 1]$, right: $u_t \in [-1, 1]$], (c) unitary noises determined by the varying perturbation rate ϵ [left: $CNOT$ bias for overrotation, right: unintentional entangling between nearby qubits], and (d) combination of amplitude damping and another type of noise [Pauli X (x), Pauli Z (z), $CNOT$ bias (c), one-hop (u1), two-hop (u2), bit-flip (b), phase-flip (pf), phase damping (pd), or depolarization (d) noise with $\epsilon = \gamma = 0.1$]. Dotted blue lines denote the ranks of QR output states. The hatched areas and non-hatched parts represent the time-variant and time-invariant components in TIPC, respectively. In panels (a), (c), and (d), the input u_t is uniformly distributed in $[0, 1]$. Reprinted figure from Fig.3 of Ref. [42] by T. Kubota, Y. Suzuki, S. Kobayashi, Q.H. Tran, N. Yamamoto, and K. Nakajima. Creative Commons Attribution 4.0 International license [DOI:<https://doi.org/10.1103/PhysRevResearch.5.023057>].

- **$CNOT$ bias:** The noise causes overrotation for the conditional X gate. That is,

$$\mathcal{N}(CNOT) = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Rx(\pi(1 + \epsilon))$$

- **Unintentional entangler (one-hop, two-hop):** The noise unintentionally apply conditional X gate with the scaling factor ϵ . That is, the following entangler gate is applied for the noise:

$$|0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes Rx(\pi\epsilon)$$

This error occurs for the nearest and the second-nearest neighboring qubits, which we call *one-hop* and *two-hop*, respectively.

Then, we build four-qubit QR systems ($m = 2$ subsystems) driven by these noise models and input-dependent unitary operators in Eq. (5.6) with Eq. (5.7).

Fig. 5.10 illustrates the TIPC decomposition of simulated 4-qubit QR systems. Here, we demonstrate the following cases: (a) the noiseless QR system, (b) the amplitude damping noise, (c) the unitary noise ($CNOT$ bias and unintended entangler), and (d) the combination of amplitude damping and another type of noise. In the noiseless situation, the QR system does not

possess time-invariant TIPC. This is consistent with our statement on how to build QR systems; the setup is considered so that the QR systems' output is trivial. Also, Fig. 5.10 (c) shows that the TIPC of QR systems under unitary noise is time-variant, which reflects the fact that the ESP does not hold for unitary noise, as stated in Sec. 5.1.2. We note that $C_{\text{tot}} = r = 0$ for single-qubit unitary, phase-flip, bit-flip, phase damping, or depolarization noise. In contrast, the amplitude damping noise can induce the time-invariant TIPC, as in Fig. 5.10 (b). Interestingly, the higher the damping rate is, the more dominant the time-invariant TIPC is. Moreover, other types of quantum noise in combination with amplitude damping can also possess the time-invariant TIPC. This indicates that amplitude damping is critical in inducing temporal information processing capabilities and can show better performance as the error rate γ increases. We note that some total capacities do not saturate the rank r because the discarded components C_k do not exceed the threshold.

Moreover, we compare the TIPC profile of simulated QRs and classical ESNs. Here, we construct 4-qubit QR systems with the input scaling $a = \pi$ in Eq. (5.7) and employ the spatial multiplexing technique [158] that combines various reservoir states to learn target sequences. More specifically, we build $2^{10} = 1,024$ QR systems using all possible combinations of 10 types of quantum noise mentioned above; then, we consider two models, the spatial multiplexing of 130 QRs and 25 QRs. As for the ESN, we consider the following as the reservoir state $x_{i,t+1}$:

$$x_{i,t+1} = \tanh \left(\sum_{j=1}^N \rho w_{ij} x_{j,t} + \nu w_{\text{in},i} u_{t+1} \right), \quad (5.25)$$

where $w_{\text{in},i}$ represents the input weight and w_{ij} is the internal weight. Also, $\nu (= 0.1)$ and $\rho (= 0.6)$ control the spectral radius of $w_{\text{in},i}$ and w_{ij} , respectively. Here, these weights are generated from the uniform distribution in the range of $[0, 1]$. As for w_{ij} , the spectral radius is set to one dividing the weight by its largest absolute value. We note that the connection probabilities of the internal and input weights are set as 0.5 and 0.1, respectively. The internal node is set as $N = 50$.

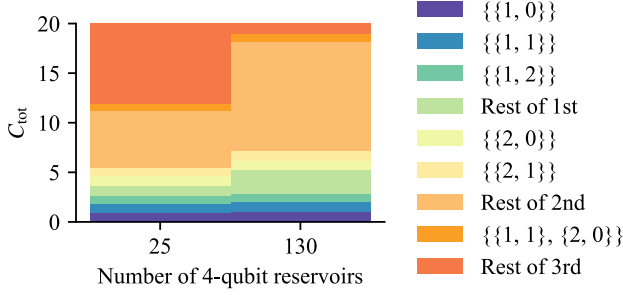
Fig. 5.11 illustrates the time-invariant TIPC for the ESN and the QR systems employing the spatial multiplexing of 130 and 25. We here consider the time-invariant TIPC, where the input is uniformly at random in $[0, 1]$ and Legendre polynomials are used as orthogonal bases. In addition, the shuffle surrogate technique introduced in Ref. [212] is used to reduce numerical errors. In Fig. 5.11, we use the notation $\{\{n_s, s\}\}$ to indicate that the Legendre polynomial that corresponds to the time-invariant TIPC term is $\prod_s P_{n_s}(u_{n-s})$ where n_s is the degree of polynomial and s is the delayed timestep of the input. We can clearly find that the TIPC profile differs for these dynamical systems. Importantly, the time-invariant TIPC component for $\{P_1(u_{t-1})P_2(u_t)\}$ (labeled $\{\{1, 1\}, \{2, 0\}\}$) does not appear in the ESN, but in the QR systems; this would indicate there might exist a temporal task that is not learnable by the ESN, but by the QR systems.

5.2.4 Benchmark Tasks

Motivated by the TIPC analysis of the ESN and the QR systems, we further investigate the relationship between the performance for benchmark tasks and the TIPC profiles. To this end, we perform two benchmark tasks: the second order NARMA task [222] (NARMA2) and a task to emulate pneumatic artificial muscle (PAM) length (we call it PAM task). As for the NARMA task, we consider the following dynamics;

$$y_{t+1} = 0.4y_t + 0.4y_t y_{t-1} + 0.6(0.3u_t)^3 + 0.1, \quad (5.26)$$

(a) Time-invariant TIPC for simulated QRs



(b) Time-invariant TIPC for ESN with 50 nodes

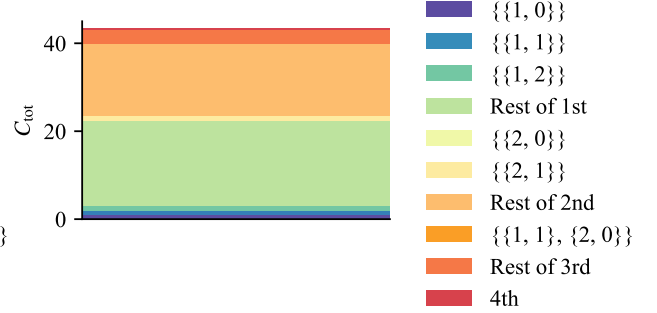


Figure 5.11: Time-invariant TIPCs for (a) simulated QR systems employing the spatial multiplexing 25 and 130, and (b) ESN with 50 nodes. The labels $\{\{n_s, s\}\}$ denote the combination of components for polynomial $\prod_s P_{n_s}(u_{n-s})$, where n_s is the degree of polynomial and s is the delayed timestep of the input. Reprinted figure from Fig.8 (c) and (d) of Ref. [42] by T. Kubota, Y. Suzuki, S. Kobayashi, Q.H. Tran, N. Yamamoto, and K. Nakajima. Creative Commons Attribution 4.0 International license [DOI:10.1103/PhysRevResearch.5.023057].

where the input u_t is uniformly at random in the range $[0, 1]$. For the PAM task, we use the dataset provided in Ref. [230]. Note that PAM is a soft actuator controlled by air pressure, and there is difficulty in measuring its length by infrared sensor; then Ref. [231] demonstrated that ESN can predict the length of PAM as accurately as infrared sensors.

We prepare the input sequence of length 49,998, where the first 9,998 timesteps are discarded for washout, and we use the following 2×10^4 steps and the remaining for training and performance evaluation, respectively. The performance is evaluated using the normalized root mean square error defined as

$$\text{NRMSE} = \frac{1}{\sigma(y)} \sqrt{\frac{1}{N_{\text{eval}}} \sum_{t=1}^{N_{\text{eval}}} (y_t - \bar{y}_t)^2}, \quad (5.27)$$

with the variance of the target sequence $\sigma^2(y)$. Here, \bar{y}_t is the prediction at time step t in N_{eval} time steps. Our numerical simulations show that the spatial multiplexing of 130 (25) QRs can accurately emulate the NARMA2 dynamic with high precision (NRMSE = 0.11 (0.21)). Its NRMSE for the PAM task is also small, NRMSE = 0.21 (0.30). The performances of the spatial multiplexing of 130 (25) QRs are the same as those of the ESN. As for the PAM task, the system combining 130 QRs can slightly perform better than the ESN with less than 520 nodes (i.e., NRMSE > 0.22).

We can analyze the performance via the TIPC. Fig. 5.12 shows the time-invariant TIPC components of the NARMA2 and PAM tasks, where Legendre polynomials are used as the orthonormal polynomial expansion. The profile clarifies the polynomials of input history required to solve these tasks: the major components are $P_1(u_t)$, $P_1(u_{t-1})$, and $P_1(u_{t-2})$ for NARMA2, and the first-order TIPCs for PAM tasks. The TIPC profiles in Fig. 5.12 show that both the ESN and QR systems possess those components, implying that the systems perform well for these tasks. We note that the PAM task includes a component $\{\{1, 1\}, \{2, 0\}\}$, which does not exist in the ESN, but in the QR systems. This might contribute to the slightly better performance for the QR systems.

(a) Time-invariant TIPC for NARMA2 task

(b) Time-invariant TIPC for PAM task

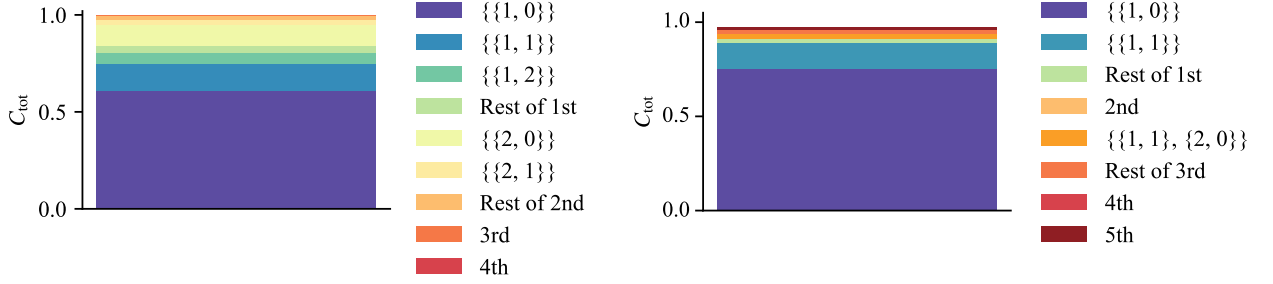


Figure 5.12: Time-invariant TIPCs for (a) NARMA2 task, and (b) PAM task. The labels $\{\{n_s, s\}\}$ represent the corresponding components of polynomial $\prod_s P_{n_s}(u_{n-s})$ with where the degree of polynomial n_s and the delayed time step of the input s . Reprinted figure from Fig.8 (a) and (b) of Ref. [42] by T. Kubota, Y. Suzuki, S. Kobayashi, Q.H. Tran, N. Yamamoto, and K. Nakajima. Creative Commons Attribution 4.0 International license [DOI:10.1103/PhysRevResearch.5.023057].

Table 5.4: Device error parameters during the experiments and the calculated total time-invariant capacities $C_{\text{tot}}^{\text{TIV}}$. The medians of error rates are shown, where only the qubits constituting the QR system are considered.

| Input type | Characteristics | Kawasaki | Toronto | Montreal | Manhattan_p1 | Manhattan_p2 |
|------------|-------------------------------|----------|---------|----------|--------------|--------------|
| Symmetric | <i>CNOT</i> error | 0.0070 | 0.0083 | 0.0095 | 0.0259 | 0.0161 |
| | Readout error | 0.0095 | 0.0300 | 0.0140 | 0.1499 | 0.0183 |
| | $C_{\text{tot}}^{\text{TIV}}$ | 0.1112 | 0.1306 | 0.1019 | 0.6248 | 0.4883 |
| Asymmetric | <i>CNOT</i> error | 0.0070 | 0.0083 | 0.0097 | 0.0252 | 0.0163 |
| | Readout error | 0.0095 | 0.0300 | 0.0138 | 0.1499 | 0.0183 |
| | $C_{\text{tot}}^{\text{TIV}}$ | 0.0703 | 0.1480 | 0.2130 | 0.6433 | 0.5352 |

5.2.5 TIPC Profile for QR Systems on Quantum Devices

We lastly perform the TIPC analysis on quantum hardware-specific QR systems demonstrated in Sec. 5.1. Recall that the QR system is driven by input-dependent unitary operators and quantum noise on hardware, as shown in Eqs. (5.1), (5.6) and (5.7) with the scaling factor $a = 2$. We consider 12-qubit QRs implemented on two types of IBM superconducting quantum processors, “Falcon” devices with 27 qubits and “Hummingbird” with 65 qubits. Specifically, we focus on two configurations of 12-qubit QRs in an *ibmq_manhattan* device (denoted as the Manhattan_p1 and Manhattan_p2) for the Hummingbird type and the *ibmq_kawasaki*, *ibmq_montreal* and *ibmq_toronto* devices (denoted as the Kawasaki, Montreal and Toronto, respectively) for the Falcon type. Note that *ibmq_melbourne* (“Canary” type) we used in the previous section was retired in July 2021, and thus we could not perform the analysis on that device. Fig. 5.13 (a) and (b) illustrate the implementation scheme and arrangement of qubits for these devices, respectively. As for the inputs, we prepare two types of sequences of total length $T = 200$: uniform random input in the symmetric range $u_t \in [-1, 1]$ and the other in the asymmetric one $u_t \in [0, 1]$. Note that we use Qiskit [172] to implement the QR systems. The characteristics of quantum devices used during the experiments are shown in Table 5.4.

Fig. 5.14 (a) illustrates time-invariant capacities for these devices, where we find that the

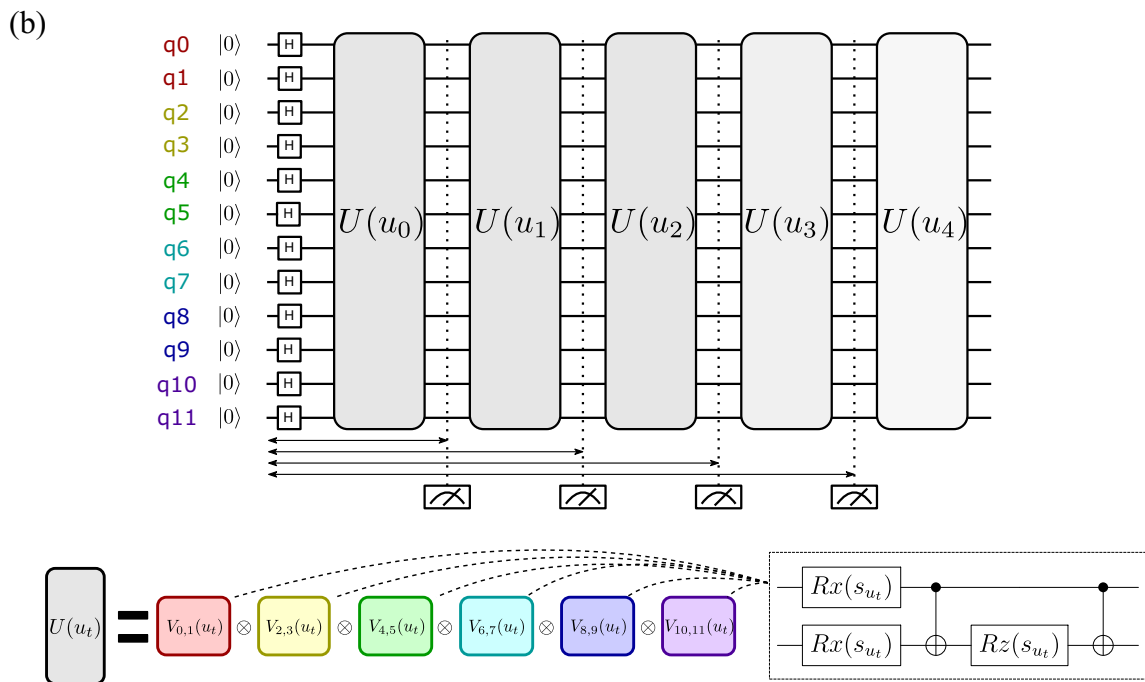
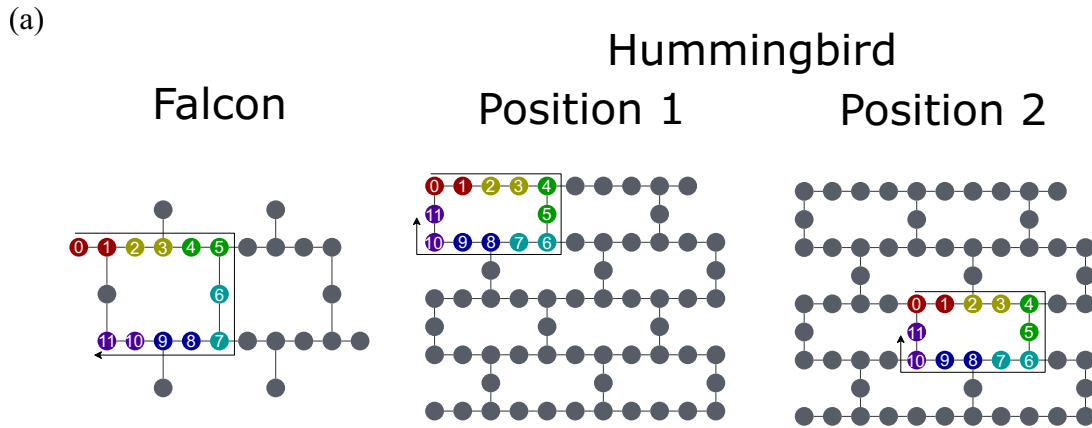


Figure 5.13: Configurations of quantum devices and the implementation scheme on quantum devices. (a) The qubit-configurations of Falcon- and Hummingbird-type quantum hardware, where nodes and edge denote qubits and physical connectivities, respectively. Colored qubits denote the positions of QR systems used for the implementation. (b) Quantum circuit representation of our scheme is depicted.

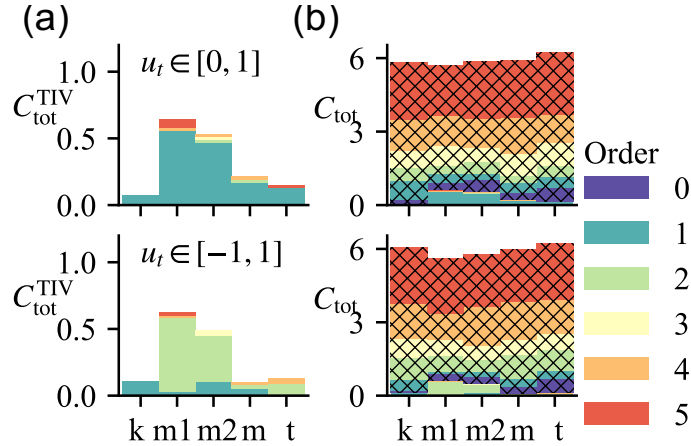


Figure 5.14: Averaged TIPC of QR systems driven by quantum noise in actual quantum devices [Kawasaki (k), Manhattan_p1 (m1), Manhattan_p2 (m2) Montreal (m), and Toronto (t)]. Panels (a) and (b) show total time-invariant capacities $C_{\text{tot}}^{\text{TIV}}$ and total capacities C_{tot} , respectively. Reprinted figure from Fig.5 (b) and (c) of Ref. [42] by T. Kubota, Y. Suzuki, S. Kobayashi, Q.H. Tran, N. Yamamoto, and K. Nakajima. Creative Commons Attribution 4.0 International license [DOI:10.1103/PhysRevResearch.5.023057].

Hummingbird type has larger capacities than the Falcon devices. We recall that the TIPC analysis on the simulated QR systems suggests that amplitude damping is essential to induce the time-invariant terms. Thus, these results imply that the amplitude damping noise occurs for all quantum systems. Note that we set a strict threshold $C_{\text{th}} = 0.14$ to obtain the results in Fig. 5.14 (b) because of the short time length. However, we can observe time-variant as well as time-invariant terms for $C_{\text{th}} = 0.1$, indicating the dominant component, i.e., $u_t x_{t-1}$, also agrees with the case for simulated QR systems. In addition, several quantum devices possess the first-order capacity in the symmetric input case $[-1, 1]$, while the component does not appear in the simulated QRs under the depolarizing noise. This suggests that quantum hardware possesses non-trivial noise other than the amplitude damping noise. Moreover, we investigate the short-term memory effect of the QR system. Fig. 5.15 (a) depicts the first-order capacity as a function of delayed step s_1 . It turns out that the QR systems mainly reflect the recent input sequence, i.e., $s_1 = 0, 1, 2$. Fig. 5.15 (b) also shows the time-invariant second-order capacities, which indicates the majority term is also the recent history of inputs u_t^2 .

Furthermore, we examine the relationship between *CNOT* error rates for each quantum device and the total-invariant capacity. We consider the *CNOT* error because this is one of the representative errors to see the performance of quantum hardware. Fig. 5.16 surprisingly show that the Manhattan device with the worst *CNOT* errors witnesses the highest capacities for temporal data processing. Also, there is a positive correlation between them as shown in Fig. 5.16, implying unavoidable noise in NISQ devices can be a useful resource to enhance the computational power from the reservoir computing perspective. In the analysis of simulated QRs, we find higher error rates (the damping rates) lead to better information processing capabilities. Hence we conjecture that the time-invariant TIPC on quantum hardware may be induced by unwanted dissipation.

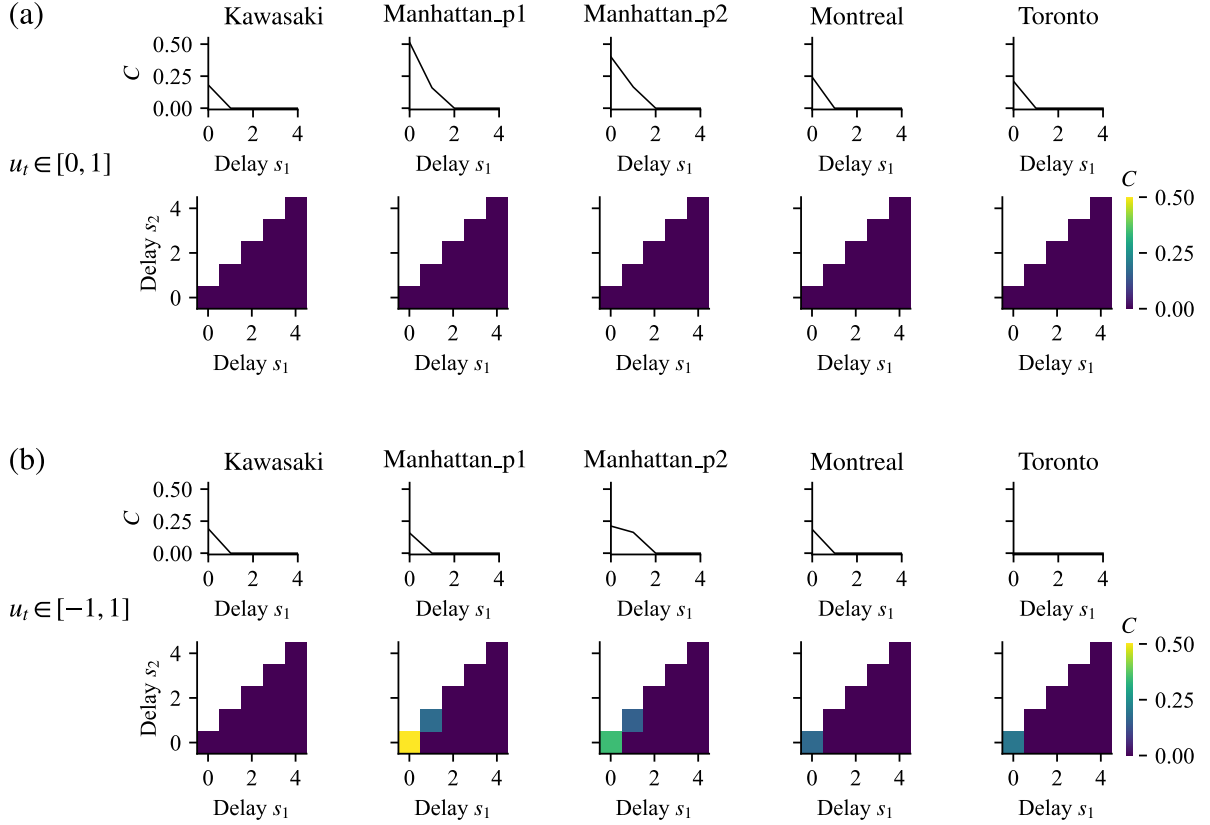


Figure 5.15: First-order and second-order capacities of QR systems on IBM quantum devices for the input range (a) $[0, 1]$ and (b) $[-1, 1]$. Note that second-order capacities do not appear for the asymmetric input case. Reprinted figure from Fig.9 of Ref. [42] by T. Kubota, Y. Suzuki, S. Kobayashi, Q.H. Tran, N. Yamamoto, and K. Nakajima. Creative Commons Attribution 4.0 International license [DOI:10.1103/PhysRevResearch.5.023057].

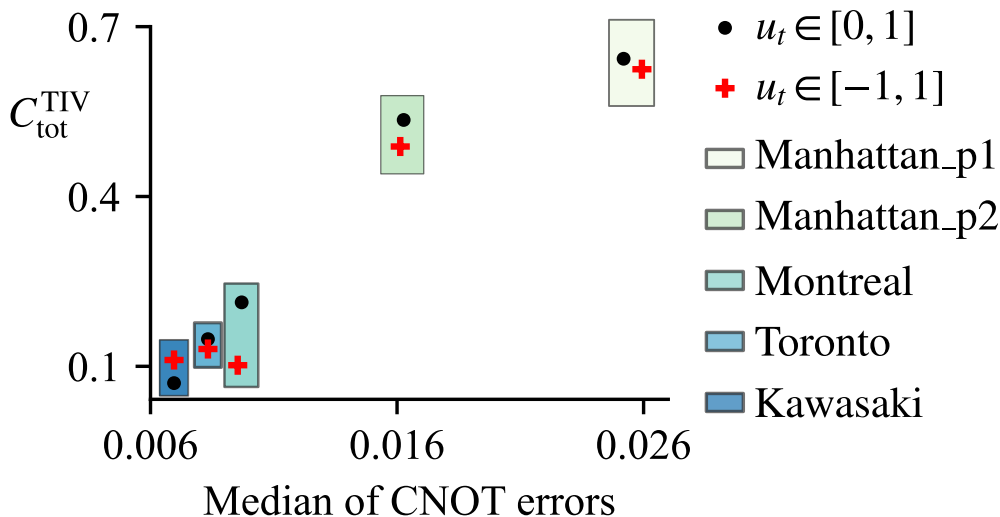


Figure 5.16: Total time-invariant capacity averaged over 10 trials against the median of $CNOT$ error rates of corresponding IBM quantum processors. Black dots and red plus symbols represent the cases for $u_t \in [0, 1]$ and $u_t \in [-1, 1]$, respectively. Reprinted figure from Fig.6 of Ref. [42] by T. Kubota, Y. Suzuki, S. Kobayashi, Q.H. Tran, N. Yamamoto, and K. Nakajima. Creative Commons Attribution 4.0 International license [DOI:10.1103/PhysRevResearch.5.023057].

5.2.6 Conclusion & Outlook

This section analyzes the temporal information processing capabilities of quantum noise-induced reservoir systems via TIPC. Numerical simulations of QR systems driven by quantum noise models elucidate that amplitude damping can induce temporal processing capacities. Moreover, the TIPC analysis of quantum hardware-specific QR systems also demonstrates the similar characteristics obtained for simulated QRs. Surprisingly, we observe a positive correlation between a representative error ($CNOT$ error) rate and the total capacity for each device, indicating the potential of unwanted quantum noise to enhance temporal information processing.

These results suggest other applications of the TIPC. First, the TIPC can be used to construct an optimal QR model. It is critical to select suitable hyperparameters of the models to achieve high performance; examples are gate sets for unitary operators, input-scaling factors, and the size of QR systems. The TIPC enables us to characterize the profile of the input-target mapping and thus can serve as a good quantity for designing performant QR systems. In addition, the TIPC could help to analyze types of quantum noise on quantum hardware. The underlying mechanism of dynamical systems can be clarified from the information processing perspective. Hence, it would be interesting to explore the TIPC approach to detect and mitigate unwanted errors in quantum computing.

Chapter 6

Conclusion and Outlook

6.1 Conclusion

This thesis analyzes two quantum-enhanced machine learning models, quantum kernel methods and quantum reservoir computing, and then provides possible guidelines to design suitable models for practical applications. The quantum-enhanced feature space can promisingly improve data quality and hence harness the performance for machine learning tasks. However, further investigations are needed to fully exploit the quantum space for real-world applications.

As for quantum kernel methods, we focus on two practical challenges: (1) choosing appropriate quantum feature maps for specific classification tasks is non-trivial, and (2) the exponentially decaying expectation value and variance cause infeasible implementation and trainability problem, as the number of qubits increases (i.e., vanishing similarity issue). Chapter 4 addressed these issues. In Sec. 4.1, focusing on the former challenge, we introduced a quantity called minimum accuracy to roughly estimate the training accuracy of classifiers based on quantum feature maps. Then, our numerical simulations demonstrate that the quantity could facilitate screening a suitable quantum feature map among many candidates. We also numerically studied the effectiveness of the synthesis approach to design a powerful quantum kernel by combining many (weak) quantum kernels. For the second challenge, Sec. 4.2 analytically and numerically demonstrated that our proposed quantum Fisher kernel can mitigate the vanishing similarity issue when shallow alternating layered ansatzes are used, whereas the commonly-used fidelity-based quantum kernels cannot, regardless of types of quantum circuits. We further demonstrate a classification task where quantum Fisher kernels can outperform the fidelity-based quantum kernel due to the absence of the vanishing similarity issue. These results will pave the way towards practical applications of quantum kernel methods.

In Chapter 5, we addressed an open question on practical applications of quantum reservoir computing: what kind of quantum reservoir systems can be performant and efficiently implementable? We here provided a new quantum reservoir computing framework that positively utilizes quantum noise to enhance the performance of temporal information processing tasks. Quantum noise is ubiquitous in NISQ devices and is considered harmful because it hinders the power of quantum computation. In stark contrast to such common thought, we use quantum noise to enrich the complexity of quantum dynamics and accordingly harness the time-series data processing abilities. Sec. 5.1 experimentally demonstrated that our quantum reservoir systems driven by quantum hardware-specific quantum noise can perform better than linear classical learning models. With a tool called temporal information processing capacity, numerical simulations in Sec. 5.2 also unveiled that dissipation noise such as amplitude damping can induce

temporal information processing capabilities. These results will provide some insights into the design principles of quantum reservoir systems that are amenable to implementation and can perform well.

6.2 Outlook

There are open problems and future works regarding our results demonstrated in Chapter 4 and 5. We summarize them for each section below.

Analysis and synthesis methods for quantum feature maps in Sec. 4.1: A main concern of the analysis method based on the minimum accuracy is the scalability with respect to the number of qubits. Considering a subset of Pauli operators can reduce computational costs and can still serve as a lower bound of training accuracy for linear classifiers. Thus, it would be interesting to investigate the efficacy of the methods for the cases of large qubit systems. In addition, there is still room for investigation in the synthesis method to build a powerful quantum kernel in a more systematic way.

Quantum Fisher kernels in Sec. 4.2: Our analysis is based on the 2-design assumptions, which might not be satisfied in realistic experimental settings. Thus, it would be interesting to analytically investigate the phenomenon using techniques that can soften or do not rely on t -design property. A technique used in a recent work [232] might be helpful. Also, thorough studies are needed to investigate practical advantages of the quantum Fisher kernel in machine learning tasks dealing with quantum or classical data.

Proof-of-Principle demonstration of quantum noise-induced reservoir computing in Sec. 5.1: An advantage of physical reservoir computing is the fast processing. However, our scheme must execute quantum circuits $N_s L$ times for the total length of time-series L and the number of measurement shots to obtain expectation values at each timestep N_s , which is time-consuming. Thus, reducing the amount of circuit executions would be imperative. Note that we can now perform mid-circuit measurements on current IBM quantum processors and thus the number of quantum circuit executions can be reduced to N_s . In addition, it would be interesting to develop quantum hardware that is tailored for the use of quantum noise-induced reservoir computing; for example, types of quantum noise can be tunable.

TIPC analysis of quantum noise-induced reservoir computing in Sec. 5.2: An open question is whether we can utilize the TIPC profile to build optimal quantum reservoir systems. The TIPC method can characterize input-target maps of time-series data and thus help us to choose hyperparameters such as a gate set for input-dependent unitary operators and types of quantum noise. Furthermore, investigating errors occurring in quantum information processors via the TIPC profile would also be exciting.

Bibliography

- [1] Kevin P. Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.
- [2] Christopher M Bishop and Nasser M Nasrabadi. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006.
- [3] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.
- [4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [5] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. *Advances in Neural Information Processing Systems*, 30, 2017.
- [7] 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.
- [8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single Shot Multibox Detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.
- [9] Pierre Baldi, Søren Brunak, and Francis Bach. *Bioinformatics: The Machine Learning Approach*. MIT Press, 2001.
- [10] Sendhil Mullainathan and Jann Spiess. Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives*, 31(2):87–106, 2017.
- [11] Matthew F Dixon, Igor Halperin, and Paul Bilokon. *Machine Learning in Finance: From Theory to Practice*. Springer, 2020.
- [12] Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhang, Yuanyuan Yang, Jiaming Tian, Hao He, Antong Li, Mengshen He, Zhengliang Liu, et al. Summary of ChatGPT-Related research and perspective towards the future of large language models. *Meta-Radiology*, 1(2):100017, 2023.

- [13] Richard P Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, 1981.
- [14] David Deutsch. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, 400(1818):97–117, 1985.
- [15] 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.
- [16] Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
- [17] Alexi Y Kitaev. Quantum measurements and the Abelian Stabilizer Problem. *arXiv preprint quant-ph/9511026*, 1995.
- [18] Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum Algorithm for Linear Systems of Equations. *Physical Review Letters*, 103(15):150502, 2009.
- [19] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.
- [20] Jarrod R McClean, Ryan Babbush, Peter J Love, and Alán Aspuru-Guzik. Exploiting Locality in Quantum Computation for Quantum Chemistry. *The Journal of Physical Chemistry Letters*, 5(24):4368–4380, 2014.
- [21] Ryan Babbush, Dominic W Berry, Ian D Kivlichan, Annie Y Wei, Peter J Love, and Alán Aspuru-Guzik. Exponentially more precise quantum simulation of fermions in second quantization. *New Journal of Physics*, 18(3):033032, 2016.
- [22] Ryan Babbush, Dominic W Berry, Jarrod R McClean, and Hartmut Neven. Quantum simulation of chemistry with sublinear scaling in basis size. *npj Quantum Information*, 5(1):92, 2019.
- [23] Patrick Rebentrost, Brajesh Gupt, and Thomas R Bromley. Quantum computational finance: Monte Carlo pricing of financial derivatives. *Physical Review A*, 98(2):022321, 2018.
- [24] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, 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] Nathan Wiebe, Daniel Braun, and Seth Lloyd. Quantum Algorithm for Data Fitting. *Physical Review Letters*, 109(5):050505, 2012.
- [27] Zhikuan Zhao, Jack K Fitzsimons, and Joseph F Fitzsimons. Quantum-assisted Gaussian process regression. *Physical Review A*, 99(5):052331, 2019.
- [28] Sanchayan Dutta, Adrien Suau, Sagnik Dutta, Suvadeep Roy, Bikash K Behera, and Prasanta K Panigrahi. Quantum circuit design methodology for multiple linear regression. *IET Quantum Communication*, 1(2):55–61, 2020.

- [29] Iordanis Kerenidis and Alessandro Luongo. Classification of the MNIST data set with quantum slow feature analysis. *Physical Review A*, 101(6):062327, 2020.
- [30] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum Support Vector Machine for Big Data Classification. *Physical Review Letters*, 113(13):130503, 2014.
- [31] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17:1013–1017, 2021.
- [32] Jonas Jäger and Roman V Krems. Universal expressiveness of variational quantum classifiers and quantum kernels for support vector machines. *Nature Communications*, 14(1):576, 2023.
- [33] Till Muser, Elias Zapusek, Vasilis Belis, and Florentin Reiter. Provable advantages of kernel-based quantum learners and quantum preprocessing based on Grover’s algorithm. *arXiv preprint arXiv:2309.14406*, 2023.
- [34] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.
- [35] Maria Schuld and Francesco Petruccione. Quantum ensembles of quantum classifiers. *Scientific Reports*, 8(1):2772, 2018.
- [36] Keisuke Fujii and Kohei Nakajima. Harnessing disordered-ensemble quantum dynamics for machine learning. *Physical Review Applied*, 8(2):024030, 2017.
- [37] 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, 2021.
- [38] Yudai Suzuki, Hideaki Kawaguchi, and Naoki Yamamoto. Quantum Fisher kernel for mitigating the vanishing similarity issue. *arXiv preprint arXiv:2210.16581*, 2022.
- [39] Supanut Thanasilp, Samson Wang, M Cerezo, and Zoë Holmes. Exponential concentration and untrainability in quantum kernel methods. *arXiv preprint arXiv:2208.11060*, 2022.
- [40] Yudai Suzuki, Hiroshi Yano, Qi Gao, Shumpei Uno, Tomoki Tanaka, Manato Akiyama, and Naoki Yamamoto. Analysis and synthesis of feature map for kernel-based quantum classifier. *Quantum Machine Intelligence*, 2:1–9, 2020.
- [41] Yudai Suzuki, Qi Gao, Ken C Pradel, Kenji Yasuoka, and Naoki Yamamoto. Natural quantum reservoir computing for temporal information processing. *Scientific Reports*, 12(1):1353, 2022.
- [42] Tomoyuki Kubota, Yudai Suzuki, Shumpei Kobayashi, Quoc Hoan Tran, Naoki Yamamoto, and Kohei Nakajima. Temporal information processing induced by quantum noise. *Physical Review Research*, 5(2):023057, 2023.
- [43] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.
- [44] Mohan Sarovar, Timothy Proctor, Kenneth Rudinger, Kevin Young, Erik Nielsen, and Robin Blume-Kohout. Detecting crosstalk errors in quantum information processors. *Quantum*, 4:321, 2020.

- [45] 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), 2019.
- [46] Göran Wendin. Quantum information processing with superconducting circuits: a review. *Reports on Progress in Physics*, 80(10):106001, 2017.
- [47] Xiu Gu, Anton Frisk Kockum, Adam Miranowicz, Yu-xi Liu, and Franco Nori. Microwave photonics with superconducting quantum circuits. *Physics Reports*, 718:1–102, 2017.
- [48] Jian-Qiang You and Franco Nori. Atomic physics and quantum optics using superconducting circuits. *Nature*, 474(7353):589–597, 2011.
- [49] Xi-Lin Wang, Yi-Han Luo, He-Liang Huang, Ming-Cheng Chen, Zu-En Su, Chang Liu, Chao Chen, Wei Li, Yu-Qiang Fang, Xiao Jiang, et al. 18-qubit entanglement with six photons’ three degrees of freedom. *Physical Review Letters*, 120(26):260502, 2018.
- [50] Hui Wang, Jian Qin, Xing Ding, Ming-Cheng Chen, Si Chen, Xiang You, Yu-Ming He, Xiao Jiang, L You, Z Wang, et al. Boson sampling with 20 input photons and a 60-mode interferometer in a 10^{14} -dimensional Hilbert space. *Physical Review Letters*, 123(25):250503, 2019.
- [51] Rainer Blatt and Christian F Roos. Quantum simulations with trapped ions. *Nature Physics*, 8(4):277–284, 2012.
- [52] Dietrich Leibfried, Rainer Blatt, Christopher Monroe, and David Wineland. Quantum dynamics of single trapped ions. *Reviews of Modern Physics*, 75(1):281, 2003.
- [53] Yu He, SK Gorman, Daniel Keith, Ludwik Kranz, JG Keizer, and MY Simmons. A two-qubit gate between phosphorus donor electrons in silicon. *Nature*, 571(7765):371–375, 2019.
- [54] Jonathan A Jones. NMR quantum computation. *Progress in Nuclear Magnetic Resonance Spectroscopy*, 38(4):325–360, 2001.
- [55] Sergey Bravyi, David Gosset, Robert Koenig, and Marco Tomamichel. Quantum advantage with noisy shallow circuits. *Nature Physics*, 16(10):1040–1045, 2020.
- [56] Youngseok Kim, Andrew Eddins, Sajant Anand, Ken Xuan Wei, Ewout Van Den Berg, Sami Rosenblatt, Hasan Nayfeh, Yantao Wu, Michael Zaletel, Kristan Temme, et al. Evidence for the utility of quantum computing before fault tolerance. *Nature*, 618(7965):500–505, 2023.
- [57] Kristan Temme, Sergey Bravyi, and Jay M Gambetta. Error mitigation for short-depth quantum circuits. *Physical Review Letters*, 119(18):180509, 2017.
- [58] Suguru Endo, Simon C Benjamin, and Ying Li. Practical Quantum Error Mitigation for Near-Future Applications. *Physical Review X*, 8(3):031027, 2018.
- [59] 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, 2021.
- [60] Zhenyu Cai, Ryan Babbush, Simon C Benjamin, Suguru Endo, William J Huggins, Ying Li, Jarrod R McClean, and Thomas E O’Brien. Quantum Error Mitigation. *arXiv preprint arXiv:2210.00921*, 2022.

- [61] Ewout Van Den Berg, Zlatko K Mineev, Abhinav Kandala, and Kristan Temme. Probabilistic error cancellation with sparse Pauli–Lindblad models on noisy quantum processors. *Nature Physics*, 19:1116–1121, 2023.
- [62] Youngseok Kim, Christopher J Wood, Theodore J Yoder, Seth T Merkel, Jay M Gambetta, Kristan Temme, and Abhinav Kandala. Scalable error mitigation for noisy quantum circuits produces competitive expectation values. *Nature Physics*, 19:752–759, 2023.
- [63] Maria Schuld and Francesco Petruccione. *Supervised Learning with Quantum Computers*, volume 17. Springer, 2018.
- [64] Ewin Tang. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 217–228, 2019.
- [65] András Gilyén, Zhao Song, and Ewin Tang. An improved quantum-inspired algorithm for linear regression. *Quantum*, 6:754, 2022.
- [66] Ewin Tang. Dequantizing algorithms to understand quantum advantage in machine learning. *Nature Reviews Physics*, 4(11):692–693, 2022.
- [67] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, 2018.
- [68] Dominik Koutný, Libor Motka, Zdeněk Hradil, Jaroslav Řeháček, and Luis L Sánchez-Soto. Neural-network quantum state tomography. *Physical Review A*, 106(1):012409, 2022.
- [69] Juan Carrasquilla and Roger G Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, 2017.
- [70] Peter Broecker, Juan Carrasquilla, Roger G Melko, and Simon Trebst. Machine learning quantum phases of matter beyond the fermion sign problem. *Scientific Reports*, 7(1):8823, 2017.
- [71] Sirui Lu, Shilin Huang, Keren Li, Jun Li, Jianxin Chen, Dawei Lu, Zhengfeng Ji, Yi Shen, Duanlu Zhou, and Bei Zeng. Separability-entanglement classifier via machine learning. *Physical Review A*, 98(1):012315, 2018.
- [72] Naema Asif, Uman Khalid, Awais Khan, Trung Q Duong, and Hyundong Shin. Entanglement detection with artificial neural networks. *Scientific Reports*, 13(1):1562, 2023.
- [73] Hsin-Yuan Huang, Richard Kueng, Giacomo Torlai, Victor V Albert, and John Preskill. Provably efficient machine learning for quantum many-body problems. *Science*, 377(6613):eabk3333, 2022.
- [74] Laura Lewis, Hsin-Yuan Huang, Viet T Tran, Sebastian Lehner, Richard Kueng, and John Preskill. Improved machine learning algorithm for predicting ground state properties. *arXiv preprint arXiv:2301.13169*, 2023.
- [75] Iris Cong, Soonwon Choi, and Mikhail D Lukin. Quantum convolutional neural networks. *Nature Physics*, 15(12):1273–1278, 2019.

- [76] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.
- [77] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum Amplitude Amplification and Estimation. *Contemporary Mathematics*, 305:53–74, 2002.
- [78] John M Martyn, Zane M Rossi, Andrew K Tan, and Isaac L Chuang. Grand Unification of Quantum Algorithms. *PRX Quantum*, 2(4):040203, 2021.
- [79] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, 2019.
- [80] Iordanis Kerenidis, Jonas Landman, Alessandro Luongo, and Anupam Prakash. q-means: A quantum algorithm for unsupervised machine learning. *Advances in Neural Information Processing Systems*, 32, 2019.
- [81] Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. Quantum speed-up for unsupervised learning. *Machine Learning*, 90:261–287, 2013.
- [82] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint arXiv:1307.0411*, 2013.
- [83] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.
- [84] Daoyi Dong, Chunlin Chen, Hanxiong Li, and Tzyh-Jong Tarn. Quantum Reinforcement Learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 38(5):1207–1220, 2008.
- [85] Vedran Dunjko, Jacob M Taylor, and Hans J Briegel. Quantum-Enhanced Machine Learning. *Physical Review Letters*, 117(13):130501, 2016.
- [86] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.
- [87] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.
- [88] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.
- [89] Edward Farhi and Hartmut Neven. Classification with Quantum Neural Networks on Near Term Processors. *arXiv preprint arXiv:1802.06002*, 2018.
- [90] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1(6):403–409, 2021.

- [91] Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit Born machines. *Physical Review A*, 98(6):062324, 2018.
- [92] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Quantum Generative Adversarial Networks for Learning and Loading Random Distributions. *npj Quantum Information*, 5(1):103, 2019.
- [93] Christa Zoufal, Aurélien Lucchi, and Stefan Woerner. Variational quantum Boltzmann machines. *Quantum Machine Intelligence*, 3:1–15, 2021.
- [94] Maria Schuld and Nathan Killoran. Quantum machine learning in feature Hilbert spaces. *Physical Review Letters*, 122(4):040504, 2019.
- [95] Louis Schatzki, Andrew Arrasmith, Patrick J Coles, and Marco Cerezo. Entangled Datasets for Quantum Machine Learning. *arXiv preprint arXiv:2109.03400*, 2021.
- [96] Elija Perrier, Akram Youssry, and Chris Ferrie. QDataSet, quantum datasets for machine learning. *Scientific Data*, 9(1):582, 2022.
- [97] Michael J Bremner, Ashley Montanaro, and Dan J Shepherd. Average-case complexity versus approximate simulation of commuting quantum computations. *Physical Review Letters*, 117(8):080501, 2016.
- [98] Leslie Ann Goldberg and Heng Guo. The complexity of approximating complex-valued Ising and Tutte partition functions. *Computational Complexity*, 26(4):765–833, 2017.
- [99] Dave Wecker, Matthew B Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, 2015.
- [100] Roeland Wiersema, Cunlu Zhou, Yvette de Sereville, Juan Felipe Carrasquilla, Yong Baek Kim, and Henry Yuen. Exploring Entanglement and Optimization within the Hamiltonian Variational Ansatz. *PRX Quantum*, 1(2):020319, 2020.
- [101] 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.
- [102] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1):1791, 2021.
- [103] Andris Ambainis, Ashwin Nayak, Ammon Ta-Shma, and Umesh Vazirani. Dense quantum coding and a lower bound for 1-way quantum automata. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 376–383, 1999.
- [104] Masahito Hayashi, Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, and Shigeru Yamashita. Quantum Network Coding. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 610–621. Springer, 2007.
- [105] Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, and Shigeru Yamashita. Unbounded-error one-way classical and quantum communication complexity. In *Automata, Languages and Programming: 34th International Colloquium, ICALP 2007, Wrocław, Poland, July 9-13, 2007. Proceedings 34*, pages 110–121. Springer, 2007.
- [106] Hiroshi Yano, Yudai Suzuki, Kohei M Itoh, Rudy Raymond, and Naoki Yamamoto. Efficient Discrete Feature Encoding for Variational Quantum Classifier. *IEEE Transactions on Quantum Engineering*, 2:1–14, 2021.

- [107] Bryan T Gard, Linghua Zhu, George S Barron, Nicholas J Mayhall, Sophia E Economou, and Edwin Barnes. Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm. *npj Quantum Information*, 6(1):10, 2020.
- [108] Panagiotis Kl Barkoutsos, Jerome F Gonthier, Igor Sokolov, Nikolaj Moll, Gian Salis, Andreas Fuhrer, Marc Ganzhorn, Daniel J Egger, Matthias Troyer, Antonio Mezzacapo, et al. Quantum algorithms for electronic structure calculations: Particle/hole Hamiltonian and optimized wave-function expansions. *Physical Review A*, 98(2):022322, 2018.
- [109] Igor O Sokolov, Panagiotis Kl Barkoutsos, Pauline J Ollitrault, Donny Greenberg, Julia Rice, Marco Pistoia, and Ivano Tavernelli. Quantum orbital-optimized unitary coupled cluster methods in the strongly correlated regime: Can quantum algorithms outperform their classical equivalents? *The Journal of Chemical Physics*, 152(12):124107, 2020.
- [110] Sofiene Jerbi, Lukas J Fiderer, Hendrik Poulsen Nautrup, Jonas M Kübler, Hans J Briegel, and Vedran Dunjko. Quantum machine learning beyond kernel methods. *Nature Communications*, 14(1):517, 2023.
- [111] Nhat A Nghiem, Samuel Yen-Chi Chen, and Tzu-Chieh Wei. Unified framework for quantum classification. *Physical Review Research*, 3(3):033056, 2021.
- [112] Dylan Herman, Rudy Raymond, Muyuan Li, Nicolas Robles, Antonio Mezzacapo, and Marco Pistoia. Expressivity of Variational Quantum Machine Learning on the Boolean Cube. *IEEE Transactions on Quantum Engineering*, 4:1–18, 2023.
- [113] Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622*, 2020.
- [114] Maria Schuld. Supervised quantum machine learning models are kernel methods. *arXiv preprint arXiv:2101.11020*, 2021.
- [115] Nachman Aronszajn. Theory of Reproducing Kernels. *Transactions of the American Mathematical Society*, 68(3):337–404, 1950.
- [116] George Kimeldorf and Grace Wahba. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications*, 33(1):82–95, 1971.
- [117] Thomas Hofmann, Bernhard Schölkopf, and Alexander J Smola. Kernel methods in machine learning. *The Annals of Statistics*, 36(3):1171–1220, 2008.
- [118] Alex J Smola and Bernhard Schölkopf. *Learning with Kernels*, volume 4. Berlin, Germany: GMD-Forschungszentrum Informationstechnik, 1998.
- [119] Tony Jebara, Risi Kondor, and Andrew Howard. Probability Product Kernels. *The Journal of Machine Learning Research*, 5:819–844, 2004.
- [120] Risi Kondor and Tony Jebara. A kernel between sets of vectors. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 361–368, 2003.
- [121] Anil Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distribution. *Bulletin of the Calcutta Mathematical Society*, 35:99–110, 1943.
- [122] Tommi Jaakkola and David Haussler. Exploiting generative models in discriminative classifiers. *Advances in Neural Information Processing Systems*, 11:487–493, 1998.

- [123] Shun-Ichi Amari. Natural Gradient Works Efficiently in Learning. *Neural Computation*, 10(2):251–276, 1998.
- [124] Florent Perronnin, Jorge Sánchez, and Thomas Mensink. Improving the Fisher Kernel for Large-Scale Image Classification. In *European Conference on Computer Vision*, pages 143–156. Springer, 2010.
- [125] Vladyslav Sydorov, Mayu Sakurada, and Christoph H. Lampert. Deep Fisher Kernels - End to End Learning of the Fisher Kernel GMM Parameters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1402–1409, 2014.
- [126] Denis Gudovskiy, Alec Hodgkinson, Takuya Yamaguchi, and Sotaro Tsukizawa. Deep Active Learning for Biased Datasets via Fisher Kernel Self-Supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9041–9049, 2020.
- [127] Laurens Van Der Maaten. Learning Discriminative Fisher Kernels. In *International Conference on Machine Learning*, volume 11, pages 217–224, 2011.
- [128] Jorge Sánchez, Florent Perronnin, Thomas Mensink, and Jakob Verbeek. Image Classification with the Fisher Vector: Theory and Practice. *International Journal of Computer Vision*, 105(3):222–245, 2013.
- [129] Harry Buhrman, Richard Cleve, John Watrous, and Ronald De Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16):167902, 2001.
- [130] Manuel Blum and Silvio Micali. How to generate cryptographically strong sequences of pseudo random bits. In *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*, pages 227–240. 2019.
- [131] Michele Mosca and Christof Zalka. Exact quantum Fourier transforms and discrete logarithm algorithms. *International Journal of Quantum Information*, 2(01):91–100, 2004.
- [132] Scott Aaronson and Andris Ambainis. Forrelation: A Problem that Optimally Separates Quantum from Classical Computing. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 307–316, 2015.
- [133] Graham R Enos, Matthew J Reagor, Maxwell P Henderson, Christina Young, Kyle Horton, Mandy Birch, and Chad Rigetti. Synthetic weather radar using hybrid quantum-classical machine learning. *arXiv preprint arXiv:2111.15605*, 2021.
- [134] Zoran Krunic, Frederik F Flöther, George Seegan, Nathan D Earnest-Noble, and Omar Shehab. Quantum Kernels for Real-World Predictions Based on Electronic Health Records. *IEEE Transactions on Quantum Engineering*, 3:1–11, 2022.
- [135] Keisuke Fujii and Kohei Nakajima. Quantum Reservoir Computing: A Reservoir Approach Toward Quantum Machine Learning on Near-Term Quantum Devices. *Reservoir Computing: Theory, Physical Implementations, and Applications*, pages 423–450, 2021.
- [136] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [137] Danilo Mandic and Jonathon Chambers. *Recurrent Neural Networks for Prediction: Learning Algorithms, Architectures and Stability*. Wiley, 2001.

- [138] Herbert Jaeger and Harald Haas. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304(5667):78–80, 2004.
- [139] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.
- [140] David Verstraeten, Benjamin Schrauwen, Michiel d’Haene, and Dirk Stroobandt. An experimental unification of reservoir computing methods. *Neural Networks*, 20(3):391–403, 2007.
- [141] Herbert Jaeger. *Tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the “Echo State Network” approach*. GMD-Forschungszentrum Informationstechnik Bonn, 2002.
- [142] Herbert Jaeger. The “echo state” approach to analysing and training recurrent neural networks—with an Erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148(34):13, 2001.
- [143] Wolfgang Maass, Thomas Natschläger, and Henry Markram. Real-Time Computing without Stable States: A New Framework for Neural Computation Based on Perturbations. *Neural Computation*, 14(11):2531–2560, 2002.
- [144] David E Rumelhart, Geoffrey E Hinton, Ronald J Williams, et al. *Learning Internal Representations by Error Propagation*. MIT Press, 1986.
- [145] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [146] Gouhei Tanaka, Toshiyuki Yamane, Jean Benoit Héroux, Ryosho Nakane, Naoki Kanazawa, Seiji Takeda, Hidetoshi Numata, Daiju Nakano, and Akira Hirose. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, 2019.
- [147] Kohei Nakajima. Physical reservoir computing—an introductory perspective. *Japanese Journal of Applied Physics*, 59(6):060501, 2020.
- [148] Kohei Nakajima, Helmut Hauser, Rongjie Kang, Emanuele Guglielmino, Darwin G Caldwell, and Rolf Pfeifer. A soft body as a reservoir: case studies in a dynamic model of octopus-inspired soft robotic arm. *Frontiers in Computational Neuroscience*, 7:91, 2013.
- [149] Kohei Nakajima, Tao Li, Helmut Hauser, and Rolf Pfeifer. Exploiting short-term memory in soft body dynamics as a computational resource. *Journal of The Royal Society Interface*, 11(100):20140437, 2014.
- [150] Kohei Nakajima, Helmut Hauser, Tao Li, and Rolf Pfeifer. Information processing via physical soft body. *Scientific Reports*, 5(1):10487, 2015.
- [151] Laurent Larger, Miguel C Soriano, Daniel Brunner, Lennert Appeltant, Jose M Gutiérrez, Luis Pesquera, Claudio R Mirasso, and Ingo Fischer. Photonic information processing beyond turing: an optoelectronic implementation of reservoir computing. *Optics Express*, 20(3):3241–3249, 2012.
- [152] Yvan Paquot, Francois Duport, Antoneo Smerieri, Joni Dambre, Benjamin Schrauwen, Marc Haelterman, and Serge Massar. Optoelectronic Reservoir Computing. *Scientific Reports*, 2(1):287, 2012.

- [153] Laurent Larger, Antonio Baylón-Fuentes, Romain Martinenghi, Vladimir S Udaltsov, Yanne K Chembo, and Maxime Jacquot. High-Speed Photonic Reservoir Computing Using a Time-Delay-Based Architecture: Million Words per Second Classification. *Physical Review X*, 7(1):011015, 2017.
- [154] Miguel C Soriano, Silvia Ortín, Lars Keuninckx, Lennert Appeltant, Jan Danckaert, Luis Pesquera, and Guy Van der Sande. Delay-Based Reservoir Computing: Noise Effects in a Combined Analog and Digital Implementation. *IEEE Transactions on Neural Networks and Learning Systems*, 26(2):388–393, 2014.
- [155] Jiayin Chen, Hendra I Nurdin, and Naoki Yamamoto. Temporal Information Processing on Noisy Quantum Computers. *Physical Review Applied*, 14(2):024065, 2020.
- [156] Makoto Negoro, Kosuke Mitarai, Kohei Nakajima, and Keisuke Fujii. Toward NMR Quantum Reservoir Computing. *Reservoir Computing: Theory, Physical Implementations, and Applications*, pages 451–458, 2021.
- [157] Rodrigo Araiza Bravo, Khadijeh Najafi, Xun Gao, and Susanne F Yelin. Quantum reservoir computing using arrays of Rydberg atoms. *PRX Quantum*, 3(3):030325, 2022.
- [158] Kohei Nakajima, Keisuke Fujii, Makoto Negoro, Kosuke Mitarai, and Masahiro Kitagawa. Boosting Computational Power through Spatial Multiplexing in Quantum Reservoir Computing. *Physical Review Applied*, 11(3):034021, 2019.
- [159] Aki Kutvonen, Keisuke Fujii, and Takahiro Sagawa. Optimizing a quantum reservoir computer for time series prediction. *Scientific Reports*, 10(1):14687, 2020.
- [160] Sanjib Ghosh, Andrzej Opala, Michał Matuszewski, Tomasz Paterek, and Timothy CH Liew. Quantum reservoir processing. *npj Quantum Information*, 5(1):35, 2019.
- [161] Jiayin Chen and Hendra I Nurdin. Learning nonlinear input–output maps with dissipative quantum systems. *Quantum Information Processing*, 18:1–36, 2019.
- [162] Quoc Hoan Tran and Kohei Nakajima. Higher-Order Quantum Reservoir Computing. *arXiv preprint arXiv:2006.08999*, 2020.
- [163] Julien Dudas, Baptiste Carles, Erwan Plouet, Frank Alice Mizrahi, Julie Grollier, and Danijela Marković. Quantum reservoir computing implementation on coherently coupled quantum oscillators. *npj Quantum Information*, 9(1):64, 2023.
- [164] Pere Mujal, Rodrigo Martínez-Peña, Gian Luca Giorgi, Miguel C Soriano, and Roberta Zambrini. Time-series quantum reservoir computing with weak and projective measurements. *npj Quantum Information*, 9(1):16, 2023.
- [165] Makoto Negoro, Kosuke Mitarai, Keisuke Fujii, Kohei Nakajima, and Masahiro Kitagawa. Machine learning with controllable quantum dynamics of a nuclear spin ensemble in a solid. *arXiv preprint arXiv:1806.10910*, 2018.
- [166] Pere Mujal, Rodrigo Martínez-Peña, Johannes Nokkala, Jorge García-Beni, Gian Luca Giorgi, Miguel C Soriano, and Roberta Zambrini. Opportunities in Quantum Reservoir Computing and Extreme Learning Machines. *Advanced Quantum Technologies*, 4(8):2100027, 2021.

- [167] Zaher Mundher Yaseen, Sadeq Olewi Sulaiman, Ravinesh C Deo, and Kwok-Wing Chau. An enhanced extreme learning machine model for river flow forecasting: State-of-the-art, practical applications in water resource engineering area and future research direction. *Journal of Hydrology*, 569:387–408, 2019.
- [168] Thomas G Dietterich. Ensemble Methods in Machine Learning. In *International Workshop on Multiple Classifier Systems*, pages 1–15. Springer, 2000.
- [169] XiMing Wang, YueChi Ma, Min-Hsiu Hsieh, and Man-Hong Yung. Quantum speedup in adaptive boosting of binary classification. *Science China Physics, Mechanics & Astronomy*, 64(2):220311, 2021.
- [170] Rupak Chatterjee and Ting Yu. Generalized coherent states, reproducing kernels, and quantum support vector machines. *Quantum Information & Computation*, 17(15–16):1292–1306, 2017.
- [171] Gert RG Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I Jordan. Learning the Kernel Matrix with Semidefinite Programming. *Journal of Machine Learning Research*, 5:27–72, 2004.
- [172] Qiskit contributors. Qiskit: An open-source framework for quantum computing, 2023.
- [173] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. API design for machine learning software: experiences from the scikit-learn project. *arXiv preprint arXiv:1309.0238*, 2013.
- [174] Aram W Harrow and Saeed Mehraban. Approximate Unitary t -Designs by Short Random Quantum Circuits Using Nearest-Neighbor and Long-Range Gates. *Communications in Mathematical Physics*, 401:1531–1626, 2023.
- [175] 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, 2009.
- [176] Joseph M Renes, Robin Blume-Kohout, Andrew J Scott, and Carlton M Caves. Symmetric informationally complete quantum measurements. *Journal of Mathematical Physics*, 45(6):2171–2180, 2004.
- [177] Jonas Kübler, Simon Buchholz, and Bernhard Schölkopf. The Inductive Bias of Quantum Kernels. *Advances in Neural Information Processing Systems*, 34:12661–12673, 2021.
- [178] Abdulkadir Canatar, Evan Peters, Cengiz Pehlevan, Stefan M Wild, and Ruslan Shaydulin. Bandwidth Enables Generalization in Quantum Kernel Models. *arXiv preprint arXiv:2206.06686*, 2022.
- [179] Yudai Suzuki and Muyuan Li. Effect of alternating layered ansatzes on trainability of projected quantum kernel. *arXiv preprint arXiv:2310.00361*, 2023.
- [180] 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, 2018.

- [181] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12(1):6961, 2021.
- [182] Marco Cerezo and Patrick J Coles. Higher order derivatives of quantum neural networks with barren plateaus. *Quantum Science and Technology*, 6(3):035006, 2021.
- [183] Andrew Arrasmith, Marco Cerezo, Piotr Czarnik, Lukasz Cincio, and Patrick J Coles. Effect of barren plateaus on gradient-free optimization. *Quantum*, 5:558, 2021.
- [184] Lorenzo Leone, Salvatore FE Oliviero, Lukasz Cincio, and Marco Cerezo. On the practical usefulness of the Hardware Efficient Ansatz. *arXiv preprint arXiv:2211.01477*, 2022.
- [185] Martin Larocca, Piotr Czarnik, Kunal Sharma, Gopikrishnan Muraleedharan, Patrick J Coles, and Marco Cerezo. Diagnosing Barren Plateaus with Tools from Quantum Optimal Control. *Quantum*, 6:824, 2022.
- [186] Zoë Holmes, Andrew Arrasmith, Bin Yan, Patrick J Coles, Andreas Albrecht, and Andrew T Sornborger. Barren Plateaus Preclude Learning Scramblers. *Physical Review Letters*, 126(19):190501, 2021.
- [187] Zoë Holmes, Kunal Sharma, Marco Cerezo, and Patrick J Coles. Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus. *PRX Quantum*, 3(1):010313, 2022.
- [188] Andrew Arrasmith, Zoë Holmes, Marco Cerezo, and Patrick J Coles. Equivalence of quantum barren plateaus to cost concentration and narrow gorges. *Quantum Science and Technology*, 7(4):045015, 2022.
- [189] Jennifer R Glick, Tanvi P Gujarati, Antonio D Corcoles, Youngseok Kim, Abhinav Kandala, Jay M Gambetta, and Kristan Temme. Covariant quantum kernels for data with group structure. *arXiv preprint arXiv:2105.03406*, 2021.
- [190] Akio Fujiwara and Hiroshi Nagaoka. Quantum Fisher metric and estimation for pure state models. *Physics Letters A*, 201(2-3):119–124, 1995.
- [191] Dénes Petz. Monotone metrics on matrix spaces. *Linear Algebra and its Applications*, 244:81–96, 1996.
- [192] Carl W Helstrom. Minimum mean-squared error of estimates in quantum statistics. *Physics Letters A*, 25(2):101–102, 1967.
- [193] Jing Liu, Haidong Yuan, Xiao-Ming Lu, and Xiaoguang Wang. Quantum Fisher information matrix and multiparameter estimation. *Journal of Physics A: Mathematical and Theoretical*, 53(2):023001, 2019.
- [194] Thomas Hofmann. Learning the Similarity of Documents: An Information-Geometric Approach to Document Retrieval and Categorization. *Advances in Neural Information Processing Systems*, 12, 1999.
- [195] Florent Perronnin and Christopher Dance. Fisher Kernels on Visual Vocabularies for Image Categorization. In *2007 IEEE conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

- [196] Cirq-Developers. Cirq, April 2022. Zenodo. doi:10.5281/zenodo.6599601. See full list of authors on Github: <https://github.com/quantumlib/Cirq/graphs/contributors>.
- [197] 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.
- [198] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay May- orov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimr- man, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.
- [199] Sukin Sim, Peter D Johnson, and Alán Aspuru-Guzik. Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.
- [200] Martín Larocca, Frédéric Sauvage, Faris M Sbahi, Guillaume Verdon, Patrick J Coles, and Marco Cerezo. Group-Invariant Quantum Machine Learning. *PRX Quantum*, 3(3):030341, 2022.
- [201] Michael Ragone, Paolo Braccia, Quynh T Nguyen, Louis Schatzki, Patrick J Coles, Fred- eric Sauvage, Martin Larocca, and M Cerezo. Representation Theory for Geometric Quan- tum Machine Learning. *arXiv preprint arXiv:2210.07980*, 2022.
- [202] Louis Schatzki, Martin Larocca, Frederic Sauvage, and Marco Cerezo. Theoretical Guarantees for Permutation-Equivariant Quantum Neural Networks. *arXiv preprint arXiv:2210.09974*, 2022.
- [203] Quynh T Nguyen, Louis Schatzki, Paolo Braccia, Michael Ragone, Patrick J Coles, Fred- eric Sauvage, Martin Larocca, and M Cerezo. Theory for Equivariant Quantum Neural Networks. *arXiv preprint arXiv:2210.08566*, 2022.
- [204] Johannes Jakob Meyer, Marian Mularski, Elies Gil-Fuster, Antonio Anna Mele, Francesco Arzani, Alissa Wilms, and Jens Eisert. Exploiting Symmetry in Variational Quantum Machine Learning. *PRX Quantum*, 4(1):010328, 2023.
- [205] Thomas Gorin, Tomaž Prosen, Thomas H Seligman, and Marko Žnidarič. Dynamics of Loschmidt echoes and fidelity decay. *Physics Reports*, 435(2-5):33–156, 2006.
- [206] Arseni Goussev, Rodolfo A Jalabert, Horacio M Pastawski, and Diego Wisniacki. Loschmidt Echo. *arXiv preprint arXiv:1206.6348*, 2012.
- [207] Koji Hashimoto, Keiju Murata, and Ryosuke Yoshii. Out-of-time-order correlators in quantum mechanics. *Journal of High Energy Physics*, 2017(138):138, 2017.
- [208] Brian Swingle. Unscrambling the physics of out-of-time-order correlators. *Nature Physics*, 14(10):988–990, 2018.

- [209] Sanjib Ghosh, Andrzej Opala, Michał Matuszewski, Tomasz Paterek, and Timothy CH Liew. Reconstructing Quantum States with Quantum Reservoir Networks. *IEEE Transactions on Neural Networks and Learning Systems*, 32(7):3148–3155, 2020.
- [210] Johannes Nokkala, Rodrigo Martínez-Peña, Gian Luca Giorgi, Valentina Parigi, Miguel C Soriano, and Roberta Zambrini. Gaussian states of continuous-variable quantum systems provide universal and versatile reservoir computing. *Communications Physics*, 4(1):53, 2021.
- [211] Gerasimos Angelatos, Saeed A Khan, and Hakan E Türeci. Reservoir Computing Approach to Quantum State Measurement. *Physical Review X*, 11(4):041062, 2021.
- [212] Tomoyuki Kubota, Hirokazu Takahashi, and Kohei Nakajima. Unifying framework for information processing in stochastically driven dynamical systems. *Physical Review Research*, 3(4):043135, 2021.
- [213] Gandhi Manjunath and Herbert Jaeger. Echo State Property Linked to an Input: Exploring a Fundamental Characteristic of Recurrent Neural Networks. *Neural Computation*, 25(3):671–696, 2013.
- [214] Izzet B Yildiz, Herbert Jaeger, and Stefan J Kiebel. Re-visiting the echo state property. *Neural Networks*, 35:1–9, 2012.
- [215] Frank Verstraete, Michael M Wolf, and J Ignacio Cirac. Quantum computation and quantum-state engineering driven by dissipation. *Nature Physics*, 5(9):633–636, 2009.
- [216] Yuxuan Du, Min-Hsiu Hsieh, Tongliang Liu, Dacheng Tao, and Nana Liu. Quantum noise protects quantum classifiers against adversaries. *Physical Review Research*, 3(2):023153, 2021.
- [217] Jonathan Foldager, Arthur Pesah, and Lars Kai Hansen. Noise-assisted variational quantum thermalization. *Scientific Reports*, 12(1):3862, 2022.
- [218] Joshua Morris, Felix A Pollock, and Kavan Modi. Non-Markovian memory in IBMQX4. *arXiv preprint arXiv:1902.07980*, 2019.
- [219] Adam Winick, Joel J Wallman, and Joseph Emerson. Simulating and Mitigating Crosstalk. *Physical Review Letters*, 126(23):230502, 2021.
- [220] Jürgen Schmidhuber, Sepp Hochreiter, et al. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
- [221] Tomoyuki Kubota, Kohei Nakajima, and Hirokazu Takahashi. Dynamical Anatomy of NARMA10 Benchmark Task. *arXiv preprint arXiv:1906.04608*, 2019.
- [222] Amir F Atiya and Alexander G Parlos. New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, 11(3):697–709, 2000.
- [223] Antonio D Córcoles, Easwar Magesan, Srikanth J Srinivasan, Andrew W Cross, Matthias Steffen, Jay M Gambetta, and Jerry M Chow. Demonstration of a quantum error detection code using a square lattice of four superconducting qubits. *Nature Communications*, 6(1):6979, 2015.

- [224] Flavio Abreu Araujo, Mathieu Riou, Jacob Torrejon, Sumito Tsunegi, Damien Querlioz, Kay Yakushiji, Akio Fukushima, Hitoshi Kubota, Shinji Yuasa, Mark D Stiles, et al. Role of non-linear data processing on speech recognition task in the framework of reservoir computing. *Scientific Reports*, 10(1):328, 2020.
- [225] Isaac L Chuang and Michael A Nielsen. Prescription for experimental determination of the dynamics of a quantum black box. *Journal of Modern Optics*, 44(11-12):2455–2467, 1997.
- [226] MPA Branderhorst, J Nunn, IA Walmsley, and RL Kosut. Simplified quantum process tomography. *New Journal of Physics*, 11(11):115010, 2009.
- [227] A Shabani, RL Kosut, M Mohseni, H Rabitz, MA Broome, MP Almeida, A Fedrizzi, and AG White. Efficient Measurement of Quantum Dynamics via Compressive Sensing. *Physical Review Letters*, 106(10):100401, 2011.
- [228] Michael J Korenberg. Identifying nonlinear difference equation and functional expansion representations: the fast orthogonal algorithm. *Annals of Biomedical Engineering*, 16:123–142, 1988.
- [229] Joni Dambre, David Verstraeten, Benjamin Schrauwen, and Serge Massar. Information Processing Capacity of Dynamical Systems. *Scientific Reports*, 2(1):514, 2012.
- [230] Nozomi Akashi, Terufumi Yamaguchi, Sumito Tsunegi, Tomohiro Taniguchi, Mitsuhiro Nishida, Ryo Sakurai, Yasumichi Wakao, and Kohei Nakajima. Input-driven bifurcations and information processing capacity in spintronics reservoirs. *Physical Review Research*, 2(4):043303, 2020.
- [231] Ryo Sakurai, Mitsuhiro Nishida, Hideyuki Sakurai, Yasumichi Wakao, Nozomi Akashi, Yasuo Kuniyoshi, Yuna Minami, and Kohei Nakajima. Emulating a sensor using soft material dynamics: A reservoir computing approach to pneumatic artificial muscle. In *2020 3rd IEEE International Conference on Soft Robotics (RoboSoft)*, pages 710–717. IEEE, 2020.
- [232] Alistair Letcher, Stefan Woerner, and Christa Zoufal. From Tight Gradient Bounds for Parameterized Quantum Circuits to the Absence of Barren Plateaus in QGANs. *arXiv preprint arXiv:2309.12681*, 2023.
- [233] Samuel L. Braunstein and Carlton M. Caves. Statistical distance and the geometry of quantum states. *Physical Review Letters*, 72:3439, 1994.
- [234] Alexander S. Holevo. *Probabilistic and Statistical Aspects of Quantum Theory*, volume 1. Springer Science & Business Media, 2011.

Appendix A

Analytical Results for Vanishing Similarity Issue in Quantum Kernels

A.1 Proof of Proposition 1

In this appendix, we give proof of Proposition 1 in Sec. 4.2. More concretely, we analytically show the expectation value and variance of the fidelity-based quantum kernel in Eq. (4.24) for two types of quantum circuits: globally-random quantum circuits and alternating layered ansatzes (ALAs). Our strategy is to integrate the quantum kernel over quantum circuits that possess the 2-design property explained in Sec. 4.2.2. For simplicity, we denote the fidelity-based quantum kernel $k_Q \equiv k_Q(\mathbf{x}, \mathbf{x}')$.

A.1.1 Case (1): Globally-Random Quantum Circuits

Expectation Value

First, we derive the expectation value. Due to the assumption that $U(\mathbf{x}, \boldsymbol{\theta})$ or $U(\mathbf{x}', \boldsymbol{\theta})$ is a t -design with $t \geq 1$, we apply Lemma 1 to obtain the expectation value. Here, without loss of generality, we consider $U(\mathbf{x}, \boldsymbol{\theta})$ forms a 1-design. Then, we have

$$\begin{aligned} \langle k_Q \rangle_{(U(\mathbf{x}, \boldsymbol{\theta}), U(\mathbf{x}', \boldsymbol{\theta}))} &= \left\langle \text{Tr} \left[U(\mathbf{x}, \boldsymbol{\theta}) \rho_0 U^\dagger(\mathbf{x}, \boldsymbol{\theta}) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \right\rangle_{(U(\mathbf{x}, \boldsymbol{\theta}), U(\mathbf{x}', \boldsymbol{\theta}))} \\ &= \left\langle \frac{1}{2^n} \text{Tr} [\rho_0] \text{Tr} [\rho_{\mathbf{x}', \boldsymbol{\theta}}] \right\rangle_{U(\mathbf{x}, \boldsymbol{\theta})} \\ &= \frac{1}{2^n}, \end{aligned} \tag{A.1}$$

where we use the property of density operators, i.e., $\text{Tr}[\rho] = 1$ in the second equality. We also utilize the assumption that $U(\mathbf{x}, \boldsymbol{\theta})$ and $U(\mathbf{x}', \boldsymbol{\theta})$ are independent.

Variance

Next, we work on the variance. By definition, the variance can be expressed as $\text{Var}[X] = \langle X^2 \rangle - \langle X \rangle^2$ for any variable X . We hence focus on the term $\langle k_Q^2 \rangle$. We again integrate the

quantity k_Q^2 over the quantum circuit $U(\mathbf{x}, \boldsymbol{\theta})$, then we have

$$\begin{aligned}
\langle k_Q^2 \rangle_{(U(\mathbf{x}, \boldsymbol{\theta}), U(\mathbf{x}', \boldsymbol{\theta}))} &= \left\langle \text{Tr} \left[U(\mathbf{x}, \boldsymbol{\theta}) \rho_0 U^\dagger(\mathbf{x}, \boldsymbol{\theta}) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \text{Tr} \left[U(\mathbf{x}, \boldsymbol{\theta}) \rho_0 U^\dagger(\mathbf{x}, \boldsymbol{\theta}) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \right\rangle_{(U(\mathbf{x}, \boldsymbol{\theta}), U(\mathbf{x}', \boldsymbol{\theta}))} \\
&= \left\langle \frac{1}{2^{2n} - 1} (\text{Tr} [\rho_0] \text{Tr} [\rho_{\mathbf{x}', \boldsymbol{\theta}}] \text{Tr} [\rho_0] \text{Tr} [\rho_{\mathbf{x}', \boldsymbol{\theta}}] + \text{Tr} [\rho_0^2] \text{Tr} [\rho_{\mathbf{x}', \boldsymbol{\theta}}^2]) \right. \\
&\quad \left. - \frac{1}{2^n (2^{2n} - 1)} (\text{Tr} [\rho_0] \text{Tr} [\rho_0] \text{Tr} [\rho_{\mathbf{x}', \boldsymbol{\theta}}^2] + \text{Tr} [\rho_0^2] \text{Tr} [\rho_{\mathbf{x}', \boldsymbol{\theta}}] \text{Tr} [\rho_{\mathbf{x}', \boldsymbol{\theta}}]) \right\rangle_{U(\mathbf{x}', \boldsymbol{\theta})} \\
&= \frac{2}{2^n (2^n + 1)}.
\end{aligned} \tag{A.2}$$

Here, we exploit Lemma 3 and the property of pure states, $\text{Tr}[\rho] = \text{Tr}[\rho^2] = 1$. As a result, the variance can be written as

$$\begin{aligned}
\text{Var}[k_Q] &= \langle k_Q^2 \rangle_{(U(\mathbf{x}, \boldsymbol{\theta}), U(\mathbf{x}', \boldsymbol{\theta}))} - \langle k_Q \rangle_{(U(\mathbf{x}, \boldsymbol{\theta}), U(\mathbf{x}', \boldsymbol{\theta}))}^2 \\
&= \frac{2}{2^n (2^n + 1)} - \frac{1}{2^{2n}} \\
&= \frac{2^n - 1}{2^{2n} (2^n + 1)}.
\end{aligned} \tag{A.3}$$

We also remark that the obtained value in Eq. (A.3) is the upper bound of the variance for the case of the mixed initial state.

A.1.2 Case (2): Alternating Layered Ansatzes

Expectation Value

The expectation value over $U(\mathbf{x}, \boldsymbol{\theta})$ can be obtained by independently integrating over local unitary blocks in quantum circuits. We remind the readers that we assume the independence of all unitary blocks. We here start with the integration over the κ -th unitary blocks in the last layer, i.e., $W_{\kappa, L}(\mathbf{x}, \boldsymbol{\theta}_{\kappa, L})$. For simplicity we will denote the local unitary blocks $W_{\kappa, L} \equiv W_{\kappa, L}(\mathbf{x}, \boldsymbol{\theta}_{\kappa, L})$ and $W'_{\kappa, L} \equiv W_{\kappa, L}(\mathbf{x}', \boldsymbol{\theta}_{\kappa, L})$. Then, we have

$$\begin{aligned}
\langle k_Q \rangle_{W_{\kappa, L}} &= \left\langle \text{Tr} \left[W_{\kappa, L} \rho_0^{(\kappa, L)} W_{\kappa, L}^\dagger \rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \right\rangle_{W_{\kappa, L}} \\
&= \frac{1}{2^m} \text{Tr} \left[\left(\text{Tr}_{S(\kappa, L)} \left[\rho_0^{(\kappa, L)} \right] \otimes \mathbb{I}_{S(\kappa, L)} \right) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right],
\end{aligned} \tag{A.4}$$

where $\rho_0^{(a, b)} = U_{a, b} \rho_0 U_{a, b}^\dagger$ with $U_{a, b} = (\prod_{k'=1}^{a-1} W_{k', b}) (\prod_{d=1}^{b-1} V_d(\mathbf{x}, \boldsymbol{\theta}))$. Note that $U_{a, b}$ means all gates up to the $(a-1)$ -th blocks in the b -th layer. Also, $\text{Tr}_{S(k, d)}$ and $\mathbb{I}_{S(k, d)}$ are the partial trace and the identity operator over the subspace $S(k, d)$ of the qubits on which $W_{k, d}$ acts, respectively. We also utilize Lemma 4 here. By iteratively integrating the quantity for all unitary blocks in the ALA, we obtain

$$\begin{aligned}
\langle k_Q \rangle_{U(\mathbf{x}, \boldsymbol{\theta})} &= \frac{1}{(2^m)^{\kappa L}} \text{Tr} \left[\left(\text{Tr}_{S(\kappa, L)} \left[\text{Tr}_{S(\kappa-1, L)} \left[\dots \text{Tr}_{S(1, 1)} \left[\rho_0^{(1, 1)} \right] \otimes \mathbb{I}_{S(1, 1)} \dots \right] \otimes \mathbb{I}_{S(\kappa-1, L)} \right) \otimes \mathbb{I}_{S(\kappa, L)} \right) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right].
\end{aligned} \tag{A.5}$$

Here, $\rho_0^{(1,1)} = \rho_0$ by definition. Let $\rho_0 = \sum_{\alpha, \alpha'} c_\alpha c_{\alpha'}^* |\alpha\rangle \langle \alpha'|$ be an arbitrary initial state where α and α' are bit-strings, and $c_\alpha, c_{\alpha'} \in \mathbb{C}$ satisfying $\sum_{\alpha, \alpha'} c_\alpha c_{\alpha'}^* = 1$. Then the quantity up to the first layer, i.e., $\text{Tr}_{S_{(\kappa,1)}}[\dots] \otimes \mathbb{I}_{S_{(\kappa,1)}}$ in Eq.(A.5), can be written as

$$\begin{aligned} & \text{Tr}_{S_{(\kappa,1)}} \left[\text{Tr}_{S_{(\kappa-1,1)}} \left[\dots \text{Tr}_{S_{(2,1)}} \left[\text{Tr}_{S_{(1,1)}} [\rho_0] \otimes \mathbb{I}_{S_{(1,1)}} \right] \otimes \mathbb{I}_{S_{(2,1)}} \right] \dots \otimes \mathbb{I}_{S_{(\kappa-1,1)}} \right] \otimes \mathbb{I}_{S_{(\kappa,1)}} \\ &= \sum_{\alpha, \alpha'} c_\alpha c_{\alpha'}^* \left(\prod_{k=1}^{\kappa} \delta_{(\alpha, \alpha')_{S_k}} \right) \times (\mathbb{I}_{S_{(1,1)}} \otimes \mathbb{I}_{S_{(2,1)}} \otimes \dots \otimes \mathbb{I}_{S_{(\kappa,1)}}) \\ &= \mathbb{I}. \end{aligned} \quad (\text{A.6})$$

We here utilize the fact that any local unitary blocks in the same layer have no overlap. Consequently, by substituting Eq. (A.6) into Eq. (A.5), we get

$$\langle k_Q \rangle_{(U(\mathbf{x}, \boldsymbol{\theta}), U(\mathbf{x}', \boldsymbol{\theta}))} = \langle k_Q \rangle_{U(\mathbf{x}, \boldsymbol{\theta})} = \frac{(2^m)^{\kappa(L-1)}}{(2^m)^{\kappa L}} = \frac{1}{2^n}. \quad (\text{A.7})$$

We note that the numerator in the first equality comes from the trace of the identity operators over the whole system by $L - 1$ times. Also, $n = m\kappa$ is used here.

Variance

We here obtain the upper bound of the variance. We here calculate the term $\langle k_Q^2 \rangle$ because of the definition of the variance. Also, we again integrate the quantity over all unitary blocks in $U(\mathbf{x}, \boldsymbol{\theta})$. First, the expectation value over $W_{\kappa, L}$ is calculated as follows;

$$\begin{aligned} & \langle k_Q^2 \rangle_{W_{\kappa, L}} \\ &= \left\langle \text{Tr} \left[W_{\kappa, L} \rho_0^{(\kappa, L)} W_{\kappa, L}^\dagger \rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \text{Tr} \left[W_{\kappa, L} \rho_0^{(\kappa, L)} W_{\kappa, L}^\dagger \rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \right\rangle_{W_{\kappa, L}} \\ &= \left\langle \sum_{\mathbf{p}, \mathbf{q}, \mathbf{p}', \mathbf{q}'} \text{Tr} \left[W_{\kappa, L} \rho_{0, \mathbf{q}\mathbf{p}}^{(\kappa, L)} W_{\kappa, L}^\dagger \rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \right] \text{Tr} \left[W_{\kappa, L} \rho_{0, \mathbf{q}'\mathbf{p}'}^{(\kappa, L)} W_{\kappa, L}^\dagger \rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] \right\rangle_{W_{\kappa, L}} \\ &= \frac{1}{2^{2m} - 1} \sum_{\mathbf{p}, \mathbf{q}, \mathbf{p}', \mathbf{q}'} \left(\text{Tr} \left[\rho_{0, \mathbf{q}\mathbf{p}}^{(\kappa, L)} \right] \text{Tr} \left[\rho_{0, \mathbf{q}'\mathbf{p}'}^{(\kappa, L)} \right] \left(\text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \right] \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] - \frac{1}{2^m} \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] \right) \right. \\ & \quad \left. + \text{Tr} \left[\rho_{0, \mathbf{q}\mathbf{p}}^{(\kappa, L)} \rho_{0, \mathbf{q}'\mathbf{p}'}^{(\kappa, L)} \right] \left(\text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] - \frac{1}{2^m} \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \right] \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] \right) \right), \end{aligned} \quad (\text{A.8})$$

where $\rho_{0, \mathbf{q}\mathbf{p}}^{(\kappa, L)} = \text{Tr}_{\bar{S}_{(\kappa, L)}} \left[\left(|\mathbf{p}\rangle \langle \mathbf{q}| \otimes \mathbb{I}_{S_{(\kappa, L)}} \right) \rho_0^{(\kappa, L)} \right]$ and $\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} = \text{Tr}_{\bar{S}_{(\kappa, L)}} \left[\left(|\mathbf{q}\rangle \langle \mathbf{p}| \otimes \mathbb{I}_{S_{(\kappa, L)}} \right) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right]$. Here, we utilize Lemma 5 in the second equality and Lemma 3 in the last equality. We repeat

the procedure for the rest of the unitary blocks in the last layer and then we end up with

$$\begin{aligned}
\langle k_Q^2 \rangle_{V_L(\mathbf{x}, \boldsymbol{\theta})} &= \frac{1}{(2^{2m} - 1)^\kappa} \times \\
&\sum_{\mathbf{p}, \mathbf{q}, \mathbf{p}', \mathbf{q}'} \sum_{S_k \in P(S^{(1:\kappa-1, L)})} \prod_{h \in \bar{S}_k \cap S^{(1:\kappa-1, L)}} \left(\delta_{(\mathbf{p}\mathbf{q})_h} \delta_{(\mathbf{p}'\mathbf{q}')_h} - \frac{1}{2^m} \delta_{(\mathbf{p}\mathbf{q}')_h} \delta_{(\mathbf{p}'\mathbf{q})_h} \right) \prod_{h \in S_k} \left(\delta_{(\mathbf{p}\mathbf{q}')_h} \delta_{(\mathbf{p}'\mathbf{q})_h} - \frac{1}{2^m} \delta_{(\mathbf{p}\mathbf{q})_h} \delta_{(\mathbf{p}'\mathbf{q}')_h} \right) \\
&\times \left(\text{Tr} \left[\text{Tr}_{\bar{S}_k} \left[\rho_0^{(1, L)} \right] \text{Tr}_{\bar{S}_k} \left[\rho_0^{(1, L)} \right] \right] \left(\text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \right] \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] - \frac{1}{2^m} \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] \right) \right. \\
&\left. + \text{Tr} \left[\text{Tr}_{\bar{S}_k \cup S_{(\kappa, L)}} \left[\rho_0^{(1, L)} \right] \text{Tr}_{\bar{S}_k \cup S_{(\kappa, L)}} \left[\rho_0^{(1, L)} \right] \right] \left(\text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] - \frac{1}{2^m} \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \right] \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] \right) \right), \tag{A.9}
\end{aligned}$$

where $P(S^{(1:\kappa-1, L)}) = \{\emptyset, \{S_{(1, L)}\}, \{S_{(2, L)}\}, \dots, \{S_{(\kappa-1, L)}\}, \{S_{(1, L)}, S_{(2, L)}\}, \{S_{(1, L)}, S_{(3, L)}\}, \dots\}$ is the power set of $S^{(1:\kappa-1, L)} = \{S_{(1, L)}, S_{(2, L)}, \dots, S_{(\kappa-1, L)}\}$. We also define $\prod_{h=\emptyset}(\dots) \equiv 1$ and $\text{Tr}_{\emptyset}[\rho_0] \equiv \rho_0$. Here $\text{Tr}[\text{Tr}_{\bar{S}_k}[\rho_0^{(1, L)}] \text{Tr}_{\bar{S}_k}[\rho_0^{(1, L)}]]$ and $\text{Tr}[\text{Tr}_{\bar{S}_k \cup S_{(\kappa, L)}}[\rho_0^{(1, L)}] \text{Tr}_{\bar{S}_k \cup S_{(\kappa, L)}}[\rho_0^{(1, L)}]]$ are regarded as the purity of the quantum state $\rho_0^{(1, L)}$ which is partially traced over \bar{S}_k and $\bar{S}_k \cup S_{(\kappa, L)}$, respectively. We remind that $\rho_0^{(1, L)}$ is the quantum state obtained by applying the ALA up to $L - 1$ layer to the initial state, i.e., $\rho_0^{(1, L)} = (\prod_{d=1}^{L-1} V_d(\mathbf{x}, \boldsymbol{\theta})) \rho_0 (\prod_{d=1}^{L-1} V_d^\dagger(\mathbf{x}, \boldsymbol{\theta}))$. Hence, due to the inequality of the purity, i.e., $1/d \leq \text{Tr}[\rho^2] \leq 1$ with the d -dimensional quantum state ρ , we have

$$\begin{aligned}
\langle k_Q^2 \rangle_{U(\mathbf{x}, \boldsymbol{\theta})} &\leq \frac{1}{(2^{2m} - 1)^\kappa} \\
&\sum_{\mathbf{p}, \mathbf{q}, \mathbf{p}', \mathbf{q}'} \sum_{S_k \in P(S^{(1:\kappa-1, L)})} \prod_{h \in \bar{S}_k \cap S^{(1:\kappa-1, L)}} \left(\delta_{(\mathbf{p}\mathbf{q})_h} \delta_{(\mathbf{p}'\mathbf{q}')_h} - \frac{1}{2^m} \delta_{(\mathbf{p}\mathbf{q}')_h} \delta_{(\mathbf{p}'\mathbf{q})_h} \right) \prod_{h \in S_k} \left(\delta_{(\mathbf{p}\mathbf{q}')_h} \delta_{(\mathbf{p}'\mathbf{q})_h} - \frac{1}{2^m} \delta_{(\mathbf{p}\mathbf{q})_h} \delta_{(\mathbf{p}'\mathbf{q}')_h} \right) \\
&\times \left(\left(\text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \right] \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] - \frac{1}{2^m} \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] \right) \right. \\
&\left. + \left(\text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] - \frac{1}{2^m} \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \right] \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] \right) \right). \tag{A.10}
\end{aligned}$$

Further, using $\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} = \text{Tr}_{\bar{S}_{(\kappa, L)}} \left[\left(|\mathbf{q}\rangle \langle \mathbf{p}| \otimes \mathbb{I}_{S_{(\kappa, L)}} \right) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right]$ and the Kronecker delta regarding bit-strings $\mathbf{p}, \mathbf{q}, \mathbf{p}', \mathbf{q}'$, we can get the following equality.

$$\begin{aligned}
&\sum_{\mathbf{p}, \mathbf{q}, \mathbf{p}', \mathbf{q}'} \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}\mathbf{q}} \right] \text{Tr} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}'\mathbf{q}'} \right] \delta_{(\mathbf{p}\mathbf{q})_{S_k}} \delta_{(\mathbf{p}'\mathbf{q}')_{S_k}} \delta_{(\mathbf{p}\mathbf{q}')_{\bar{S}_k}} \delta_{(\mathbf{p}'\mathbf{q})_{\bar{S}_k}} \\
&= \sum_{\mathbf{p}, \mathbf{q}, \mathbf{p}', \mathbf{q}'} \text{Tr} \left[\left(|\mathbf{q}\rangle \langle \mathbf{p}| \otimes \mathbb{I}_{S_{(\kappa, L)}} \right) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \text{Tr} \left[\left(|\mathbf{q}'\rangle \langle \mathbf{p}'| \otimes \mathbb{I}_{S_{(\kappa, L)}} \right) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \delta_{(\mathbf{p}\mathbf{q})_{S_k}} \delta_{(\mathbf{p}'\mathbf{q}')_{S_k}} \delta_{(\mathbf{p}\mathbf{q}')_{\bar{S}_k}} \delta_{(\mathbf{p}'\mathbf{q})_{\bar{S}_k}} \\
&= \text{Tr} \left[\text{Tr}_{S_k \cup S_{(\kappa, L)}} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \text{Tr}_{S_k \cup S_{(\kappa, L)}} \left[\rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \right], \tag{A.11}
\end{aligned}$$

$$\begin{aligned}
& \sum_{\mathbf{p}, \mathbf{q}, \mathbf{p}', \mathbf{q}'} \text{Tr} [\rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p} \mathbf{q}} \rho_{\mathbf{x}', \boldsymbol{\theta}, \mathbf{p}' \mathbf{q}'}] \delta_{(\mathbf{p} \mathbf{q})_{S_k}} \delta_{(\mathbf{p}' \mathbf{q}')_{S_k}} \delta_{(\mathbf{p} \mathbf{q}')_{\bar{S}_k}} \delta_{(\mathbf{p}' \mathbf{q})_{\bar{S}_k}} \\
&= \sum_{\mathbf{p}, \mathbf{q}, \mathbf{p}', \mathbf{q}'} \text{Tr} \left[\text{Tr}_{\bar{S}_{(\kappa, L)}} \left[\left(|\mathbf{q}\rangle \langle \mathbf{p}| \otimes \mathbb{I}_{S_{(\kappa, L)}} \right) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \text{Tr}_{\bar{S}_{(\kappa, L)}} \left[\left(|\mathbf{q}'\rangle \langle \mathbf{p}'| \otimes \mathbb{I}_{S_{(\kappa, L)}} \right) \rho_{\mathbf{x}', \boldsymbol{\theta}} \right] \right] \\
& \hspace{20em} \times \delta_{(\mathbf{p} \mathbf{q})_{S_k}} \delta_{(\mathbf{p}' \mathbf{q}')_{S_k}} \delta_{(\mathbf{p} \mathbf{q}')_{\bar{S}_k}} \delta_{(\mathbf{p}' \mathbf{q})_{\bar{S}_k}} \\
&= \text{Tr} \left[\text{Tr}_{S_k} [\rho_{\mathbf{x}', \boldsymbol{\theta}}] \text{Tr}_{S_k} [\rho_{\mathbf{x}', \boldsymbol{\theta}}] \right].
\end{aligned} \tag{A.12}$$

This means that Eq. (A.10) can also be represented using the purity of quantum states. Therefore we have

$$\langle k_Q^2 \rangle_{(U(\mathbf{x}, \boldsymbol{\theta}), U(\mathbf{x}', \boldsymbol{\theta}))} = \langle k_Q^2 \rangle_{U(\mathbf{x}, \boldsymbol{\theta})} \leq \frac{2^\kappa}{(2^{2m} - 1)^\kappa}, \tag{A.13}$$

where we use $\text{Tr}[\rho^2] \leq 1$. Also, we assume $2^m \gg 1$ here. Thus, the upper bound of the fidelity-based quantum kernel using the ALA is described as

$$\text{Var} [k_Q] \leq \frac{2^\kappa}{(2^{2m} - 1)^\kappa} - \frac{1}{2^{2n}} \approx \frac{1}{2^{n(2 - \frac{1}{m})}}. \tag{A.14}$$

This result is valid for the case where a mixed state is used as the initial state since the upper bound is derived using the purity of quantum states.

A.2 Proof of Theorem 1

We provide the proof of Theorem 1. Here, we analytically calculate the expectation value of the variance for the i -th term of the quantum Fisher kernel (QFK), i.e.,

$$k_{QF}^{(i)} \equiv \text{Tr}[\rho_0 \{ \tilde{B}_{\mathbf{x}, \theta_i}, \tilde{B}_{\mathbf{x}', \theta_i} \}] / 2. \tag{A.15}$$

where $\tilde{B}_{\mathbf{x}, \theta_i} = U_{1:i}^\dagger(\mathbf{x}, \boldsymbol{\theta}) B_{\theta_i} U_{1:i}(\mathbf{x}, \boldsymbol{\theta})$ with $B_{\mathbf{x}, \theta_i} = 2i(\partial_{\theta_i} U(\mathbf{x}, \boldsymbol{\theta})) U^\dagger(\mathbf{x}, \boldsymbol{\theta})$.

In addition, we present that the variance scaling of the i -th term is the same even if $\text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}]$ and $\text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}]$ are taken into account, i.e.,

$$k_{QF'}^{(i)} \equiv \text{Tr}[\rho_0 \{ \tilde{B}_{\mathbf{x}, \theta_i} - \text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}], \tilde{B}_{\mathbf{x}', \theta_i} - \text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}] \}] / 2. \tag{A.16}$$

That is, these terms hold the following equality;

$$\text{Var} [k_{QF'}^{(i)}] = \text{Var} [k_{QF}^{(i)}] + \text{Var} [T^{(i)}] + 2\text{Cov} [k_{QF}^{(i)}, T^{(i)}] \tag{A.17}$$

with $T^{(i)} = -\text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}]$. Here, $\text{Cov}[A, B] = \langle AB \rangle - \langle A \rangle \langle B \rangle$ denotes the covariance of A and B . Therefore, we show that terms $\text{Var}[T^{(i)}] + 2\text{Cov}[k_{QF}^{(i)}, T^{(i)}]$ do not contribute the scaling of variance so much later.

A.2.1 Case (1): Globally-Random Quantum Circuits

Expectation Value

We compute the expectation value of the i -th term of the QFK, assuming either $U_{1:i}(\mathbf{x}, \boldsymbol{\theta})$ or $U_{1:i}(\mathbf{x}', \boldsymbol{\theta})$ is a 1-design. Without loss of generality, we assume $U_{1:i}(\mathbf{x}, \boldsymbol{\theta})$ is a 1-design due to the

symmetry. Also we denote the unitary operators $U_{1:i} \equiv U_{1:i}(\mathbf{x}, \boldsymbol{\theta})$ and $U'_{1:i} \equiv U_{1:i}(\mathbf{x}', \boldsymbol{\theta})$. Then, we obtain

$$\begin{aligned}
\langle k_{QF}^{(i)} \rangle_{(U_{1:i}, U'_{1:i})} &= \frac{1}{2} \left\langle \text{Tr} \left[\rho_0 U_{1:i}^\dagger B_{\theta_i} U_{1:i} \tilde{B}_{\mathbf{x}', \theta_i} \right] \right\rangle_{(U_{1:i}, U'_{1:i})} + \frac{1}{2} \left\langle \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} U_{1:i}^\dagger B_{\theta_i} U_{1:i} \right] \right\rangle_{(U_{1:i}, U'_{1:i})} \\
&= \frac{1}{2} \left\langle \text{Tr} \left[U_{1:i}^\dagger B_{\theta_i} U_{1:i} \tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \right\rangle_{(U_{1:i}, U'_{1:i})} + \frac{1}{2} \left\langle \text{Tr} \left[U_{1:i}^\dagger B_{\theta_i} U_{1:i} \rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \right] \right\rangle_{(U_{1:i}, U'_{1:i})} \\
&= \frac{1}{2 \cdot 2^n} \text{Tr} [B_{\theta_i}] \left\langle \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \right\rangle_{U'_{1:i}} + \frac{1}{2 \cdot 2^n} \left\langle \text{Tr} [B_{\theta_i}] \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \right] \right\rangle_{U'_{1:i}} \\
&= 0,
\end{aligned} \tag{A.18}$$

where Lemma 1 and the traceless property of the Pauli operators are utilized. As

$$\left\langle \text{Tr} \left[\rho_0 U_{1:i}^\dagger B_{\theta_i} U_{1:i} \right] \right\rangle_{U_{1:i}} = 0 \tag{A.19}$$

using Lemma 1, we can also show that $\langle k_{QF'}^{(i)} \rangle = 0$.

Variance

We compute the term $\langle k_{QF}^{(i)2} \rangle$, as the variance is equal to this term. Because of the independence of $U_{1:i}$ and $U'_{1:i}$, we first calculate the expectation value over $U_{1:i}$. The expectation value consists of three terms;

$$\begin{aligned}
\text{Var}[k_{QF}^{(i)}] &= \langle k_{QF}^{(i)2} \rangle_{U_{1:i}} \\
&= \frac{1}{4} \left\langle \left(\text{Tr} \left[\rho_0 U_{1:i}^\dagger B_{\theta_i} U_{1:i} \tilde{B}_{\mathbf{x}', \theta_i} \right] + \text{Tr} \left[U_{1:i}^\dagger B_{\theta_i} U_{1:i} \rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \right] \right)^2 \right\rangle_{U_{1:i}} \\
&= \frac{1}{4} \left\langle \text{Tr} \left[\rho_0 U_{1:i}^\dagger B_{\theta_i} U_{1:i} \tilde{B}_{\mathbf{x}', \theta_i} \right] \text{Tr} \left[\rho_0 U_{1:i}^\dagger B_{\theta_i} U_{1:i} \tilde{B}_{\mathbf{x}', \theta_i} \right] \right\rangle_{U_{1:i}} \\
&\quad + \frac{1}{2} \left\langle \text{Tr} \left[\rho_0 U_{1:i}^\dagger B_{\theta_i} U_{1:i} \tilde{B}_{\mathbf{x}', \theta_i} \right] \text{Tr} \left[U_{1:i}^\dagger B_{\theta_i} U_{1:i} \rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \right] \right\rangle_{U_{1:i}} \\
&\quad + \frac{1}{4} \left\langle \text{Tr} \left[U_{1:i}^\dagger B_{\theta_i} U_{1:i} \rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \right] \text{Tr} \left[U_{1:i}^\dagger B_{\theta_i} U_{1:i} \rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \right] \right\rangle_{U_{1:i}} \\
&= \text{Var}_{r,1} + \text{Var}_{r,2} + \text{Var}_{r,3},
\end{aligned} \tag{A.20}$$

where $\text{Var}_{r,i}$ represents the i -th term of the right-hand side of the second equality. Thus, we independently calculate these terms.

The first term can be obtained as

$$\begin{aligned}
\text{Var}_{r,1} &= \frac{1}{4} \left\langle \text{Tr} \left[\rho_0 U_{1:i}^\dagger B_{\theta_i} U_{1:i} \tilde{B}_{\mathbf{x}', \theta_i} \right] \text{Tr} \left[\rho_0 U_{1:i}^\dagger B_{\theta_i} U_{1:i} \tilde{B}_{\mathbf{x}', \theta_i} \right] \right\rangle_{U_{1:i}} \\
&= \frac{1}{4 \cdot (2^{2n} - 1)} \left(\text{Tr} [B_{\theta_i}] \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \text{Tr} [B_{\theta_i}] \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] + \text{Tr} [B_{\theta_i}^2] \text{Tr} \left[\left(\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right)^2 \right] \right) \\
&\quad - \frac{1}{4 \cdot 2^n (2^{2n} - 1)} \left(\text{Tr} [B_{\theta_i}] \text{Tr} [B_{\theta_i}] \text{Tr} \left[\left(\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right)^2 \right] + \text{Tr} [B_{\theta_i}^2] \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \right) \\
&= \frac{2^n}{4 \cdot (2^{2n} - 1)} \left(\text{Tr} \left[\left(\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right)^2 \right] - \frac{1}{2^n} \left(\text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \right)^2 \right) \\
&= \frac{2^n}{4 \cdot (2^{2n} - 1)} \left(1 - \frac{1}{2^n} \right) \left(\text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \right)^2,
\end{aligned} \tag{A.21}$$

where we exploit Lemma 3 and the properties of the Pauli operators, $\text{Tr}[B] = 0$ and $\text{Tr}[B^2] = 2^n$. In addition, we use the equality $\text{Tr}[(\tilde{B}_{\mathbf{x}',\theta_i}\rho_0)^2] = (\text{Tr}[\tilde{B}_{\mathbf{x}',\theta_i}\rho_0])^2$ due to the assumption that the initial state is pure.

Similarly, the second and the third terms are calculated in the following way.

$$\text{Var}_{r,2} = \frac{2^n}{2 \cdot (2^{2n} - 1)} \left(\text{Tr} [\tilde{B}_{\mathbf{x}',\theta_i}^2 \rho_0] - \frac{1}{2^n} \left(\text{Tr} [\tilde{B}_{\mathbf{x}',\theta_i} \rho_0] \right)^2 \right), \quad (\text{A.22})$$

$$\text{Var}_{r,3} = \frac{2^n}{4 \cdot (2^{2n} - 1)} \left(1 - \frac{1}{2^n} \right) \left(\text{Tr} [\tilde{B}_{\mathbf{x}',\theta_i} \rho_0] \right)^2. \quad (\text{A.23})$$

Consequently, we obtain

$$\begin{aligned} \langle k_{QF}^{(i)2} \rangle_{U_{1:i}} &= \text{Var}_{r,1} + \text{Var}_{r,2} + \text{Var}_{r,3} \\ &= \frac{2^n}{2^{2n} - 1} \cdot \frac{1}{2} \left(\left(1 - \frac{1}{2^n} \right) \left(\text{Tr} [\tilde{B}_{\mathbf{x}',\theta_i} \rho_0] \right)^2 + \left(\text{Tr} [\tilde{B}_{\mathbf{x}',\theta_i}^2 \rho_0] - \frac{1}{2^n} \left(\text{Tr} [\tilde{B}_{\mathbf{x}',\theta_i} \rho_0] \right)^2 \right) \right). \end{aligned} \quad (\text{A.24})$$

Next, we integrate the quantity over $U'_{1:i}$. As $U'_{1:i}$ is involved in $\text{Tr}[\tilde{B}_{\mathbf{x}',\theta_i}^2 \rho_0]$ and $(\text{Tr}[\tilde{B}_{\mathbf{x}',\theta_i} \rho_0])^2$ in Eq. (A.24), we consider these terms. The expectation values of these terms are calculated as

$$\begin{aligned} \langle \text{Tr} [\tilde{B}_{\mathbf{x}',\theta_i}^2 \rho_0] \rangle_{U'_{1:i}} &= \langle \text{Tr} [U_{1:i}^\dagger B_{\theta_i}^2 U'_{1:i} \rho_0] \rangle_{U'_{1:i}} \\ &= \frac{1}{2^n} \text{Tr} [B_{\theta_i}^2] \text{Tr} [\rho_0] \\ &= 1, \end{aligned} \quad (\text{A.25})$$

$$\begin{aligned} \left\langle \left(\text{Tr} [\tilde{B}_{\mathbf{x}',\theta_i} \rho_0] \right)^2 \right\rangle_{U'_{1:i}} &= \left\langle \text{Tr} [U_{1:i}^\dagger B_{\theta_i} U'_{1:i} \rho_0] \text{Tr} [U_{1:i}^\dagger B_{\theta_i} U'_{1:i} \rho_0] \right\rangle_{U'_{1:i}} \\ &= \frac{1}{2^{2n} - 1} \left(\text{Tr} [B_{\theta_i}] \text{Tr} [\rho_0] \text{Tr} [B_{\theta_i}] \text{Tr} [\rho_0] + \text{Tr} [B_{\theta_i}^2] \text{Tr} [\rho_0^2] \right) \\ &\quad - \frac{1}{2^n (2^{2n} - 1)} \left(\text{Tr} [B_{\theta_i}] \text{Tr} [B_{\theta_i}] \text{Tr} [\rho_0^2] + \text{Tr} [B_{\theta_i}^2] \text{Tr} [\rho_0] \text{Tr} [\rho_0] \right) \\ &= \frac{1}{2^{2n} - 1} (2^n - 1) \\ &= \frac{1}{2^n + 1}. \end{aligned} \quad (\text{A.26})$$

Here, we utilize Lemmas 1 and 3 and the property of the Pauli operators and the pure state. Therefore, substituting the terms into Eq. (A.24), we have

$$\text{Var}[k_{QF}^{(i)}] = \langle k_{QF}^{(i)2} \rangle_{(U_{1:i}, U'_{1:i})} = \frac{2^n}{2(2^{2n} - 1)} \left(1 + \frac{2^n - 2}{2^n(2^n + 1)} \right) \approx \frac{1}{2^{n+1}}. \quad (\text{A.27})$$

Also, we compute $\text{Var}[T^{(i)}]$ and $2\text{Cov}[k_{QF}^{(i)}, T^{(i)}]$ to obtain $\text{Var}[k_{QF}^{(i)}]$. We remind the readers

that $T^{(i)} = -\text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr}[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}]$. As $\langle T^{(i)} \rangle = 0$ utilizing Eq. (A.2.1), we can obtain

$$\begin{aligned}
\text{Var} [T^{(i)}] &= \left\langle \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}] \right\rangle_{(U_{1:i}, U'_{1:i})} \\
&= \left\langle \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \right\rangle_{U_{1:i}} \left\langle \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}] \right\rangle_{U'_{1:i}} \\
&= \left(\frac{2^n}{2^{2n} - 1} \left(1 - \frac{1}{2^n} \right) \right)^2 \\
&= \frac{1}{(2^n + 1)^2}.
\end{aligned} \tag{A.28}$$

As for the covariance term, we only focus on the term $\langle k_{QF}^{(i)}, T^{(i)} \rangle$ as the remaining term is zero. Thus, we can compute

$$\begin{aligned}
\text{Cov} [k_{QF}^{(i)}, T^{(i)}] &= -\frac{1}{2} \left\langle \text{Tr} [\rho_0 \{ \tilde{B}_{\mathbf{x}, \theta_i}, \tilde{B}_{\mathbf{x}', \theta_i} \}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}] \right\rangle_{(U_{1:i}, U'_{1:i})} \\
&= -\frac{1}{2} \left\langle \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i} \tilde{B}_{\mathbf{x}', \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}] \right\rangle_{(U_{1:i}, U'_{1:i})} \\
&\quad - \frac{1}{2} \left\langle \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}] \right\rangle_{(U_{1:i}, U'_{1:i})} \\
&= -\frac{1}{2} (\text{Cov}_{r,1} + \text{Cov}_{r,2}).
\end{aligned} \tag{A.29}$$

These terms respectively read

$$\begin{aligned}
\text{Cov}_{r,1} &= \left\langle \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i} \tilde{B}_{\mathbf{x}', \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}] \right\rangle_{(U_{1:i}, U'_{1:i})} \\
&= \frac{1}{(2^n + 1)^2},
\end{aligned} \tag{A.30}$$

$$\begin{aligned}
\text{Cov}_{r,2} &= \left\langle \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_i}] \right\rangle_{(U_{1:i}, U'_{1:i})} \\
&= \frac{1}{(2^n + 1)^2},
\end{aligned} \tag{A.31}$$

where we utilize Lemma 3 and the property of the pure states, i.e., $\rho^2 = \rho$. Consequently, we have

$$\begin{aligned}
\text{Cov} [k_{QF}^{(i)}, T^{(i)}] &= -\frac{1}{2} (\text{Cov}_{r,1} + \text{Cov}_{r,2}) \\
&= -\frac{1}{(2^n + 1)^2}.
\end{aligned} \tag{A.32}$$

Therefore, the variance of $k_{QF}^{(i)}$ is

$$\begin{aligned}
\text{Var} [k_{QF}^{(i)}] &= \text{Var} [k_{QF}^{(i)}] + \text{Var} [T^{(i)}] + 2\text{Cov} [k_{QF}^{(i)}, T^{(i)}] \\
&= \frac{2^n}{2(2^{2n} - 1)} \left(1 + \frac{2^n - 2}{2^n(2^n + 1)} \right) + \frac{1}{(2^n + 1)^2} - \frac{2}{(2^n + 1)^2} \\
&= \frac{2^n}{2(2^{2n} - 1)} \left(1 - \frac{1}{2^n + 1} \right) \\
&\approx \frac{1}{2^{n+1}},
\end{aligned} \tag{A.33}$$

which indicates the same scaling as $\text{Var} [k_{QF}^{(i)}]$.

A.2.2 Case (2): Alternating Layered Ansatzes

Expectation Value

Assuming $\tilde{W}_{k,d}(\mathbf{x}, \theta_i)$, $\tilde{W}_{k,d}(\mathbf{x}', \theta_i)$ and all unitary blocks in the light-cones are t -designs, we compute the expectation value of $k_{QF}^{(i)}$. For simplicity, we denote unitary operators $\tilde{W}_{k,d} \equiv \tilde{W}_{k,d}(\mathbf{x}, \theta_i)$, $\tilde{W}'_{k,d} \equiv \tilde{W}_{k,d}(\mathbf{x}', \theta_i)$, $V_r \equiv V_r(\mathbf{x}, \boldsymbol{\theta})$ and $V'_r \equiv V_r(\mathbf{x}', \boldsymbol{\theta})$. Then, due to the independence of these unitary operators, integrating the term over $\tilde{W}_{k,d}$ leads to

$$\begin{aligned}
\langle k_{QF}^{(i)} \rangle_{\tilde{W}_{k,d}} &= \frac{1}{2} \left\langle \text{Tr} \left[\rho_0 V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \tilde{W}_{k,d} V_r \tilde{B}_{\mathbf{x}', \theta_i} \right] \right\rangle_{\tilde{W}_{k,d}} + \frac{1}{2} \left\langle \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \tilde{W}_{k,d} V_r \right] \right\rangle_{\tilde{W}_{k,d}} \\
&= \frac{1}{2} \left\langle \text{Tr} \left[\tilde{W}_{k,d} V_r \tilde{B}_{\mathbf{x}', \theta_i} \rho_0 V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \right] \right\rangle_{\tilde{W}_{k,d}} + \frac{1}{2} \left\langle \text{Tr} \left[\tilde{W}_{k,d} V_r \rho_0 \tilde{B}_{\mathbf{x}', \theta_i} V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \right] \right\rangle_{\tilde{W}_{k,d}} \\
&= \frac{1}{2 \cdot 2^m} \text{Tr} \left[\text{Tr}_{S(k,d)} \left[V_r \rho_0 \tilde{B}_{\mathbf{x}', \theta_i} V_r^\dagger \right] \text{Tr}_{S(k,d)} [B_{\theta_i}] \right] \\
&\quad + \frac{1}{2 \cdot 2^m} \text{Tr} \left[\text{Tr}_{S(k,d)} \left[V_r \rho_0 \tilde{B}_{\mathbf{x}', \theta_i} V_r^\dagger \right] \text{Tr}_{S(k,d)} [B_{\theta_i}] \right] \\
&= 0,
\end{aligned} \tag{A.34}$$

where $\text{Tr}_{S(k,d)}[\cdot]$ represents a partial trace over the space $S(k,d)$ on which $\tilde{W}_{k,d}$ acts. Also, we utilize Lemma 4 and the traceless property of the Pauli operators. This means that the expectation value $\langle k_{QF}^{(i)} \rangle$ is zero irrespective of the remaining unitary blocks in $U_{1:i}$, $\tilde{W}'_{k,d}$ and $U'_{1:i}$. Similarly, we can obtain

$$\left\langle \text{Tr} \left[\rho_0 U_{1:i}^\dagger B_{\theta_i} U_{1:i} \right] \right\rangle_{\tilde{W}_{k,d}} = 0 \tag{A.35}$$

using Lemma 1, and thus $\langle k_{QF'}^{(i)} \rangle = 0$.

Variance

The expectation value of $k_{QF}^{(i)}$ is zero as shown above, and thus the variance is equivalent to $\langle k_{QF}^{(i)2} \rangle$. As we assume unitary operators are independent, we integrate the term over the unitary operators in the following order; $\tilde{W}_{k,d}$, V_r , $\tilde{W}'_{k,d}$ and V'_r . We first work on the integration over $\tilde{W}_{k,d}$. Then, the term can be decomposed into three terms as follows; Then, we have

$$\begin{aligned}
\langle k_{QF}^{(i)2} \rangle_{\tilde{W}_{k,d}} &= \frac{1}{4} \left\langle \left(\text{Tr} \left[\rho_0 V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \tilde{W}_{k,d} V_r \tilde{B}_{\mathbf{x}', \theta_i} \right] + \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \tilde{W}_{k,d} V_r \right] \right)^2 \right\rangle_{\tilde{W}_{k,d}} \\
&= \frac{1}{4} \left\langle \text{Tr} \left[\rho_0 V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \tilde{W}_{k,d} V_r \tilde{B}_{\mathbf{x}', \theta_i} \right] \text{Tr} \left[\rho_0 V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \tilde{W}_{k,d} V_r \tilde{B}_{\mathbf{x}', \theta_i} \right] \right\rangle_{\tilde{W}_{k,d}} \\
&\quad + \frac{1}{2} \left\langle \text{Tr} \left[\rho_0 V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \tilde{W}_{k,d} V_r \tilde{B}_{\mathbf{x}', \theta_i} \right] \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \tilde{W}_{k,d} V_r \right] \right\rangle_{\tilde{W}_{k,d}} \\
&\quad + \frac{1}{4} \left\langle \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \tilde{W}_{k,d} V_r \right] \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} V_r^\dagger \tilde{W}_{k,d}^\dagger B_{\theta_i} \tilde{W}_{k,d} V_r \right] \right\rangle_{\tilde{W}_{k,d}}. \\
&= \text{Var}_{a,1} + \text{Var}_{a,2} + \text{Var}_{a,3},
\end{aligned} \tag{A.36}$$

where $\text{Var}_{a,i}$ is the i -th term of the right-hand side of the second equality.

We first focus on the first term $\text{Var}_{a,1}$. The integration of the term over $\tilde{W}_{k,d}$ results in

$$\begin{aligned}
\text{Var}_{a,1} &= \frac{1}{4} \left\langle \text{Tr} \left[\tilde{W}_{k,d} \tilde{\rho}_{0,B_l}^{(1)} \tilde{W}_{k,d}^\dagger B_{\theta_i} \right] \text{Tr} \left[\tilde{W}_{k,d} \tilde{\rho}_{0,B_l}^{(1)} \tilde{W}_{k,d}^\dagger B_{\theta_i} \right] \right\rangle_{\tilde{W}_{k,d}} \\
&= \frac{1}{4} \sum_{\mathbf{p}, \mathbf{q}, \mathbf{p}', \mathbf{q}'} \left\langle \text{Tr} \left[\tilde{W}_{k,d} \tilde{\rho}_{0,B_l, \mathbf{qp}}^{(1)} \tilde{W}_{k,d}^\dagger B_{\theta_i, \mathbf{pq}} \right] \text{Tr} \left[\tilde{W}_{k,d} \tilde{\rho}_{0,B_l, \mathbf{q'p}'}^{(1)} \tilde{W}_{k,d}^\dagger B_{\theta_i, \mathbf{p'q}'} \right] \right\rangle_{\tilde{W}_{k,d}} \\
&= \frac{1}{4} \cdot \frac{2^m}{2^{2m} - 1} \sum_{\mathbf{p}, \mathbf{p}'} \left(\text{Tr} \left[\tilde{\rho}_{0,B_l, \mathbf{pp}}^{(1)} \tilde{\rho}_{0,B_l, \mathbf{p'p}'}^{(1)} \right] - \frac{1}{2^m} \text{Tr} \left[\tilde{\rho}_{0,B_l, \mathbf{pp}}^{(1)} \right] \text{Tr} \left[\tilde{\rho}_{0,B_l, \mathbf{p'p}'}^{(1)} \right] \right),
\end{aligned} \tag{A.37}$$

where we define $\tilde{\rho}_{0,B_l, \mathbf{qp}}^{(1)} = \text{Tr}_{\tilde{S}_{(k,d)}} [(|\mathbf{p}\rangle \langle \mathbf{q}| \otimes \mathbb{I}_{S_{(k,d)}}) \tilde{\rho}_{0,B_l}^{(1)}]$ with $\tilde{\rho}_{0,B_l}^{(1)} = V_r \tilde{B}_{\mathbf{x}', \theta_i} \rho_0 V_r^\dagger$ and $B_{\theta_i, \mathbf{pq}} = \text{Tr}_{\tilde{S}_{(k,d)}} [(|\mathbf{q}\rangle \langle \mathbf{p}| \otimes \mathbb{I}_{S_{(k,d)}}) B_{\theta_i}]$. Here the following two equalities are utilized;

$$\begin{aligned}
\text{Tr} [B_{\theta_i, \mathbf{pq}}] &= \text{Tr} \left[\text{Tr}_{\tilde{S}_{(k,d)}} \left[(|\mathbf{q}\rangle \langle \mathbf{p}| \otimes \mathbb{I}_{S_{(k,d)}}) B_{\theta_i} \right] \right] = 0, \\
\text{Tr} [B_{\theta_i, \mathbf{pq}} B_{\theta_i, \mathbf{p'q}'}] &= \text{Tr} \left[\text{Tr}_{\tilde{S}_{(k,d)}} \left[(|\mathbf{q}\rangle \langle \mathbf{p}| \otimes \mathbb{I}_{S_{(k,d)}}) B_{\theta_i} \right] \text{Tr}_{\tilde{S}_{(k,d)}} \left[(|\mathbf{q}'\rangle \langle \mathbf{p}'| \otimes \mathbb{I}_{S_{(k,d)}}) B_{\theta_i} \right] \right] \\
&= \delta_{(\mathbf{p}, \mathbf{q})} \delta_{(\mathbf{p}', \mathbf{q}')} \text{Tr} [B_{\theta_i}^2] \\
&= \delta_{(\mathbf{p}, \mathbf{q})} \delta_{(\mathbf{p}', \mathbf{q}')} 2^m.
\end{aligned} \tag{A.38}$$

The first and second terms in the last equality of Eq. (A.37) can be rewritten as

$$\sum_{\mathbf{p}, \mathbf{p}'} \text{Tr} \left[\tilde{\rho}_{0,B_l, \mathbf{pp}}^{(1)} \right] \text{Tr} \left[\tilde{\rho}_{0,B_l, \mathbf{p'p}'}^{(1)} \right] = \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \tag{A.39}$$

and

$$\sum_{\mathbf{p}, \mathbf{p}'} \text{Tr} [\tilde{\rho}_{0,B_l, \mathbf{pp}}^{(1)} \tilde{\rho}_{0,B_l, \mathbf{p'p}'}^{(1)}] = \text{Tr} \left[\text{Tr}_{\tilde{S}_{(k,d)}} [V_r \tilde{B}_{\mathbf{x}', \theta_i} \rho_0 V_r^\dagger] \text{Tr}_{\tilde{S}_{(k,d)}} [V_r \tilde{B}_{\mathbf{x}', \theta_i} \rho_0 V_r^\dagger] \right], \tag{A.40}$$

respectively. This indicates that V_r can be excluded from the expectation value calculation for the quantity in Eq. (A.39), but not from the calculation for the other in Eq. (A.40).

Then we integrate the second quantity in Eq. (A.40) over the unitary V_r . We remind that V_r contains all unitary blocks in the light-cone of $W_{k,d}$. Hence, the quantity is iteratively integrated over every unitary block. To do so, we consider the following situations: (1) a unitary block w_s acting on (1) a subspace of S' , (2) a subspace of \bar{S}' , (3) a subspace of both S' and \bar{S}' and (4) S' and a subspace of \bar{S}' . Then, for arbitrary operator $A : S' \otimes \bar{S}' \rightarrow S' \otimes \bar{S}'$, the expectation value of $\text{Tr}[\text{Tr}_{\bar{S}'} [w_s A w_s^\dagger] \text{Tr}_{\bar{S}'} [w_s A w_s^\dagger]]$ over $w_s : S_s \rightarrow S_s$ can be obtained as follows;

(1) $S_s \subseteq S'$

$$\begin{aligned}
\left\langle \text{Tr} \left[\text{Tr}_{\bar{S}'} \left[w_s A w_s^\dagger \right] \text{Tr}_{\bar{S}'} \left[w_s A w_s^\dagger \right] \right] \right\rangle_{w_s} &= \left\langle \text{Tr} \left[w_s \text{Tr}_{\bar{S}'} [A] w_s^\dagger w_s \text{Tr}_{\bar{S}'} [A] w_s^\dagger \right] \right\rangle_{w_s} \\
&= \text{Tr} [\text{Tr}_{\bar{S}'} [A] \text{Tr}_{\bar{S}'} [A]]
\end{aligned} \tag{A.41}$$

(2) $S_s \subseteq \bar{S}'$

$$\begin{aligned}
\left\langle \text{Tr} \left[\text{Tr}_{\bar{S}'} \left[w_s A w_s^\dagger \right] \text{Tr}_{\bar{S}'} \left[w_s A w_s^\dagger \right] \right] \right\rangle_{w_s} &= \left\langle \text{Tr} \left[\text{Tr}_{\bar{S}'} \left[A w_s^\dagger w_s \right] \text{Tr}_{\bar{S}'} \left[A w_s^\dagger w_s \right] \right] \right\rangle_{w_s} \\
&= \text{Tr} [\text{Tr}_{\bar{S}'} [A] \text{Tr}_{\bar{S}'} [A]]
\end{aligned} \tag{A.42}$$

(3) $S_s = S_h \otimes S_{\bar{h}}$ with $d^{1/2}$ -dimensional spaces $S_h \subset S'$ and $S_{\bar{h}} \subset \bar{S}'$

$$\begin{aligned}
& \left\langle \text{Tr} \left[\text{Tr}_{\bar{S}'} \left[w_s A w_s^\dagger \right] \text{Tr}_{\bar{S}'} \left[w_s A w_s^\dagger \right] \right] \right\rangle_{w_s} \\
&= \left\langle \text{Tr} \left[\left(w_s A w_s^\dagger \otimes w_s A w_s^\dagger \right) \left(\text{Swap}_{S'_1 \otimes S'_2} \otimes \mathbb{I}_{\bar{S}'_1 \otimes \bar{S}'_2} \right) \right] \right\rangle_{w_s} \\
&= \frac{1}{d^2 - 1} \left(\text{Tr} \left[\left(\mathbb{I}_{S_{s,1} \otimes S_{s,2}} \otimes \text{Tr}_{S_{s,1}} [A] \otimes \text{Tr}_{S_{s,2}} [A] \right) \left(\text{Swap}_{S'_1 \otimes S'_2} \otimes \mathbb{I}_{\bar{S}'_1 \otimes \bar{S}'_2} \right) \right] \right. \\
&\quad \left. + \text{Tr} \left[\left(\text{Swap}_{S_{s,1} \otimes S_{s,2}} \otimes \text{Tr}_{S_{s,1} \cup S_{s,2}} \left[A \otimes A \left(\text{Swap}_{S_{s,1} \otimes S_{s,2}} \otimes \mathbb{I}_{S_{s,1}^- \otimes S_{s,2}^-} \right) \right] \right) \left(\text{Swap}_{S'_1 \otimes S'_2} \otimes \mathbb{I}_{\bar{S}'_1 \otimes \bar{S}'_2} \right) \right] \right) \\
&\quad - \frac{1}{d(d^2 - 1)} \left(\text{Tr} \left[\left(\mathbb{I}_{S_{s,1} \otimes S_{s,1}} \otimes \text{Tr}_{S_{s,1} \cup S_{s,2}} \left[A \otimes A \left(\text{Swap}_{S_{s,1} \otimes S_{s,2}} \otimes \mathbb{I}_{S_{s,1}^- \otimes S_{s,2}^-} \right) \right] \right) \left(\text{Swap}_{S'_1 \otimes S'_2} \otimes \mathbb{I}_{\bar{S}'_1 \otimes \bar{S}'_2} \right) \right] \right) \\
&\quad \left. + \text{Tr} \left[\left(\text{Swap}_{S_{s,1} \otimes S_{s,2}} \otimes \text{Tr}_{S_{s,1}} [A] \otimes \text{Tr}_{S_{s,2}} [A] \right) \left(\text{Swap}_{S'_1 \otimes S'_2} \otimes \mathbb{I}_{\bar{S}'_1 \otimes \bar{S}'_2} \right) \right] \right) \\
&= \frac{d^{1/2}}{d+1} \left(\text{Tr} \left[\text{Tr}_{\bar{S}' \cup S_h} [A] \text{Tr}_{\bar{S}' \cup S_h} [A] \right] + \text{Tr} \left[\text{Tr}_{\bar{S}' \setminus S_{\bar{h}}} [A] \text{Tr}_{\bar{S}' \setminus S_{\bar{h}}} [A] \right] \right)
\end{aligned} \tag{A.43}$$

(4) $S_s = S' \otimes S_{\bar{h}}$ with $d^{1/2}$ -dimensional spaces $S' \subset S'$ and $S_{\bar{h}} \subset \bar{S}'$

$$\begin{aligned}
\left\langle \text{Tr} \left[\text{Tr}_{\bar{S}'} \left[w_s A w_s^\dagger \right] \text{Tr}_{\bar{S}'} \left[w_s A w_s^\dagger \right] \right] \right\rangle_{w_s} &= \left\langle \text{Tr} \left[\left(w_s A w_s^\dagger \otimes w_s A w_s^\dagger \right) \left(\text{Swap}_{S'_1 \otimes S'_2} \otimes \mathbb{I}_{\bar{S}'_1 \otimes \bar{S}'_2} \right) \right] \right\rangle_{w_s} \\
&= \frac{d^{1/2}}{d+1} \left(\text{Tr} [A] \text{Tr} [A] + \text{Tr} \left[\text{Tr}_{\bar{S}' \setminus S_{\bar{h}}} [A] \text{Tr}_{\bar{S}' \setminus S_{\bar{h}}} [A] \right] \right)
\end{aligned} \tag{A.44}$$

Here, $\mathbb{I}_{S_1 \otimes S_2}$ and $\text{Swap}_{S_1 \otimes S_2}$ denote the identity operator and the swap operator acting on the systems S_1, S_2 , respectively. Also, the subspace labeled with the number in the subscript (for example, $S_{s,i}$ with $i \in \{1, 2\}$) represents one of the duplicated subsystems. Thus, the following result can be obtained;

$$\begin{aligned}
& \left\langle \text{Tr} \left[\text{Tr}_{\bar{S}(k,d)} [V_r \tilde{B}_{\mathbf{x}', \theta_i} \rho_0 V_r^\dagger] \text{Tr}_{\bar{S}(k,d)} [V_r(\mathbf{x}, \boldsymbol{\theta}) \tilde{B}_{\mathbf{x}', \theta_i} \rho_0 V_r^\dagger] \right] \right\rangle_{V_r} \\
&= \sum_{h \in P_U(S^{(k_s: k_l, 1)})} t_h \text{Tr} \left[\text{Tr}_{\bar{h}} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0] \text{Tr}_{\bar{h}} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0] \right],
\end{aligned} \tag{A.45}$$

where $t_h \in \mathbb{R}^+$ and $P_U(S^{(k_s: k_l, 1)}) = \{\emptyset, S_{(k_s, 1)}, S_{(k_s+1, 1)}, \dots, S_{(k_l, 1)}, S_{(k_s, 1)} \cup S_{(k_s+1, 1)}, \dots\}$ is the set of subspace, each element of which is the union of the spaces in a subset of $P(S^{(k_s: k_l, 1)})$. Here, $k_s(k_l)$ is the smallest (largest) label of the unitary blocks in the first layer of $V_r(\mathbf{x}, \boldsymbol{\theta})$. Note that every t_h is equal to or greater than $(2^{\frac{m}{2}}/2^m + 1)^{2(d-1)}$. Then, the expectation value of the first term over $U_{1:i}$ can be written as

$$\frac{1}{4} \cdot \frac{2^m}{2^{2m} - 1} \left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} t_h \text{Tr} \left[\text{Tr}_{\bar{h}} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0] \text{Tr}_{\bar{h}} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0] \right] - \frac{1}{2^m} \text{Tr} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0] \text{Tr} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0] \right). \tag{A.46}$$

Next, we compute the expectation value of Eq. (A.46) over $U'_{1:j}$. Here we begin with the integration for $W'_{k,d}$. The expectation value of $\text{Tr}[\text{Tr}_{\bar{h}} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0] \text{Tr}_{\bar{h}} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0]]$ in the first term of Eq. (A.46) can be calculated as

$$\begin{aligned}
& \left\langle \text{Tr} \left[\text{Tr}_{\bar{h}} \left[\tilde{B}_{x',\theta_i} \rho_0 \right] \text{Tr}_{\bar{h}} \left[\tilde{B}_{x',\theta_i} \rho_0 \right] \right] \right\rangle_{\tilde{W}'_{k,d}} \\
&= \left\langle \text{Tr} \left[\text{Tr}_{\bar{h}} \left[V_r'^{\dagger} \tilde{W}'_{k,d}{}^{\dagger} B_{\theta_i} \tilde{W}'_{k,d} V_r' \rho_0 \right] \text{Tr}_{\bar{h}} \left[V_r'^{\dagger} \tilde{W}'_{k,d}{}^{\dagger} B_{\theta_i} \tilde{W}'_{k,d} V_r' \rho_0 \right] \right] \right\rangle_{\tilde{W}'_{k,d}} \\
&= \frac{2^m}{2^{2m}-1} \left(\text{Tr} \left[\left(V_r'^{\dagger} \otimes V_r'^{\dagger} \right) \left(\text{Swaps}_{S_{(k,d),1} \otimes S_{(k,d),2}} \otimes \mathbb{I}_{\bar{S}_{(k,d),1} \otimes \bar{S}_{(k,d),2}} \right) \right. \right. \\
&\quad \left. \left. \times \left(V_r' \otimes V_r' \right) \left(\rho_0 \otimes \rho_0 \right) \left(\text{Swap}_{h_1 \otimes h_2} \otimes \mathbb{I}_{\bar{h}_1 \otimes \bar{h}_2} \right) \right] \right) - \frac{1}{2^m} \text{Tr} \left[\text{Tr}_{\bar{h}} [\rho_0] \text{Tr}_{\bar{h}} [\rho_0] \right], \tag{A.47}
\end{aligned}$$

where we utilize the equality,

$$\begin{aligned}
& \left\langle V^{\dagger} w_s^{\dagger} A w_s V A' \otimes V^{\dagger} w_s^{\dagger} A w_s V A' \right\rangle_{w_s} \\
&= \frac{1}{2^{2m}-1} \left(\left(V^{\dagger} \otimes V^{\dagger} \right) \left(\mathbb{I}_{S_{s,1} \otimes S_{s,2}} \otimes \text{Tr}_{S_{s,1}} [A] \otimes \text{Tr}_{S_{s,2}} [A] \right) \left(V \otimes V \right) \left(A' \otimes A' \right) \right. \\
&\quad \left. + \left(V^{\dagger} \otimes V^{\dagger} \right) \left(\text{Swap}_{S_{s,1} \otimes S_{s,2}} \otimes \text{Tr}_{S_{s,1} \cup S_{s,2}} \left[A \otimes A \left(\text{Swaps}_{S_{s,1}, S_{s,2}} \otimes \mathbb{I}_{\bar{S}_{s,1}, \bar{S}_{s,2}} \right) \right] \right) \left(V \otimes V \right) \left(A' \otimes A' \right) \right) \\
&\quad - \frac{1}{2^m(2^{2m}-1)} \left(\left(V^{\dagger} \otimes V^{\dagger} \right) \left(\text{Swaps}_{S_{s,1} \otimes S_{s,2}} \otimes \text{Tr}_{S_{s,1}} [A] \otimes \text{Tr}_{S_{s,2}} [A] \right) \left(V \otimes V \right) \left(A' \otimes A' \right) \right. \\
&\quad \left. + \left(V^{\dagger} \otimes V^{\dagger} \right) \left(\mathbb{I}_{S_{s,1} \otimes S_{s,2}} \otimes \text{Tr}_{S_{s,1} \cup S_{s,2}} \left[A \otimes A \left(\text{Swaps}_{S_{s,1}, S_{s,2}} \otimes \mathbb{I}_{\bar{S}_{s,1}, \bar{S}_{s,2}} \right) \right] \right) \left(V \otimes V \right) \left(A' \otimes A' \right) \right) \tag{A.48}
\end{aligned}$$

for arbitrary operator A, A' and the properties of the Pauli operators. Since the first term in Eq.(A.47) still includes V_r' , we integrate the quantity over all unitary blocks in V_r' . Then, using the equality in Eq. (A.48), we have

$$\begin{aligned}
& \left\langle \left(V_r'^{\dagger} \otimes V_r'^{\dagger} \right) \left(\text{Swaps}_{S_{(k,d),1} \otimes S_{(k,d),2}} \otimes \mathbb{I}_{\bar{S}_{(k,d),1} \otimes \bar{S}_{(k,d),2}} \right) \left(V_r' \otimes V_r' \right) \right\rangle_{V_r'} \\
&= \sum_{h' \in P_U(S^{(k_s:k_l,1)})} t_{h'} \left(\text{Swap}_{h'_1 \otimes h'_2} \otimes \mathbb{I}_{\bar{h}'_1 \otimes \bar{h}'_2} \right), \tag{A.49}
\end{aligned}$$

where $t_{h'} \in \mathbb{R}^+$. Note that a set of the coefficients $\{t_{h'}\}$ is the same as $\{t_h\}$. Thus, substituting the above equation into the first term in Eq. (A.47), the following result can be obtained.

$$\begin{aligned}
& \left\langle \text{Tr} \left[\text{Tr}_{\bar{h}} \left[\tilde{B}_{x',\theta_i} \rho_0 \right] \text{Tr}_{\bar{h}} \left[\tilde{B}_{x',\theta_i} \rho_0 \right] \right] \right\rangle_{U_{1:i}} \\
&= \frac{2^m}{2^{2m}-1} \left(\left(\sum_{h' \in P_U(S^{(k_s:k_l,1)})} t_{h'} \text{Tr} \left[\text{Tr}_{(h \cup h') \setminus (h \cap h')} [\rho_0] \text{Tr}_{(h \cup h') \setminus (h \cap h')} [\rho_0] \right] \right) - \frac{1}{2^m} \text{Tr} \left[\text{Tr}_{\bar{h}} [\rho_0] \text{Tr}_{\bar{h}} [\rho_0] \right] \right). \tag{A.50}
\end{aligned}$$

As for the second term in Eq. (A.46), the integration for $\tilde{W}'_{k,d}$ can be calculated in the following way;

$$\begin{aligned}
& \left\langle \text{Tr} \left[\tilde{B}_{x',\theta_i} \rho_0 \right] \text{Tr} \left[\tilde{B}_{x',\theta_i} \rho_0 \right] \right\rangle_{\tilde{W}'_{k,d}} = \left\langle \text{Tr} \left[V_r'^{\dagger} \tilde{W}'_{k,d}{}^{\dagger} B_{\theta_i} \tilde{W}'_{k,d} V_r' \rho_0 \right] \text{Tr} \left[V_r'^{\dagger} \tilde{W}'_{k,d}{}^{\dagger} B_{\theta_i} \tilde{W}'_{k,d} V_r' \rho_0 \right] \right\rangle_{\tilde{W}'_{k,d}} \\
&= \frac{2^m}{2^{2m}-1} \left(\text{Tr} \left[\text{Tr}_{\bar{S}_{(k,d)}} [\rho_{V_{x'}}] \text{Tr}_{\bar{S}_{(k,d)}} [\rho_{V_{x'}}] \right] - \frac{1}{2^m} \right), \tag{A.51}
\end{aligned}$$

where $\rho_{V_{x'}, \mathbf{p}\mathbf{q}} = \text{Tr}_{\tilde{S}(k,d)} [(|\mathbf{p}\rangle \langle \mathbf{q}| \otimes \mathbb{I}_{S(k,d)}) \rho_{V_{x'}}]$ with $\rho_{V_{x'}} = V_r' \rho_0 V_r'^{\dagger}$. Here we use Lemmas 4 and 5, $\text{Tr}[B_{\theta_i, \mathbf{p}\mathbf{q}}] = 0$ and $\text{Tr}[B_{\theta_i, \mathbf{p}\mathbf{q}} B_{\theta_i, \mathbf{p}'\mathbf{q}'}] = \delta_{(\mathbf{p}, \mathbf{q})} \delta_{(\mathbf{p}', \mathbf{q}')} 2^m$. Then, integrating the quantity over V_r' using Eqs. (A.41)- (A.44), we have

$$\left\langle \text{Tr} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0] \text{Tr} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0] \right\rangle_{U_{1:i}'} = \frac{2^m}{2^{2m} - 1} \left(\left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} t_h \text{Tr} [\text{Tr}_{\bar{h}}[\rho_0] \text{Tr}_{\bar{h}}[\rho_0]] \right) - \frac{1}{2^m} \right). \quad (\text{A.52})$$

Therefore we obtain

$$\begin{aligned} \text{Var}_{a,1} = \frac{1}{4} \left(\frac{2^m}{2^{2m} - 1} \right)^2 & \left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} \sum_{h' \in P_U(S^{(k_s: k_l, 1)})} t_h t_{h'} \text{Tr} \left[\text{Tr}_{\overline{(h \cup h') \setminus (h \cap h')}}[\rho_0] \text{Tr}_{\overline{(h \cup h') \setminus (h \cap h')}}[\rho_0] \right] \right. \\ & \left. - \frac{2}{2^m} \left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} t_h \text{Tr} [\text{Tr}_{\bar{h}}[\rho_0] \text{Tr}_{\bar{h}}[\rho_0]] \right) + \frac{1}{2^{2m}} \right). \end{aligned} \quad (\text{A.53})$$

Similarly, we can get

$$\begin{aligned} \text{Var}_{a,2} = \frac{1}{2} \left(\frac{2^m}{2^{2m} - 1} \right)^2 & \left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} \sum_{h' \in P_U(S^{(k_s: k_l, 1)})} t_h t_{h'} \text{Tr} [\mathbb{I}_{h \cap h'}] \text{Tr} [\text{Tr}_{\overline{h \cup h'}}[\rho_0] \text{Tr}_{\overline{h \cup h'}}[\rho_0]] \right. \\ & \left. - \frac{2}{2^m} \left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} t_h \text{Tr} [\text{Tr}_{\bar{h}}[\rho_0] \text{Tr}_{\bar{h}}[\rho_0]] \right) + \frac{1}{2^{2m}} \right), \end{aligned} \quad (\text{A.54})$$

$\text{Var}_{a,3}$

$$\begin{aligned} = \frac{1}{4} \left(\frac{2^m}{2^{2m} - 1} \right)^2 & \left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} \sum_{h' \in P_U(S^{(k_s: k_l, 1)})} t_h t_{h'} \text{Tr} \left[\text{Tr}_{\overline{(h \cup h') \setminus (h \cap h')}}[\rho_0] \text{Tr}_{\overline{(h \cup h') \setminus (h \cap h')}}[\rho_0] \right] \right. \\ & \left. - \frac{2}{2^m} \left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} t_h \text{Tr} [\text{Tr}_{\bar{h}}[\rho_0] \text{Tr}_{\bar{h}}[\rho_0]] \right) + \frac{1}{2^{2m}} \right). \end{aligned} \quad (\text{A.55})$$

Consequently, by summing up Eqs. (A.53), (A.54) and (A.55), the variance is expressed as

$$\begin{aligned} \text{Var} [k_{QF}^{(i)}] & = \text{Var}_{a,1} + \text{Var}_{a,2} + \text{Var}_{a,3} \\ & = \frac{1}{2} \left(\frac{2^m}{2^{2m} - 1} \right)^2 \left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} \sum_{h' \in P_U(S^{(k_s: k_l, 1)})} t_h t_{h'} \left(\text{Tr} \left[\text{Tr}_{\overline{(h \cup h') \setminus (h \cap h')}}[\rho_0] \text{Tr}_{\overline{(h \cup h') \setminus (h \cap h')}}[\rho_0] \right] \right. \right. \\ & \quad \left. \left. + \text{Tr} [\mathbb{I}_{h \cap h'}] \text{Tr} [\text{Tr}_{\overline{h \cup h'}}[\rho_0] \text{Tr}_{\overline{h \cup h'}}[\rho_0]] \right) - \frac{4}{2^m} \left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} t_h \text{Tr} [\text{Tr}_{\bar{h}}[\rho_0] \text{Tr}_{\bar{h}}[\rho_0]] \right) + \frac{2}{2^{2m}} \right). \end{aligned} \quad (\text{A.56})$$

Furthermore, by assuming the initial states can be represented as the tensor product states of arbitrary single-qubit pure states $\{\rho_{0,i}\}_{i=1}^n$, i.e., $\rho_0 = \rho_{0,1} \otimes \rho_{0,2} \otimes \dots \otimes \rho_{0,i} \otimes \dots \otimes \rho_{0,n}$, then

the lower bound of Eq. (A.56) can be written as

$$\begin{aligned}
& \text{Var} \left[k_{QF}^{(i)} \right] \\
& \geq \frac{1}{2} \left(\frac{2^m}{2^{2m} - 1} \right)^2 \left(\sum_{h \in P_U(S^{(k_s:k_l,1)})} \sum_{h' \in P_U(S^{(k_s:k_l,1)})} t_h t_{h'} (1 + \text{Tr} [\mathbb{I}_{h \cap h'}]) - \frac{4}{2^m} \left(\sum_{h \in P_U(S^{(k_s:k_l,1)})} t_h \right) + \frac{2}{2^{2m}} \right) \\
& = \frac{1}{2} \left(\frac{2^m}{2^{2m} - 1} \right)^2 \left(2 \left(\left(\sum_{h \in P_U(S^{(k_s:k_l,1)})} t_h \right) - \frac{1}{2^m} \right)^2 + \sum_{h \in P_U(S^{(k_s:k_l,1)})} \sum_{h' \in P_U(S^{(k_s:k_l,1)})} t_h t_{h'} (\text{Tr} [\mathbb{I}_{h \cap h'}] - 1) \right) \\
& \geq \frac{1}{2} \left(\frac{2^m}{2^{2m} - 1} \right)^2 \left(\sum_{h \in P_U(S^{(k_s:k_l,1)})} \sum_{h' \in P_U(S^{(k_s:k_l,1)})} t_h t_{h'} (\text{Tr} [\mathbb{I}_{h \cap h'}] - 1) \right) \\
& \geq \frac{1}{2} \left(\frac{2^m}{2^{2m} - 1} \right)^2 t_{S^{(k_s:k_l,1)}}^2 \left(\text{Tr} [\mathbb{I}_{S^{(k_s:k_l,1)}}] - 1 \right) \\
& = \frac{2^{2md} (2^{md} - 1)}{2 (2^{2m} - 1)^2 (2^m + 1)^{4(d-1)}}.
\end{aligned} \tag{A.57}$$

Moreover, suppose the initial state satisfies the following equalities;

$$\begin{aligned}
& \text{Tr} \left[\text{Tr}_{(\overline{h \cup h'}) \setminus (h \cap h')} [\rho_0] \text{Tr}_{(\overline{h \cup h'}) \setminus (h \cap h')} [\rho_0] \right] \geq \text{Tr} [\text{Tr}_{\bar{h}} [\rho_0] \text{Tr}_{\bar{h}} [\rho_0]] \text{Tr} [\text{Tr}_{\bar{h}'} [\rho_0] \text{Tr}_{\bar{h}'} [\rho_0]], \\
& \text{Tr} [\text{Tr}_{\overline{h \cup h'}} [\rho_0] \text{Tr}_{\overline{h \cup h'}} [\rho_0]] \geq \text{Tr} [\text{Tr}_{\bar{h}} [\rho_0] \text{Tr}_{\bar{h}} [\rho_0]] \text{Tr} [\text{Tr}_{\bar{h}'} [\rho_0] \text{Tr}_{\bar{h}'} [\rho_0]].
\end{aligned} \tag{A.58}$$

Then, the lower bound of the variance can be written as

$$\text{Var} \left[k_{QF}^{(i)} \right] \geq \frac{2^{md} - 1}{2 (2^{2m} - 1)^2 (2^m + 1)^{4(d-1)}}. \tag{A.59}$$

Note that the initial states that satisfy the above conditions include the tensor product states of arbitrary single-qubit pure states $\{\rho_{0,i}\}_{i=1}^n$, i.e., $\rho_0 = \rho_{0,1} \otimes \rho_{0,2} \otimes \dots \otimes \rho_{0,i} \otimes \dots \otimes \rho_{0,n}$, and the completely mixed states, while it is unclear if any quantum states fulfill the properties.

Lastly, we present that the variance scaling of $k_{QF}^{(i)}$ is the same as the above results. With the assumption that the initial state is the tensor product of arbitrary single-qubit states, a similar calculation process leads to the following results;

$$\text{Var} \left[T^{(i)} \right] = \frac{2^{2m}}{(2^{2m} - 1)^2} \left(\left(\sum_{h \in P_U(S^{(k_s:k_l,1)})} t_h \right) - \frac{1}{2^m} \right)^2, \tag{A.60}$$

$$\text{Cov} \left[k_{QF}^{(i)}, T^{(i)} \right] = -\frac{2^{2m}}{(2^{2m} - 1)^2} \left(\left(\sum_{h \in P_U(S^{(k_s:k_l,1)})} t_h \right) - \frac{1}{2^m} \right)^2. \tag{A.61}$$

Therefore, by substituting Eqs. (A.56), (A.60) and (A.61) into Eq. (A.17), we get

$$\text{Var} \left[k_{QF}^{(i)} \right] \geq \frac{2^{2md} (2^{md} - 1)}{2 (2^{2m} - 1)^2 (2^m + 1)^{4(d-1)}}. \tag{A.62}$$

A.3 Further Analytical Results

We here address the variance scaling of the QFK in Eq. (4.38). Specifically, we discuss the variance of the QFK summed over all terms, i.e., $\text{Var}[\sum_i k_{Q_{F'}}^{(i)}]$, and the effect of the quantum Fisher information matrix.

First, we show the following inequality;

$$\text{Var} \left[\sum_i k_{Q_{F'}}^{(i)} \right] \geq \sum_i \text{Var} \left[k_{Q_{F'}}^{(i)} \right]. \quad (\text{A.63})$$

Because of the definition of the variance, this is equivalent to demonstrating $\text{Cov}[k_{Q_{F'}}^{(i)}, k_{Q_{F'}}^{(j)}] \geq 0$. We thus show the covariance term is equal to or greater than zero for globally-random quantum circuits and ALAs. The covariance term can be decomposed as follows;

$$\begin{aligned} \text{Cov} \left[k_{Q_{F'}}^{(i)}, k_{Q_{F'}}^{(j)} \right] &= \text{Cov} \left[k_{Q_F}^{(i)} + T^{(i)}, k_{Q_F}^{(j)} + T^{(j)} \right] \\ &= \left\langle k_{Q_F}^{(i)} k_{Q_F}^{(j)} \right\rangle + \left\langle k_{Q_F}^{(i)} T^{(j)} \right\rangle + \left\langle T^{(i)} k_{Q_F}^{(j)} \right\rangle + \left\langle T^{(i)} T^{(j)} \right\rangle \end{aligned} \quad (\text{A.64})$$

We note that we here utilize $\langle k_{Q_F}^{(i)} \rangle = \langle T^{(i)} \rangle = 0$. Thus, we focus on these four terms in the following. Also, without loss of generality, we assume $i < j$.

Globally-Random Quantum Circuits

We consider the following situations: U_i and U_j (i) are in the same layer and (ii) in the different layers and the unitary operator between them $U_{i,j} \equiv U_{i,j}(\mathbf{x}, \boldsymbol{\theta})$ can form a 1-design. We here demonstrate the calculation for $\langle k_{Q_F}^{(i)} k_{Q_F}^{(j)} \rangle$. The expectation value over $U_{1,i}$ is expressed as

$$\begin{aligned} \left\langle k_{Q_F}^{(i)} k_{Q_F}^{(j)} \right\rangle_{U_{1,i}} &= \left\langle \left[\text{Tr} \left[\rho_0 \left\{ \tilde{B}_{\mathbf{x}, \theta_i}, \tilde{B}_{\mathbf{x}', \theta_i} \right\} \right] \text{Tr} \left[\rho_0 \left\{ \tilde{B}_{\mathbf{x}, \theta_j}, \tilde{B}_{\mathbf{x}', \theta_j} \right\} \right] \right] \right\rangle_{U_{1,i}} \\ &= \left\langle \text{Tr} \left[\rho_0 U_{1,i}^\dagger B_{\theta_i} U_{1,i} \tilde{B}_{\mathbf{x}', \theta_i} \right] \text{Tr} \left[\rho_0 U_{1,i}^\dagger B'_{\theta_j, \mathbf{x}} U_{1,i} \tilde{B}_{\mathbf{x}', \theta_j} \right] \right\rangle_{U_{1,i}} \\ &\quad + \left\langle \text{Tr} \left[U_{1,i}^\dagger B_{\theta_i} U_{1,i} \rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \right] \text{Tr} \left[\rho_0 U_{1,i}^\dagger B'_{\theta_j, \mathbf{x}} U_{1,i} \tilde{B}_{\mathbf{x}', \theta_j} \right] \right\rangle_{U_{1,i}} \\ &\quad + \left\langle \text{Tr} \left[\rho_0 U_{1,i}^\dagger B_{\theta_i} U_{1,i} \tilde{B}_{\mathbf{x}', \theta_i} \right] \text{Tr} \left[U_{1,i}^\dagger B'_{\theta_j, \mathbf{x}} U_{1,i} \rho_0 \tilde{B}_{\mathbf{x}', \theta_j} \right] \right\rangle_{U_{1,i}} \\ &\quad + \left\langle \text{Tr} \left[U_{1,i}^\dagger B_{\theta_i} U_{1,i} \rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \right] \text{Tr} \left[U_{1,i}^\dagger B'_{\theta_j, \mathbf{x}} U_{1,i} \rho_0 \tilde{B}_{\mathbf{x}', \theta_j} \right] \right\rangle_{U_{1,i}} \\ &= \text{Var}'_{r,1} + \text{Var}'_{r,2} + \text{Var}'_{r,3} + \text{Var}'_{r,4}, \end{aligned} \quad (\text{A.65})$$

with $B'_{\theta_j, \mathbf{x}} = U_{i,j}^\dagger(\mathbf{x}, \boldsymbol{\theta}) B_{\theta_j} U_{i,j}(\mathbf{x}, \boldsymbol{\theta})$. Similarly to the expectation value calculation shown in Appendix A.2, we can obtain

$$\text{Var}'_{r,1} = \frac{\text{Tr} \left[B_{\theta_i} B'_{\theta_j, \mathbf{x}} \right]}{(2^{2n} - 1)} \left(1 - \frac{1}{2^n} \right) \left(\text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_j} \rho_0 \right] \right), \quad (\text{A.66})$$

$$\text{Var}'_{r,2} = \frac{\text{Tr} \left[B_{\theta_i} B'_{\theta_j, \mathbf{x}} \right]}{(2^{2n} - 1)} \left(\text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} \tilde{B}_{\mathbf{x}', \theta_j} \right] - \frac{1}{2^n} \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_j} \rho_0 \right] \right), \quad (\text{A.67})$$

$$\text{Var}'_{r,3} = \frac{\text{Tr} \left[B_{\theta_i} B'_{\theta_j, \mathbf{x}} \right]}{(2^{2n} - 1)} \left(\text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_j} \tilde{B}_{\mathbf{x}', \theta_i} \right] - \frac{1}{2^n} \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_i} \rho_0 \right] \text{Tr} \left[\tilde{B}_{\mathbf{x}', \theta_j} \rho_0 \right] \right), \quad (\text{A.68})$$

$$\text{Var}'_{r,4} = \frac{\text{Tr} [B_{\theta_i} B'_{\theta_j, \mathbf{x}}]}{(2^{2n} - 1)} \left(1 - \frac{1}{2^n}\right) \left(\text{Tr} [\tilde{B}_{\mathbf{x}', \theta_i} \rho_0] \text{Tr} [\tilde{B}_{\mathbf{x}', \theta_j} \rho_0]\right). \quad (\text{A.69})$$

Subsequently, by integrating the quantity over $U'_{1:i}$, we get

$$\begin{aligned} \left\langle k_{QF}^{(i)} k_{QF}^{(j)} \right\rangle_{(U_{1:i}, U'_{1:i})} &= \text{Var}'_{r,1} + \text{Var}'_{r,2} + \text{Var}'_{r,3} + \text{Var}'_{r,4} \\ &= \frac{2 \text{Tr} [B_{\theta_i} B'_{\theta_j, \mathbf{x}}] \text{Tr} [B_{\theta_i} B'_{\theta_j, \mathbf{x}'}]}{2^{2n} (2^{2n} - 1) (2^n + 1)} (2^n (2^n + 1) + 2^n - 2). \end{aligned} \quad (\text{A.70})$$

Now, we investigate the expectation value for the two cases mentioned above. In case (i), $U_{i:j}$ and $U'_{i:j}$ are the identity operators and thus we have

$$\left\langle k_{QF}^{(i)} k_{QF}^{(j)} \right\rangle_{(U_{1:i}, U'_{1:i})} = \frac{2 (\text{Tr} [B_{\theta_i} B_{\theta_j}])^2}{2^{2n} (2^{2n} - 1) (2^n + 1)} (2^n (2^n + 1) + 2^n - 2) \geq 0. \quad (\text{A.71})$$

Also, in case (ii), integration of the term over $U_{i:j}(\mathbf{x}, \boldsymbol{\theta})$ end up with

$$\begin{aligned} \left\langle k_{QF}^{(i)} k_{QF}^{(j)} \right\rangle_{(U_{1:i}, U'_{1:i})} &= \frac{2 \left\langle \text{Tr} [B_{\theta_i} B'_{\theta_j, \mathbf{x}}] \text{Tr} [B_{\theta_i} B'_{\theta_j, \mathbf{x}'}] \right\rangle_{(U_{1:i}, U'_{1:i})}}{2^{2n} (2^{2n} - 1) (2^n + 1)} (2^n (2^n + 1) + 2^n - 2) \\ &= 0, \end{aligned} \quad (\text{A.72})$$

where we utilized Lemma 1 and the traceless property of B_{θ_i} .

Similarly, the remaining terms in Eq. (A.64) can be calculated as follows;

Case (i)

$$\left\langle k_{QF}^{(i)} T^{(j)} \right\rangle = \left\langle T^{(i)} k_{QF}^{(j)} \right\rangle = -\frac{(\text{Tr} [B_{\theta_i} B_{\theta_j}])^2}{2^{2n} (2^n + 1)^2}, \quad (\text{A.73})$$

$$\left\langle T^{(i)} T^{(j)} \right\rangle = \frac{(\text{Tr} [B_{\theta_i} B_{\theta_j}])^2}{2^{2n} (2^n + 1)^2}. \quad (\text{A.74})$$

Case (ii)

$$\left\langle k_{QF}^{(i)} T^{(j)} \right\rangle = \left\langle T^{(i)} k_{QF}^{(j)} \right\rangle = 0, \quad (\text{A.75})$$

$$\left\langle T^{(i)} T^{(j)} \right\rangle = 0. \quad (\text{A.76})$$

Consequently, by substituting these terms into Eq. (A.64), we can show that $\text{Cov}[k_{QF'}^{(i)}, k_{QF'}^{(j)}] \geq 0$.

Alternating Layered Ansatzes

As in the case of the globally-random quantum circuits, we consider the following situations: the gates containing θ_i and θ_j are (i) in the different local unitary blocks, and (ii) in the same local unitary block and the quantum gate between them is the identity matrix, and (iii) in the same local unitary block and the quantum gate between them forms 1-design.

In what follows, we compute the term $\langle k_{QF}^{(i)} k_{QF}^{(j)} \rangle$ in Eq. (A.64) for these cases. In case (i), we integrate the quantity over $\tilde{W}_{k_j, d_j} \equiv \tilde{W}_{k_j, d_j}(\mathbf{x}, \theta_j)$ and then we get

$$\begin{aligned}
& \left\langle \left[\text{Tr} \left[\rho_0 \left\{ \tilde{B}_{\mathbf{x}, \theta_i}, \tilde{B}_{\mathbf{x}', \theta_i} \right\} \right] \text{Tr} \left[\rho_0 \left\{ \tilde{B}_{\mathbf{x}, \theta_j}, \tilde{B}_{\mathbf{x}', \theta_j} \right\} \right] \right] \right\rangle_{\tilde{W}_{k_j, d_j}} \\
&= \text{Tr} \left[\rho_0 V_r^\dagger \tilde{W}_{k_j, d_j}^\dagger B_{\theta_i} \tilde{W}_{k_j, d_j} V_r \tilde{B}_{\mathbf{x}', \theta_i} \right] \left\langle \text{Tr} \left[\rho_0 V_r^\dagger \tilde{W}_{k_j, d_j}^\dagger B_{\theta_j} \tilde{W}_{k_j, d_j} V_r \tilde{B}_{\mathbf{x}', \theta_j} \right] \right\rangle_{\tilde{W}_{k_j, d_j}} \\
&\quad + \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} V_r^\dagger \tilde{W}_{k_j, d_j}^\dagger B_{\theta_i} \tilde{W}_{k_j, d_j} V_r \right] \left\langle \text{Tr} \left[\rho_0 V_r^\dagger(\mathbf{x}, \boldsymbol{\theta}) \tilde{W}_{k_j, d_j}^\dagger B_{\theta_j} \tilde{W}_{k_j, d_j} V_r \tilde{B}_{\mathbf{x}', \theta_j} \right] \right\rangle_{\tilde{W}_{k_j, d_j}} \\
&\quad + \text{Tr} \left[\rho_0 V_r^\dagger \tilde{W}_{k_j, d_j}^\dagger B_{\theta_i} \tilde{W}_{k_j, d_j} V_r \tilde{B}_{\mathbf{x}', \theta_i} \right] \left\langle \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_j} V_r^\dagger \tilde{W}_{k_j, d_j}^\dagger B_{\theta_j} \tilde{W}_{k_j, d_j} V_r \right] \right\rangle_{\tilde{W}_{k_j, d_j}} \\
&\quad + \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_i} V_r^\dagger \tilde{W}_{k_j, d_j}^\dagger B_{\theta_i} \tilde{W}_{k_j, d_j} V_r \right] \left\langle \text{Tr} \left[\rho_0 \tilde{B}_{\mathbf{x}', \theta_j} V_r^\dagger \tilde{W}_{k_j, d_j}^\dagger B_{\theta_j} \tilde{W}_{k_j, d_j} V_r \right] \right\rangle_{\tilde{W}_{k_j, d_j}} \\
&= 0.
\end{aligned} \tag{A.77}$$

We utilize the fact that \tilde{W}_{k_j, d_j} is in the different local unitary block in the first equality and $\langle k_{QF}^{(j)} \rangle = 0$ in the second equality. As for case (ii) and (iii), we perform a similar calculation to derive the variance of the QFK for ALAs (see Appendix A.2) and then we can have

$$\begin{aligned}
\left\langle k_{QF}^{(i)} k_{QF}^{(j)} \right\rangle_{(U_{1:i}, U'_{1:i})} &\geq \frac{2^{2m(d-1)-1} (2^{md} - 1)}{(2^{2m} - 1)^2 (2^m + 1)^{4(d-1)}} \left\langle \text{Tr} [B_{\theta_i} B'_{\theta_j, \mathbf{x}}] \text{Tr} [B_{\theta_i} B'_{\theta_j, \mathbf{x}'}] \right\rangle_{(\tilde{W}_{k_i, d_i}, \tilde{W}'_{k_i, d_i})} \\
&\quad + \frac{\left\langle \text{Tr} [B_{\theta_i} B'_{\theta_j, \mathbf{x}}] \text{Tr} [B_{\theta_i} B'_{\theta_j, \mathbf{x}'}] \right\rangle_{(\tilde{W}_{k_i, d_i}, \tilde{W}'_{k_i, d_i})}}{(2^{2m} - 1)^2} \left(\left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} t_h \right) - \frac{1}{2^m} \right)^2.
\end{aligned} \tag{A.78}$$

Then, analogous to the case of globally-random quantum circuits, the expectation values for these cases are respectively written as

$$\left\langle k_{QF}^{(i)} k_{QF}^{(j)} \right\rangle_{(U_{1:i}, U'_{1:i})} = 0, \tag{A.79}$$

$$\begin{aligned}
\left\langle k_{QF}^{(i)} k_{QF}^{(j)} \right\rangle_{(U_{1:i}, U'_{1:i})} &\geq \frac{2^{2m(d-1)-1} (2^{md} - 1)}{(2^{2m} - 1)^2 (2^m + 1)^{4(d-1)}} (\text{Tr} [B_{\theta_i} B_{\theta_j}])^2 \\
&\quad + \frac{(\text{Tr} [B_{\theta_i} B_{\theta_j}])^2}{(2^{2m} - 1)^2} \left(\left(\sum_{h \in P_U(S^{(k_s: k_l, 1)})} t_h \right) - \frac{1}{2^m} \right)^2.
\end{aligned} \tag{A.80}$$

As for the remaining terms, expectation values read as follows;

Case (i)

$$\left\langle k_{QF}^{(i)} T^{(j)} \right\rangle = \left\langle T^{(i)} k_{QF}^{(j)} \right\rangle = 0, \tag{A.81}$$

$$\left\langle T^{(i)} T^{(j)} \right\rangle = 0. \tag{A.82}$$

Case (ii)

$$\langle k_{QF}^{(i)} T^{(j)} \rangle = \langle T^{(i)} k_{QF}^{(j)} \rangle = 0, \quad (\text{A.83})$$

$$\langle T^{(i)} T^{(j)} \rangle = 0. \quad (\text{A.84})$$

Case (iii)

$$\langle k_{QF}^{(i)} T^{(j)} \rangle = \langle T^{(i)} k_{QF}^{(j)} \rangle = -\frac{(\text{Tr} [B_{\theta_i} B_{\theta_j}])^2}{(2^{2m} - 1)^2} \left(\left(\sum_{h \in P_U(S^{(k_s:k_l,1)})} t_h \right) - \frac{1}{2^m} \right)^2, \quad (\text{A.85})$$

$$\langle T^{(i)} T^{(j)} \rangle = \frac{(\text{Tr} [B_{\theta_i} B_{\theta_j}])^2}{(2^{2m} - 1)^2} \left(\left(\sum_{h \in P_U(S^{(k_s:k_l,1)})} t_h \right) - \frac{1}{2^m} \right)^2. \quad (\text{A.86})$$

Therefore, we can show $\text{Cov}[k_{QF'}^{(i)}, k_{QF'}^{(j)}] \geq 0$.

Lastly, we present the effect of the quantum Fisher information matrix on the variance scaling. By the eigendecomposition of the inverse of the quantum Fisher information matrix, i.e., $\mathcal{F}_A^{-1} = VD^{-1}V^{-1}$ with D the diagonal matrix containing eigenvalues of QFIM and V the unitary matrix, the QFK can be expressed as the QFK in Eq. (4.38) can be rewritten as

$$\begin{aligned} k_{QF}(\mathbf{x}, \mathbf{x}') &= \frac{1}{2} \sum_k D_{kk}^{-1} \sum_{i,j} v_{ik} v_{jk}^* \text{Tr} \left[\rho_0 \left\{ \tilde{B}_{\mathbf{x}, \theta_i} - \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}, \theta_i}] \right\}, \tilde{B}_{\mathbf{x}', \theta_j} - \text{Tr} [\rho_0 \tilde{B}_{\mathbf{x}', \theta_j}] \right] \\ &= \frac{1}{2} \sum_k D_{kk}^{-1} \sum_{i,j} v_{ik} v_{jk}^* \phi_{QFK,i}(\mathbf{x})^T \phi_{QFK,j}(\mathbf{x}') \end{aligned} \quad (\text{A.87})$$

where v_{ij} represents (i, j) element of the unitary V and $\phi_{QFK,i}(\mathbf{x})$ represents the feature vector corresponding to the i -th term of the QFK, i.e., $k_{QF}^{(i)} = \phi_{QFK,i}(\mathbf{x})^T \phi_{QFK,j}(\mathbf{x}')$. As V is a unitary operator, the term $\sum_{i,j} v_{ik} v_{jk}^* \phi_{QFK,i}(\mathbf{x})^T \phi_{QFK,j}(\mathbf{x}')$ can be regarded as the basis transformation. From the discussion above, the summation would not reduce the variance scaling. Thus, the diagonal terms of D determine whether the variance decreases exponentially. If $D_{kk} \in \mathcal{O}(c^n)$ with a constant $c > 1$ and the number of qubits n for all k , the variance might vanish. However, this assumption means that, from the quantum Cramer-Rao inequality [233,234], the estimation error of all the parameters of the parametrized quantum circuit is lower bounded by exponentially small numbers, which contradicts the result of quantum channel tomography. Thus, even with the non-identity quantum Fisher information matrix, the statement in Theorem 1 would remain unchanged for the QFK in Eq. (4.38).