Characterizing and harnessing complex quantum dynamics for quantum machine learning

Aoi HAYASHI

Doctor of Philosophy

Department of Informatics
School of Multidisciplinary Sciences

The Graduate University for Advanced Studies, SOKENDAI

March 2026 (version of January 9, 2026)

Acknowledgements

Firstly, I would like to express my profound gratitude to my supervisor, Professor Kae Nemoto, for her guidance, which provided not only scientific ideas and directions but also many valuable opportunities for me to consider how to shape a story based on scientific results, and helped me grasp the long-term research direction of the field. It has been fortunate for me to be a PhD student in her group, which encompasses stimulating interactions with people from a variety of disciplines. Besides, it was also an enjoyable time to chat about violins with her.

I would like to express my gratitude to Professor William John Munro. He always provided valuable critical comments on my results and possible directions. His support also improved my English and scientific writing skills.

I am deeply grateful to Professor Akihito Soeda. The discussions with him helped not only in considering better research directions, but also in summarizing and refining the story of my thesis work.

I would like to extend my sincere gratitude to my former supervisor, Professor Bhanu Pratap Das. Thanks to his guidance, my internationality began to grow through the international collaborations I experienced with India and Germany. I am also profoundly thankful for his continued support, even after I finished my Bachelor's degree.

I am very grateful to Dr. Akitada Sakurai for his generous support. As what he focuses on is unique, discussions with him always inspire me. His help with my writing was also invaluable. I would not have been able to accomplish my thesis work without his warm and patient support.

I gratefully thank Ms. Yukiko Sanaka for her huge administrative support over the past five years. Thanks to her help, I was able to process thesis evaluations smoothly. Besides, I would not have gained invaluable experiences during my business trips without her extensive support. I would like to extend my gratitude to other administrative staff at NII, SOKENDAI, and OIST for their help.

I gratefully thank all other members (including former members) of our group, especially (in alphabetical order) Dr. Ananga Datta, Ms. Chaimae El Bouazizi, Dr. Henry Nourse, Dr. Hon Wai Lau, Dr. Josephine Dias, Dr. Nicholas Connolly, Dr. Nicolo Lo Piparo, Dr. Peizhe Li, Dr. Shin Nishio, Dr. Thomas Scruby, Dr. Victor Bastidas, and Dr. Zhu Yuwei. The discussions I had with them provided new and fresh angles and ideas for my research, as this group has diverse expertise. I also thank them for their valuable feedback on my presentation rehearsals. I would like to extend my thanks to the members of Bill's unit at OIST, especially (in alphabetical order) Dr. Heitor Peres Casagrande and Dr. Steven Sagona-Stophel, for their scientific support and feedback on my defense rehearsals.

I thank all the members of the string ensemble I have belonged to since 2018, as well as all the composers who wrote the chamber pieces we performed. Their music and all the rehearsals and concerts we have done have always refreshed and encouraged me to concentrate on my research.

Lastly, I would like to express my deepest gratitude to my parents for their understanding of my passion for research. My father especially gave me the chance for exposure to science when I was a child. My mother has always been understanding and supportive of my desire to study in a PhD program. I acknowledge their precious love.

Aoi Hayashi

Abstract

Quantum computation is a form of computation utilizing quantum mechanical phenomena. A formal formulation of quantum computer was proposed in 1985, leading to novel quantum algorithms whose computation time scales better, with respect to the problem size, than the classical counterparts. However, running these quantum algorithms requires fault-tolerant implementations, and hence, such engineering in quantum systems can be an overhead in utilizing quantum systems for information processing. This suggests, in addition to seeking faster quantum algorithms, the necessity of exploring how to harness quantum systems more naturally without extensive engineering, leading to another form of quantum information processing.

Quantum machine learning (QML) is expected to be a promising approach for harnessing quantum systems for information processing. This is motivated by the fact that the success of neural networks has led to the possibility of solving ML tasks more efficiently with non-digital computer designs. Several QML algorithms that resemble neural networks utilize quantum systems in a manner distinct from digital quantum algorithms. However, training quantum systems is fundamentally challenging, suggesting that it still remains necessary to explore an alternative way of harnessing quantum systems for ML.

Along with this direction, the quantum reservoir model harnesses quantum systems in a unique way: nontrivial outputs from an untrained quantum system are leveraged by a linear classical machine learning model to solve various tasks. Hence, the quantum part does not need to be programmable and can be analog quantum dynamics. Without extensive engineering in the quantum part, a quantum reservoir model was shown to be capable of performing nontrivial ML tasks using physical dynamics, which had already been realized experimentally. Therefore, the quantum reservoir model should be one of the promising directions for pursuing another form of quantum information processing.

In this thesis, I will investigate the computational capability of natural quantum dynamics in the quantum reservoir model. I will tackle the following critical open questions in this context: (i) how well natural dynamics perform nontrivial ML tasks compared to engineered systems, and (ii) what kind of engineering in quantum dynamics is essential and appropriate for a given task. To tackle the first question (i), I will benchmark the learning performance of various natural quantum dynamics against random circuits that require complicated engineering, thereby revealing the computational power of natural dynamics. For the second question (ii), I will present a characterization method of quantum dynamics that has the potential to bridge quantum properties and their machine learning roles, providing concrete insights into essential and appropriate engineering to design quantum systems for ML.

Contents

Chapter 1

Introduction

Quantum computation is the idea of harnessing quantum mechanics to perform computational tasks, originating from Feynman in 1981 [1]. He was especially interested in simulating quantum dynamics on a fully controllable quantum system, which is generally challenging for classical digital computers since the number of degrees of freedom in quantum mechanics scales exponentially with respect to the system size. As of today, there have been various experimental demonstrations of nontrivial quantum dynamics [2–12] that are almost intractable on classical computers, thanks to the rapid development of controllable quantum platforms within reach of thousands of physical quantum bits (qubits) [13–15].

Towards more algorithmic tasks in addition to quantum simulation, a formal formulation of quantum computer was proposed in 1985 [16], followed by the proposal of its logical circuit model in 1989 [17]. It performs computation with quantum logic gates, which are obtained by designing and digitizing quantum time evolution. Novel quantum algorithms for several computational tasks have been proposed [18–22], and the number of computation steps scales significantly better with respect to the problem size in these quantum algorithms than in their corresponding classical ones. Solving these algorithmic tasks on a quantum computer requires advanced engineering in the quantum system, such as quantum error correction and fault-tolerant implementation, to reduce noise in qubits and achieve accurate and reliable computation.

However, it has been discussed that such advanced engineering may prevent us from benefiting from those quantum algorithms [23–26]. For instance, quantum error correction and fault-tolerant implementations slow down each computation step, thereby suggesting that the actual computation time of some of those quantum algorithms may be longer than that of their classical counterparts in realistic circumstances [23–26]. Such advanced engineering

in quantum systems, which is required for a digital quantum computer, can thus be an overhead in utilizing quantum systems for information processing. This fact strongly encourages us to reconsider how we should utilize quantum systems more efficiently. Therefore, in addition to seeking faster quantum algorithms, it is also essential to explore how to harness quantum systems more naturally without the need for extensive engineering.

It seems nontrivial what kind of information processing can be performed by less engineered quantum systems, which may contain noise. In this sense, machine learning (ML) may have the potential due to the impact that neural networks have made on hardware design [27]. ML is a type of heuristic algorithm that learns and models a structure behind empirical data to make reasonable predictions on future data. There are various ML algorithms, such as clustering analysis [28], support vector machine [29], and neural networks [30]. Notably, the study of neural networks has led to computer hardware designs that differ from the standard digital one due to the noise tolerance of neural networks [31–33].

Quantum machine learning (QML) was independently coined by Chrisley [27] and Kak [34] in 1995. Especially, Chrisley has pointed out the importance of considering how to harness quantum systems for ML in addition to a better time scale of computation, based on the impact of neural networks on hardware. Indeed, QML models proposed soon after 1995 [35–39] focused on how quantum dynamics can be utilized or interpreted as neural networks, whose form is different from the digital quantum computation model [16]. Those models are unfortunately not well-formulated enough for performing general ML tasks.

A QML model was proposed in 2018 [40], which resembles neural networks and is sufficiently definitive for general ML tasks. It is called quantum circuit learning (QCL) or variational quantum algorithms (VQAs) [41]. In these algorithms, input information is encoded in quantum states, processed through a parametrized quantum circuit, and measured to extract classical outputs. The learning process is performed by tuning the parameter values in the quantum circuit to obtain desired outputs, just like neural networks. Thus, QCL and VQAs have garnered considerable attention for utilizing quantum circuits over noisy qubits rather than logical qubits protected by quantum error correction.

However, it has been found that the trainability in QCL and VQAs is a fundamental obstacle [42–44]. The difficulty of training significantly limits the applicability of quantum systems, while they inherently possess a huge space and complex dynamics in current quantum devices with tens or hundreds of noisy qubits. This suggests that it still remains necessary to explore an alternative way of harnessing quantum systems for ML.

In contrast to the pure circuit models such as QCL and VQAs, the quantum reservoir model [45–49] was proposed to solve ML tasks by harnessing quantum dynamics. Its idea is to utilize a fixed quantum dynamics as a non-trivial information processing and to process outputs from the quantum part with a simple classical ML to be trained. As the fixed quantum part is not trained, it can be implemented using analog Hamiltonian dynamics instead of a circuit implementation with quantum gates. Integrating such dynamics directly into ML algorithms might be more natural and require less engineering, as quantum gate operations are inherently implemented by designing and digitizing Hamiltonian dynamics.

Even though the quantum reservoir model significantly reduces the engineering required for solving several ML tasks, this QML model may still require complicated engineering to design Hamiltonian dynamics. However, using physical dynamics that have already been experimentally realized [2], it has been shown that one of the quantum reservoir models can achieve mid-scale image classification tasks [50]. Therefore, this natural way of utilizing quantum systems should be one of the promising angles for pursuing an alternative and more effective form of quantum information processing for ML.

In this thesis, I will explore the potential computational capability of natural quantum dynamics for performing ML by focusing on a quantum reservoir model, which is the one that can solve image classification tasks. I will tackle the following critical open questions in this context: (i) how well natural quantum dynamics perform nontrivial ML tasks such as image classification compared to engineered systems, and (ii) what kind of engineering in quantum systems is essential and appropriate for those tasks. Regarding (i), in previous studies, natural quantum dynamics are rarely compared with engineered quantum systems, such as random circuits [51, 52]. However, it is essential to determine whether such natural dynamics are already sufficient or additional engineering is required for a given task. I will thus benchmark the learning performance of natural quantum dynamics on image classification tasks against random circuits, thereby investigating the computational power of natural quantum dynamics.

Regarding the second question (ii), it has been reported that various physical properties of quantum systems are correlated with learning performance [48, 50, 53–56], suggesting empirical evidence on what kind of engineering is sufficient, but not on what is essential or appropriate. Their mechanisms are not yet clear, as those studies considered physics-oriented analysis, lacking a direct connection to the machine learning roles of quantum properties. On the other hand, there are several machine-learning theoretical analyses [45, 57–59], but they often lack concrete insights into what

engineering is required in quantum systems. I will therefore present a characterization method of quantum dynamics that has the potential to bridge quantum properties and their machine learning roles.

For these purposes, the following chapters will be organized as follows. In Chapter 2, mathematical preparation and essential notions in quantum information science will be provided. In Chapter 3, preliminaries on machine learning and several quantum machine learning models will be outlined, along with existing characterizations of their learning behaviors. The quantum reservoir model adopted throughout this thesis will also be described. Through Chapters 4, 5, and 6, we will investigate the learning performance by simplifying the quantum dynamics used in the QML model while benchmarking it against random circuits, aiming to elucidate the beneficial properties in the quantum dynamics that are essential for high performance. Several characterization methods of unitary matrices representing quantum dynamics will be introduced, and their impacts on learning performance will be examined. In Chapter 7, motivated by an ML theory, an information-theoretic complexity of quantum dynamics will be proposed, which has a connection to a fundamental complexity of quantum objects from the quantum information perspective. Finally, in Chapter 8, this thesis will be concluded, and potential future directions will also be discussed.

Chapter 2

Preliminaries

2.1 Mathematical preparation

In this section, we introduce some mathematical basics and notations that will be used in this thesis. We begin by introducing several basic notions in group theory and providing essential examples of groups in quantum information.

2.1.1 Groups

In quantum mechanics, group theory plays a central role, particularly the unitary group, since unitary operators represent quantum operations.

Definition 2.1.1 (group)**.** A tuple $(G, \cdot)$ of a set G and a binary operation $\cdot : G \times G \to G$ is called a *group* when they satisfy the followings:

1. **(associativity)** For any three elements $g_1, g_2, g_3 \in G$, $(g_1 \cdot g_2) \cdot g_3 = g_1 \cdot (g_2 \cdot g_3)$.

2. **(identity)** There exists a unique element $e \in G$ such that $e \cdot g = g \cdot e = g$ for all $g \in G$.

3. **(inverse)** For any element $g \in G$, there uniquely exists an element $h \in G$ such that $g \cdot h = h \cdot g = e$. h is usually denoted by g^{-1}.

The symbol for the binary operation may be omitted for simplicity: for example, $g_1 \cdot (g_2 \cdot g_3)$ may be written as $g_1(g_2 g_3)$.

(Ex.1) The general linear group on a vector space

Given a d-dimensional vector space V, a set of all invertible linear transformations (or operators) $L : V \to V$ forms a group with the binary operation being the composition of the operators. This group is called the *general linear group*, denoted by $\mathrm{GL}(V)$. By choosing a basis set in V, a linear operator $L \in \mathrm{GL}(V)$ can be represented as an invertible complex matrix. The matrix representation depends on the choice of the basis set.

Definition 2.1.2 (subgroup)**.** Let $(G, \cdot)$ be a group. A *subgroup* H of G is defined as a subset of G satisfying the definition of a group with the same binary operation, that is,

1. Any three elements of $H \subseteq G$ satisfy the associativity.

2. H contains the identity.

3. The inverse of any element in H is included in H.

(Ex.2) The unitary group

Consider a finite-dimensional (complex) vector space equipped with an inner product $\langle \cdot, \cdot \rangle$, which forms a Hilbert space $\mathcal{H}$. A set of all invertible linear operators $L : \mathcal{H} \to \mathcal{H}$ that preserve the inner product between $x, y \in \mathcal{H}$ satisfies the definition of a group with the binary operation being the composition of those linear operators. Thus, this group is a subgroup of the general linear group $\mathrm{GL}(\mathcal{H})$.

For a linear operator L preserving the inner product, it follows that

$$\forall x, y \in \mathcal{H} \quad \langle x, y \rangle = \langle Lx, Ly \rangle = \langle x, L^\dagger L y \rangle \tag{2.1}$$

where $L^\dagger$ is the adjoint operator of L. Thus, $L^\dagger L = L L^\dagger = I$, where I is the identity, or equivalently $L^\dagger = L^{-1}$. This operator is called a *unitary operator*. This group is referred to as the *unitary group* and denoted by $\mathrm{U}(d)$, where d is the dimension of the Hilbert space.

As the unitary group $\mathrm{U}(d)$ is a subgroup of $\mathrm{GL}(\mathcal{H})$, a unitary operator can be represented by a $d \times d$ unitary matrix U, once a basis set is chosen, satisfying that $(U^{-1})_{jk} = (U^\dagger)_{jk} = (U)^*_{kj}$, where $\dagger, *$ are Hermite and complex conjugates, and $(\cdot)_{jk}$ denotes the (j, k)-entry of the matrix.

(Ex.3) The Pauli group

The Pauli group is a subgroup of the unitary group U(2) and is given by

$$\mathcal{P} = \{\pm I, \pm iI, \pm\sigma^x, \pm i\sigma^x, \pm\sigma^y, \pm i\sigma^y, \pm\sigma^z, \pm i\sigma^z\}, \tag{2.2}$$

where i is the imaginary unit. The operators σ^α ($\alpha = x, y, z$) are called the *Pauli operators*, which are Hermitian, that is, $\sigma^\alpha = (\sigma^\alpha)^\dagger$. They satisfy that $\sigma^x\sigma^y = i\sigma^z$, $\sigma^y\sigma^z = i\sigma^x$, $\sigma^z\sigma^x = i\sigma^y$, and $\sigma^\alpha\sigma^\alpha = I$.

The elements of $\mathcal{P}$ are not independent since, for instance, $i\sigma^z$ can be obtained by the composition of σ^x and σ^y. In fact, these three operators $\sigma^x, \sigma^y, \sigma^z$ are enough to express any element in the Pauli group with a finite composition of those operators (and their inverses) under the group operation. Such operators are called *generators* of the group.

A matrix representation of the Pauli operators is often given by the following:

$$I \doteq \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \sigma^x \doteq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma^y \doteq \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma^z \doteq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.3}$$

where the basis set of the matrix representation is chosen to be the eigenvectors of σ^z.

(Ex.4) N-fold tensor product of the Pauli group

For convenience, let the Pauli operators denoted as $\sigma^0 = I, \sigma^1 = \sigma^x, \sigma^2 = \sigma^y$, and $\sigma^3 = \sigma^z$. The N-fold tensor product of the Pauli group, $\mathcal{P}_N$, which is a subgroup of $\mathrm{U}(2^N)$, can be generated by

$$\{\sigma^{n_0} \otimes \sigma^{n_1} \otimes \cdots \otimes \sigma^{n_{N-1}} |\ \ n_0 n_1 \cdots n_{N-1} \in \{0, 1, 2, 3\}^N\}. \tag{2.4}$$

Thus, the elements of $\mathcal{P}_N$ have the form of $s\sigma^{n_0} \otimes \cdots \otimes \sigma^{n_{N-1}}$ with s being ± 1 or $\pm i$. The N-fold tensor products of the Pauli operators are often referred to as *Pauli strings*, and they sometimes are denoted by $\sigma_0^{n_0}\sigma_1^{n_1}\cdots\sigma_{N-1}^{n_{N-1}}$ where $\sigma_j^{n_j} = I \otimes \cdots \otimes I \otimes \sigma^{n_j} \otimes I \otimes \cdots \otimes I$.

(Ex.5) The Clifford group

The Clifford group $\mathcal{C}_N$ is a subgroup of the unitary group $\mathrm{U}(2^N)$ that normalizes the N-fold tensor product of the Pauli group: $\mathcal{C}_N = \{U \in \mathrm{U}(2^N) |\ \forall P \in \mathcal{P}_N, UPU^\dagger \in \mathcal{P}_N\}$.

For example, for $N = 1$, an operator $h = (\sigma^x + \sigma^z)/\sqrt{2}$ is not a member of the Pauli group, and it normalizes the Pauli group as

$$h\sigma^x h^\dagger = \sigma^z,\ \ h\sigma^y h^\dagger = -\sigma^y,\ \ h\sigma^z h^\dagger = \sigma^x. \tag{2.5}$$

Hence, h is a member of $\mathcal{C}_1$.

The matrix representation of the Clifford operator we have seen, $h = (\sigma^x + \sigma^z)/\sqrt{2} \in \mathcal{C}_1$, is represented in the Pauli-z basis set as follows:

$$h \doteq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{2.6}$$

2.1.2 Exponential form of unitary matrices

It is known that any unitary matrix can be written in the form of $\exp(S) = \sum_{n=0}^{\infty} S^n/n!$ with a skew-Hermitian matrix S [60]. For two skew-Hermitian matrices R, S, it can be verified that there exists another skew-Hermitian matrix T such that $\exp(T) = \exp(R)\exp(S)$ by virtue of the Baker-Campbell-Hausdorff formula [61]. It states that T is given by a sequence of iterative commutators of R and S: $T = R + S + [R, S]/2 + [R, [R, S]]/12 + \cdots$, where $[R, S] = RS - SR$. As the commutator $[R, S]$ is skew-Hermitian, $[R, S] = (SR - RS)^\dagger = -[R, S]^\dagger$, T is skew-Hermitian.

Consider a real vector space formed by $d \times d$ skew-Hermitian matrices and let $S_1, \cdots, S_{d^2}$ be the basis set of the space. These bases may be characterized through the commutator, as it can be spanned by the bases:

$$[S_j, S_k] = \sum_{l=1}^{d^2} f_{jkl} S_l, \tag{2.7}$$

where $f_{jkl} \in \mathbb{R}$. One of the choices of the bases is the generalized Gell-Mann matrices [62, 63] with the imaginary unit, $i\lambda^x_{lm}, i\lambda^y_{lm}, i\lambda^z_k$:

$$\begin{aligned}
(\lambda^x_{lm})_{jk} &= \delta_{lj}\delta_{mk} + \delta_{lk}\delta_{mj}, && (2.8)\\
(\lambda^y_{lm})_{jk} &= -i\delta_{lj}\delta_{mk} + i\delta_{lk}\delta_{mj}, && (2.9)\\
\lambda^z_n &= \sqrt{\frac{2}{n(n+1)}}\ \mathrm{diag}(\underbrace{1, \cdots, 1}_{n \text{ times}}, -n, 0, \cdots, 0), && (2.10)\\
\lambda^z_d &= I, && (2.11)
\end{aligned}$$

where $1 \le l < m \le d$, $1 \le j, k \le d$, $1 \le n \le d$, $(M)_{jk}$ indicates the (j, k)-entry of the matrix M, and δ_{jk} is the Kronecker delta. For instance, $d = 3$, by relabeling the bases as

$$\begin{aligned}
&S_1 = \lambda^x_{01},\ S_2 = \lambda^y_{01},\ S_3 = \lambda^z_1,\ S_4 = \lambda^x_{02}, && (2.12)\\
&S_5 = \lambda^y_{02},\ S_6 = \lambda^x_{12},\ S_7 = \lambda^y_{12},\ S_8 = \lambda^z_2,\ S_9 = \lambda^z_3, && (2.13)
\end{aligned}$$

the commutator has the following structure:

$$f_{123} = -1,\ f_{458} = f_{678} = -\frac{\sqrt{3}}{2}, \tag{2.14}$$

$$f_{147} = f_{165} = f_{246} = f_{257} = f_{345} = f_{376} = -\frac{1}{2}, \tag{2.15}$$

and the others are zero. For another choice of the basis set, the generators of $\mathcal{P}_N$ given by Eq. 2.4 can be used for the case with d being 2^N.

2.2 The postulates of quantum mechanics

In this section, we summarize the postulates [64] that quantum-mechanical objects must obey.

In physics, a collection of physical objects of interest is referred to as a *system*. For simplicity, we consider systems with finite degrees of freedom in the present thesis. The first postulate mathematically determines quantum systems and the state of isolated systems, which are those without any interaction with the outside of the system:

Postulate 1 Any quantum system is associated with a complex Hilbert space. A unit vector in the Hilbert space associated with an isolated system represents a *pure state* of the system.

The simplest example of quantum systems is a two-level system associated with a two-dimensional Hilbert space $\mathcal{H}_2$. A pure state of the system is represented by a unit vector $|\psi\rangle \in \mathcal{H}_2$, which can be determined with two complex numbers as the quantum state $|\psi\rangle$ can be written in a linear combination form, so called *superposition*, of orthonormal bases $\{|\phi_0\rangle, |\phi_1\rangle\}$ of $\mathcal{H}_2$: $|\psi\rangle = \alpha|\phi_0\rangle + \beta|\phi_1\rangle$ with complex numbers α and β satisfying that $|\alpha|^2 + |\beta|^2 = 1$. A physical interpretation of the two complex numbers determining the state will be clear in the postulate that will be explained later.

The second postulate of quantum mechanics formalizes how one can observe quantum states, that is, the measurement of quantum states.

Postulate 2 Measurements are described by a set of operators $\{M_m\}$ with the completeness condition $\sum_m M_m^\dagger M_m = I$. The measurement of a quantum state $|\psi\rangle$ yields a *measurement outcome* m with probability $p_m = \langle\psi|M_m^\dagger M_m|\psi\rangle$. When an outcome m occurred, the state $|\psi\rangle$ is transformed to another quantum state, called a *post-measurement state*, as

$$|\psi\rangle \to \frac{M_m|\psi\rangle}{\sqrt{p_m}}. \tag{2.16}$$

The completeness condition guarantees that the sum of the probabilities p_m is unity for any state $|\psi\rangle$ to be measured. The significant concepts in this postulate are that a measurement outcome is probabilistically obtained in a single measurement operation, and that the measurement transforms (or destroys) the current quantum state.

An important class of measurement is *projective measurements*. A projective measurement corresponds to a set of orthogonal projectors $\{P_m\}$ with $P_m P_n = \delta_{m,n} P_m$ where $\delta_{m,n}$ is the Kronecker delta and P_m are Hermitian. This measurement class gives a physical interpretation of α and β seen in the example in the previous postulate. Let us consider a projective measurement on the quantum state $|\psi\rangle = \alpha |\phi_0\rangle + \beta |\phi_1\rangle$ considered in the previous example. One performs a projective measurement $\{P_0 = |\phi_0\rangle\langle\phi_0| , P_1 = |\phi_1\rangle\langle\phi_1|\}$, and it yields a measurement outcome m corresponding to P_m or the basis $|\phi_m\rangle$. The probabilities of getting the measurement outcome $m = 0, 1$ are given by $p_0 = |\alpha|^2$ and $p_1 = |\beta|^2$. The complex numbers α and β therefore characterize the probability of getting the outcome corresponding to the basis states, and the condition that $\langle\psi|\psi\rangle = 1$ can be viewed as the normalization condition of probability: $p_0 + p_1 = 1$.

The next postulate is about composite systems.

Postulate 3 A composite system of two subsystems is associated with a Hilbert space obtained by the tensor product of the Hilbert spaces of the subsystems.

This may be motivated by the following observation. Let us consider a composition of two systems. If these systems are independent of each other, the states of the two systems should be determined independently by a unit vector in the Hilbert spaces of the systems. Fix an orthonormal basis set for each system and consider the projective measurement in that basis for each system. Remembering the probabilistic nature of quantum measurement, possible pairs of measurement outcomes in the two systems are essentially all possible combinations of outcomes. As the systems are independent, the probability corresponding to a pair of outcomes (m, n) should be $p_m \times p_n$. This can be explained by the tensor product structure.

While the above observation follows the tensor product structure, it simultaneously admits non-independent measurement outcomes. It may happen when two systems quantumly interact, which is determined by the next postulate on the time evolution of quantum states:

Postulate 4 The time evolution of a quantum state of a closed system obeys the (time-dependent) Schrödinger's equation, that is,

$$i\hbar \frac{\partial}{\partial t} |\psi(t)\rangle = H(t) |\psi(t)\rangle . \tag{2.17}$$

Here, $\hbar$ is the Dirac constant and $|\psi(t)\rangle$ represents the quantum state at time t. $H(t)$ is called the Hamiltonian operator, which determines the energy of the system and governs the behavior of the physical objects in the system. The Hamiltonian may include interactions between those objects, an external magnetic field, or a quench (a sudden change of parameters in the Hamiltonian).

It is noteworthy that the time evolution preserves the norm of the state vector, $\langle\psi(t)|\psi(t)\rangle$, as follows:

$$\begin{aligned}\frac{\partial}{\partial t}\langle\psi(t)|\psi(t)\rangle &= \left(\frac{\partial}{\partial t}\langle\psi(t)|\right)|\psi(t)\rangle + \langle\psi(t)|\left(\frac{\partial}{\partial t}|\psi(t)\rangle\right) \\ &= \frac{1}{i\hbar}(-\langle\psi(t)|H(t)|\psi(t)\rangle + \langle\psi(t)|H(t)|\psi(t)\rangle) = 0.\end{aligned} \tag{2.18}$$

As the Schrödinger's equation is linear, the preservation of the norm of the state vector essentially states that the time evolution is *unitary*, that is, there is a unitary operator $U(t)$ that maps an initial state to the state at time t: $|\psi(t)\rangle = U(t)\,|\psi(0)\rangle$.

2.3 Quantum information processing

We now introduce essential elements for quantum information processing based on the four postulates of quantum mechanics summarized in Sec. 2.2.

2.3.1 Qubits

A (classical) bit is the most elementary object in classical computers. Bits represent information, such as numbers or letters, in binary form, consisting of 0s and 1s. In quantum information processing, the analog of the classical bit is a quantum bit or *qubit*, which is a two-level system associated with a two-dimensional Hilbert space $\mathcal{H}_2$. Similar to the classical bit, we fix 0 and 1 quantum states, $|0\rangle, |1\rangle \in \mathcal{H}_2$ that form an orthonormal basis set, which is referred to as the *computational basis set*. The computational basis states are typically chosen to be the eigenstates of the two-level system implemented on a physical platform at hand.

Any state of a qubit can be written in a linear combination of the computational basis states. A convenient parametrization of a qubit state $|\psi\rangle$ is the following form:

$$|\psi\rangle = \cos\left(\frac{\theta}{2}\right)|0\rangle + e^{i\phi}\sin\left(\frac{\theta}{2}\right)|1\rangle, \tag{2.19}$$

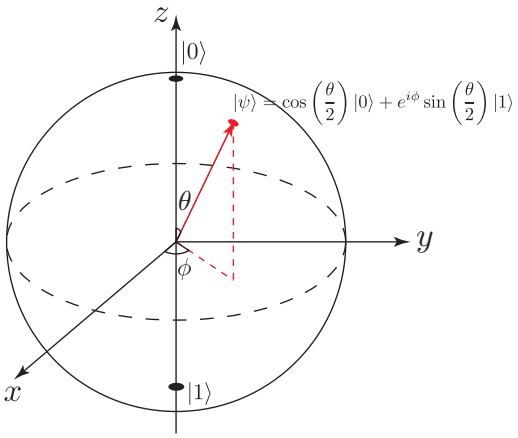

Figure 2.1: Visualization of the Bloch sphere.

where $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi)$. This parametrization allows us to associate a qubit state with a point on the sphere in three-dimensional Euclidean space, so-called the *Bloch sphere*, as shown in Fig. 2.1.

For a multiple-qubit system consisting of N qubits, the associated Hilbert space is given by the tensor product of the Hilbert spaces associated with each qubit. It induces a computational basis set of the N-qubit system,

$$\{|000\cdots0\rangle\,,\ |100\cdots0\rangle\,,\ |010\cdots0\rangle\,,\ |110\cdots0\rangle\,,\ \cdots\,,\ |1\cdots1\rangle\}, \tag{2.20}$$

whose cardinality is 2^N. Here, the inside of $|\cdot\rangle$ is filled by a binary string of length N. Any state of the N-qubit system $|\Psi\rangle$ can be written as a linear combination of the basis states with 2^N complex numbers $\alpha_0, \cdots \alpha_{2^N-1}$ satisfying $\sum_i |\alpha_i|^2 = 1$:

$$|\Psi\rangle = \sum_{i=0}^{2^N-1} \alpha_i \, |i_0 i_1 \cdots i_{2^N-1}\rangle\,, \tag{2.21}$$

where $i_0 i_1 \cdots i_{2^N-1}$ is the binary representation of integer i.

2.3.2 Quantum gate operations

By Postulate 4 of quantum mechanics introduced in Sec. 2.2, qubit states can be controlled according to the Schrödinger's equation of Eq. 2.17. Namely, one can perform an operation on an initial qubit state by designing a Hamiltonian operator for which the evolved state at a particular time t coincides with the state resulting from the action of the desired operation on the initial state. Remembering that the time evolution is unitary and there is a

unitary operator associated with the time evolution, it is convenient to consider the unitary operation and introduce *quantum gates* without seeing the Hamiltonian-based time evolution.

Pauli gates and single-qubit Clifford gates

Let us first consider a simple yet essential set of quantum gates, known as the Pauli gates. There are three single-qubit Pauli gates, which are often denoted by X, Y, and Z with the following matrix representation in the computational basis:

$$X \doteq \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},\ Y \doteq \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},\ Z \doteq \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.22}$$

The action of the X operation on the computational basis states is the bit-flip or NOT operation: $X|0\rangle = |1\rangle, X|1\rangle = |0\rangle$. The action of Z on a superposition state given by $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ is to flip the sign of the coefficient of $|1\rangle$: $Z|\psi\rangle = \alpha|0\rangle - \beta|1\rangle$. Together with Y and the identity operator, those operations form the Pauli group.

Clifford gates are unitary operations that form the Clifford group. Important examples of single-qubit Clifford gates are the Hadamard gate H and the phase gate S, whose matrix representations in the computational basis are given by

$$H \doteq \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},\ S \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/2} \end{pmatrix}. \tag{2.23}$$

Rotation gates

The next example of quantum gates is the rotation gates, which are rotational operations along the x-, y-, and z-axes in the Bloch sphere shown in Fig. 2.1. These operations allow us to map the computational basis states to any state on the Bloch sphere. The rotation gates are defined as follows:

$$R_x(\theta) = \exp\left(-iX\frac{\theta}{2}\right), R_y(\theta) = \exp\left(-iY\frac{\theta}{2}\right), R_z(\theta) = \exp\left(-iZ\frac{\theta}{2}\right), \tag{2.24}$$

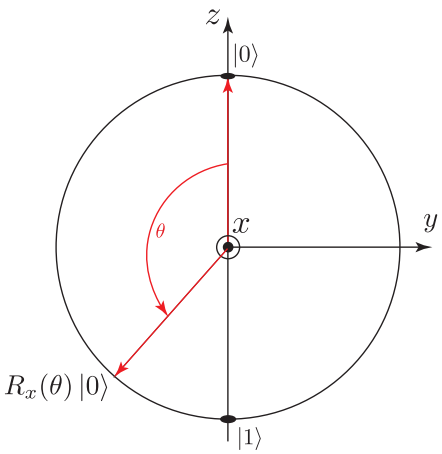

Figure 2.2: Visualization of the action of $R_x(\theta)$ on $|0\rangle$ in the Bloch sphere.

where $\theta \in [0, 2\pi)$. The exponential of operators is defined by the Taylor series, and one can rewrite Eq. 2.24 as follows:

$$R_x(\theta) = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) X, \tag{2.25}$$

$$R_y(\theta) = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) Y, \tag{2.26}$$

$$R_z(\theta) = \cos\left(\frac{\theta}{2}\right) I - i \sin\left(\frac{\theta}{2}\right) Z. \tag{2.27}$$

The rotation gate $R_x(\theta)$ can map the computational basis state $|0\rangle$ to any state parametrized as in Eq. 2.19 with a fixed phase being $\phi = -\pi/2$ as shown in Fig. 2.2. The phase can be changed by another rotation gate $R_z(\theta)$. One can hence generate arbitrary states on the Bloch sphere (up to a global phase) from $|0\rangle$ by sequentially applying $R_x(\theta)$ and $R_z(\phi)$.

Controlled gates

We next introduce important multi-qubit gates, which are controlled gates. A simple example of controlled gates is the controlled NOT gate, also known as the controlled X gate, acting on two qubits. The controlled X gate is a two-qubit Clifford gate that causes the X gate to act on the state of one qubit depending on the state of the other qubit. The qubit referred to is called a *control qubit* and the qubit on which X may act is called a *target qubit*. More precisely, the action of the controlled X gate, CX, is summarized as follows:

$$\mathrm{C}X\,|00\rangle = |00\rangle\,,\ \mathrm{C}X\,|01\rangle = |01\rangle\,,\ \mathrm{C}X\,|10\rangle = |11\rangle\,,\ \mathrm{C}X\,|11\rangle = |10\rangle\,, \tag{2.28}$$

where the first and second qubits are the control and target qubits, respectively.

Multi-qubit gates, such as the controlled X gate, are necessary for generating a general state in multi-qubit systems. For example, the following two-qubit state cannot be generated using only single-qubit operations:

$$\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) = \mathrm{C}X(H \otimes I)\,|00\rangle\,. \tag{2.29}$$

Universal gate sets

In practice, finite kinds of quantum gates are available for information processing, and one applies those quantum gates sequentially to realize a desired unitary operation. As there are uncountably many unitary operations, the set of quantum gates available may not be able to generate any unitary operation within arbitrary precision. It is then essential to check whether the given gate set can cover all unitary operations. The property of gate sets is *universality*, and a gate set with universality is called a *universal gate set*.

An example of a universal gate set is given by an arbitrary single-qubit rotation gate and the controlled X gate. As the combination of R_x and R_z can realize arbitrary rotational operations on the Bloch sphere, the universal gate set is $\{R_x, R_z, \mathrm{C}X\}$.

Another example is the set of Clifford gates and a non-Clifford gate, such as the T gate whose matrix representation in the computational basis is given by

$$T \doteq \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}. \tag{2.30}$$

By applying the T gate many times together with the Hadamard gate, one can approximate any single-qubit rotation operation within an arbitrary precision [65]. Since the controlled X gate and the single-qubit rotation gate with arbitrary angles form a universal gate set, $\{T, H, \mathrm{C}X\}$ forms a universal gate set, which is essentially equivalent to the set containing the T gate and the generators of the Clifford group, $H, S, \mathrm{C}X$.

2.3.3 Measurements

After quantum information represented by qubits is processed by unitary operations, we need to extract the computation result by measurement. In quantum information processing, projective measurements in the computational basis are often adopted. As a measurement destroys the current quantum state, one should note that, for a single run of the measurement,

one probabilistically obtains an outcome associated with each computational basis to which the state is projected. This nature is crucial to designing quantum algorithms.

2.3.4 Pictorial representation of quantum circuits

As qubits, quantum gates, and measurements have been introduced, one can construct a *quantum circuit* in which one prepares an initial state, applies quantum gates, and performs measurements. A quantum circuit is often represented in a pictorial form, such as shown in Fig. 2.3 (a). The picture is read from left to right. The horizontal lines represent qubits, which are *initialized* to be $|0\rangle$ in the beginning. The white box lying down on the qubits is the unitary operation U being performed. The boxes at the end of the circuit mean the measurement, which is often the projective measurement in the computational basis.

The unitary operation U represented by the white box may be comprised of more elementary quantum gates, such as single-qubit gates and the controlled X gate. They are drawn in Fig. 2.3 (b), where V can be X, H, $R_z(\theta)$, and so on. An example of a quantum circuit is shown in Fig. 2.3 (c) depicting the operations in Eq. 2.29.

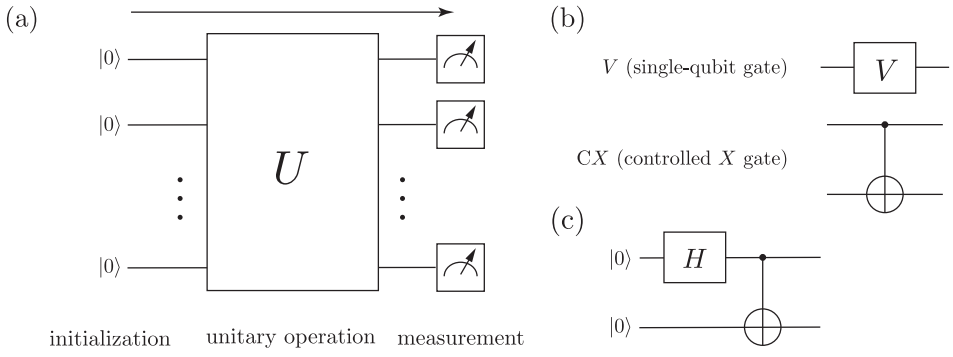

Figure 2.3: (a) A general pictorial representation of a quantum circuit. (b) Pictorial representations for single-qubit gates $V = X, H, R_z(\theta), \cdots$ and the controlled X gate. (c) The example of the representation of Eq. 2.29.

2.4 Characterization of quantum states and unitary operations

As basic concepts and tools for quantum information processing have been introduced, one can generate, process, and measure various quantum states. To investigate the properties of quantum states and unitary operations, we summarize several important notions for characterizing quantum states and unitary operations.

2.4.1 Entanglement entropy

One of the important properties of quantum states is *entanglement*. Suppose that we have two qubits A and B, associated with Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, in a state $|\Psi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$. The state is said to be a product state if $|\Psi\rangle_{AB}$ can be written in a tensor product of two states of A and B, i.e., there exist $|\psi\rangle_A \in \mathcal{H}_A$ and $|\psi\rangle_B \in \mathcal{H}_B$ such that $|\Psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B$. If $|\Psi\rangle_{AB}$ is not a product state, it is called an *entangled* state.

An example of an entangled state is $|\Psi\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$, which we have seen previously in Eq. 2.29. This state exhibits a non-trivial correlation in measurement. Consider the projective measurement for the first qubit, that is, $\{P_0 = |0\rangle\langle 0| \otimes I, P_1 = |1\rangle\langle 1| \otimes I\}$. The probability p_m of getting the outcome $m = 0, 1$ is equal to $1/2$, and the associated post-measurement state becomes

$$|\Psi\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \rightarrow \frac{P_m |\Psi\rangle}{\sqrt{p_m}} = |m\rangle \otimes |m\rangle \,, \tag{2.31}$$

where the post-measurement state of the second qubit is certainly correlated to the measurement outcome of the *first* qubit. While no operation is performed on the second qubit, the measurement on the first qubit affects the state of the second qubit even if they are spatially apart as long as they remain coherent. This non-local correlation is a key signature of quantumness for distinguishing quantum physics and classical physics [66, 67]. The entanglement has also been utilized in quantum information processing and quantum communication in various ways [68].

To quantify entanglement of pure states, we introduce the *entanglement entropy* using the Schmidt decomposition. Consider a bipartite system with two subsystems A and B associated with Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$ whose dimensions are d_A and $d_B (\geq d_A)$, respectively. Using the Schmidt decomposition, any pure state $|\Psi\rangle_{AB} \in \mathcal{H}_A \otimes \mathcal{H}_B$ in the bipartite system can be

written in the following form:

$$|\Psi\rangle_{AB} = \sum_{i=0}^{d_A-1} \sqrt{\lambda_i}\, |a_i\rangle_A\, |b_i\rangle_B\,, \tag{2.32}$$

where $\lambda_i \geq 0$, and $\{|a_i\rangle_A\}$ and $\{|b_i\rangle_B\}$ are orthonormal states in $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively. The entanglement entropy $S(|\Psi\rangle_{AB})$ of the state with respect to the bipartition is now defined as

$$S(|\Psi\rangle_{AB}) = -\sum_{i=0}^{d_A-1} \lambda_i \log_2 \lambda_i. \tag{2.33}$$

When one performs the projective measurement on the subsystem A using $\{|a_i\rangle_A\}$, each λ_i is understood as the probability of getting the measurement outcome corresponding to $|a_i\rangle_A$, which is certainly correlated to the state of the other subsystem B. Hence, the entanglement entropy quantifies the statistical complexity of pairs of correlated orthonormal states. For example, the quantum state in Eq. 2.31 can be Schmidt-decomposed with $\lambda_0 = \lambda_1 = 1/2$, and hence the entanglement entropy is 1, while product states have zero entanglement entropy as they always have a single pair of correlated states.

This discussion motivates us to consider a statistical mixture of pure states. As the projective measurement with $\{|a_i\rangle_A\}_i$ pins the state of the subsystem B in the corresponding state, we have a probabilistic mixture $\{(\lambda_i, |b_i\rangle_B)\}_i$ with the corresponding probability being λ_i. An alternative expression of quantum states, referred to as the *density operator*, allows for dealing with such a mixture of pure states in a simple way. The density operator ρ of a mixture $\{(p_i, |\phi_i\rangle)\}_{i=1}^M$ is defined as

$$\rho = \sum_{i=1}^{M} p_i\, |\phi_i\rangle\langle\phi_i|\,. \tag{2.34}$$

For an observable O, its expectation value on the mixture can be calculated as $\mathrm{Tr}[\rho O]$, which is compatible with the sum of the expectation values on the individual pure states as

$$\mathrm{Tr}[\rho O] = \sum_i p_i\, \mathrm{Tr}[|\phi_i\rangle\langle\phi_i|\, O] = \sum_i p_i\, \langle\phi_i|O|\phi_i\rangle\,. \tag{2.35}$$

Now, the entanglement entropy of pure states can be defined with the density operator expression. Consider a pure state $|\Psi_{AB}\rangle$ in a bipartite system AB. Another definition of the entanglement entropy is given as

$$S(|\Psi_{AB}\rangle) = -\,\mathrm{Tr}_A[\rho_A \log_2 \rho_A] = -\,\mathrm{Tr}_B[\rho_B \log_2 \rho_B], \tag{2.36}$$

where $\mathrm{Tr}_A, \mathrm{Tr}_B$ are partial traces over the Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$, respectively. ρ_A and ρ_B are reduced density operators: $\rho_A = \mathrm{Tr}_B[|\Psi_{AB}\rangle\langle\Psi_{AB}|], \rho_B = \mathrm{Tr}_A[|\Psi_{AB}\rangle\langle\Psi_{AB}|]$. The logarithm of density operators is defined as $\log_2 \rho = \sum_i (\log_2 \lambda_i) |e_i\rangle\langle e_i|$ where $|e_i\rangle$ and λ_i are the i-th eigenvector and eigenvalue of ρ. Thus, the entropy $S(|\Psi_{AB}\rangle)$ is equivalent to

$$S(|\Psi_{AB}\rangle) = -\sum_i \lambda_i \log_2 \lambda_i, \tag{2.37}$$

where λ_i's are the eigenvalues of ρ_A or ρ_B.

2.4.2 The presence of non-Clifford gates

Another important characteristic that unitary operations can have is the presence of non-Clifford gates required to implement them. Simulating quantum circuits on a classical computer is hard in general, since it requires considering an exponentially large number of degrees of freedom with respect to the number of qubits. However, the evolution of the initial state $|00\cdots0\rangle$ under a quantum circuit consisting of only Clifford gates and Pauli measurements can be described in the *stabilizer form*, which significantly reduces the number of degrees of freedom that have to be taken into account [69, 70]. In this sense, the presence of non-Clifford gates makes simulating the quantum circuit complex.

The stabilizer formalism is one to describe quantum states with Pauli operators. For instance, there exists a two-qubit quantum state $|\Psi\rangle$ *stabilized by* $\sigma_1^x\sigma_2^x$ and $\sigma_1^z\sigma_2^z$ simultaneously, that is,

$$\sigma_1^x\sigma_2^x |\Psi\rangle = |\Psi\rangle\,, \text{ and } \sigma_1^z\sigma_2^z |\Psi\rangle = |\Psi\rangle\,, \tag{2.38}$$

The quantum state with this condition can be uniquely determined (up to the global phase), which is $(|00\rangle + |11\rangle)/\sqrt{2}$. Note that the quantum state is also stabilized by $-\sigma_1^y\sigma_2^y = (\sigma_1^x\sigma_2^x)(\sigma_1^z\sigma_2^z) = (\sigma_1^z\sigma_2^z)(\sigma_1^x\sigma_2^x)$. Furthermore, there are no other Pauli operators that stabilize the quantum state except the identity I_1I_2, implying that a subgroup of the 2-fold Pauli group is induced: $\{I_1I_2, \sigma_1^x\sigma_2^x, -\sigma_1^y\sigma_2^y, \sigma_1^z\sigma_2^z\}$. Thus, the quantum state $(|00\rangle + |11\rangle)/\sqrt{2}$ is *the* stabilizer state associated with this subgroup.

More formally, the stabilizer state is a quantum state $|\psi\rangle$ for which there exists a subgroup $\mathcal{S}$ of the N-fold Pauli group such that $S|\psi\rangle = |\psi\rangle$ for all $S \in \mathcal{S}$ and $[S, R] = 0$ for all pairs (S, R) of distinct operators $S \neq R$ in $\mathcal{S}$. The stabilizer state associated with $\mathcal{S}$ is uniquely determined when there are N generators in $\mathcal{S}$, which is much more efficient than representing the state as a complex-valued vector in the 2^N-dimensional Hilbert space. Furthermore,

the action of a Clifford operator on the generators defines another stabilizer group, indicating that the evolution of a stabilizer state under a Clifford circuit can be fully tracked by only looking at the evolution of the generators.

Not all quantum states are the stabilizer state, for example, $TH|0\rangle = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$. One can simply check that none of the Pauli operators, except for the identity, stabilizes the state. Due to the presence and action of the non-Clifford gate, T, the generators of the stabilizer group associated with $|+\rangle = H|0\rangle$ are mapped to non-Pauli operators, implying that the stabilizer formalism will break down. In this sense, the presence of non-Clifford gates is a type of complexity of unitary operations. It is worth noting that this is a distinct aspect of quantum characteristics from entanglement, as it can be generated by Clifford operations, just like $(|00\rangle + |11\rangle)/\sqrt{2}$.

Measuring how far a given state is from stabilizer states, i.e., quantifying *non-stabilizerness*, is not straightforward in general [71,72]. A simple measure is the *stabilizer Rényi entropy* $M_2(|\psi\rangle)$ [73], defined as

$$M_2(|\psi\rangle) = -\log \sum_{n=0}^{d-1} \left(\frac{\langle\psi|P_n|\psi\rangle^2}{d} \right)^2 - \log d, \tag{2.39}$$

where P_n is the Pauli string $\sigma^{n_0} \otimes \sigma^{n_1} \otimes \cdots \otimes \sigma^{n_{N-1}}$, and $d = 2^N$. This quantity measures how nontrivially the expectation values of P_n distribute. The stabilizer entropy is zero for stabilizer states, but not for non-stabilizer states, for instance, for $d = 2$,

$$M_2(|+\rangle) = -\log\left(\frac{1}{d^2} + \frac{1}{d^2} + 0 + 0\right) - \log d = 0, \tag{2.40}$$

$$\begin{aligned} M_2(T|+\rangle) &= -\log\left(\frac{1}{d^2} + \frac{\cos^4(\pi/4)}{d^2} + \frac{\sin^4(\pi/4)}{d^2} + 0\right) - \log d \\ &= \log\frac{4}{3}. \end{aligned} \tag{2.41}$$

2.4.3 Haar-random unitaries and t-designs

We also introduce an essential notion in quantum information, the uniform random sampling of unitary operators from the unitary group. The uniform randomness is often referred to as *Haar randomness*. It is considered to yield unitaries that are complicated in many different senses, including entanglement and non-stabilizerness [74,75]. It therefore plays versatile roles in various contexts, such as quantum cryptography [76, 77], quantum system benchmarking [78–80], efficient observable estimation [81], information scrambling [82], and characterization of complex quantum dynamics [83].

The randomness is also known as a fast scrambler of information in the initial quantum state.

As it is generally challenging to uniformly sample unitaries as a quantum circuit with a given universal gate set, the construction of random circuits and ones with a certain level of randomness has been a central interest in this context [51, 52]. The exact Haar sampling may not be necessary for some applications; rather, it may be of interest to compute the expectation values of polynomials in matrix elements of Haar-random unitaries [79]. Such expectation values can be calculated with $f_t(U, U^\dagger) = \mathrm{Tr}\big[BU^{\otimes t}A(U^\dagger)^{\otimes t}\big]$ for t order polynomials. Thus, the goal is to construct a unitary ensemble $\mathcal{E} = \{(p_i, U_i)\}_i$ that reproduces the expectation values of any t-th order polynomials f_t over Haar-random unitaries:

$$\sum_i p_i f_t(U_i, U_i^\dagger) = \mathbb{E}_{U\sim\mathrm{Haar}} f_t(U, U^\dagger), \tag{2.42}$$

where $p_i \geq 0$ and $\sum_i p_i = 1$. Such a unitary ensemble is referred to as a *unitary t-design*.

Unitary t-designs are hierarchical in the sense that, if $\mathcal{E}$ is a t-design, $\mathcal{E}$ forms s-designs ($s = 1, 2, \cdots, t-1$). If an ensemble $\mathcal{E}$ is a t-design for any t, $\mathcal{E}$ is exactly Haar random. In this sense, unitary designs are one of the complexity axes for unitaries along this statistical property.

2.4.4 Networks in the Hilbert space

Considering that unitary operators map a computational basis state to a superposition state of the basis states, unitary operators can be viewed as a network whose nodes and edges represent the basis states and their transition amplitudes, respectively. Network-scientific characterization of unitary operators has been one of the central interests in many-body quantum physics and quantum information [84–89].

Some of the network-scientific studies of unitary operators consider converting a Hermite operator associated with the unitary into an undirected network, instead of defining a network directly from the unitary. A unitary matrix U represented in the computational basis can be written in the form of

$$U = e^{-iG}, \tag{2.43}$$

where G is Hermitian. There are many different Hermitian matrices that generate the same unitary matrix. To uniquely determine it, the principal natural logarithm is usually adopted, that is, $G = i\mathrm{Log}(U)$ where all eigenvalues of G are in the range $[-\pi, \pi)$. As a symmetric matrix represents an

undirected network, the Hermitian matrix G can be considered an undirected network.

To illustrate it more precisely, G can be decomposed by the generalized Gell-Mann matrices $\lambda^x_{lm}, \lambda^y_{lm}, \lambda^z_k$:

$$G = \sum_{l<m}^{d} (a_{lm}\lambda^x_{lm} + b_{lm}\lambda^y_{lm}) + \sum_{n=1}^{d} c_n \lambda^z_n, \tag{2.44}$$

where a_{lm}, b_{lm}, c_n are real. As a_{lm} and b_{lm} determine the real and imaginary part of the (l, m)-entry of G, they represent the edge weights between the computational basis states $|l\rangle$ and $|m\rangle$ $(l \neq m)$. The weight of the self-loop on $|n\rangle$ is then given by c_n. This decomposition allows for interpreting G as a weighted network.

Another network interpretation can be obtained by the percolation rule [90], which provides an unweighted network. Consider the Hermitian matrix G in the form of $G = \sum_k G_k |k\rangle\langle k| + \sum_{l\neq m} G_{lm} |l\rangle\langle m|$. We now define a matrix A whose elements are determined through the following percolation process:

$$(A)_{jk} = \begin{cases} 1 & \text{if } |G_j - G_k| \leq |G_{jk}| \\ 0 & \text{if } |G_j - G_k| > |G_{jk}| \end{cases}, \tag{2.45}$$

implying that A encodes the effective transitions between the computational basis states.

Chapter 3

Classical and quantum machine learning

In this chapter, we begin introducing a machine learning setting that will be considered throughout the present thesis. We also provide a brief summary of neural networks and discuss a key phenomenon in learning that will be of interest in the latter chapters. On top of these notions in machine learning, we will summarize several quantum machine learning models and their characterizations of learning behavior, to clarify the motivation behind the work presented in this thesis.

3.1 Supervised learning

In this section, we provide the definition and example models of *supervised learning* [30], as it is an important class of machine learning and will be of interest throughout the present thesis.

In supervised learning, we typically consider input data $x \in \mathbb{R}^N$ and label data $y \in \mathbb{R}^M$ that are represented by real vectors in the Euclidean spaces of dimension N and M, respectively. The input and label spaces $\mathcal{X}, \mathcal{Y}$ are hence subspaces of the Euclidean spaces: $\mathcal{X} \subseteq \mathbb{R}^N, \mathcal{Y} \subseteq \mathbb{R}^M$. The collection of empirical data, called *training dataset* $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}_n$, contains inputs and labels as a pair $(x^{(n)}, y^{(n)}) \in \mathcal{X} \times \mathcal{Y}$. It is assumed that there exists some true function $F : \mathcal{X} \to \mathcal{Y}$ that generated the dataset with some noise. The goal of supervised learning is to model and infer the true function using the dataset. Defining a model function $f(\cdot|\theta) : \mathcal{X} \to \mathcal{Y}$ parametrized by $\theta \in \mathbb{R}^L$, one computes the quality of the model function on the *loss function* $\mathcal{L}(\theta|\mathcal{D}) : \mathbb{R}^L \to [0, \infty)$ that essentially measures the difference between the predicted label $f(x|\theta)$ and the correct label y for all $(x, y) \in \mathcal{D}$. We omit $\mathcal{D}$

in the notation of the loss function for readability unless explicitly needed. The learning process is tuning the parameter θ to minimize the loss value $\mathcal{L}(\theta)$.

A simple example of supervised learning is the regression problem. For instance, consider a dataset $\mathcal{D} = \{(x^{(n)}, y^{(n)}) | y^{(n)} = F(x^{(n)}) + \epsilon_n, n = 1, \cdots, D\}$, where $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}$, $\epsilon_n \in \mathbb{R}$ is noise, and $F : \mathcal{X} \to \mathcal{Y}$ is the true function. The simplest model function is a linear function, $f(x|a, b) = ax + b$, where $a, b \in \mathbb{R}$ are tunable parameters. A typical loss function is the mean squared loss $\mathcal{L}(a, b) = (1/D) \sum_{(x,y)\in\mathcal{D}} |y - f(x|a, b)|^2$.

Another important example of supervised learning is the classification problem. The task is to classify input data into the corresponding class. The label is typically given by the *one-hot vector*, which is usually interpreted as indicating the likelihood of the input data belonging to each class. For instance, a dataset of a binary classification problem is typically given by $\mathcal{D} = \{(x^{(n)}, y^{(n)}) \in \mathcal{X} \times \mathcal{Y} \mid \mathcal{X} \subseteq \mathbb{R}^N, \mathcal{Y} = \Delta_1, y^{(n)} \in \{l_0, l_1\} \subset \mathcal{Y}, n = 1, \cdots, D\}$, where $\Delta_1 = \{(p, 1-p)^\mathsf{T} | 0 \leq p \leq 1\}$, $l_0 = (1, 0)^\mathsf{T}$ and $l_1 = (0, 1)^\mathsf{T}$ are called *one-hot vectors* and indicate to which class the corresponding input data is classified. For a model function $f(\cdot|\theta) : \mathcal{X} \to \mathcal{Y} = \Delta_1$, the loss function is usually chosen to be the cross-entropy loss:

$$\mathcal{L}(\theta) = -\frac{1}{D} \sum_{(x,y)\in\mathcal{D}} \sum_{c=1}^{2} (y)_c \ln [(f(x|\theta))_c + \delta], \tag{3.1}$$

where $(\cdot)_c$ indicates the c-th entry of the vector, and δ is a small positive value for computational stability.

3.1.1 Neural networks

One of the popular ML models for supervised learning is *neural networks* (NNs). An NN processes input information x with a layer structure where each layer consists of some nodes, and nodes in adjacent layers are connected by weighted edges. Mathematically speaking, the collection of the nodes in the l-th layer is given by a vector $x^{[l]}$, $l = 1, 2, \cdots, L$, where $x^{[1]} = x$. Each node is equipped with a non-linear activation function a that acts on $x^{[l]}$, $z^{[l]} = a(x^{[l]})$. The node values $z^{[l]}$ are called *features*, and the vector space in which they live is the *feature space*. The features in the l-th and $(l+1)$-th layers are then associated by a weight matrix $W^{[l]}$ together with a bias vector $b^{[l]}$, that is,

$$x^{[l+1]} = W^{[l]} z^{[l]} + b^{[l]}, \quad z^{[l+1]} = a(x^{[l+1]}). \tag{3.2}$$

Thus, once input information is given for the node vector $x^{[1]}$ in the first layer, it is processed by the weight matrices $W^{[l]}$ and bias vectors $b^{[l]}$ through

all the layers. These weight and bias parameters are to be trained to obtain a desired output at the final layer.

At the final layer, another activation function $a_{\rm f}$ may be chosen based on the ML task under consideration. For example, for a C-class classification task, $a_{\rm f}$ is typically chosen to be the softmax function:

$$\left(a_{\rm f}(x^{[L]})\right)_i = \frac{e^{x_i^{[L]}}}{\sum_{j=1}^{C} e^{x_j^{[L]}}},\ (i = 1, 2, \cdots, C), \tag{3.3}$$

so that the range of the model function becomes compatible with the label space $\mathcal{Y} = \Delta_C = \{y \in \mathbb{R}^C | \sum_{i=1}^{C} (y)_i = 1, y_i \geq 0\}$.

The learning procedure is to minimize the loss $\mathcal{L}(\theta|\mathcal{D})$ by varying θ, where θ represents all the weight and bias parameters in the NN. It is often performed by a gradient-based method, such as gradient descent. It shifts the current parameter values θ by some amount proportional to the gradient of the loss function:

$$\theta \leftarrow \theta - \eta \boldsymbol{\nabla}_\theta \mathcal{L}(\theta|\mathcal{D}), \tag{3.4}$$

where η (< 1) is called a *learning rate*, and this single update is called an *epoch*. For updated parameters, the loss $\mathcal{L}(\theta|\mathcal{D})$ will be computed again. By iterating this update, having many epochs, one aims to find better parameter values that yield a smaller loss.

3.1.2 Overfitting

The goal of ML is not to perfectly minimize the loss for the training dataset, but to obtain reasonable predictions for unseen data that are not included in the training dataset. It leads to an important concept in ML, *overfitting*. It is a phenomenon where the model function fits the training data too much to infer the underlying true function. Such a model function can have an almost zero loss value for the training dataset, but could not be capable of making reasonable predictions for unseen data.

To illustrate it numerically, let us introduce a polynomial regression problem. In Fig. 3.1, we assume that a true quadratic function F (dashed curve) generated nine data points with small random noise ϵ: $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{9}$, where $y^{(n)} = F(x^{(n)}) + \epsilon_n$. Our model function is a polynomial f_d with degree d. For a fixed d, there are $d + 1$ trainable parameters to minimize the mean squared loss $(1/9)\sum_n (y^{(n)} - f_d(x^{(n)}))^2$.

An example of training results for $d = 1, 3$, and 7 is shown by the blue, orange, and green curves in Fig. 3.1. For $d = 3$ (orange), the model polynomial fits the data points and gives a reasonable function, whereas the model

polynomial with $d = 1$ does not fit them at all. It essentially suggests that the model polynomial with $d = 1$ does not have a sufficient number of parameters to fit (or express) the data points, but the polynomial with $d = 3$ does. There are even more parameters in the model polynomial with $d = 7$, and in fact, the green curve in Fig. 3.1 is on all the data points. It means that the total loss is minimized to almost zero. However, the trained model polynomial does not yield a reasonable function and fails to infer the true function F, which may result in a significant loss for unseen data. The model polynomial with $d = 7$, thus, overfits the training data and cannot be generalized to unseen data.

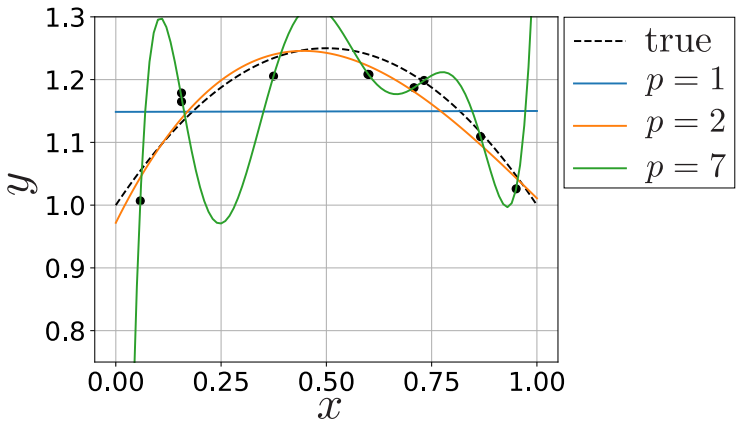

Figure 3.1: A simple demonstration of underfitting and overfitting in polynomial regression.

In practice, it is however impossible to check if the model function is overfitting, as the true function is unknown, but one can evaluate the tendency of overfitting by dividing the dataset into two subsets, the *training* and *validation* datasets. The training dataset is used to update training parameters, implying that these parameters explicitly contain information from the dataset. The validation dataset is not used for this purpose, but rather to monitor the generalizability or *generalization performance* of the model function for validation data. Depending on the monitored generalization performance, one may adjust several parameters, known as hyperparameters, such as the learning rate or the degree of the model polynomial, as in the above example. This is involved in learning, but hyperparameters are not changed during the update of training parameters. In this sense, the validation dataset implicitly contributes to the overall learning performance. To evaluate the overall performance, one may use another sub-dataset that is not included in either the training or validation process.

Various prescriptions for avoiding overfitting have been extensively discussed in the field [91, 92], but an essential key idea is *regularization*. The *regularization function* or *regularizer* is a function of training parameters that is additionally introduced in the loss function. Namely, the minimization problem of the loss function $\mathcal{L}(\theta)$ is now given by

$$\min_{\theta} \Big(\mathcal{L}(\theta|\mathcal{D}) + \lambda \mathcal{R}(\theta) \Big), \tag{3.5}$$

where $\mathcal{R}(\theta)$ is the regularizer, which usually does not depend on the training data explicitly. λ is a constant and can be chosen through the validation process. The regularizer effectively provides a constraint in the parameter space. For instance, $\mathcal{R}(\theta)$ may be the L2-norm of the parameters: $\mathcal{R}(\theta) = |\theta|^2$ in polynomial regression, suppressing the contributions of less relevant degrees in the following sense. The model polynomial of degree d is given by $f_d(x) = \sum_{i=0}^{d} \theta_i x^i$. When all degrees x^i equally contribute to fitting data points, $|\theta_i|$ are almost equal, yielding large $|\theta|^2$. If we have too large d for the dataset, there are many different parameter configurations for which $\mathcal{L}(\theta) \approx 0$. It implies that minimizing $\mathcal{L}(\theta)$ together with $|\theta|^2$ will choose a configuration in which some of θ_i may be zero. In this sense, the regularizer could allow for expressing the data points with essential degrees, while disregarding the other degrees.

3.2 Quantum machine learning

The brief idea of quantum machine learning (QML) is to integrate quantum systems or quantum algorithms into ML frameworks. The quantum system may be utilized for a feature map, i.e., the *quantum feature map*. There are various types of such QML models [41, 93, 94], but in this chapter, we summarize two types: *quantum circuit learning* and *quantum reservoir model*, focusing on the required engineering to clarify the motivation of this thesis.

We explain the QML models in the supervised learning setting with classical input information, while those models also work for quantum inputs. We begin with introducing several methods for encoding classical information into quantum states [95].

3.2.1 Encoding schemes for QML

One straightforward method to encode classical inputs is the *amplitude encoding*, whose schematic diagram is depicted in Fig. 3.2(a). Fix the number of qubits, denoted by N. As an N-qubit quantum state can be spanned by

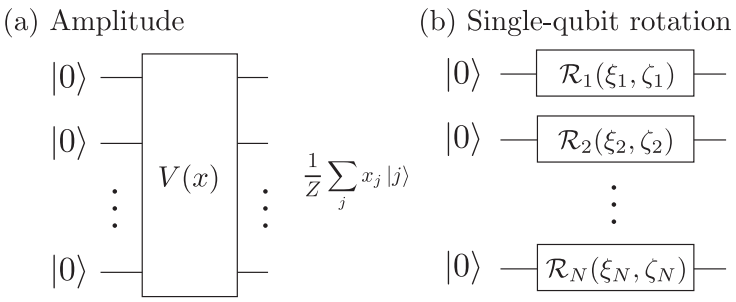

Figure 3.2: Schematic representation of (a) the amplitude encoding and (b) the single-qubit rotation encoding.

the computational basis states, one can load $(2^N - 1)$ independent variables $x_0, \cdots, x_{2^N-2}$ into the amplitudes of the basis states:

$$|0\rangle \xrightarrow{V(x)} \frac{1}{Z}\big(x_0 |0\rangle + x_1 |1\rangle + \cdots + x_{2^N-1} \big|2^N - 1\big\rangle\big), \tag{3.6}$$

where $V(x)$ is the unitary operator to prepare the state and Z is the normalization factor, $Z = \sqrt{\sum_{j=0}^{2^N-1} x_j}$. If the dimension of the input space $\mathcal{X}$ of our dataset is more than 2^N, dimensionality reduction algorithms are necessary, but one may not often face the case, as this encoding allows us to load exponentially many variables. It then seems useful in terms of the amount of information that can be encoded; however, if arbitrary input is allowed, this encoding method may require a deep circuit [64], involving two-qubit gates. This method may hence be experimentally demanding, as two-qubit gates generally have lower fidelity than single-qubit gates in general [96, 97].

A popular encoding method for QML is the *single-qubit rotation encoding*, which associates a two-dimensional value to a point on the Bloch sphere in each qubit. Using the Bloch sphere representation, two variables $\xi \in [0, \pi], \zeta \in [0, 2\pi)$ can be encoded into a qubit as follows:

$$\cos\left(\frac{\xi}{2}\right)|0\rangle + e^{i\zeta}\sin\left(\frac{\xi}{2}\right)|1\rangle\,. \tag{3.7}$$

The single-qubit rotation encoding method prepares this state for each qubit to encode $2N$ variables ξ_j, ζ_j $(j = 1, \cdots, N)$. As shown in Fig. 3.2(b), letting $\mathcal{R}_j(\xi_j, \zeta_j)$ denote the single-qubit rotation acting on the j-th qubit to prepare the state of Eq. 3.8, the total encoding unitary $V(\xi_1, \cdots, \xi_N, \zeta_1, \cdots, \zeta_N)$ is given by

$$V(\xi_1, \cdots, \xi_N, \zeta_1, \cdots, \zeta_N) = \bigotimes_{j=1}^{N} \mathcal{R}_j(\xi_j, \zeta_j). \tag{3.8}$$

Dimensionality reduction of input information is often required in this encoding method, as it has a polynomially large number of degrees of freedom with respect to N, in contrast to the amplitude encoding. The single-qubit rotation encoding may however be easier and more accurate to perform since it consists of only single-qubit operations. Loading information in both the amplitude and phase is not always considered, and sometimes one of them is adopted. The ranges for the variables ξ, ζ may also be restricted to a smaller space [50].

3.2.2 Quantum circuit learning

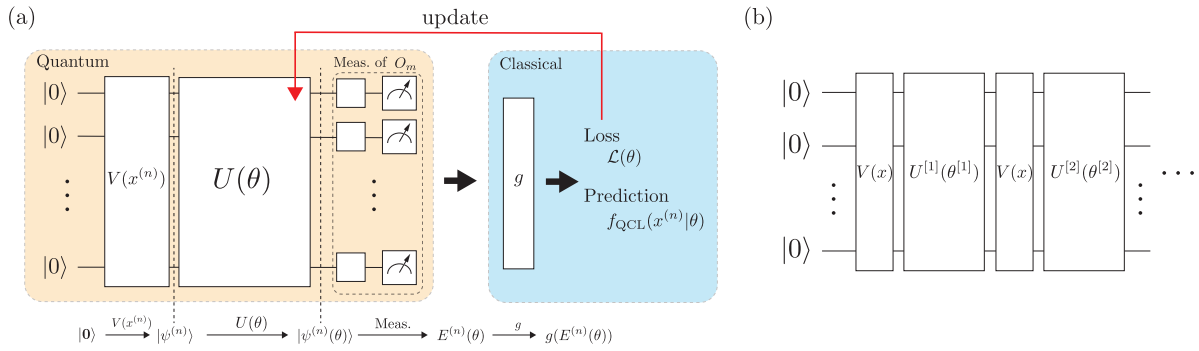

Figure 3.3: (a) Schematic diagram of QCL. (b) shows the re-uploading method for encoding.

Quantum circuit learning (QCL) [40, 41] is the QML algorithm using parametrized quantum circuits (PQCs) [1]. The schematic architecture of QCL is depicted in Fig. 3.3. Consider a dataset $\mathcal{D} = \{(x^{(n)}, y^{(n)}) \in \mathcal{X} \times \mathcal{Y} \mid \mathcal{X} \subseteq \mathbb{R}^{d_{\mathrm{i}}}, \mathcal{Y} \subseteq \mathbb{R}^{d_{\mathrm{o}}}\}_{n=1}^{D}$. We encode an input data in a quantum state $|\psi^{(n)}\rangle$ by applying a unitary operation $V(x^{(n)})$ to the initial state prepared to be $|\mathbf{0}\rangle = |0\rangle^{\otimes N}$, $|\psi^{(n)}\rangle = V(x^{(n)})|\mathbf{0}\rangle$. The encoding state is now mapped to another quantum state by a parametrized unitary operation $U(\theta)$ where θ represents training parameters. $U(\theta)$ is often given by a sequence of two-qubit gates and parametrized single-qubit gates. On the final state $|\psi^{(n)}(\theta)\rangle = U(\theta)|\psi^{(n)}\rangle$, we measure the expectation values of M observables $\{O_m\}_{m=1}^{M}$ that are some of the Pauli strings. Let us denote the expectation values by $E^{(n)}(\theta) = (E_1^{(n)}(\theta), \cdots, E_M^{(n)}(\theta))^{\mathsf{T}}$ where $E_m^{(n)}(\theta)$ is associated with the expectation value of O_m with respect to $|\psi^{(n)}(\theta)\rangle$. The final output of the

[1] This type of QML is often referred to as variational quantum algorithms as well.

QCL algorithm is then given by

$$f_{\mathrm{QCL}}(x^{(n)}|\theta) = g(E^{(n)}(\theta)), \tag{3.9}$$

where g is a fixed function that transforms $E^{(n)}(\theta)$ to be compatible with the label y_n. For instance, g can be chosen to be the softmax function for classification tasks. The quality of $f_{\mathrm{QCL}}(x^{(n)}|\theta)$ is measured by the loss function,

$$\mathcal{L}(\theta) = \frac{1}{D}\sum_n l(y^{(n)}, f_{\mathrm{QCL}}(x^{(n)}|\theta)), \tag{3.10}$$

where l is a loss for each data sample. The parameter values θ in the parametrized circuit are then trained to minimize the value of the loss function.

The encoding method in QCL can be the amplitude encoding or the single-qubit rotation encoding. There is another encoding method popular in QCL, known as *data reuploading*, where input information is injected repeatedly. The reuploading scheme results in alternating layers of the encoding unitary $V(x)$ and parametrized unitaries $U^{[l]}(\theta^{[l]})$, where $\theta^{[l]}$ represents the tunable parameters in the l-th layer, resulting in the following unitary for the L-layer circuit:

$$U^{[L]}(\theta^{[L]})V(x)\cdots U^{[2]}(\theta^{[2]})V(x)U^{[1]}(\theta^{[1]})V(x). \tag{3.11}$$

This method can generate a non-trivial non-linear map for inputs that cannot be implemented with a single injection of inputs, leading to the universal approximation capability [57].

We, however, should note that measurement is crucial to training behavior. First, the expectation values $E_m^{(n)}(\theta)$ must be precisely estimated to accurately evaluate the current loss value. Additionally, the training process is typically performed using a gradient-based method, which involves estimating the gradient based on measurements. If the precision of the estimation is insufficient, the current parameter values may not be updated along the desired direction.

QCL formally resembles NNs and works like them. QCL also appears robust to noise in the quantum device, since the training procedure may find suitable parameters that renormalize the noise effects. This suggests the possibility of implementing QCL on physical qubits without having the overheads of fault-tolerant computation [23–26].

It was however turned out that training quantum circuits is a fundamental obstacle [42, 43, 98]. This is illustrated as the *expressibility-trainability trade-off*. The expressibility is defined by the search space in the Hilbert space

of the PQC. The tradeoff then states that the amplitude of the parameter update in the learning process is likely to be very small at any initial parameters when they are randomly chosen in a highly expressive circuit, implying that the training may be very slow or may not even start. Thus, while a framework for trainable quantum neural networks could be developed in the future, at present, this PQC approach may still not be effectively utilizing quantum systems.

3.2.3 Quantum extreme learning machine

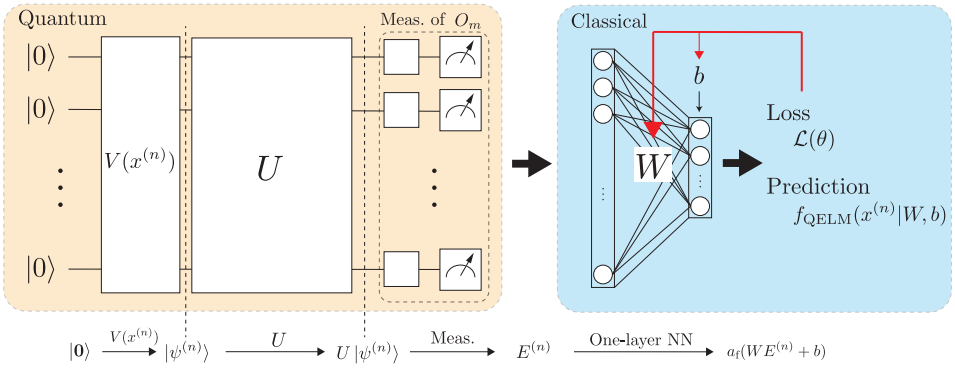

Figure 3.4: Schematic picture of the QELM algorithm.

We now introduce the quantum reservoir model, which performs ML differently from the PQC approach in QCL. The quantum reservoir model can be divided into two types, which are *quantum extreme learning machines* (QELM) and *quantum reservoir computing* (QRC) [2] . In this section, we first outline QELMs, which are typically used for static-data processing [47,50,99].

QELM is the quantum extension of extreme learning machines, which are essentially a two-layer neural network where the first layer is not trained. The network weights in the first layer of extreme learning machines are fixed values that are randomly sampled. In QELM, the quantum unitary evolution is responsible for the random network as the first layer, and the second layer is given by a classical one-layer NN.

[2] I have another classification between QELM and QRC, which is different from a review paper [94]. In the paper, they distinguish these types of the quantum reservoir model by the machine learning tasks under consideration in reservoir models, whereas I identify those types with the presence or absence of (intentionally introduced) memory mechanisms. The quantum reservoir model with/without memory is identified as QRC/QELM in this thesis.

More formally, input information is encoded in the same manner as in QCL: $|\psi^{(n)}\rangle = V(x^{(n)})\,|\mathbf{0}\rangle$. Now we apply a fixed unitary operation U, called the *quantum reservoir*, to the encoding state and measure the expectation values of M observables O_m. Let $E^{(n)}$ denote the expectation values. They are then fed into a classical one-layer NN with a non-linear function a_{f}, and the final output of QELM is given by:

$$f_{\mathrm{QELM}}(x^{(n)}|W,b) = a_{\mathrm{f}}\big(WE^{(n)} + b\big), \tag{3.12}$$

where W and b are the tunable weight matrix and bias vector in the NN. These parameters are tuned to minimize the loss function,

$$\mathcal{L}(\theta) = \frac{1}{D}\sum_{n} l(y^{(n)}, f_{\mathrm{QELM}}(x^{(n)}|W,b)). \tag{3.13}$$

The most significant characteristic of QELM, in comparison to QCL, is that the quantum reservoir does not need to be parametrized and composed of quantum gates. This characteristic allows for harnessing natural Hamiltonian dynamics to generate non-trivial features $E^{(n)}$. This removes the engineering associated with gate implementations, which may allow for a more efficient implementation of nontrivial unitary maps for generating features.

The classical one-layer NN for processing the expectation values $E^{(n)}$ has a notable advantage in the presence of decoherence and statistical error in measurement. In contrast to QCL, the measurement step in QELM is not a precise estimation of the expectation values of observables O_m. Instead, it is sufficient to obtain features $E^{(n)}$ that are meaningful enough for the trainable classical one-layer NN to solve ML problems. QELM thus requires a smaller number of measurement shots compared with QCL. It is also suggested that QELM is robust to decoherence thanks to the classical NN. These characteristics enable harnessing natural quantum dynamics without extensive engineering, although the design of the quantum reservoir may still be required.

3.2.4 Quantum reservoir computing

The *quantum reservoir computing* (QRC) is the other kind of the quantum reservoir model, which is a quantum extension of reservoir computing [45,46, 48,100,101]. It is essentially similar to an ELM in the sense that the first layer is not trained; however, the first layer is equipped with memory. The memory effect is utilized for processing temporal data, such as emulating or predicting non-linear functions. Since such tasks require storing a certain amount of history, the memory effect is implemented by changing the weights in the

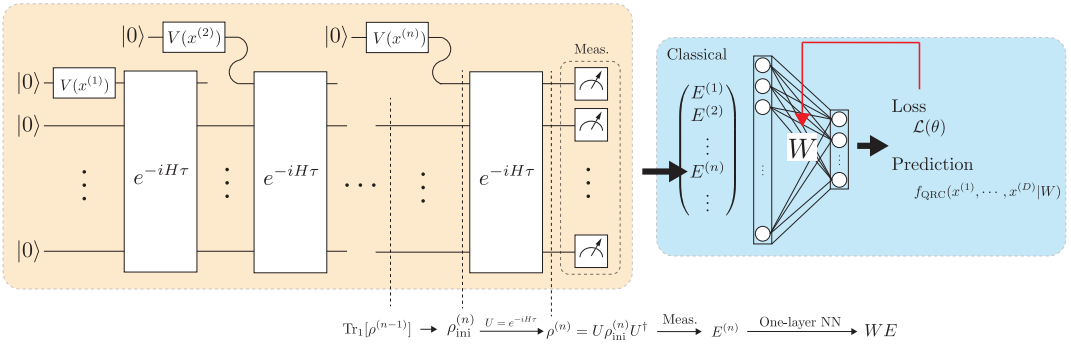

Figure 3.5: Overview of the procedure of QRC.

first layer depending on past inputs, so that the current input is nontrivially mixed with past information.

In QRC, the memory mechanism is introduced in quantum dynamics, i.e., the quantum reservoir, which was designed in the original proposal [45] as follows. Let us consider sequential data $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(D)}, y^{(D)})$ where $\mathcal{X}, \mathcal{Y} \subseteq \mathbb{R}$. Starting with an initial state $\left|\Psi^{(1)}\right\rangle = \left(V(x^{(1)})\left|0\right\rangle\right) \otimes \left|0\right\rangle^{\otimes(N-1)}$, in which the first qubit is used for encoding, a Hamiltonian dynamics determined by H is applied for time τ:

$$\rho^{(1)} = U\rho_{\mathrm{ini}}^{(1)}U^\dagger, \tag{3.14}$$

where $U = e^{-iH\tau}$ and $\rho_{\mathrm{ini}}^{(1)} = \left|\Psi^{(1)}\right\rangle\!\left\langle\Psi^{(1)}\right|$. We then measure the expectation values of σ_j^z $(j = 1, \cdots, N)$, which are denoted by $E^{(1)}$. The initial state for the next step is then set to be

$$\rho_{\mathrm{ini}}^{(2)} = \left(V(x^{(2)})\left|0\right\rangle\!\left\langle 0\right|V(x^{(2)})^\dagger\right) \otimes \mathrm{Tr}_1\left[\rho^{(1)}\right], \tag{3.15}$$

which depends on the previous input. Here, Tr_1 indicates the partial trace that removes the degrees of freedom associated with the first qubit. The other qubits can hence retain past input information to mix it with future inputs. The preparation of the initial state $\rho_{\mathrm{ini}}^{(n+1)}$ for the $(n+1)$-th step usually cannot be performed consecutively after measuring $E^{(n)}$ and requires running the entire evolution from the first step $(n = 1)$ without measuring the intermediate states, $\rho^{(1)}, \cdots, \rho^{(n)}$ [100, 101].

The expectation values obtained are now fed into a linear network to get the final output,

$$f_{\mathrm{QRC}}(x^{(1)}, x^{(2)}, \cdots, x^{(D)}|W) = WE, \tag{3.16}$$

$$E = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ E^{(1)} & E^{(2)} & \cdots & E^{(D)} \end{pmatrix}. \tag{3.17}$$

The loss function is often chosen to be the mean squared loss:

$$\mathcal{L}(W) = \frac{1}{D}\sum_{n}\left(y^{(n)} - (f_{\mathrm{QRC}}(x^{(1)}, \cdots, x^{(D)}|W))_n\right)^2. \tag{3.18}$$

QRC is also expected to be robust to noise and decoherence, and they can even be integrated as a part of QRC [100, 102]. However, QRC requires a nontrivial design of the quantum reservoir dynamics to process time-series inputs while capturing the correlation between different time-steps. We then discuss what kinds of designs are necessary in QELM and QRC by reviewing past investigations on the relationships between quantum system properties and learning behaviors.

3.2.5 Characterization of learning behavior in QELM and QRC

In QELM, the choice of quantum systems for the quantum reservoir U is crucial for learning performance. The properties of U can be investigated using Fourier analysis [59]. As in QCL, the Fourier frequencies are fully determined by the encoding process, and the Fourier coefficients are produced initially by the quantum reservoir U and controlled by the trainable weights of the one-layer NN. The role of the reservoir is then to make the frequencies accessible [59]. For instance, if we choose a unitary that yields zero coefficients in specific frequencies, they cannot be utilized, resulting in a loss of expressiveness, while those frequencies inherently exist in the encoding part. It was reported [59] that Haar random unitaries have good Fourier expressiveness, and more interestingly, the 1D Ising model can have comparable expressiveness in a particular parameter regime. Hamiltonian dynamics can potentially produce rich expressiveness, but this is not always the case in all parameter regimes. This suggests that there may be certain physical properties in the Hamiltonian when it becomes expressive.

Such properties have been investigated from various perspectives [49, 50, 56]. For instance, in [50], it was reported that a near-optimal learning performance was obtained in the parameter regime where the quantum reservoir dynamics exhibit non-trivial network structure in the Hilbert space. In another work, the impact of information scrambling on learning performance was explored [56]. It essentially shows that the more capable the Hamiltonian dynamics is of scrambling information, the higher the learning performance is. It was also observed that an information-theoretic correlation between input and output quantum states is a key quantity for predicting learning performance in their task [56].

In QRC, the design of the quantum reservoir is more complicated. Fading memory is a crucial property that the reservoir should possess for temporal data processing. In temporal data, it is assumed that the data at time step n significantly depends on recent past data, and the dependence becomes weaker for data further in the past. To capture this, the reservoir should retain recent input information while gradually forgetting information from past inputs. This property is called *fading memory*, but it is highly non-trivial how much the property can actually be exploited for ML tasks. Information processing capacity (IPC) and total IPC are typically used to measure the expressiveness of dynamical systems with fading memory [103].

In QRC, the fading memory property is implemented by the coherent dynamics generated by H and the non-coherent operation, that is, the partial trace. The partial trace reduces the amount of information in the state, but in this algorithm, the coherent dynamics part has a more significant effect. As we measure only N observables, σ_j^z, out of 4^N independent observables, the quantum output $E^{(n)}$ may look very random and independent of the input $x^{(n)}$ if H provides complex dynamics. This has been explored in [53] in relation to dynamical phase transitions in the quantum reservoir. The authors selected a Hamiltonian that exhibits various dynamical properties for different parameter configurations, which are classified as dynamical phases by analyzing a statistical property of the Hamiltonian's spectrum. It was numerically shown that the total IPC is low in a regular dynamical phase and becomes high in a complex phase [53]. This suggests that the physical complexity of quantum dynamics is connected to nontrivial memory for temporal data processing.

To implement fading memory, other types of QRC algorithms have been proposed [100, 101]. For instance, in [100], the algorithm utilizes decoherence noise that naturally exists in quantum devices, instead of introducing nontrivial dynamics in addition to the partial trace. When a quantum state evolves in the presence of decoherence, the information of the initial state will be attenuated as the evolution time becomes longer. Such an effect, therefore, can be utilized to implement fading memory. It suggests the possibility of studying quantum reservoir properties using rich tools in the theory of open quantum systems, and this was carried out in [102].

In [55], the learning performance of QRC was explored from a different angle. The authors investigated the effect of coherence and entanglement in quantum states, rather than the Hamiltonian, on the performance. Coherence and entanglement are two distinct types of quantumness, and the motivation is to clarify which kind of quantumness is more crucial to learning. The authors numerically revealed a more significant correlation between

the amount of coherence and learning performance, compared with entanglement.

Limitations in those characterizations of QML

In this chapter, several QML models and their characterizations were outlined. While those characterizations capture a certain correlation between physical properties and learning behaviors, it remains challenging to reveal appropriate engineering of the quantum reservoir. The characterization of QML models can be physics-based or ML-theoretic, and the vast majority of the characterization methods are either of these two approaches. For instance, the Fourier expressiveness evaluates an aspect of the computational capability of the quantum reservoir as a mathematical object, lacking concrete insight into what engineering is necessary to control the expressiveness. In contrast, the impact of information scrambling on the learning performance of QELM exhibits a clear correlation between the physical property and its ML behavior; however, it only provides empirical evidence of sufficient engineering, which however may be unnecessary or redundant. Both physics- and ML-oriented approaches thus should be taken simultaneously to provide a definitive guideline for appropriate engineering.

In this thesis, I therefore explore such a characterization of quantum dynamics in the quantum reservoir model, while investigating the computational power of natural quantum dynamics without extensive engineering. For this purpose, we consider one of the QELM models, known as *quantum extreme reservoir computation* (QERC) [50]. It was proposed for solving classification tasks, such as image classification, while there had been no attempts to apply QELMs to such classical tasks [94]. QERC achieved high classification performance with ten qubits on the modified National Institute of Standards and Technology (MNIST) dataset of handwritten images, which is one of the important benchmark datasets in computer vision. Interestingly, QERC exhibited various learning behaviors for a Hamiltonian in different parameter settings. Thus, it is expected to acquire beneficial insights for realistic problems by analyzing the contribution of quantum dynamics to learning performance in QERC. Let us now outline the architecture of QERC [50] in the following section.

3.2.6 QERC

Consider a classification task with the training dataset $\{(x^{(n)}, y^{(n)})\}_{n=1}^{D}$ where $x^{(n)} \in \mathbb{R}^K$, $y^{(n)}$ is a one-hot vector in Δ_C, and C is the number of classes.

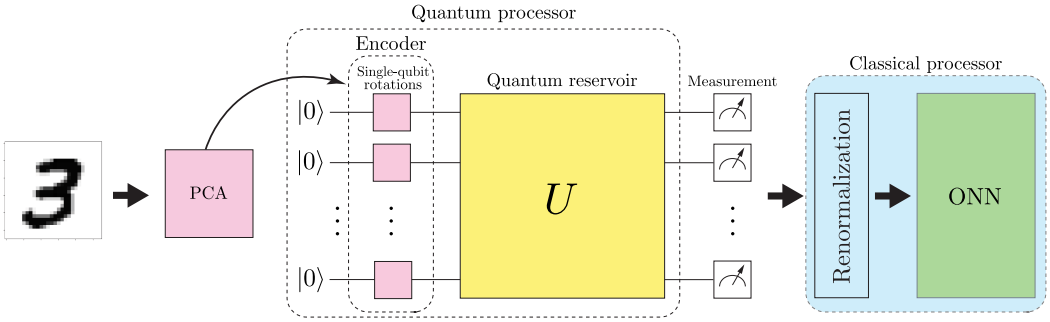

Figure 3.6: Schematic picture of QERC.

QERC essentially follows the QELM framework introduced above, but the characteristics of QERC are summarized below:

Preprocess Input data to be classified is preprocessed (if necessary) by the principal component analysis (PCA) [104]. PCA is a basis transformation to maximize the variance of the data distribution projected onto each basis. Define a data matrix $X = ((x^{(1)})^\mathsf{T}, \cdots, (x^{(D)})^\mathsf{T})^\mathsf{T} \in \mathbb{R}^{D\times K}$, and consider a covariance matrix $X^\mathsf{T}X$ of the data. The data matrix after the basis transformation is given by $XT = R\Sigma$, where $R\Sigma T^\mathsf{T} = X$ is the singular value decomposition and Σ is the diagonal matrix with the singular values in decreasing order: $(\Sigma)_{11} \geq (\Sigma)_{22} \geq \cdots$. The data matrix is then reduced to a matrix of size $D \times 2N$, where N is the number of qubits, by keeping only the first $2N$ singular values. Each reduced input data $\chi^{(n)} \in \mathbb{R}^{2N}$ is now transformed to an angle $\xi^{(n)} \in [0, \pi]$ as follows:

$$(\xi^{(n)})_j = \pi \times \frac{\chi_j^{(n)} - \min_m \chi_j^{(m)}}{\max_m \chi_j^{(m)} - \min_m \chi_j^{(m)}} \in [0, \pi]. \tag{3.19}$$

Encoder The angles $\xi^{(n)}$ associated with the n-th input data are encoded with the single-qubit rotation encoding. The encoding state $\left|\psi^{(n)}\right\rangle$ is given by

$$\left|\psi^{(n)}\right\rangle = \bigotimes_j \left(\cos\left(\xi_j^{(n)}/2\right)|0\rangle + e^{i\xi_{j+N}^{(n)}} \sin\left(\xi_j^{(n)}/2\right)|1\rangle\right). \tag{3.20}$$

In each qubit, half of the Bloch sphere is employed.

Quantum reservoir The encoding state evolves under the quantum reservoir dynamics U, which serves as the quantum feature map in QERC.

Measurement The final state $U\left|\psi^{(n)}\right\rangle$ is now measured by the projective measurement into the computational basis states. This process will be repeated to obtain the probabilities of getting measurement outcomes associated with each basis state, and we have a 2^N-dimensional probability vector $p^{(n)}$: $\sum_j p_j^{(n)} = 1,\ p_j^{(n)} \geq 0$.

Classical processor The probability vector is renormalized to have mean zero and variance one, and is fed to a one-layer NN (ONN). The ONN is trained on the training dataset using a gradient-based method.

For validation and testing datasets, the same basis transformation T calculated with the training dataset is applied for the dimensionality reduction. In the reduced data, one may find larger or smaller values than $\max_m \chi_j^{(m)}$ or $\min_m \chi_j^{(m)}$ obtained in the training dataset. Such value are set to be $\max_m \chi_j^{(m)}$ or $\min_m \chi_j^{(m)}$.

In the original proposal [50], a periodically driven system was employed as the quantum reservoir. A high accuracy rate was achieved on the MNIST dataset with the system, and the learning behavior was shown to be correlated with a network property of the system. The system was analyzed with the percolation rule explained in the background chapter. It was then revealed that the unitary matrix for two periods possesses a non-trivial network at a particular parameter configuration [90]. At the configuration where the non-trivial network appears, the QERC performance was found to be the highest [50]. While this interesting correlation was observed, several aspects of the learning behavior of QERC remain unclear, motivating us to look into how the periodically driven dynamics serve as an effective quantum reservoir in the next chapter.

Chapter 4

Impact of the form of weighted networks on the learning performance of QERC

In QERC, a periodically-driven quantum system, known as the DTC model, was originally adopted as the quantum reservoir [50], where two different unitary maps are alternately applied. Although the DTC model achieved a testing accuracy rate of 97.1% with 11 qubits on the MNIST dataset, several points remain unclear. The first point is the learning behavior in relation to the number of periods. The high accuracy rate was observed at a configuration of parameters in the DTC Hamiltonians, where the two-period effective Hamiltonian exhibits a non-trivial network structure [50, 90]. Although the characteristic of the two-period effective Hamiltonian has a clear correlation with the accuracy rate, more than 50 periods are required for the performance to grow and saturate. This then seems like a necessary engineering in the DTC model. This growth and saturation are rather intuitive since, for example, the Haar sampling from the unitary group usually requires a deep quantum circuit [22,105], but the characterization of the two-period effective Hamiltonian does not capture such a growth of complexity. Another point that requires clarification is how well the DTC model performs classification compared to other unitary maps, such as random unitaries, which require extensive engineering.

In this chapter, we introduce a characterization method for unitary maps to capture the performance behavior against time in the case of the DTC model. We will also benchmark the classification performance of the DTC model against random and artificial unitary maps, and explore the differences in performance behavior using the characterization method.

The rest of the present chapter is organized as follows. In Sec. 4.1,

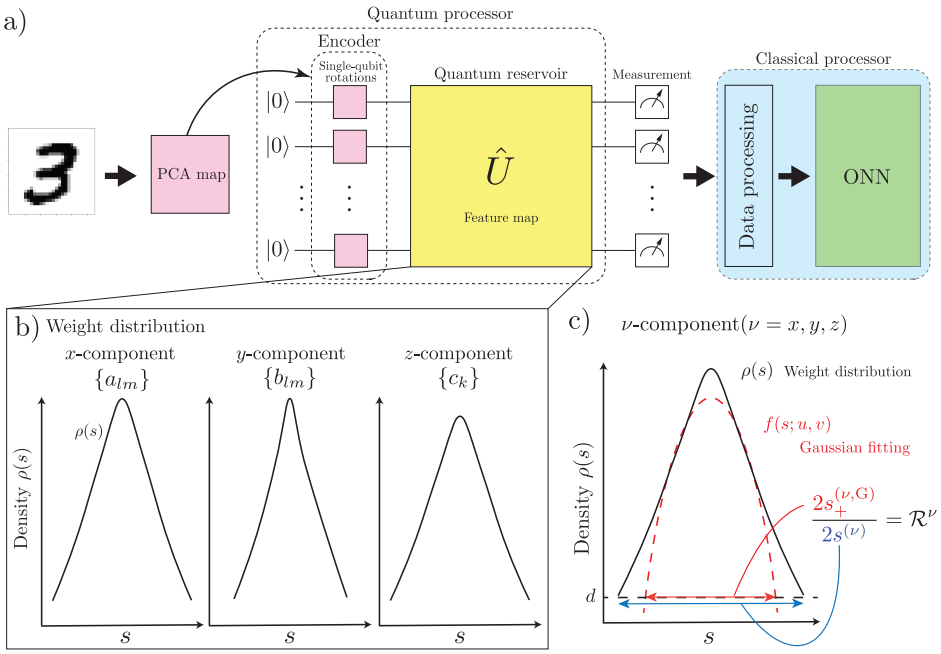

Figure 4.1: a) Schematic architecture of QERC. b) Schematics of the weight distributions for the x-, y-, and z-components. The panel c) summarizes the definition of the ratio $\mathcal{R}^\nu = s_+^{(\nu,\mathrm{G})}/s^{(\nu)}$ for the ν-component, where $s^{(\nu)}$ is given by $s^{(\nu)} = (s_{\max}^{(\nu)} - s_{\min}^{(\nu)})/2$ with $s_{\max\ (\min)}^{(\nu)}$ being the maximum (minimum) value of s in the ν-component. Next one can consider a Gaussian fitting function $f(s;u,v) = u\exp(-vs^2)$, which allows us to define $s_+^{(\nu,\mathrm{G})} := \sqrt{\ln(u/d)/v}$ where d corresponds to the possible minimum nonzero height of the density function $\rho(s) = h(s)/N_\nu ds$, that is, $d = (N_\nu ds)^{-1}$.

the DTC dynamics used for QERC [50] is explained. The characterization method of unitary maps will also be introduced. In Sec. 4.2 and 4.3, using our method, we will investigate the properties of the DTC model and other random unitary maps while benchmarking their classification performance. We will also propose an experimentally feasible procedure to enhance the performance of the DTC model in Sec. 4.3.1. At last, we summarize our results in the last section.

4.1 The quantum reservoir and its characterization in QERC

In this work, we employ QERC shown in Fig. 4.1a), which was explained in the previous chapter. Our focus in this work is on the properties required for the quantum reservoir and their influence on the performance of QERC. For this purpose, we first investigate the DTC model used in [50] as our choice of the quantum reservoir dynamics, whose Hamiltonian is given by

$$\hat{H}(t) = \begin{cases} \hat{H}_1 = \hbar g(1-\epsilon)\sum_l \hat{\sigma}_l^x & t \in [0, T/2) \\ \hat{H}_2 = \hbar \sum_{lm} J_{lm}\hat{\sigma}_l^z\hat{\sigma}_m^z + \hbar\sum_l D_l\hat{\sigma}_l^z & t \in [T/2, T) \end{cases} \tag{4.1}$$

where $\hat{\sigma}_l^a$ $(a = x, y, z)$ are the Pauli operators on the l-th qubit, T is the cycle of driving, while the DTC cycle is $2T$, and g is the rotation strength set to $gT = \pi$. $J_{lm} = J_0/|l-m|^\alpha$ is the coupling strength between the qubits l and m with a power-law decay that scales with a constant α. D_l is a disordered external field for each qubit l, while all the $D_l T$ are set to zero in this work unless explicitly stated.

The imperfection of rotation ϵ is a parameter that controls its dynamical property or complexity. Starting with the perfect DTC when the rotation parameter error $\epsilon = 0$, the dynamics gradually deviates as ϵ increases. The time evolution of the computational basis states hence can be periodic or regular for small ϵ and, for large ϵ, non-trivial.

The time-periodic system is conveniently characterized by the Floquet operator $\hat{\mathcal{F}} = \exp\left[-i\hat{H}_2 T/2\hbar\right]\exp\left[-i\hat{H}_1 T/2\hbar\right]$ where the stroboscopic time-evolution can be obtained by the unitary operator $\hat{U}(nT, 0) = \hat{\mathcal{F}}^n$ for $n \in \mathbb{N}$. We are interested in how the unitary operator $\hat{U}(nT, 0)$ behaves as a feature map as n increases.

4.1.1 Characterization of the unitary matrices

The next step is the characterization of the unitary operator $\hat{U}(nT, 0)$. This unitary operator serves as a map between the input and output states, which can be considered a weighted network [39, 106]. We then consider the matrix representation of unitary operators in the computational basis. Still, if unitary matrices are naively converted into a weighted network, it will result in a directed network, which may not be straightforward to investigate. More simply, we thus apply the generator decomposition for representing unitary matrices as a weighted network. The generators of the unitary group $\mathrm{U}(d)$ are skew Hermitian matrices forming the Lie algebra. A unitary matrix $U \in$

U(d) can then be written in the form $U = e^{-iG}$ where G is a Hermitian matrix. This Hermitian matrix is determined by real coefficients a_{lm}, b_{lm}, and c_k in the decomposition of G with respect to the generators λ as

$$G = \sum_{l<m} (a_{lm}\lambda^x_{lm} + b_{lm}\lambda^y_{lm}) + \sum_k c_k \lambda^z_k, \tag{4.2}$$

where those λ generators are the generalized Gell-Mann matrices [62, 63]

$$(\lambda^\nu_{lm})_{ij} = \delta_{li}\delta_{mj}\sigma^\nu_{12} + \delta_{lj}\delta_{mi}\sigma^\nu_{21} \quad (\nu = x, y), \tag{4.3}$$

$$\lambda^z_k = \sqrt{\frac{2}{k(k+1)}}\ \mathrm{diag}(\underbrace{1, \cdots, 1}_{k \text{ times}}, -k, 0, \cdots, 0) \tag{4.4}$$

$$\lambda^z_N = I. \tag{4.5}$$

Here δ_{ij} represents the Kronecker delta, σ^ν_{ij} ($\nu = x, y$) is the (i, j)-component of the Pauli matrices and I the identity matrix. G as a weight matrix has three components x, y, and z with $\{a_{lm}\}, \{b_{lm}\}$ and $\{c_k\}$ being the x, y and z contributions of the weight matrix, respectively.

The weight matrix G obtained from the unitary matrix U is not necessarily unique. To uniquely determine G, we employ the *principal logarithm* for matrices [107] in our numerical analysis. If A is a complex-valued matrix of dimension d with no eigenvalues on the negative real line $\mathbb{R}_{<0}$, then there is a unique logarithm X of the matrix A such that all of its eigenvalues lie in the strip $\{z : -\pi < \mathrm{Im}(z) < \pi\}$. X is called the *principal logarithm* of A and denoted by $X = \mathrm{Log}(A)$. For the DTC model, we compute the Hermitian matrix for period n, $G(n) = i\mathrm{Log}(U(nT, 0))$, noting that $G(n)$ is not simply equal to $n \times G(1)$.

To convert the weight matrix to its weight distribution, we first count how many coefficients are in a certain value window $(s, s + ds]$ for $s, ds \in \mathbb{R}$. This gives us a histogram $h(s)$ that shows how likely the coefficients are to take a certain value $(s, s + ds]$ in the weight matrix. In numerical calculations, we take 100 segments for each coefficient set to determine the value of ds: let the support of the histogram denoted by a sequence $S = \{s_0, s_1, \cdots, s_{M-1}\}$, where $s_i - s_{i-1} = ds > 0$ for all $i = 0, 1, \cdots, M-1$ and M is the number of segments in the range $(s_0, s_{M-1}]$. We then have $ds = (s_{M-1} - s_0)/M$. After obtaining the histogram, we have a density function $\rho(s) = h(s)/N_\nu ds$ where N_ν is the number of elements in the component $\nu = x, y$ or z, that is, $N_\nu = \sum_{i=0}^{M-2} h(s_i)$. We refer to this as the weight distribution of the ν components ($\nu = x, y, z$) (see Fig. 4.1b)).

4.1.2 Characterization of the weight distributions

We will show weight distributions for the DTC model in different configurations and other models. To quantitatively characterize those weight distributions, we calculate two quantities for each weight distribution. The first is that the empirical standard deviation σ_ν is given by the standard deviation of the weight distribution of component ν.

Our second quantity is a ratio that represents how far the weight distribution reaches from its center compared to a Gaussian function approximating the weight distribution, which is analogous to the MP rank defined in Ref. [108]. The ratio $\mathcal{R}^\nu$ for the weight distribution of the ν component is given by

$$\mathcal{R}^\nu = \frac{s_+^{(\nu,\mathrm{G})}}{s^{(\nu)}}. \tag{4.6}$$

The denominator $s^{(\nu)}$ is defined as a quantity representing how far the weight distribution reaches from its center on the horizontal axis. Since the weight distribution $\rho(s)$ is not necessarily symmetric with respect to $s = 0$, $s^{(\nu)}$ is given by $s^{(\nu)} = (s_{\max}^{(\nu)} - s_{\min}^{(\nu)})/2$ with $s_{\max\ (\min)}^{(\nu)}$ being the maximum (minimum) value of s in the ν component. $s_+^{(\nu,\mathrm{G})}$ represents how far the Gaussian distribution function obtained by a Gaussian fitting to the weight distribution reaches from $s = 0$. The Gaussian function reaches $s = \infty$ in general, and we thus introduce a cutoff for the Gaussian function. Let $f(s; u, v)$ denote the Gaussian function $f(s; u, v) = u \exp(-vs^2)$ obtained by the Gaussian fitting with fitting parameters u, v. To introduce the cutoff, we consider the positive cross point $s_+^{(\nu,\mathrm{G})}$ between the function $f(s; u, v)$ and a horizontal line at height d. Namely, $s_+^{(\nu,\mathrm{G})} = \sqrt{\ln(u/d)/v}$. In our numerical calculations, we set $d = (N_\nu ds)^{-1}$, which corresponds to the minimum nonzero height of the density function $\rho(s) = h(s)/N_\nu ds$. In Fig. 4.1c), the definition of $\mathcal{R}^\nu$ is summarized.

4.1.3 Simulation setup for QERC

We will evaluate the properties of the feature map generated by the DTC dynamics using the method outlined above. Setting a computational task is not essential to do the analysis, but it is extremely useful when we later compare these properties to the performance of QERC. For this purpose, we perform QERC to solve the ten-class classification problem on the well-known modified National Institute of Standards and Technology (MNIST) dataset [109] of handwritten digit images, where each image has 784 ($= 28 \times 28$) pixels. 60,000 images out of 70,000 are used for training, while the

remaining 10,000 images will be used for testing. To optimize the parameters of the ONN, we employ the stochastic gradient descent method used in [50]. Throughout this work, our parameters are set as $N = 10$, $J_0T = 0.12$ with $\alpha = 1.51$. These are compatible with the current ion trap experiments [2]. We also set $\epsilon = 0.03$, where the highest accuracy rate has been reported in [50].

4.2 Quantum reservoir weight distribution

It is important to emphasize that an initial state evolving under the DTC dynamics deviates over time unless it is in a periodic regime. This allows for the growth of complexity in the system. To observe the complexity growth, we evaluate the weight distribution of the DTC dynamics $\hat{\mathcal{F}}^n$ for various time periods: $n = 2, 10, 50$, and 100. Calculating the weight matrix $G(n)$ from $\hat{\mathcal{F}}^n$ for each n, we can determine the real coefficients a_{lm}, b_{lm} and c_k, yielding the weight distribution. The Hermitian weight matrix $G(n)$ is equivalent to the n-period effective Hamiltonian up to the constant factor $\hbar/nT$. Thus, the diagonal (corresponding to $\{c_k\}$) and off-diagonal (corresponding to $\{a_{lm}\}, \{b_{lm}\}$) entries of $G(n)$ are associated with the energies of the basis states and the transition energies between the basis states, respectively.

In Fig 4.2a) we plot the weight distribution for the x (a_{lm}), y (b_{lm}) and z (c_k) components of $G(n)$ respectively. Each color (or marker) represents a different period of the time evolution. For $n = 2$ (blue curve), we observe very sharp peaks at $s = 0$ for the x- and y- components. As these components correspond to the off-diagonal entries of the Hermitian matrix $G(n)$, the sharp peak around $s = 0$ indicates very few transitions between the basis states for this time period. In contrast, for large periods $n = 50$ (brown curve) and 100 (orange curve), the weight distribution for the x- and y-components is broader. As the z-component has a similar broadness to that for $n = 2$, the broader distribution for the x- and y-components implies more active transitions between the basis states. We also note that the x- and y weight distributions for $n = 50$ and 100 converge to a similar shape that is approximately quadratic in the log-scaled plots (Gaussian in linear plot). In the middle time region, at $n = 10$ (green curve), the x and y components have already converged to the typical distribution, however, the z component has broadened the most. This suggests that there is a tradeoff in this time regime; the stationary elements of the z component significantly suppress the effect of the x and y components. This trade-off captures the dynamics of the DTC melting slowly in time in this ($\epsilon = 0.03$) parameter regime.

The behavior for those components can be quantitatively observed by

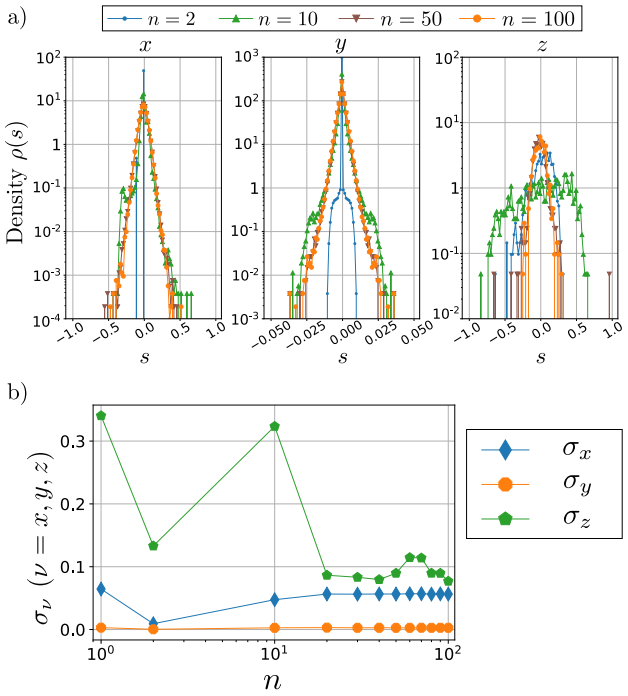

Figure 4.2: a) Convergence of the weight distribution in the DTC model. From left to right panels, the weight distributions for the x-, y-, and z-components are depicted, where the colors correspond to different periods. The blue (dot), green (upward triangle), brown (downward triangle), and orange (filled circle) curves are for $n = 2, 10, 50$, and 100, respectively. b) shows the n-dependency of the empirical standard deviations of the weight distributions, σ_x (blue with diamonds), σ_y (orange with hexagons), and σ_z (green with pentagons).

the empirical standard deviation for the weight distributions, defined in Sec. 4.1.2. In Fig. 4.2b), the empirical standard deviations are depicted against the number of periods, n. While the z-component is broadened at $n = 10$, the x- and y-components gradually get higher standard deviations, and then they converge.

4.2.1 Comparing the weight distribution

To capture the characteristics of the weight distribution for the DTC model, we first introduce the Haar random sampling of unitary operators [110–112]. The Haar measure sampling can be considered to exhibit a typical complex-

ity that a quantum computer may provide, and its gate implementation is usually given through unitary t-designs [22, 105]. Hence the similarity and disparity in the weight distributions for these cases would give us valuable insights into understanding the DTC dynamics and its role in QERC. In our analysis, to obtain the typical distribution given by the Haar random sampling, the $d \times d$ unitary matrix U_{H} is created using the QR decomposition [113] where $d = 2^N = 2^{10} = 1024$ in our case. We compare this unitary map, which we refer to as the Haar-random model, to the converged weight distribution of the DTC model.

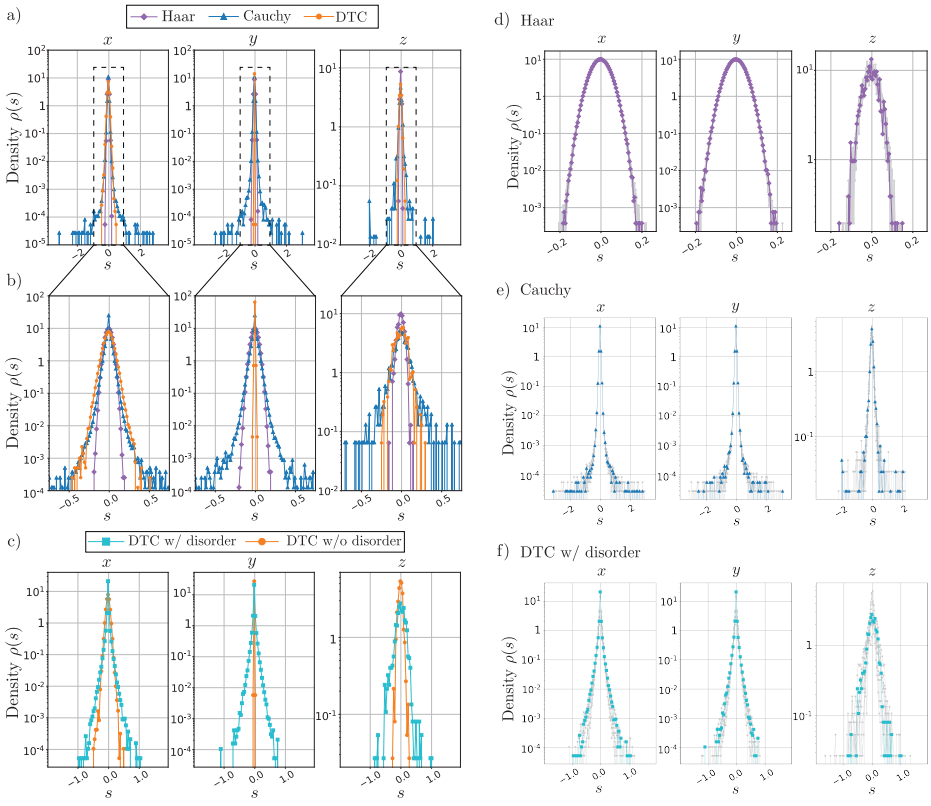

Figure 4.3: Comparison of the Haar-random, Cauchy-random, and DTC models for $n = 100$ in a). In each case from the left to right panels, the distributions of the x-, y-, and z-components are depicted respectively. In b), a) is zoomed in for the range $[-0.75, 0.75]$. In c), the DTC model with and without disorder for $n = 100$ is shown. In d,e,f), Many realizations of the random models are shown. The total number of realizations in each case is ten. The colored curves correspond to each curve in a,b,c) respectively. The other realizations are plotted as gray curves.

	σ_x	σ_y	σ_z	$\overline{\mathcal{R}}$
Haar	0.0400	0.0402	0.0406	0.8592
Cauchy	0.0402	0.0394	0.2733	0.0978
DTC	0.0564	0.0028	0.0770	0.5189
DDTC	0.0410	0.0393	0.1914	0.2245

Table 4.1: The empirical standard deviations and the averated ratios $\overline{\mathcal{R}}$ for different models, where $\overline{\mathcal{R}} = \sum_{\nu=x,y,z} \mathcal{R}^\nu/3$ for each model.

As shown in Fig. 4.3a) and b), the typical weight distributions are approximately Gaussian for all components x, y, and z. Here we use only a single realization of the Haar-random model since one sample and not the average of many realizations will be used for QERC. We will further show later that the characteristics of the weight distribution we observe are not realization-dependent.

We can now compare the DTC model to the Haar-random model, and we first characterize the weight distributions of the DTC model in terms of two properties: broadness and tail. Although the DTC's y-component has a narrower distribution compared to the Haar-random model, the broadness of the distribution for the x- and z- components is comparable as shown by the empirical standard deviations in Table. 4.1.

However, only the DTC model has a tail in the weight distribution of the x component: a few elements at the edge of the weight distribution, which indicate the presence of a few large values in the weight matrix $G(n)$. It essentially implies a few strong transitions between certain basis states. To quantitatively observe tails in the weight distribution, we calculate the ratio $\mathcal{R}$ defined in Sec. 4.1.2. Table. 4.1 shows the averages of the ratios for the DTC and Haar models where $\overline{\mathcal{R}}$ denotes the average, that is, $\sum_{\nu=x,y,z} \mathcal{R}^\nu/3$. One can see that the averaged ratio of the DTC model is smaller than that of the Haar model, which is close to unity. It implies that the weight distribution of the DTC model deviates from that of the Haar model in terms of the tail.

To further explore the tail we found in the DTC model's distribution, we employ the Cauchy distribution. One reason for this is that the Cauchy distribution has a power-law tail, which is longer than the exponentially decaying tail in the Gaussian distribution. Another reason is as follows. In classical reservoir computation, the Cauchy distribution was used to obtain the edge of chaos where the reservoir computation should be optimal [114]. Hence, it is interesting to see the properties of the feature map generated by

the Cauchy distribution. The Cauchy distribution is given by

$$\text{Cauchy}(x;\gamma) = \left(\frac{1}{\pi\gamma}\right)\frac{1}{1+(x/\gamma)^2}, \tag{4.7}$$

where γ is the scale parameter. Since the Cauchy distribution has a power-law tail, one may expect that the weight distribution exhibits a long tail. The unitary matrix U_{C} for this Cauchy-random model is defined as follows. First, we generate an $d \times d$ Hermitian matrix A whose real and imaginary parts in each independent entry are drawn from the Cauchy distribution (4.7). Then we define the unitary matrix U_{C} as $U_{\mathrm{C}} = e^{-iA}$. We set $\gamma = 0.04$ in Eq. 4.7 for consistency with the corresponding parameter of the Haar-random model ($\sigma \approx 0.04$).

Applying the decomposition (4.2) to $G_{\mathrm{C}} = i\log(U_{\mathrm{C}})$, we obtain each weight distribution for the three components x, y and z., which are depicted in Fig. 4.3a) and b). The weight distributions for the Cauchy-random model definitely have a tail, much longer than that in the DTC model, for all the components x, y, and z. Table. 4.1 shows the averaged ratio of the Cauchy-random model, which is even smaller than the other two models. This reflects the nature of the Cauchy distribution (4.7), and we will come back to this point later.

Here, we note that the characteristic form of the weight distribution for those random unitaries is not realization-dependent. In Fig. 4.3d,e,f), the weight distribution of each model is depicted with ten realizations. The colored curves here correspond to those in Fig. 4.3 a,b,c). We clearly observe that the width and tail properties of the weight distribution are not strongly dependent on particular realizations.

4.3 Relation between the QERC performance and the weight distribution

As we have characterized the three models through the weight distribution, let us now turn our attention to the performance of QERC employing these three different models as its feature space. In the previous work [50] it was shown that the accuracy of QERC for solving an image classification task increases with the number of time periods of the DTC model saturating near $n = 50$. The behavior of the accuracy rate coincides with the time evolution of the weight distributions seen in Fig. 4.2, as the unitary map of the DTC model acquires the typical complexity around $n = 50$. To illustrate this further, Fig. 4.4 summarizes the comparison between the accuracy rate

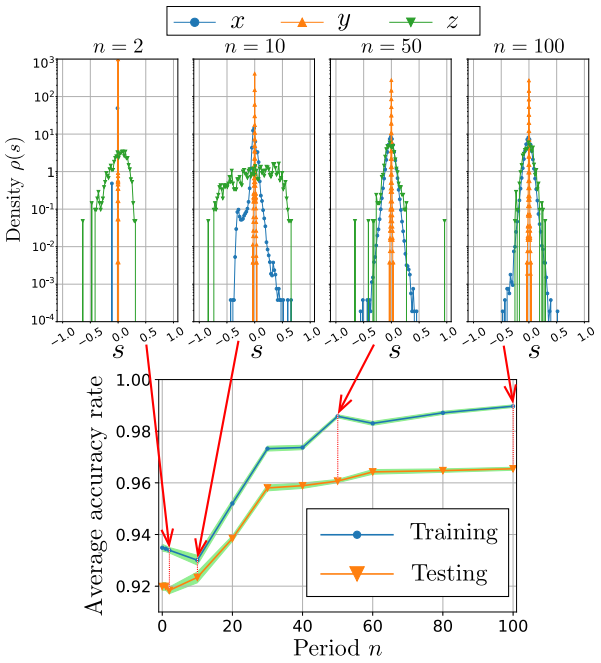

Figure 4.4: Average accuracy rates for training and testing with the associated standard deviation against the period in the DTC model. The blue (dot) and orange (downward triangle) curves correspond to training and testing, respectively. At each datapoint, the average and the standard deviation are taken for 250 to 300 epochs in the ONN optimization.

and the weight distribution. Here, we plot the accuracy rates for training (blue dot) and testing (orange downward triangle) against the time period n in the DTC model, with the weight distributions for $n = 2, 10, 50, 100$. The accuracy rate grows with the period, as the weight distribution becomes broader and the transition between the basis states gets more complex. The trade-off between the x, y components and z component discussed in Fig. 4.2 is reflected in the accuracy rates. This suggests that even if the system dynamics is complex enough, within a short coherent regime, the system does not evolve enough to achieve the computational power the system would promise.

Unlike unitary maps from the Haar-random sampling, the DTC model does not achieve such high randomness, allowing the system to exhibit certain tendencies in its dynamics. Next, we further investigate the effect of this difference between these models on the performance of the QERC.

a) MNIST	testing acc. (std.)	training acc. (std.)	$\Delta_{\text{acc.}}$
PCA	0.8635($\pm$0.0009)	0.8688($\pm$0.0005)	0.0053
Haar	0.9657($\pm$0.0005)	0.9949($\pm$0.0003)	0.0292
Cauchy	0.9673($\pm$0.0005)	0.9945($\pm$0.0003)	0.0272
DTC	0.9655($\pm$0.0006)	0.9897($\pm$0.0005)	0.0242
DDTC	0.9671($\pm$0.0005)	0.9911($\pm$0.0005)	0.0240
b) FashionMNIST			
Haar	0.9427($\pm$0.0017)	0.9744($\pm$0.0017)	0.0317
Cauchy	0.9440($\pm$0.0029)	0.9720($\pm$0.0023)	0.0280
DTC	0.9388($\pm$0.0039)	0.9662($\pm$0.0049)	0.0274
DDTC	0.9407($\pm$0.0050)	0.9589($\pm$0.0044)	0.0182

Table 4.2: Average accuracy rates with the associated standard deviation of the models on a) the MNIST and b) FashionMNIST datasets. For the FashionMNIST case, three classes are selected: T-shirt, Pullover, and Dress. The average and the standard deviation are taken from 250 to 300 epochs, and $\Delta_{\text{acc.}}$ denotes the gap between training and testing. The label "PCA" denotes the case where the PCA components are directly fed into the ONN (without any quantum feature maps). The results for the DTC cases with/without disorder are for the period $n = 100$. In the random models, the accuracy rates are for a single realization.

4.3.1 Tails in the Distribution and the Performance

The unitary operators we characterized through the weight distribution directly serve as the feature map for QERC. We will now explore how these feature maps equipped with the different weight distributions affect the performance of the QERC with the MNIST dataset.

Table. 4.2a) presents the accuracy rates for each model. Before comparing the quantum feature maps, we first present the performance of the case where the PCA components are directly fed into the ONN (without a quantum feature map). One can immediately observe that the case, denoted by "PCA" in Table. 4.2a), has the lower accuracy rates in both training and testing than any other cases with quantum feature maps. It states that those quantum feature maps significantly help the QERC achieving a high performance.

We now compare the accuracy rates of the quantum feature maps. In testing, the DTC ($n = 100$) model is comparable to the Haar-random model, whereas not so in training. Moreover, it is interesting that the Haar-random model does not yield the best testing accuracy rate in this setting, and the Cauchy-random model does among these feature maps. These observations

may suggest that the tail in the weight distribution of the DTC and Cauchy models contributes to the higher testing accuracy rate.

We show the correlation between the testing accuracy rate and the quantities we used to characterize the weight distribution: the averaged empirical standard deviation ($\overline{\sigma}$) and ratio ($\overline{\mathcal{R}}$) in Fig. 4.6a). The quantities correspond to the vertical and horizontal axes, respectively, and the color of the markers indicates the testing accuracy rate. It can be observed that the testing accuracy rate improves as the standard deviation and the ratio $\overline{\mathcal{R}}$ increase and decrease, respectively.

To investigate this further, we introduce the t-random model, whose unitary is defined using the Student's t-distribution in the same way as generating the Cauchy-random model. The Student's t-distribution is defined as

$$\mathrm{t}(x;0,\gamma_\mathrm{t},\nu)=\frac{\Gamma(\frac{\nu+1}{2})}{\gamma_\mathrm{t}\sqrt{\pi\nu}\Gamma(\frac{\nu}{2})}\left(1+\frac{(x/\gamma_\mathrm{t})^2}{\nu}\right)^{-(\nu+1)/2} \tag{4.8}$$

where $\Gamma(\cdot)$ denotes the Gamma function, and γ_t is a scale parameter. ν is a parameter determining how heavy the tail of the distribution is. This parameter connects the standard Cauchy ($\nu = 1$) and normal ($\nu = \infty$) distributions when $\gamma_\mathrm{t} = 1$. In our context, this parameter allows us to generate the weight distribution located in between the Haar- and Cauchy-random models in Fig. 4.6a) with $\gamma_\mathrm{t} = \sigma = \gamma = 0.04$. Fig. 4.5 shows the accuracy rates for training and testing against ν. The dashed lines and shaded areas correspond to the accuracy rates averaged over ten realizations for each value of ν and the associated standard deviations, respectively. The horizontal solid lines are the averaged accuracy rates of the Haar-random model. As ν increases, that is, as the effect of the tail decreases, the accuracy rates for both training and testing converge to the Haar-random model.

In Fig. 4.6a), the star-shaped markers correspond to the averaged data points over ten realizations of the unitary for each value of ν ($\nu = 1, 2, 3, 5, 10,$ and 100). The error bars correspond to the standard deviations of each data point. One can find that the ratio and the testing accuracy rate tend to be smaller and darker, respectively, as ν gets smaller. Moreover, data points close to each other in the plot have similar testing accuracy rates. This simulation with the Student's t-distribution illustrates the correlation between the testing accuracy rate and the tail of the weight distribution.

Considering the implementability of QERC with an effective quantum feature map, one may wonder what physical system can achieve a high accuracy rate comparable to the artificial models: the Cauchy- and t-random (for $\nu = 1$) models. They are far distinct from the DTC model as the unitary map, but there still seems to be room to improve the DTC model by

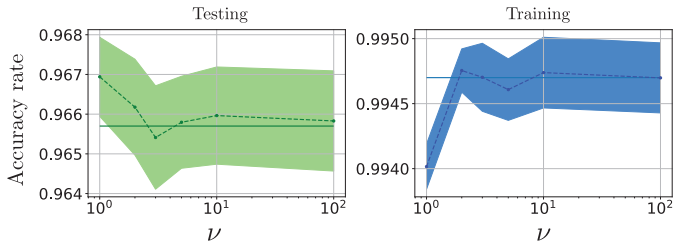

Figure 4.5: Accuracy rates for testing and training of the t-random model against ν with the MNIST dataset. The dashed lines and shaded areas correspond to the accuracy rates averaged over ten realizations for each value of ν. The horizontal solid lines are the averaged accuracy rates of the Haar-random model.

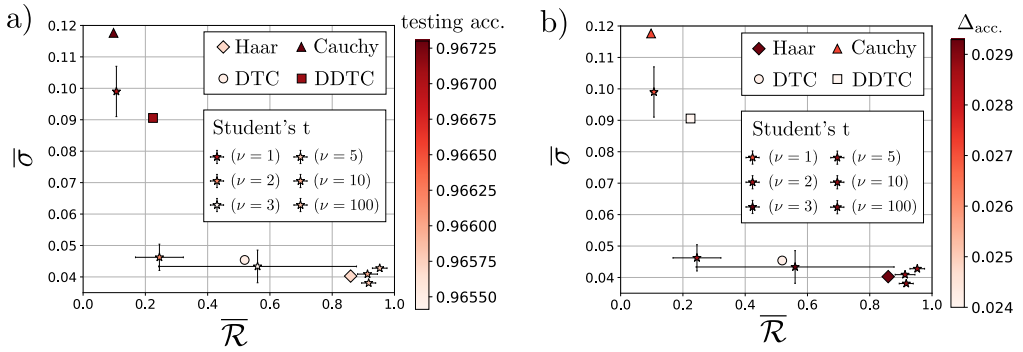

Figure 4.6: Scatter plot of the models we consider in this paper against $\overline{\sigma}$ and $\overline{\mathcal{R}}$ with colored markers indicating a) the testing accuracy rate and b) the gap $\Delta_{\text{acc.}}$ of accuracy rates between training and testing.

considering the tail of the weight distribution. Here, we consider disorder in the DTC model, as presented in Eq. 4.1.

In Eq. 4.1, we have set $D_l T = 0$ to enable us to consider the DTC model without the disorder. That constraint can now be relaxed. The disorder in Floquet systems has been considered crucial for suppressing thermalization and stabilizing the DTCs [2, 10]. Thus, the disorder is expected to introduce strong transitions between certain basis states, yielding a longer tail. Furthermore, it is more realistic to have a certain degree of disorder in such quantum systems in actual implementation. We choose the disorder terms $D_l T$ in Eq.4.1 independently drawn from a uniform distribution on $[0, 2\pi)$. As illustrated in Fig.4.3c), the disorder changes the form of the weight distribution, and it is characterized by the empirical standard deviation and the

MNIST	testing (std.)	training (std.)	$\Delta_{\text{acc.}}$
Haar	0.9657 ($\pm$0.0011)	0.9947 ($\pm$0.0003)	0.0290
Cauchy	0.9678 ($\pm$0.0013)	0.9943 ($\pm$0.0002)	0.0265
DTC w/ disorder	0.9668 ($\pm$0.0005)	0.9910 ($\pm$0.0002)	0.0242

Table 4.3: Realization- and epoch-averaged accuracy rates of the models. The epoch average spans 51 epochs, ranging from 250 to 300 epochs. The realization average is the epoch-averaged accuracy rate averaged over 10 realizations for each model. $\Delta_{\text{acc.}}$ denotes the difference between the means for training and testing.

ratio (see Table. 4.1), and the DDTC model is now located around the upper left region of Fig. 4.6a). One can find that the model achieves a similar testing accuracy rate to that of the Cauchy- and t-random (with $\nu = 1$) models (see also Table. 4.2a)).

So far, we have only discussed our testing accuracy, and we now turn our attention to the training accuracy and the difference between the testing and training accuracy rates. The difference $\Delta_{\text{acc.}}$ is an important quantity in terms of the overfitting phenomenon and the generalization performance of this machine learning model. Neural networks often show the overfitting phenomenon [91,115] where the neural networks are too well optimized to the training data and lose their flexibility in dealing with the testing data. The generalization performance is also an important factor in designing QNNs.

In Table.4.2, we also provide the training accuracy and difference $\Delta_{\text{acc.}}$ for the various feature models we have considered. We also plot the correlation between the generalization performance and the properties of the weight distribution in Fig. 4.6b). One can observe that the Cauchy-random model has the smallest gap $\Delta_{\text{acc.}}$ among the artificial models, and the value of $\Delta_{\text{acc.}}$ tends to increase with a larger value of the ratio. It is strongly suggested that the tail in the weight distribution helps QERC achieve high generalization performance while suppressing the training accuracy rate. One can also find that the physical models (DTC and DDTC) have even smaller gaps. Therefore, from our analysis, the DDTC model achieves the best generalization performance and a nearly-optimal testing accuracy rate among the quantum feature maps we have considered. It is an encouraging fact that a simple Hamiltonian system could perform at least as good as Haar random unitary maps, which makes the implementation of such QNNs much simpler and more feasible.

We also note that the tendency of the QERC performance with those random models does not strongly depend on particular realizations of the mod-

els. The realization-averaged accuracy rates for the Haar-random, Cauchy-random, and disordered DTC models are also shown in Table. 4.3. As the mean difference $\Delta_{\text{acc.}}$ shows, the models that have a tail in their weight distributions potentially avoid overlearning. The disordered DTC model achieves the highest generalization performance.

4.3.2 Simulations in other settings

In this section, we consider different numbers of qubits, another optimizer for ONN, and a different image dataset to verify whether our findings above can be observed in various settings. Let us begin with Fig. 4.7, where the weight distributions of the Haar, DTC, and DDTC models are shown for $N = 9, 10, 11$ with the corresponding accuracy rates on the MNIST classification task. One can observe the same findings for different numbers of qubits. The Haar-random model exhibits Gaussian-like distributions, and the DTC distribution deviates from them as it has a longer tail. It can also be observed that the disorder in the DTC model extends the tail length for $N = 9$ and 11. In Fig. 4.7(d), the Cauchy model has a better testing accuracy rate than the Haar model for $N = 9$ and 11, as well as $N = 10$, suggesting the correlation between the tail length and the testing accuracy rate.

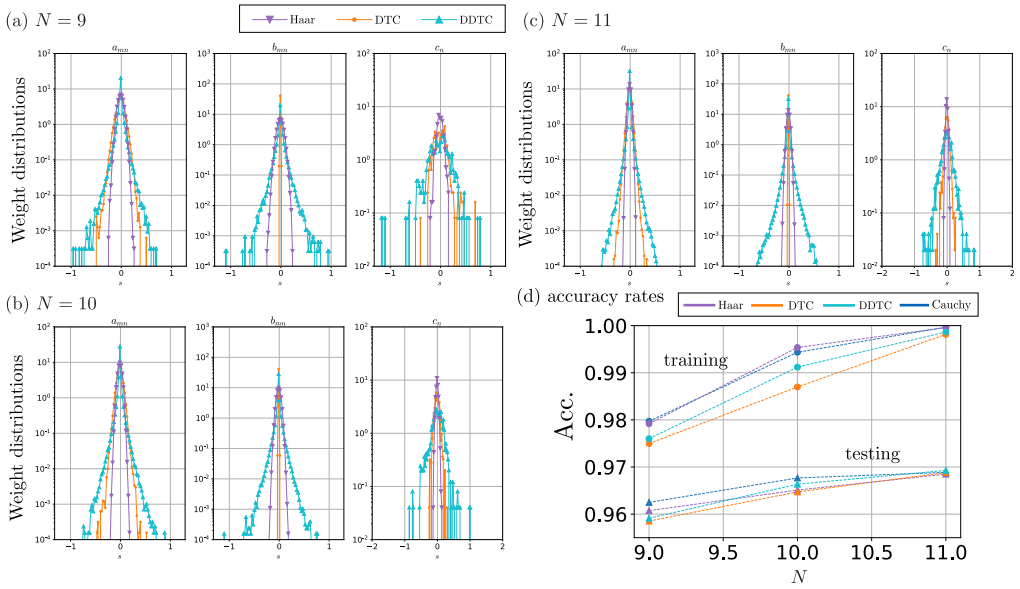

Figure 4.7: The weight distributions of the Haar-random, DTC, and DDTC models for (a) $N = 9$, (b) $N = 10$, and (c) $N = 11$. (d) shows the corresponding accuracy rates for training and testing.

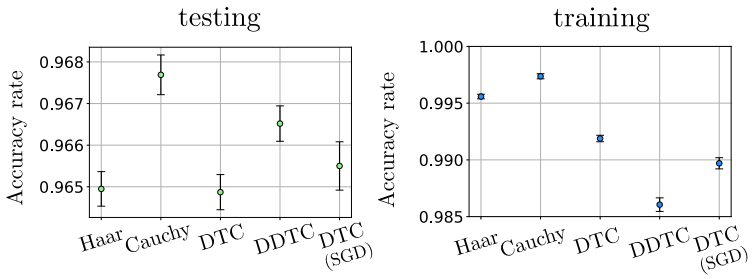

Figure 4.8: Accuracy rates for testing and training of various models with AdaGrad. The same specific realizations are used for the random models as in the numerical simulations in the text. At each data point, the average is taken from 250 to 300 epochs. The error bar indicates the associated standard deviation.

For the optimizer, we used the stochastic gradient descent method in the previous numerical simulations. The method has been broadly employed in many situations, but there are other optimizers, for instance, a more technical optimizer, AdaGrad [116]. It is an adaptive algorithm that updates the parameters in the ONN based on the loss values from earlier iteration steps. Hence, it is considered that the training loss converges more quickly than with SGD. In Fig. 4.8, the accuracy rates for training and testing are shown for various models. The accuracy rates for the DTC model with SGD are also plotted for comparison. In fact, the training accuracy rate tends to be higher than that of the SGD case, and consequently, the generalization is poorer. In this case, it can still be observed that the tail affects the testing accuracy rate and the generalization performance. Furthermore, the DDTC model achieves a nearly optimal testing accuracy rate and exhibits the best generalization with AdaGrad as well as SGD.

Next, we direct our attention to the dataset. So far, we have used the MNIST dataset of handwritten digit images to benchmark quantum feature maps, and we have concluded that the tail in the weight distribution contributes to the testing accuracy rate and generalization performance. In this section, we examine, using another dataset, the difference in QERC performance between the quantum feature maps we have considered. Here, we need to carefully select a dataset. If it is a simple dataset, such as the 2D isotropic Gaussian samples demonstrated in Ref. [50], all the feature maps may have high accuracy rates. In contrast, if we choose a hard dataset, such as the Fashion MNIST [117], all the models may achieve poor performance, and there should be no room to discuss generalization.

For our calculations, we chose the Fashion MNIST dataset with a limited

number of classes. We picked three classes: T-shirt, Pullover, and Dress, and show the accuracy rate in Table. 4.2b). It can be observed that the same trend we obtained with the MNIST dataset: the models with the tail tend to have higher testing accuracy rates and better generalization.

4.4 Summary

In this chapter, we have analyzed the weight distribution of the quantum reservoir in QERC while benchmarking the DTC model against random unitaries. The benchmark results showed that the DTC model achieved a high testing accuracy rate comparable to that of the Haar-random model, and the disordered DTC model attained nearly optimal accuracy and best generalization performance among other random unitaries that we considered. This observation strongly suggests that QERC can leverage physical dynamics such as the (disordered) DTC dynamics for solving practical medium-scale datasets. While such a dynamics is much simpler to implement than the random unitaries, it possesses the nontrivial properties in its weight distribution, implying that the Haar randomness or t-designs are not necessarily the type of complexity required for the quantum reservoir in QERC. This implies that the extensive engineering required for such randomness is excessive and redundant for these ML tasks considered in this chapter. One may then expect that even simpler Hamiltonian dynamics can be powerful enough for high performance in QERC. We thus investigate other simpler Hamiltonian dynamics in the next chapter.

Chapter 5

Two-body Hamiltonian dynamics as a powerful quantum reservoir

In this chapter, we investigate QERC performance using other Hamiltonians for the quantum reservoir and examine physical properties that are beneficial for QERC. The disordered discrete time crystal (DDTC) model has been shown to achieve a high testing accuracy rate, which is significantly better than the Haar-random model. This suggests that QERC does not require random complicated quantum circuits for the quantum reservoir dynamics. It is then worth investigating how much the quantum reservoir can be simplified. A possible simplification is to consider analog Hamiltonian dynamics with two-body interactions, in contrast to the DDTC model whose effective Hamiltonian includes interactions over more than two bodies. We thus evaluate the power of Ising-type Hamiltonian dynamics as the quantum reservoir and explore their learning behavior. We will also consider several circuit-oriented models to explore what properties common in quantum information contribute to QERC performance, aiming to narrow down essential properties in the quantum reservoir. We will then examine a physics-based characterization of the quantum reservoir, which has a direct connection to classification tasks in our setting.

The rest of this chapter is organized as follows: in Sec. 5.1, we introduce various models of interest. In Sec. 5.2, we will summarize their performance behaviors. In Sec. 5.3, we will introduce a characterization quantity of the quantum reservoir and evaluate those models with this quantity. We will summarize this chapter in the last section.

5.1 Quantum reservoir models

We summarize the quantum systems employed for the quantum reservoir in this chapter and begin by introducing two Ising-type Hamiltonians: the ZZ-Ising model,

$$H_{ZZ} = \hbar J_0 \sum_{k>l} \frac{1}{|k-l|^\alpha} \sigma_k^z \sigma_l^z + \hbar g \sum_m \sigma_m^x, \tag{5.1}$$

and the XX-Ising model,

$$H_{XX} = \hbar J_0 \sum_{k>l} \frac{1}{|k-l|^\alpha} \sigma_k^x \sigma_l^x + \hbar g \sum_m \sigma_m^z, \tag{5.2}$$

where σ_l^μ ($\mu = x, y, z$) are the Pauli matrices at site l represented in the eigenvector space of the Pauli-z operator. J_0 and g are the strength parameters of the interaction and magnetic field. α is the exponent for the power-law decay of the interaction strength. Note that the dynamics provided by those Hamiltonians are essentially identical since H_{XX} can be obtained by rotating H_{ZZ} as

$$H_{XX} = \exp\left(\frac{i\pi}{4} \sum_k \sigma_k^y\right) H_{ZZ} \exp\left(-\frac{i\pi}{4} \sum_k \sigma_k^y\right). \tag{5.3}$$

As a variant of the ZZ-Ising model, we will also consider the ZZX model whose unitary map is given as

$$U_{ZZX} = \exp\left(-\theta^x \sum_m \sigma_m^x\right) \exp\left(-\theta^z \sum_{k>l} \frac{1}{|k-l|^\alpha} \sigma_k^z \sigma_l^z\right), \tag{5.4}$$

where θ^x and θ^z are angles. This unitary also resembles the DTC model we have considered in the previous chapter, although U_{ZZX} does not have a number of drives. We hence expect that a large value of θ^z is necessary. We will also consider several models that are more circuit-like: the random Clifford, Clifford+T, and random single-qubit rotation (RSR) models. The random Clifford model is defined as a random circuit consisting of only Clifford operations, whereas, in the Clifford+T model, non-Clifford single-qubit gates HTH are applied to each qubit after a random Clifford circuit. The RSR model consists of only single-qubit rotation operations uniformly sampled from U(2). We will also use the Haar-random model, whose unitary is uniformly sampled from U(2^N) as a reference.

5.1.1 Simulation setup for QERC

We perform the classification problem on the MNIST dataset [109] of handwritten digit images, where each image has 784 ($= 28 \times 28$) pixels, with ten qubits in QERC. 60,000 images out of 70,000 are used for training, while the remaining 10,000 images will be used for testing. To optimize the parameters of the ONN, we employ the adaptive gradient (AdaGrad) algorithm [116] with the mini-batch method. Accuracy rates that will be shown are averaged over 40-50 epochs, and their associated standard deviations will also be provided if necessary.

5.2 Performance behaviors

model	Train acc. (std.)	Test acc. (std.)
Haar	0.990 ± 0.0004	0.969 ± 0.0004
RSR	0.812 ± 0.0011	0.811 ± 0.0015
ZZ-Ising	0.987 ± 0.0004	0.970 ± 0.0006
XX-Ising	0.986 ± 0.0004	0.968 ± 0.0006
ZZX	0.986 ± 0.0003	0.972 ± 0.0005
Random Clifford	0.902 ± 0.001	0.884 ± 0.002
Clifford+T	0.990 ± 0.000	0.970 ± 0.001

Table 5.1: A list of training and testing accuracy rates achieved with different quantum reservoirs. The accuracy rates are averaged over epochs, and std. indicates their associated standard deviations.

We now perform QERC with the ZZ-Ising model and plot its performance in Fig. 5.1. α and the evolution time t are set as $\alpha = 1.5$ and $J_0 t = 3.5$. For $g/J_0 \approx 1$, a peak of the accuracy rates is observed, but the decrease of the accuracy rate in the regime of $g/J_0 > 1$ can be improved by randomizing the initial state before the quantum reservoir as depicted in Fig. 5.1. For $g/J_0 > 1$, the accuracy rate for both training and testing becomes comparable to the Haar-random case. It thus shows that the ZZ-Ising model at this parameter configuration is capable enough of performing the MNIST task at comparable accuracy rates to the Haar-random model.

A reason why a certain strength g of magnetic fields is necessary is that diagonal unitaries (in the computational basis) are functionally identical to the identity in QERC, yielding the same accuracy rate as the case with no reservoir dynamics. It can be seen more clearly in Fig. 5.1. The QERC performance with the XX-Ising model does not strongly depend on the strength

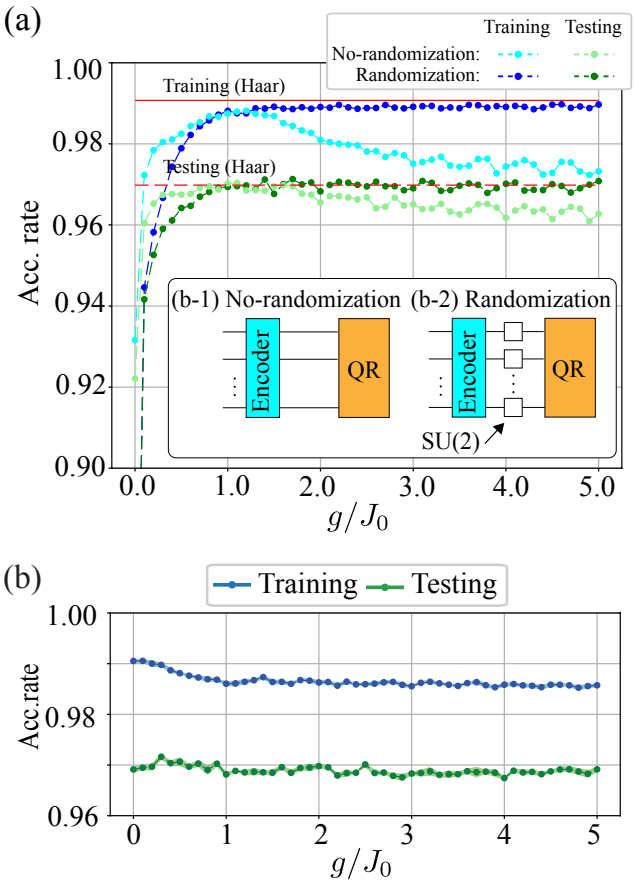

Figure 5.1: The behavior of the epoch-averaged accuracy rate against g in (a) the ZZ-Ising and (b) XX-Ising models. The solid and dashed horizontal lines, colored red, indicate the training and testing accuracy rates of the Haar-random model.

of the magnetic fields g, although the dynamics is just a rotated ZZ-Ising model (Eq. 5.3). In this sense, the role of the transversal magnetic field g in H_{ZZ} can be viewed as making information processed by the diagonal dynamics accessible through interference in our measurement basis space.

Let us now consider the ZZX model to check how much interference is necessary to obtain nontrivial output from the quantum reservoir. As the ZZX model has the Pauli-x rotation unitary followed by the diagonal unitary, one can examine how well performance can be achieved by varying the rotation angle θ^x. In Fig. 5.2(a), the accuracy rate for training and testing is plotted for different θ^z against θ^x. It is observed that a symmetric concave behavior against θ^x for those different θ^z. For $\theta^x = 0$, one can-

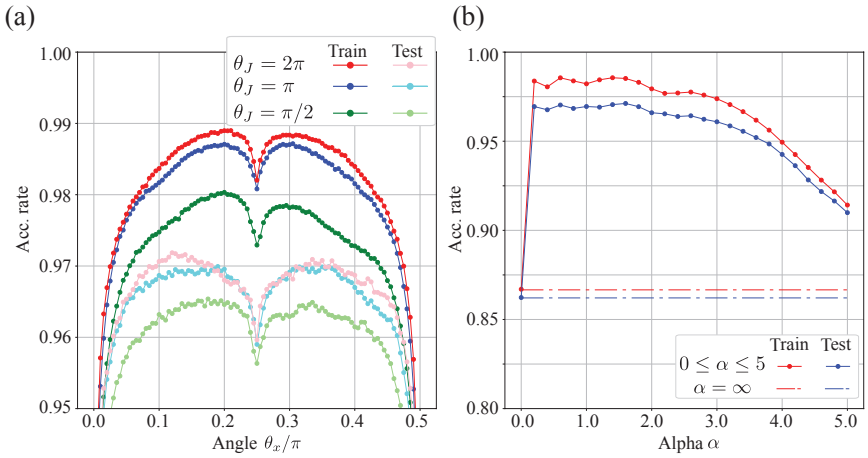

Figure 5.2: The accuracy behavior of the ZZX model. (a) The θ^x-dependency with different θ^z, $\theta^z = \pi/2, \pi$ and 2π. (b) The α-dependency of the accuracy rate for training and testing. the dash-dotted horizontal lines represent the accuracy with $\alpha = \infty$.

not extract nontrivial outputs since the unitary evolution is diagonal in our measurement basis. As θ^x increases, one can obtain informative outputs, resulting in a higher accuracy rate. Around $\theta^x = \pi/8$, the testing accuracy rate becomes highest, while the training one is still increasing. Thus, the underfitting regime in $0 \leq \theta \leq \pi/8$ and the overfitting regime in $\pi/8 \leq \theta < \pi/4$ can be observed. There is a dip at $\theta^x = \pi/4$, which will be revisited in Sec. 5.3. It is also observed that the accuracy rate becomes higher for larger θ^z in consistency with the ZZ-Ising case, where we need a sufficiently long time evolution. It is now clearer that the diagonal unitary with the ZZ interactions maps input states to intermediate states, which are potentially informative for learning, and the Pauli-x rotations extract nontrivial features from the states.

We next explore the behavior of QERC performance for different α in the XX-Ising model. In Fig. 5.3, we provide the training accuracy rate for $\alpha = 0, 1.5, \infty$ against $J_0 t$. A periodic behavior of the training accuracy rate is observed for $\alpha = 0$ and ∞. This is because the unitary operator defined by H_{XX} has a periodicity. Namely, $U(t) = \exp(-itH_{XX}/\hbar)$ becomes the identity at $t = n\pi/J_0$ for $n \in \mathbb{Z}$. Noting that $[\sigma_j^x \sigma_k^x, \sigma_l^x \sigma_m^x] = 0$ for any sites

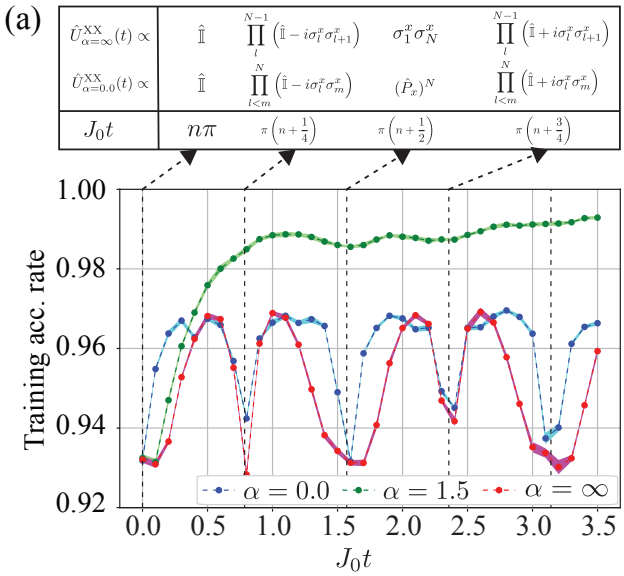

Figure 5.3: The training accuracy behavior of the XX-Ising model against t for different α. In the inset table, the forms of the unitary operator with $\alpha = 0, \infty$ at specific t are summarized.

j, k, l, m, the unitary operator can be written in the following form:

$$\begin{aligned} U(t) &= \prod_{j>k} \left(\cos\left(\frac{J_0 t}{|j-k|^\alpha} \right) I - i \sin\left(\frac{J_0 t}{|j-k|^\alpha} \right) \sigma_j^x \sigma_k^x \right), \\ &= \begin{cases} \prod_j \left(\cos(J_0 t) I - i \sin(J_0 t) \sigma_j^x \sigma_{j+1}^x \right) & (\text{for} \quad \alpha = \infty) \\ \prod_{j>k} \left(\cos(J_0 t) I - i \sin(J_0 t) \sigma_j^x \sigma_k^x \right) & (\text{for} \quad \alpha = 0) \end{cases} . \end{aligned} \tag{5.5}$$

Thus, the unitary operators with α being 0 and ∞ have the periodicity, $U(t) = U(t + n\pi/J_0)$ for $n \in \mathbb{Z}$. This periodicity yields the periodic behavior in the accuracy rate, as shown in Fig. 5.3, which prevents the accuracy rate from growing, in contrast to the case with $\alpha = 1.5$.

This periodicity crucially affects the QERC performance with the ZZX model as well as shown in Fig. 5.2(b). By replacing σ^x with σ^z in Eq. 5.5 for $\alpha = 0$ and $\alpha = \infty$, those reduced forms become ones of the ZZX model. Thus, if the complexity of the interaction unitary in the ZZX model is bounded due to the periodicity, the QERC performance does not become decent, as the Pauli-x rotations in the ZZX model do not provide nontrivial outputs on their own.

To connect the performance behavior of QERC with more common quantities in quantum information, we now consider the random Clifford, Clif-

ford+T, and RSR models. The random Clifford model can generate entanglement, but it is not universal due to the lack of non-Clifford gates. The RSR model on the other hand has non-stabilizerness without entanglement. The Clifford+T model possesses both these properties. In Table 5.1, neither the random Clifford nor the RSR models achieves decent performance, while the Clifford+T model achieves a high performance comparable with the Haar-random model. This suggests that neither non-stabilizerness nor entanglement is sufficient on its own, and both are required for the quantum reservoir. It is also worth noting that the Clifford+T is not as random as the Haar-random model in the sense of unitary designs. Hence, the common characterizations of unitaries in quantum information based on non-stabilizerness, entanglement, or unitary designs do not directly capture the learning behavior of QERC. Thus, let us consider another characterization suitable for classification settings in the next section.

5.3 Participation ratio for a set of quantum states

In QERC, the probability vector obtained in the measurement step is what the classifier sees. It suggests us to look into the probability vector. For this purpose, we employ the participation ratio (PR), which is widely used in condensed matter physics [118]. The PR of a quantum state $|\psi\rangle$ in the computational basis is defined as

$$\mathrm{PR}(|\psi\rangle) = \frac{1}{\sum_j |\langle j|\psi\rangle|^4}. \tag{5.6}$$

The PR is high for quantum states distributing equality over the basis states, and it is low for localized states.

For our classification setting, let us define two quantities based on the PR for a set $\mathbb{D}$ of quantum states, which are the PR averaged over $\mathbb{D}$ (APR) and the PR of the probability averaged over $\mathbb{D}$ (IAPR),

$$\mathrm{APR}(\mathbb{D}) = \left\langle \frac{1}{\sum_j |\langle j|\psi\rangle|^4} \right\rangle_{|\psi\rangle\in\mathbb{D}}, \text{ and } \mathrm{IAPR}(\mathbb{D}) = \frac{1}{\sum_j \langle |\langle j|\psi\rangle|^2 \rangle^2_{|\psi\rangle\in\mathbb{D}}}, \tag{5.7}$$

where $\langle\cdot\rangle_{|\psi\rangle\in\mathbb{D}}$ denotes averaging the quantity inside the bracket over all quantum states $|\psi\rangle$ in $\mathbb{D}$. Here, we consider a set of quantum states that are obtained from the initial encoding states through a quantum reservoir on the MNIST dataset. If many quantum states in $\mathbb{D}$ are localized on certain

computational basis states, the APR becomes small. The IAPR on the other hand can be either large or small. If those quantum states are localized on the same basis states, the IAPR may be comparable with the APR value. However, if the localization is happening on different bases, the averaged probability can be flatter, yielding a higher IAPR value. Thus, if a quantum reservoir produces a low APR and a high IAPR value, the probability vectors are expected to have a narrow overlap, suggesting ease for the ONN in capturing and distinguishing them.

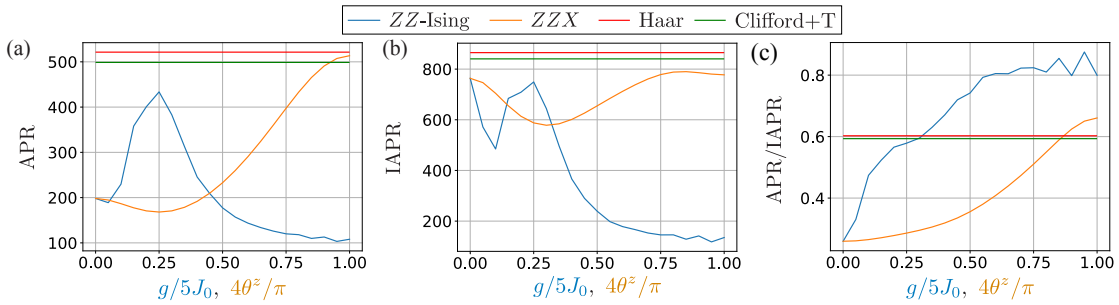

Figure 5.4: The behavior of (a) APR, (b) IAPR, and (c) the ratio of those is depicted for the ZZ-Ising, ZZX, Haar, and Clifford+T models. For the ZZ-Ising model, the horizontal axis is associated with g scaled as $g/5J_0$, while, for the ZZX model, it is so with θ^x scaled as $4\theta^x/\pi$.

In Fig. 5.4(a), the APR for ZZ-Ising, ZZX, Haar, and Clifford+T models is shown. For the ZZ-Ising model, the curve has a peak around $g/J_0 = 1$, indicating that each state has a support over many computational basis states. In the ZZX model, for a larger angle θ^x, the APR tends to be higher. The APR of the Haar-random and Clifford+T models is higher than that of the others. In comparison with the behavior of the accuracy rate we have observed, the APR tends to follow the training accuracy rate. This is because a higher APR value may indicate more degrees of freedom as the quantum state is distributed over more basis states.

In Fig. 5.4(b), the IAPR for ZZ-Ising, ZZX, Haar, and Clifford+T models is shown. For the ZZ-Ising model, the curve has a peak around $g/J_0 = 1$, implying that the largest number of the computational basis states is in participation. For $g/J_0 > 1$, the IAPR is small, and many quantum states share common basis states. For the ZZX model, there is a dip around $\theta^x = \pi/16$, while the IAPR does not drop as in the ZZ-Ising case. The IAPR of the Haar-random and Clifford+T models is higher than the others. For the ZZ-Ising and ZZX models with $g/J_0 = 0$ and $\theta^x = 0$, the IAPR value is relatively high, probably because of the PCA.

In Fig. 5.4(c), we show the ratio, APR/AIPR, for those models. This ratio represents how much dimension is used by each quantum state on average, out of the effective dimension occupied by all the states. Although this quantity does not reflect everything about learning behavior, it still captures certain properties of the accuracy rate behavior. An interesting observation is that, in the Haar-random and Clifford-T models, each quantum state occupies 60% of the effective dimension. Then, the quantum states still have sufficient distinguishability for the ONN, given the fact that the accuracy rate of those models is high. At $\theta^x = \pi/4$ in the ZZX model, the ratio exceeds 60% while its IAPR is lower than that of the Haar-random model. The quantum states may be overlapping with each other too much, as the accuracy rate curve of the ZZX model against θ^x has the dip at $\theta^x = \pi/4$.

5.4 Summary

In this chapter, we demonstrated that the ZZ- and XX-Ising Hamiltonian dynamics can achieve a high accuracy rate comparable to that of the Haar-random model, implying that many-body interactions involving more than two bodies are not necessary. This result also provides a fundamental understanding of physical resources for QERC that two-body interactions in physical Hamiltonian dynamics can be exploited to achieve sufficiently high performance. In the ZZX model, it also became clear that the layer of Pauli-x rotations can extract informative features generated by the ZZ interactions.

We also investigated the learning behavior of more circuit-oriented models to gain insights into the type of engineering required. Examining the random Clifford, RSR, and Clifford+T models, both entanglement and non-stabilizerness are shown to be required for an effective quantum reservoir, while it is still unclear how much each is needed. This implies that characterizing those properties on their own does not capture the learning behavior, suggesting different approaches for characterizing the quantum reservoir.

We introduced the APR and IAPR, which are two different extensions of the PR to a set of quantum systems in our classification setting. Although those quantities do not capture all trends of learning behavior perfectly, APR tends to follow the training accuracy rate, and the ratio (APR/IAPR) can reasonably provide clear insights into how the ONN sees the probability vectors obtained from the quantum reservoir. As the PR has been originally employed for investigating localized states and ergodic states [118], this characterization of the quantum reservoir is bridging such physical properties and ML behaviors. However, the characterization does not have guarantees from

the ML theoretical point of view, suggesting further exploration of the learning behavior of QERC.

From an engineering perspective, those results suggest a more simplified implementation of an effective quantum reservoir, particularly the availability of intrinsic dynamics in several quantum platforms, such as trapped ions and neutral atoms, without the need for extensive engineering. However, regarding the α-dependency, long-range interactions are still preferable, which limits the ease of implementation in other physical platforms, such as superconducting qubits [10].

Based on the discussion above, one may expect a significant advancement in understanding QERC by investigating the impacts of network topology that connects qubits through two-body interactions in the quantum reservoir, which will be discussed in the next chapter.

Chapter 6

Roles of the interaction network of qubits and symmetries in QERC

In this chapter, we analyze the roles of the interaction network of qubits in the quantum reservoir of QERC. In the quantum reservoirs discussed in the previous chapter, long-range interactions are necessary for high QERC performance. In particular, the periodic property in time emerges in the unitary operator with nearest-neighbor interactions, yielding a periodic feature in the training accuracy rate and bounding it. In this sense, the periodicity, rather than short-range interactions, seems more intrinsic to poor performance.

Therefore, we investigate QERC performance using a quantum reservoir with various interaction networks in a different Hamiltonian, focusing on the periodicity. The analysis is simplified by virtue of spectral graph theory, which exploits the symmetries in the Hamiltonian. We will also explore an inherent ML property of those symmetries in QERC and demonstrate how to exploit the property.

This chapter is organized as follows: in Sec. 6.1, we introduce our Hamiltonian and summarize our numerical simulation settings. In Sec. 6.2, we will outline the mathematical properties of the Hamiltonian arising from its symmetry and the interaction network. In Sec. 6.3, we will evaluate the behavior of the QERC performance with different networks. In Sec. 6.4, we will investigate an inherent generalization of learning in the Hamiltonian and show a prescription for extracting it. At last, we will summarize our results in Sec. 6.5.

6.1 The XY model on graphs for QERC

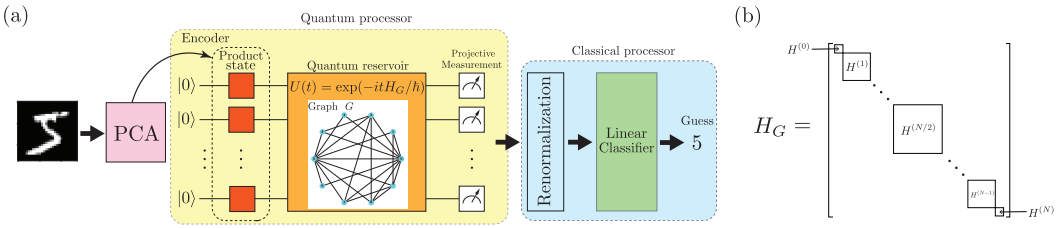

Figure 6.1: (a) A schematic illustration of QERC considering the interaction network in the quantum reservoir. (b) An illustration for the block-diagonal structure of the XY Hamiltonian for even N. Each block $H^{(k)}$ coincides with the adjacency matrix of $\wedge^k G$.

We consider QERC to investigate the effect of the interaction connectivity in the reservoir on QERC performance, shown in Fig. 6.1. For this purpose, we employ the following Hamiltonian on N qubits, referred to as the XY Hamiltonian [119], for the quantum reservoir.

$$H_G = \frac{\hbar J}{2} \sum_{(i,j)\in E} \left(\sigma_i^x \sigma_j^x + \sigma_i^y \sigma_j^y\right). \tag{6.1}$$

where $G = (V, E)$ is a graph with V, E being the sets of vertices and edges, respectively, while J is the coupling strength of the edges.

The dynamics of this Hamiltonian on different graphs has been extensively investigated in a number of contexts, for instance, perfect state transfer and continuous-time quantum walks [120–122]. We hence employ the XY Hamiltonian dynamics $U(t) = \exp\left(-itH_G/\hbar\right)$ for the quantum reservoir to discuss the properties of the quantum feature map that different graphs induce.

6.1.1 Numerical simulation settings

We perform QERC using the XY model with ten qubits ($N = 10$) to solve the MNIST image classification task [109]. Our training procedure follows the original one [50] with the following settings. 60,000 images out of 70,000 are used for training with the AdaGrad optimizer [116]. We employ the mini-batch training method, with a batch size of 100, where the training parameters are updated using 100 randomly chosen samples per iteration. We keep training up to 51 epochs, where one epoch consists of 600 iterations

in our case. The other 10,000 images are used for testing to examine how well the trained QERC classifies images that are not used for training. The accuracy rate of training and testing indicates how many images are correctly classified out of the training and testing image sets, respectively.

6.2 The periodicity of unitary maps

We first begin by considering the periodicity of unitary maps given by Eq. 6.1. In the XX Ising model with certain parameter settings, the unitary operator exhibits a periodicity such that the unitary map evolves back to the identity at a certain time $\tilde{t}$ that is $U(\tilde{t}) = \exp\big(-i\tilde{t}H_{XX}/\hbar\big) = I$ as we have seen in the previous chapter. Such a periodicity bounds the dynamics, and hence the unitary map can grow only within the bound. The periodicity prevented the XX model from achieving high QERC performance when the interaction network in the Hamiltonian was restricted to nearest neighbors in 1D. Thus, it is worth exploring what graphs yield the periodicity in the XY dynamics.

6.2.1 The periodicity and Hamiltonian spectra

In the XY dynamics, such periodicity can be analyzed through eigenvalues of the Hamiltonian [119, 123]. The spectral condition for the Hamiltonian to exhibit the periodicity is as follows: let μ_j $(j = 0, 1, \cdots, 2^{N-1})$ be the eigenvalues of the Hamiltonian H where N is the number of qubits. Then *the unitary matrix given by H becomes the identity at $t = \tilde{t} \neq 0$ if and only if all the ratios of any two eigenvalues of H are rational, that is, $\mu_j/\mu_{j'} \in \mathbb{Q}$ for all j and j' with $\mu_{j'} \neq 0$* [123].

Using this spectral condition, we will characterize unitary maps on various graphs and investigate the QERC performance achieved with them.

6.2.2 The wedge product of graphs

To study the spectral condition for the periodicity in the XY model, one can exploit the symmetries in the Hamiltonian [123]. We first divide the Hamiltonian into symmetry sectors where the symmetry is given by $[H_G, \sum_{i=0}^{N-1} \sigma_i^z] = 0$, with σ_i^z being the Pauli-z operator on the i-th qubit. The entire Hilbert space is divided into $N+1$ subspaces $\mathcal{H}^{(k)}$ $(k = 0, 1, \cdots N)$. Each subspace $\mathcal{H}^{(k)}$ is spanned by some basis states for which the expectation value of the total magnetization satisfies that $\langle \sum_{i=0}^{N-1} \sigma_i^z \rangle = N - 2k$. For example, the Pauli-z basis set, which is our computational basis set, can be divided into $N+1$ subsets $\Gamma^{(k)}(k = 0, 1, \cdots, N)$ whose of each elements satisfy that

$\langle \sum_{i=0}^{N-1} \sigma_i^z \rangle = N - 2k$:

$$\begin{aligned}
\Gamma^{(0)} &= \{|00\cdots 0\rangle\}, \\
\Gamma^{(1)} &= \{|10\cdots 0\rangle, |010\cdots 0\rangle, \cdots, |0\cdots 01\rangle\}, \\
\Gamma^{(2)} &= \{|110\cdots 0\rangle, |0110\cdots 0\rangle, \cdots, |0\cdots 011\rangle\}, \\
&\cdots \ , \\
\Gamma^{(N)} &= \{|11\cdots 1\rangle\}.
\end{aligned} \tag{6.2}$$

Each $\Gamma^{(k)}$ spans $\mathcal{H}^{(k)}$. One can rewrite the Hamiltonian matrix H_G represented in our computational basis set as $H^{(0)} \oplus H^{(1)} \oplus \cdots \oplus H^{(N)}$ with an appropriate permutation of the computational basis states where the submatrices $H^{(k)}$ live in $\mathcal{H}^{(k)}$ and $\oplus$ denotes the direct sum of matrices (see Fig. 6.1(b) for a schematic illustration). In other words, H_G is block-diagonalized in our computational basis set. We use this property through this paper. Thus, the entire unitary $U_G(t) = \exp(-itH_G/\hbar)$ exhibits the periodicity if and only if $U^{(k)}(t) = \exp\big(-itH^{(k)}/\hbar\big)$ is periodic for all k with the same period. Furthermore, the Hamiltonians $H^{(k)}$ acting on each symmetry sector are *identical* to the adjacency matrices of the wedge products of the original graph G, $\wedge^k G$ [119]. Thus, the spectrum of $H^{(k)}$ is given by that of $\wedge^k G$ meaning the spectral condition for periodic unitary maps is equivalent to the fact that the spectra of graphs $\wedge^k G$ for all $k = 0, \cdots, N$ are integers up to a common irrational multiplicative factor. This structure simplifies our analysis of the unitary map for the quantum reservoir using the XY dynamics.

6.3 Periodicity and QERC performance

Now as an example of a graph that brings the periodicity to the unitary map is the complete graph K_N where all the vertices are fully connected (see Fig. 6.2(a)). Using the properties of the XY model discussed in the previous section, the unitary matrix $U_{K_N}(t)$ exhibits periodicity as it evolves to $\sigma^z \otimes \cdots \otimes \sigma^z$ and I at $Jt = \pi$ and 2π, respectively. Both are diagonal in the computational basis which means they do not work well in QERC. We here explain that the unitary matrix of the XY model on K_N with even N reduces to P^z_{N-1} at $Jt = \pi$ and the identity at $Jt = 2\pi$. According to [119], the blocks $H^{(k)}$ of the XY Hamiltonian with K_N are identical to the Johnson graphs with the following spectra

$$\begin{aligned}
\mathrm{Spec}(H^{(k)}) = \{ \ \ \mu_j \ \ &| \ j = 0, 1, \cdots, k\}, \\
\text{where } \mu_j \ &= \ k(N-k) - j(N+1-j).
\end{aligned} \tag{6.3}$$

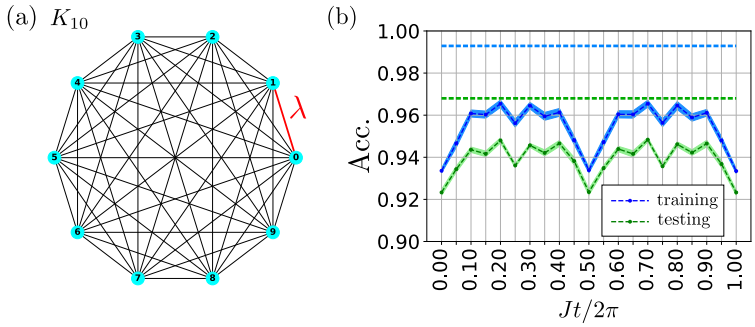

Figure 6.2: Periodic feature in accuracy rate with K_N (a) The $N = 10$ complete graph K_N, where red edge indicates the edge weighted by $\lambda \in [0.5, 1.0]$. Unless explicitly mentioned, λ is set to 1 and equivalent to other edges. (b) The accuracy rates for training (blue) and testing (green) of the QERC with the complete graph K_{10}. Those accuracy rates are averaged over epochs from 41 to 51. The shaded areas represent the standard deviation over the same epoch range at each data point. The two horizontal dashed lines depict the training and testing accuracy rates for the Haar-random case where the unitary matrix is sampled from the unitary group with the Haar measure.

If N is even, all the eigenvalues μ_j are either odd for odd k or even for even k. Thus, the corresponding unitary $U^{(k)}(t) = \exp\left(-itH^{(k)}/\hbar\right)$ is equivalent to the identity up to the sign (I or $-I$) at $Jt = \pi$ for all k. Hence, the entire unitary $U_G(t = \pi/J)$ is diagonal with elements being either 1 or -1, and it is equivalent to P^z_{N-1} with an appropriate permutation in the computational basis. Therefore, $U_G(t = 2\pi/J) = I$.

The periodicity brings a periodic feature in the accuracy rate of QERC as shown in Fig. 6.2(b). The reason for the periodic feature is that the unitary matrices at $Jt = \pi$ and 2π are both diagonal in the computational basis, noting that diagonal matrices do not produce a nontrivial feature map in QERC. The accuracy rates at $Jt = \pi, 2\pi$ are almost equivalent to those of the identity ($Jt = 0$).

Another interesting feature of this accuracy behavior is that the top of the accuracy rate between $Jt = 0$ and π, does not become comparable to the level of the Haar-random case. This suggests that the periodicity of the unitary prevents the unitary from growing as a good quantum feature map in QERC.

The periodicity is very sensitive to the connectivity of the underlying graph since the spectrum property must hold strictly for all k. To examine this, we consider two types of perturbations in the weights of the edges.

One of the two perturbations is examined as follows. We parametrize an interaction strength of the red edge in Fig. 6.2(a) in the range $\lambda \in [0, 1]$. If $\lambda = 1$, it indicates the complete graph K_N while the red edge no longer exists for $\lambda = 0$. In Fig. 6.3(a), the training and testing accuracy curves for $\lambda = 1$ and $\lambda < 1$ are depicted by dashed curves and filled areas, respectively. The periodic feature of the accuracy rates in the case with $\lambda = 1$ is immediately broken even with a small deviation of λ from one.

The other perturbation is a random perturbation in all the edges. Now, the weight of each edges is given by $1 + wr$ where w is the amplitude and r is a uniform random variable taken from the range $[-1, 1)$. The accuracy rates for different w are depicted in Fig. 6.3(b). For $w = 0.01$, the periodicity does still remain, even though the spectral condition is not satisfied. However, for $w = 0.1$, the accuracy curves are off from the periodic one ($w = 0$). For those two different perturbations, one can find that the periodicity can be broken for finite perturbations in edges while the accuracy rates do not reach the same level as the Haar-random case.

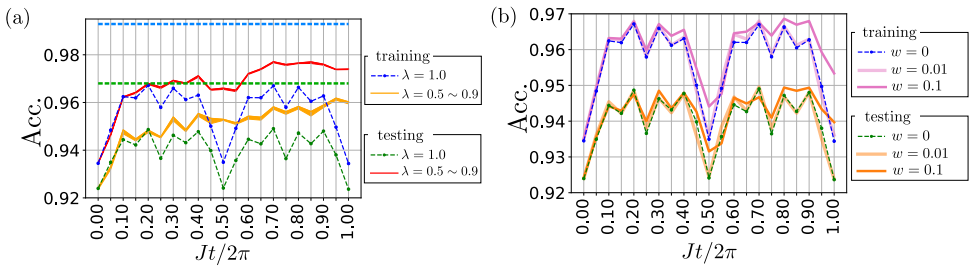

Figure 6.3: Sensitivity of the periodicity of the accuracy rate to a perturbation in edge connections. (a) The dashed blue and green curves depict the training and testing accuracy rates with $\lambda = 1.0$, respectively. The upper and lower boundaries of the shaded areas are given by the maximum and minimum accuracy rates among the cases of $\lambda = 0.5, 0.6, 0.7, 0.8$, and 0.9. (b) all edges are perturbed with random weights whose amplitude is w. The weight of each edges is given by $1 + wr$ where r is a uniform random variable in $[-1, 1)$.

6.3.1 On the path graph and cycle graph

We have observed that the periodicity of unitary does not provide a proper growth of a quantum feature map even if perturbations are introduced in the interaction strengths. Fortunately, we can achieve non-periodic unitary

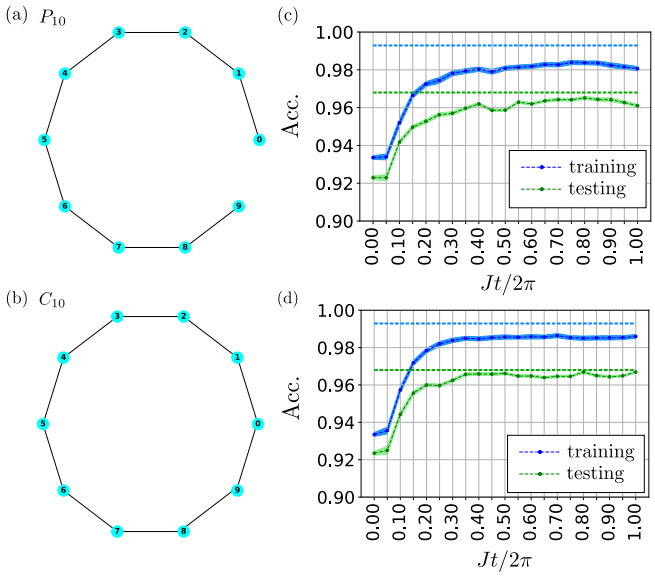

Figure 6.4: Illustration of the path graph P_N in (a) and cycle graph C_N in (b) for $N = 10$. The accuracy rates for training (blue) and testing (green) of the QERC with the XY Hamiltonian (6.1) on the two graphs (c) P_{10} and (d) C_{10}. At each data point, the accuracy rate is averaged over epochs from 41 to 51. The shaded areas represent the associated standard deviation. The blue and green horizontal dashed lines depict the accuracy rates of the Haar-random case.

maps on simple graphs such as a path graph P_N and a cycle graph C_N shown in Fig. 6.4(a), (b) as the spectral condition is not satisfied.

Non-periodicity of the path and cycle graphs

Let us briefly explain that the path graph does not satisfy the spectrum condition for unitary maps to exhibit the periodicity of which the unitary goes back to the identity in a certain period (see [121] for the proof). It is sufficient to see that the condition does not hold for a certain k. For the path graph P_N, the spectrum of the wedge product $\wedge^k P_N$ can be determined by one of the Cartesian k-th power of P_N, denoted by $\square^k P_N$ [119]. Thus, the the eigenvalues $\mu_{j_0,\cdots,j_{k-1}}$ of the Hamiltonian $H^{(k)}$ for a fixed k are given by

$$\mu_{j_0,\cdots,j_{k-1}} = 2\sum_{l=0}^{k-1} \cos\left[\frac{\pi(j_l+1)}{N+1}\right] \tag{6.4}$$

with $j_0 < \cdots < j_{k-1}$ and $j_l \in \{0, 1, \cdots, N-1\}$. For $k = 1$, two of the eigenvalues are $\mu_0 = 2\cos\theta$ and $\mu_1 = 2\cos(2\theta)$ where $\theta = \pi/(N+1)$. It has been shown that $\cos 2\theta / \cos\theta$ is irrational for $N \geq 4$ where $\theta = \pi/(N+1)$ [121]. Hence, the spectral condition for $H^{(1)}$ does not hold.

We next prove that the unitary matrix of the XY Hamiltonian on C_N does not satisfy the spectrum condition for $N = 5$ or $N \geq 7$. In the proof, we will use the following proposition.

Proposition 6.3.1 (Corollary 3.12 of [124]). *If $\theta = 2\pi r$ and $r \in \mathbb{Q}$, the only rational values of $\cos\theta$ are $0, \pm 1/2, \pm 1$.*

As we mentioned in the previous section, it is sufficient to consider a certain k, and here we chose $k = 1$. Now the spectrum of $H^{(1)}$ is the adjacency matrix of the cycle graph C_N and, for even N, is given [125] by

$$\mu_j = 2\cos\left(\frac{2\pi j}{N}\right), \quad j = 0, 1, \cdots, N-1. \tag{6.5}$$

Then, to prove that the spectrum condition does not hold, it is sufficient to show that $\cos(2\pi/N)$ is irrational for $N = 5$ or $N \geq 7$ since we can choose $\mu_0 = 2$. It is immediately shown by Prop. 6.3.1, noting that $0 < 2\pi/N \leq 2\pi$.

QERC performance with the path and cycle graphs

The QERC accuracy rates with the XY model on those graphs are depicted in Fig. 6.4(c), (d). Interestingly, the accuracy rates with the two graphs are growing in time t, whereas the XX model with the nearest-neighbor interactions has the periodicity. In addition, the testing accuracy rate of the cycle graph C_{10} is comparable to the one of the Haar-random case at a certain time, but the path graph P_{10} is not so. On the other hand, the training accuracy rates of both graphs do not reach the same level as the Haar-random case. Hence, the XY model with C_{10} exhibits a testing accuracy rate and a better generalization performance compared with the Haar-random case. These results, including the case with the complete graph, suggest that the periodicity should be avoided for an effective quantum feature map and that the existence of the periodicity can be verified by the spectral condition based on the underlying graph representing the interaction network.

6.3.2 Disconnected graphs

Now let us consider another impact of the interaction network on the properties of the QFM. Fig. 6.5(a) depicts a specific graph we will consider here

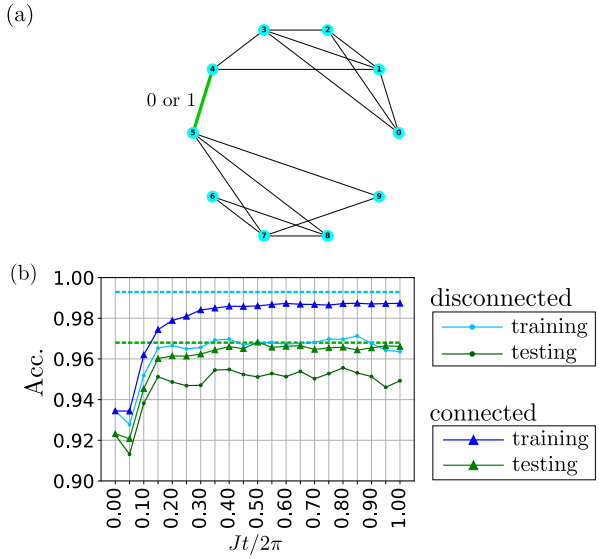

Figure 6.5: Difference in the accuracy rate between connected and disconnected graphs (a) where it can become either disconnected or connected via the green edge. If its weight is 0 (1), the graph is disconnected (connected). In each cluster, the edges have been randomly chosen. (b) The accuracy rates for training and testing of the graph depicted in (a). The disconnected case is marked by dots, and the connected case is marked by upward triangles. The blue and green horizontal dashed lines represent the accuracy rates of the Haar-random case.

which can be either connected or disconnected by switching on or off the weight of the green edge between qubits 4 and 5. When the weight is zero, the graph is disconnected while it becomes a connected graph for the weight being one.

The accuracy rates with the graph are shown in Fig. 6.5(b) where we observe the accuracy rates of the disconnected case are inferior to that of the connected case in both training and testing. One of the reasons is that we encode classical information into product states with single-qubit rotations, and hence the final states are still product states in the disconnected case, implying that the entire Hilbert space is not utilized. It can be clearly illustrated in our graph analysis. The unitary matrix can be further block-diagonalized into smaller blocks in the disconnected case. As the XY dynamics preserves the total magnetization, that of each disconnected region should be respectively preserved. In fact, $\wedge^k G$ is disconnected for all k if G is disconnected (see Theorem 2 in Ref. [126]). Such a smaller block structure could prevent QERC from achieving a higher accuracy rate as examined in the following

section.

6.4 The block structure and its impact on QERC performance

In this section, we explore the impact of the block size on the QERC performance and how to improve the performance of the XY model. In the first subsection, we first investigate how the block size generally affects the training accuracy rate of QERC with artificial unitary matrices. Based on that, in the following subsection, we will discuss how to improve the QERC performance with the XY model on the simple graph, C_{10}.

6.4.1 Influence of the block size to accuracy rate

Let us begin by introducing artificial unitary matrices with equitable partitions for the basis set. An equitable partition here is denoted by $[S \times M]$, where S is the size of each partition and M is the number of partitions, and hence $S \times M = d = 2^N$. We define a unitary matrix $U_{[S\times M]}$ with a block structure as $U_{[S\times M]} = U^{(1)} \oplus \cdots \oplus U^{(M)}$, as illustrated in Fig. 6.6(a), where each $U^{(m)}$ is sampled using the Haar random sampling from the unitary group $\mathrm{U}(S)$.

The plot of Fig. 6.6(b) shows the accuracy rates of the equitable partition model for different M. It is observed that the training accuracy rate decreases as the number of partitions M increases (the block size decreases). This behavior can be explained by the participation ratio since it is closely related to the training accuracy rate as investigated in the previous chapter of the thesis. We here provide a theoretical analysis of the relation between the block structure and the inverse participation ratio, showing that the block structure increases the inverse participation ratio, which implies that the training accuracy rate tends to be lower. We use the inverse PR (IPR) instead of the PR as it is a measure closely related to the training accuracy rate and is easier to theoretically deal with. The IPR is given by [127]

$$\mathrm{IPR}(|\psi\rangle) = \sum_j |\langle j|\psi\rangle|^4, \tag{6.6}$$

where $|\psi\rangle$ is a quantum state for which we want to calculate the IPR. Here, we calculate the overlap with the computational basis, $|j\rangle$ since the QERC measurement is a projective measurement in the computational basis. We define the averaged IPR (AIPR) for a set of quantum states since the quantum

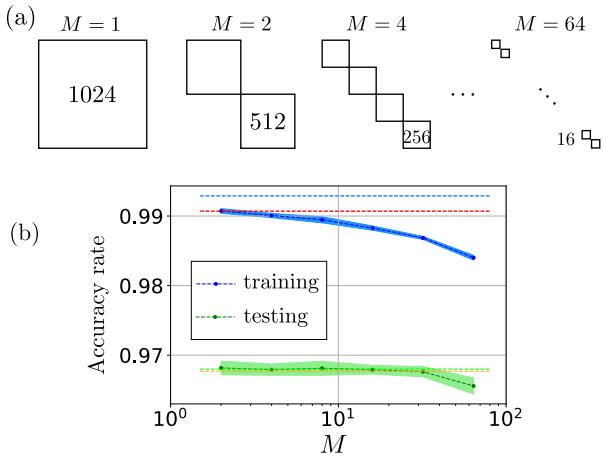

Figure 6.6: (a) Schematic illustration of the equitable partition for different M. A number in a block (or near a block) denotes the block size. (b) Robustness of testing accuracy rate to equitable partitions. The behavior of the accuracy rate in training (blue) and testing (green) for different partitions M. The dashed curves represent the averaged accuracy rates averaged over ten realizations of random unitary matrices at each data point. The shaded areas are the corresponding standard deviation. The dashed four horizontal lines depict the accuracy rates of the Haar-random model U_{Haar} (corresponding to $M = 1$) and $U_{\mathrm{Haar}}^{(\mathrm{BD})}$ as follows: training of U_{Haar}, training of $U_{\mathrm{Haar}}^{(\mathrm{BD})}$, testing of $U_{\mathrm{Haar}}^{(\mathrm{BD})}$, and testing of U_{Haar} from above (blue, red, green and orange). $U_{\mathrm{Haar}}^{(\mathrm{BD})}$ denotes the case where the unitary has the same block structure as the XY model, and each block is Haar-randomly sampled.

reservoir processes many quantum states encoding different classical information. Denoting the set of the initial states as $\mathbb{D} = \{|\psi^{(n)}\rangle\}_{n=1}^{D}$, the AIPR for the final states obtained by a unitary map U is given by

$$\mathrm{AIPR}(\mathbb{D}) = \frac{1}{D}\sum_{n=1}^{D} \mathrm{IPR}(U|\psi^{(n)}\rangle). \tag{6.7}$$

For simplicity, we compare the AIPR between two unitary matrices with the same initial state. The first unitary matrix U_1 is block-diagonalizable with two blocks in our computational basis set, $U_1 = U^{(0)} \oplus U^{(1)}$ where $U^{(0)}$ and $U^{(1)}$ act on $\mathcal{H}^{(0)}$ and $\mathcal{H}^{(1)}$, respectively. The second one U_2 is not block-diagonalizable in the computational basis set. We assume that $U^{(k)}(k = 0, 1)$ and U_2 are sampled from the unitary groups $\mathrm{U}(d_k)$ and $\mathrm{U}(d)$ with the Haar measure, respectively, where $d_k = \dim(U^{(k)})$ and $d = d_0 + d_1$. For those unitary matrices, we let an initial state $|\psi_0\rangle$ evolve, which lies over the whole

Hilbert space. We define p_k $(k = 0, 1)$ as $p_k = \sum_{|i\rangle \in \Gamma^{(k)}} |\langle i|\psi_0\rangle|^2$, and hence $p_0 + p_1 = 1$, where $\Gamma^{(k)}$ denotes a set of the computational basis states that span $\mathcal{H}^{(k)}$. Then, for the first unitary U_1, which is block-diagonalizable, the IPR in average $\overline{r}^{(\mathrm{BD})}$ is given by

$$\overline{r}^{(\mathrm{BD})} = \mathbb{E}_{U^{(0)}} \mathbb{E}_{U^{(1)}} \left[\mathrm{IPR}(U_1|\psi_0\rangle) \right] = \sum_{k=0}^{1} \frac{2}{d_k + 1} p_k^2. \tag{6.8}$$

However, for the second unitary U_2, the IPR in average $\overline{r}$ can be expressed as

$$\overline{r} = \mathbb{E}_{U_2} \left[\mathrm{IPR}(U_2|\psi_0\rangle) \right] = \frac{2}{1 + \sum_{k=0}^{1} d_k} = \frac{2}{1 + d}. \tag{6.9}$$

Here, these IPRs correspond to the fixed initial state $|\psi_0\rangle$. Taking the average of the IPRs over the initial state set $\mathbb{D}$, we obtain the AIPRs associated with the unitaries,

$$\langle \overline{r}^{(\mathrm{BD})} \rangle_{\mathbb{D}} = \sum_{k=0}^{1} \frac{2}{d_k + 1} \langle p_k^2 \rangle_{\mathbb{D}}, \text{ and } \langle \overline{r} \rangle_{\mathbb{D}} = \overline{r}, \tag{6.10}$$

where $\langle \cdot \rangle_{\mathbb{D}}$ denotes the average over $\mathbb{D}$.

When one varies p_k for all initial states in $\mathbb{D}$, the following inequality for those AIPRs is given by applying the Lagrange multiplier method:

$$\langle \overline{r}^{(\mathrm{BD})} \rangle_{\mathbb{D}} \geq \left(1 - \frac{2}{d + 2} \right) \langle \overline{r} \rangle_{\mathbb{D}}. \tag{6.11}$$

Hence, for a large d, the lower bound of $\langle \overline{r}^{(\mathrm{BD})} \rangle_{\mathbb{D}}$ and $\langle \overline{r} \rangle_{\mathbb{D}}$ match well. Furthermore, the equality is satisfied with p_k being a certain value for *all the initial states* in $\mathbb{D}$, but it is not realistic in the QERC since we encode spreading PCA components into the hemispheres of Bloch spheres. Thus, it is expected that $\langle \overline{r}^{(\mathrm{BD})} \rangle_{\mathbb{D}} > \langle \overline{r} \rangle_{\mathbb{D}}$ in the QERC (In fact, it holds as numerically shown below).

The above discussion suggests that the block-diagonalizability in the computational basis set increases the IPR on average. It implies that the block-diagonalizability yields a lower training accuracy rate.

In Fig. 6.7, we show the validity of the inequality $\langle \overline{r}^{(\mathrm{BD})} \rangle_{\mathbb{D}} > \langle \overline{r} \rangle_{\mathbb{D}}$ for the MNIST dataset. The right hand side of the inequality is represented by a horizontal dashed line colored blue. The blue curve represents the numerically computed AIPR with a block-diagonalized unitary. The orange curve represents $\langle \overline{r}^{(\mathrm{BD})} \rangle_{\mathrm{MNIST}}$ of Eq. 6.10. Comparing the blue and orange curves, one can confirm that the expectations executed in Eq. 6.8,6.9 are

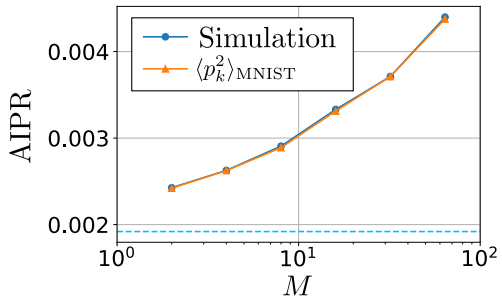

Figure 6.7: The validity of the inequality $\langle \overline{r}^{(\mathrm{BD})} \rangle_{\mathrm{MNIST}} > \langle \overline{r} \rangle_{\mathrm{MNIST}}$ for the MNIST dataset. The horizontal dashed line colored blue indicates the AIPR with a random unitary sampled from the unitary group $\mathrm{U}(2^{10})$, which corresponds to $\langle \overline{r} \rangle_{\mathrm{MNIST}}$. The blue curve shows the numerically obtained AIPR with the block-diagonalized unitaries with partition size M. The orange curve shows $\langle \overline{r}^{(\mathrm{BD})} \rangle_{\mathrm{MNIST}}$ of Eq. 6.10 with $\langle p_k^2 \rangle_{\mathrm{MNIST}}$ numerically obtained.

valid. As both curves are above the horizontal dashed line, $\langle \overline{r}^{(\mathrm{BD})} \rangle_{\mathbb{D}} > \langle \overline{r} \rangle_{\mathbb{D}}$ actually holds. Hence, it implies that the block-diagonalizability yields a lower training accuracy rate, and the behavior of the AIPR matches that of the training accuracy rate shown in Fig. 6.6(a).

Another interesting feature in Fig. 6.6(b) is the robustness of the testing accuracy rate. As the block size decreases, the training accuracy rate becomes lower, while the testing accuracy rate remains almost the same until the block size is reduced to about 32. This could suggest that the block structure can be exploited to avoid overfitting in QERC.

6.4.2 Improving the XY model with a simple interaction network

We now discuss the QERC performance of the XY model on a graph G from the point of view of the block structure. The block sizes that the XY model has are $(1, 10, 45, 120, 210, 252, 210, 120, 45, 10, 1)$. The accuracy rates of the XY model with the cycle graph C_{10} are 0.9853 for training and 0.9654 for testing, which are averaged over $Jt/2\pi = 0.75 \sim 1$. Both the training and testing accuracy rates are lower than that of a random unitary model $U_{\mathrm{Haar}}^{(\mathrm{BD})}$, which are 0.9907 and 0.9677, shown in Fig. 6.6(b). It implies that there is room to improve the XY model for higher performance. One of the possible ways is to find and implement a graph G yielding high performance. However, even if it were possible to find such a graph, it could have a complex

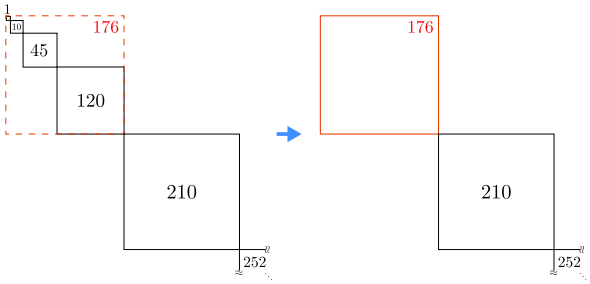

Figure 6.8: Schematic illustration of merging the first four blocks explained in the main text.

qubit interaction and be impractical.

Another way to improve the XY model could be to adjust the block sizes in the XY model, inspired by the training accuracy's behavior we have shown in Fig. 6.6(b). By regarding the first four blocks in the XY model as one block, the original block structure can be viewed as $(176, 210, 252, 210, 176)$, while the blocks of size 176 have finer block structure $(1, 10, 45, 120)$. If it were possible to merge those finer blocks and obtain a single dense block of size 176 such as shown in Fig. 6.8, the training accuracy rate of the unitary matrix could increase. However, if every block is sufficiently merged, the unitary would be as complex as the Haar-random model U_{Haar}, which suffers from overfitting. Thus, the block structure should still exist when merging smaller blocks for a good generalization performance. Therefore, it implies that "weakly" merging those block sectors could improve the QERC training accuracy rate while suppressing overfitting. In the following subsection, we will discuss how to effectively achieve this.

6.4.3 Small perturbations in the block structure

Here, we consider perturbations in the block structure with feasible interaction networks while still exploiting the merit of the block structure. As shown in Fig. 6.9(a), we apply the Pauli-x rotation gates $R_x(\theta) = \exp(-i\theta\sigma^x/2)$ to each qubit with the identical parameter $\theta \in [0, \pi/2]$ after the XY Hamiltonian dynamics on C_{10}. These rotation gates could break the block structure and merge the blocks, and hence the training accuracy rate is expected to be higher as suggested in Fig 6.6. The expected behavior of the testing accuracy rate is not very obvious, and the QERC might suffer from overfitting. However, the merit of the block structure to generalization performance may be obtained for a small value of θ since the blocks may be sparsely connected to each other, which might induce a similar contirbution to generalization

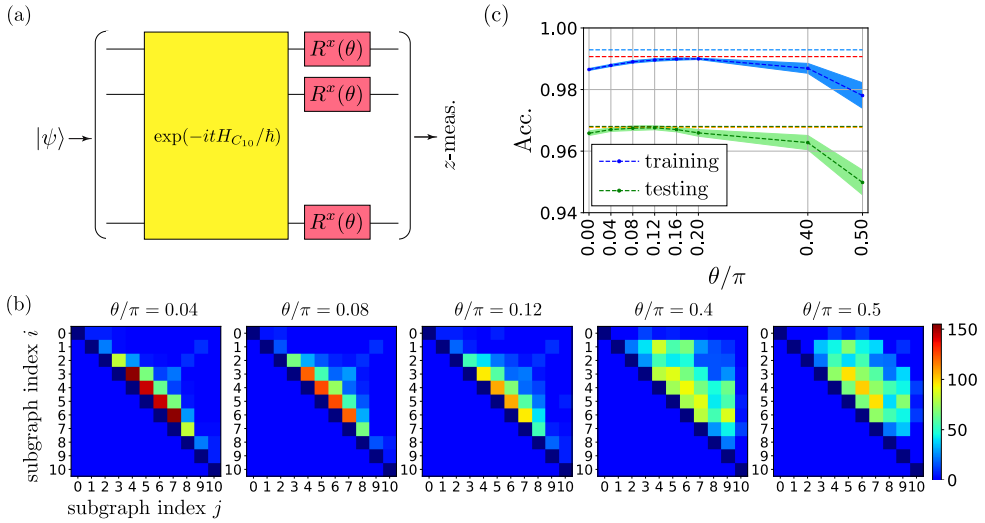

Figure 6.9: (a) Schematic illustration of the application of the Pauli-x rotation gates to each qubit after the XY Hamiltonian dynamics on the graph C_{10}. The angles θ for the rotation all are identical. (b) Heat map matrices whose entry (i, j) representing h_{ij} for different θ/π. Those values of h_{ij} are averaged over $Jt/2\pi \in [0.75, 1.0]$. (c) The behavior of the accuracy rate in training and testing against $\theta \in [0, \pi/2]$. The blue and green dashed curves depict the accuracy rate in training and testing, respectively, averaged over $Jt/2\pi \in [0.75, 1.0]$. The shaded areas represent the corresponding standard deviation. The four horizontal dashed lines represent the accuracy rate in training and testing of $U^{(\mathrm{BD})}_{\mathrm{Haar}}$ and the case with partition $[128 \times 8]$. The orange and red ones depict the accuracy rate in training and testing of $U^{(\mathrm{BD})}_{\mathrm{Haar}}$, and the blue and green show the accuracy rate in training and testing of $[128 \times 8]$.

performance to the ideal block structure.

First, we observe how the initial independent blocks given by $\Gamma^{(k)}$ of Eq. 6.2 merge as θ increases. To do so, we need to extract a graph in the Hilbert space from the unitary matrix $U_{\mathrm{tot}}(\theta, t) = \mathcal{R}_x(\theta) U_{C_N}(t)$ where $\mathcal{R}_x(\theta)$ is given by the tensor product of the x-rotation acting on each qubit, that is, $R_x(\theta) \otimes \cdots \otimes R_x(\theta)$. We then apply the percolation rule [90] to the effective Hamiltonian for $U_{\mathrm{tot}}(\theta, t)$, that is, $H_{\mathrm{eff}} = i\hbar\ \mathrm{Log}(U_{\mathrm{tot}}(\theta, t))/t$ and obtain the unweighted adjacency matrix of a graph $\mathcal{G}$ in the Hilbert space, whose vertices are given by the computational basis. For $\theta = 0$, the subgraphs of the graph, $\mathcal{G}^{(k)}$, whose vertices are given by $\Gamma^{(k)}$, are disconnected to each other since the total unitary matrix $U_{\mathrm{tot}}(0, t)$ can be block-diagonalized and

each block are represented in the basis $\Gamma^{(k)}$. On the other hand, for $\theta \neq 0$, the subgraphs $\mathcal{G}^{(k)}$ could be connected.

To measure the connection between the subgraphs for a given θ, we introduce the following quantity for a pair of the subgraphs $(\mathcal{G}^{(i)}, \mathcal{G}^{(j)})$ with $i \neq j$:

$$h_{ij} = \frac{|\{(a,b) \in E(\mathcal{G}) | a \in V(\mathcal{G}^{(i)}), b \in V(\mathcal{G}^{(j)})\}|}{\min(|V(\mathcal{G}^{(i)})|, |V(\mathcal{G}^{(j)})|)}, \tag{6.12}$$

where $V(\cdot)$ and $E(\cdot)$ denote the set of edges and vertices in a given graph, respectively. This quantity may remind one the Cheeger constant known as one of the important measures for expander graphs [128]. Similarly to the Cheeger constant, our quantity h_{ij} can measure how strongly coupled the pair of the subgraphs $(\mathcal{G}^{(i)}, \mathcal{G}^{(j)})$ is since its numerator is nothing but the number of edges between those subgraphs.

In Fig. 6.9(b), we show the quantity h_{ij} as a heat map matrix for different values of θ. Each value of the quantity h_{ij} is averaged over $Jt/2\pi \in [0.75, 1.0]$. The color of each entry (i, j) of the matrix in Fig. 6.9(b) represents the value of h_{ij}. For $\theta/\pi = 0.4$ and 0.5, one can find that the coupling amplitudes between each pair of the subgraphs is almost evenly spreading. This implies that a given basis state on $\mathcal{G}^{(i)}$ randomly can move to another subgraph $\mathcal{G}^{(j)}$, $(j \neq i)$.

In contrast, for smaller values of θ/π, $\theta/\pi = 0.04, 0.08$, and 0.12, a few pairs of the subgraphs are strongly coupled and the others are weakly coupled. A chain structure of the subgraphs $\mathcal{G}^{(k)}(k = 3, 4, 5, 6, 7)$ emerges since $h_{3,4}$, $h_{4,5}$, $h_{5,6}$ and $h_{6,7}$ are significanlty larger than others. This structure allows basis states to move over particular subgraphs, but not evenly over all the subgraphs. These results also suggest the transition of the graph structure mentioned above.

Now let us explore the behavior of the accuracy rates in training and testing against rotation gate angle θ, which are depicted by the blue and green curves respectively in Fig. 6.9(c). The accuracy rates are averaged over $t/2\pi \in [0.75, 1.0]$, and the shaded areas represent the corresponding standard deviation. The accuracy rates in both training and testing improves as the angle θ increases and start going down at a certain θ, while the peaks are slightly shifted for training and testing. In testing, the accuracy rate is the highest around $\theta/\pi \approx 0.12$, while the training accuracy rate is still increasing and not the highest. Thus, the generalization performance is the best around $\theta/\pi \approx 0.12$. In Fig. 6.9(b), four horizontal dashed lines are depicted for comparison to the accuracy rates of $U^{(\mathrm{BD})}_{\mathrm{Haar}}$ and the case with partition $[128 \times 8]$. For $\theta/\pi = 0.04, 0.08$, and 0.12, the XY dynamics on C_{10} with the rotation gates achieves a comparable testing accuracy rate to that

of $U_{\mathrm{Haar}}^{(\mathrm{BD})}$ and $[128 \times 8]$. The training accuracy rate is smaller than that of $U_{\mathrm{Haar}}^{(\mathrm{BD})}$ and the partition case with $[128 \times 8]$. This result suggests that it is possible to achieve a higher testing accuracy with a good generalization with the XY dynamics on a feasible interaction network by applying the Pauli-x rotation with a small certain angle after the XY dynamics. Moreover, it could be implied that such a graph structure induced by the symmetries can be exploited for higher generalization performance of the QERC.

6.5 Summary

In this work, we have explored the performance of quantum extreme reservoir computation in relation to the XY Hamiltonian dynamics for various interaction networks over qubits. In contrast to other physical systems we have investigated in the previous chapters, the nearest-neighbor connectivity (Path or cycle graphs) in the XY model allows for sufficiently nontrivial dynamics, enabling QERC to achieve a high testing accuracy rate comparable to the Haar-random case. For instance, in the XX model, the nearest neighbor connectivity induces periodic dynamics, yielding a limited accuracy rate. On the other hand, the XY model with the nearest neighbor connectivity does not have the periodicity, and it can be analyzed by utilizing spectral graph theory. This suggests that intrinsic connectivity in several quantum platforms, such as superconducting qubits, can be leveraged to perform this task without advanced engineering to generate complex connectivity. We further demonstrated that the block structure in Hamiltonians, which is associated with symmetries, contributes to generalization performance.

Generalization performance of ML, in general, can be improved by introducing a constraint or a regularizer, which simplifies the ML model. As symmetries simplify quantum dynamics in the sense that the whole Hilbert space is divided into mutually exclusive subspaces, they might behave as regularizers in ML, suppressing overfitting in QERC. Therefore, quantifying the complexity of the quantum reservoir as a machine learning model is expected to capture the connection between complex quantum dynamics and their machine learning function, leading to an understanding of what engineering is essential. In the next chapter, we then aim to formulate and illustrate this idea with an information-theoretic measure.

Chapter 7

The quantum reservoir complexity by data compression

In this chapter, we characterize quantum systems motivated by an information-theoretic framework for quantifying model complexity in classical machine learning. A machine learning model that is too complex, compared with a given dataset, may suffer from overfitting, which can be mitigated by simplifying the model with a regularizer. Frameworks for quantifying this phenomenon are of essential interest in classical machine learning [129–132]. As we observed a similar phenomenon in the previous chapter, these frameworks should be useful for understanding what quantum properties and engineering can appropriately contribute to quantum reservoir complexity.

A framework for quantifying the balance of fitting quality and complexity of machine learning models is the minimum description length (MDL) principle [131, 133, 134]. It is closely related to coding theory and data compression, providing a framework for quantifying model complexity by compression [135]: the model complexity is defined by the number of bits required to represent the model. It has been numerically and theoretically demonstrated that the model complexity captures generalization behavior in classical machine learning [136–138].

In this chapter, therefore, we compress unitary matrices to quantify the reservoir complexity by compression within QERC. We will explore how QERC performance is captured by the complexity. We will also thoroughly consider the methodology for compressing classically represented quantum objects to properly quantify the compression complexity of quantum objects.

The remainder of this chapter is organized as follows: In Sec. 7.1, the MDL framework and our approach will be explained. In Sec. 7.2, the quan-

tum reservoir unitaries of interest will be presented. Our compression methods will also be introduced. We will then show the compression results in Sec. 7.3. In Sec. 7.4, we will discuss developing our compression methods using a different representation of quantum objects. In Sec. 7.5, we will then discuss that the improved compression method could capture not only ML behaviors, but also fundamental quantum properties. Lastly, we will summarize our results and discuss the operational interpretation of this compression complexity, showing the potential of the complexity to illustrate appropriate engineering for the quantum reservoir.

7.1 Compression complexity of machine learning models

We first explain our MDL framework. Starting from the original paper by Rissanen [133], there are several alternatives to the original MDL [134], but we focus on the following setting introduced by Vitányi and Li in 2000 [135], called the *ideal* MDL principle.

Consider a training dataset $\mathcal{D} = \{(x^{(n)}, y^{(n)})\}_{n=1}^{D}$ and a class $\mathcal{M}$ of statistical models $M_\gamma \in \mathcal{M}$, $\gamma \in \Gamma$, where Γ is a set of model indeces equipped with a probability distribution $\mathfrak{p}(\gamma)$ over Γ. The quality of fitting the dataset is quantified with the negative log-likelihood,

$$L(Y|\gamma, X) = -\sum_n \log_2 M_\gamma(y = y^{(n)}|x^{(n)}), \tag{7.1}$$

where X and Y are ordered lists of inputs and labels, $X = (x^{(1)}, x^{(2)}, \cdots, x^{(n)})$ and $Y = (y^{(1)}, y^{(2)}, \cdots, y^{(n)})$. Our interest is to select the best model, and thus, we minimize the negative log-likelihood $L(\gamma|Y, X)$ of M_γ given the dataset based on Bayes' rule:

$$\begin{aligned} \min_{\gamma\in\Gamma} L(\gamma|Y, X) &\stackrel{+}{=} \min_{\gamma\in\Gamma} [L(\gamma) + L(Y|\gamma, X)] &(7.2)\\ &= \min_{\gamma\in\Gamma} \left(\log_2 \frac{1}{\mathfrak{p}(\gamma)} + \sum_n \log_2 \frac{1}{M_\gamma(y^{(n)}|x^{(n)})} \right). &(7.3) \end{aligned}$$

where $\stackrel{+}{=}$ indicates euality up to an additive constant.

This objective can be interpreted in the context of data compression as those negative log-likelihoods correspond to the codeword lengths of, for example, the Shannon-Fano code [135] or the arithmetic code [137]. Vitányi and Li then consider the (prefix-free) Kolmogorov complexity, known as the

theoretical and ideal lower bound of data compression, which is defined as the length of the shortest program to output the bitstring of interest (see [139]). After all, the ideal MDL objective is given by [135]:

$$K(M) + K(Y|M, X), \tag{7.4}$$

where $K(\cdot)$ and $K(\cdot|\cdot)$ are the prefix-free Kolmogorov complexity and the conditional one. However, the Kolmogorov complexity is not computable [139], but it can be estimated from above using data compression, which may require very efficient compression methods. Several demonstrations of compression of machine learning models have been conducted to quantify their complexity [136–138]. For instance, if we have a neural network with some parameter values on a fixed network structure, the compression complexity can be estimated from above by efficiently compressing the parameter values. Note that the compression has to be conducted with a self-delimiting code [135,137,139]. The second term $K(Y|M_\gamma, X)$ is bounded from above by the negative log-likelihood $L(Y|\gamma, X)$ [138], thereby enabling the evaluation of the ideal MDL objective from above.

7.1.1 Compression complexity in QERC

We evaluate the compression complexity of QERC, C_{QERC} ($\overset{+}{\geq} K(\mathrm{QERC})$), inspired by the ideal MDL principle. The reason why we are not going to directly consider the (nonideal) MDL principle is that, in our case, the distribution $\mathfrak{p}(\gamma)$ of models is not trivial, or even defining it may not be justified.

As the main interest is in the impact of various quantum reservoirs on the learning behavior, we consider compressing the quantum reservoir part while ignoring PCA and ONN. This may be justified as follows. The PCA part can be considered a prior knowledge [137] independent of the choice of the quantum reservoir. The parameters in the ONN can be encoded in bits by keeping a fixed number of digits in their binary representation, yielding a fixed constant.

We will evaluate the reservoir complexity by compression. However, we need to note that we will not directly carry out the MDL minimization in QERC; instead, we will provide a proof-of-concept demonstration and discuss how to further develop this framework within the quantum reservoir model.

7.2 Quantum reservoir models and our compression methods

(a)

Unitary maps	brief descriptions
ZZX	$U_{ZZX}(\theta^x) = \exp\left(-i\theta^x \sum_l \sigma_l^x\right) \exp\left(-i \sum_{j<k} \theta_{jk}^z \sigma_j^z \sigma_k^z\right)$
Haar	Sampled from the unitary group with the Haar measure.
Cauchy	Defined using Cauchy-random variables.
Random Clifford	Uniformly sampled from the Clifford group.
Clifford+T	$\left(\bigotimes_j H_j T_j H_j\right) V$ H_j, T_j : Hadamard and T gates acting on the j-th qubit. V : Random Clifford
Random single-qubit rotation (RSR)	Tensor product of Haar-random SU(2).

(b)

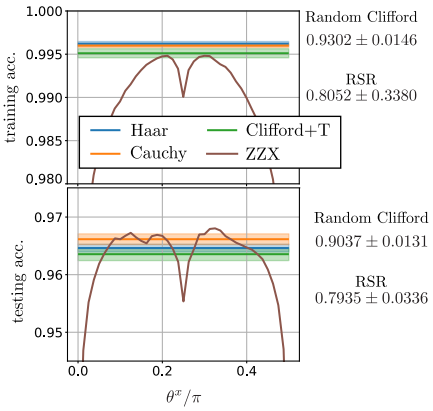

Figure 7.1: (a) The list of the quantum reservoir models of interest, and (b) QERC performance with them. The numerical simulations were carried out under the following settings: ten qubits, the MNIST dataset [109], and mini-batch learning with a batch size of 100, using the AdaGrad optimizer [116]. For the random reservoirs, QERC was performed with 20 different realizations of the reservoir. The solid horizontal lines are the average with the associated standard deviation.

We employ several quantum reservoir models summarized in Fig. 7.1(a), which have been considered in the previous chapters. The training and testing accuracy rates of those models are also summarized in Fig. 7.1(b). Quantifying the complexity of these models is interesting because they possess distinct quantum properties and exhibit different behaviors in generalization performance. The Haar-random model tends to overfit compared with the Cuahcy-random model. The ZZX model exhibits a transition between underfitting and overfitting while varying θ^x. The random Clifford, RSR, and Clifford+T models show the necessity of both entanglement and non-stabilizerness.

7.2.1 Compression methods for quantum reservoirs

Now, we consider two compression methods for unitary matrices represented in the computational basis. In both of those methods, we compress the real and imaginary parts of the matrix entries separately, as a unitary matrix is

complex-valued. The first method is a naive one, where the first τ digits in the binary representation of the modulus of each entry are stored. The sign is specified with 1 bit separately, yielding a self-delimiting codeword of length $(3 + 2\tau) \times 2$ for the real and imaginary parts of a matrix entry. We refer to this method as Method 1.

For the second method, which is expected to be more efficient, we employ arithmetic coding [140] as used in [137, 138] in the context of compressing classical machine learning. Arithmetic coding uses the frequency distribution of the entry values in a unitary matrix to define codewords of a proper length. The distribution $f^{(u)}$ for the real parts ($u = \mathrm{r}$) and the imaginary parts ($u = \mathrm{i}$) is obtained as follows: (i) set the number of bins, M, (ii) divide the range $[-1, 1]$ equally into M segments, (iii) count the number of entries, $f_m^{(u)}$, on the m-th segment. Once the distributions $f^{(\mathrm{r})}$ and $f^{(\mathrm{i})}$ are calculated, the matrix is then represented [140] by a bitstring whose length is less than

$$\left(\sum_{u=\mathrm{r,i}} \sum_{m=1}^{M} f_m^{(u)} \log_2 \frac{d}{f_m^{(u)}} \right) + 4 + \sum_{u=\mathrm{r,i}} \sum_{m=1}^{M} \log_2 \frac{d}{f_m^{(u)}} + 2 \log_2 M + 1, \qquad (7.5)$$

where $d = 2^N$. The first two terms come from arithmetic coding, the second and third terms are for storing codewords, and the other terms represent the number of bits required to reproduce the bins. We refer to this method as Method 2.

The parameters τ and M determine the compression precision in Methods 1 and 2. We choose these parameters to ensure that the inprecision is lower than a fixed threshold ϵ (> 0). Here, the inprecision is calculated with the Frobenius norm, $(1/d)\|U - \tilde{U}\|_{\mathrm{F}} \leq \epsilon$, where U is the unitary matrix to be compressed and $\tilde{U}$ is the reconstructed matrix from the compressed information. We set $\epsilon = 0.01$.

7.3 Compression of quantum reservoirs

We now apply our compression methods to the quantum reservoirs listed in Fig. 7.1. In Fig. 7.2, the compression results with Methods 1 and 2 are compared. Looking at the scale of the horizontal axis, Method 2 is more efficient. The compression results for many quantum reservoirs using Method 2 capture the performance behavior shown in Fig. 7.1. For instance, the observation that the Haar-random model is overfitting compared with the Cauchy-random model in Fig. 7.1 can be explained as follows. The Haar model has a higher complexity according to Method 2, while its training accuracy rate is comparable to that of the Cauchy model. Thus, the MDL principle suggests

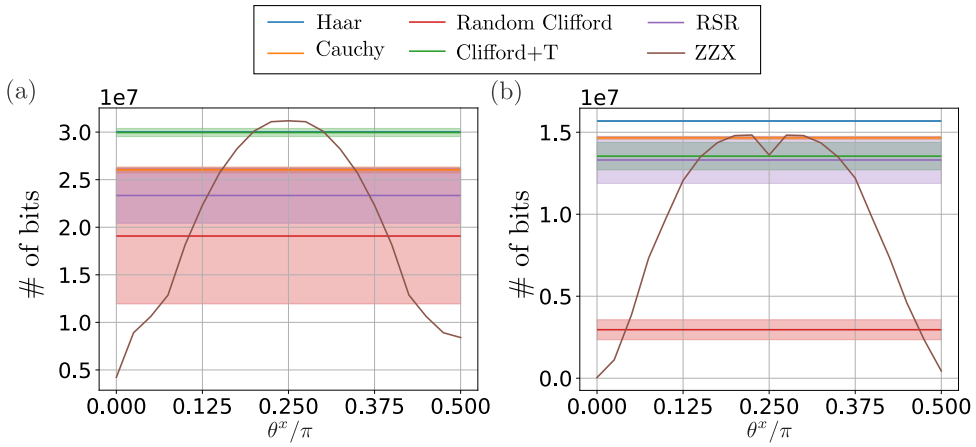

Figure 7.2: The compression results using (a) Method 1 and (b) Method 2. The vertical axis is the number of bits in the compressed information. The horizontal axis is the parameter θ^x in the ZZX model. For random models, the compression results are averaged over 100 different realizations, which are represented by the horizontal lines with the associated standard deviations. $\epsilon = 0.01$.

that the Cauchy model has better generalization, which matches what we observe in Fig. 7.1.

The difference in the compression complexity between the random Clifford and Clifford+T models is also reflected in their QERC performance. The random Clifford model has a very low complexity, suggesting a poor training accuracy rate. As the layer of T gates in the Clifford+T model causes a higher complexity than that of the random Clifford model, we observe an improvement in the training accuracy rate in the Clifford+T model. The Clifford+T model is more complex than the random Clifford model, but less than the Cauchy model, yielding a slightly lower training accuracy rate.

The learning behavior of the ZZX model against θ^x is also predicted by the complexity curve with Method 2, whereas Method 1 does not reflect the dip of the accuracy at $\theta^x = \pi/4$. Hence, by efficiently compressing those quantum reservoirs, the QERC performance can be captured, except for the RSR model.

In Fig. 7.2(b), the RSR model is as complex as the Clifford+T model. However, this does not match the learning behavior of the RSR model, implying that the complexity of the RSR model is overestimated. Thus, a more efficient compression method may be desired. We discuss this further in the following section.

7.4 The compression complexity and representation of quantum objects

It was observed that the complexity of the RSR model was overestimated. It may be because our compression is not sufficiently efficient, as compression just estimates the ideal MDL objective from above. The RSR model can be ideally compressed more efficiently than Method 2, since it has a tensor product structure, and each U(2) unitary can be compressed and stored individually. However, it cannot be carried out without any prior knowledge of the structure, which caused the overestimation in Method 2. It implies that Method 2 does not properly capture the complexity of the quantum reservoir. This may suggest the need to explore more efficient compression algorithms; however, it must be noted that the classical representation of unitary maps does not necessarily reflect their actual complexity.

We have been interested in compressing unitary matrices, but to simply illustrate this problem, let us consider compressing two physically *distinct* quantum states:

$$|\Phi_{AB}\rangle = \frac{1}{\sqrt{2}}(|0_A\rangle\,|0_B\rangle + |1_A\rangle\,|1_B\rangle), \quad |\Psi_C\rangle = \frac{1}{\sqrt{2}}(|0_C\rangle + |3_C\rangle). \tag{7.6}$$

$|\Phi\rangle_{AB}$ is defined over a system composed of two-level subsystems A and B associated with two-dimensional Hilbert spaces $\mathcal{H}_A$ and $\mathcal{H}_B$. On the other hand, the system C over which $|\Psi_C\rangle$ is defined is a four-level system associated with a four-dimensional Hilbert space $\mathcal{H}_C$. $|\Phi\rangle_{AB}$ has entanglement, and the subsystems A and B are entangled, while entanglement is not well-defined in the system C, as it is not necessarily a composite system.

Those states may have the identical vector representation, as it mathematically holds that $\mathcal{H}_C \cong \mathcal{H}_A \otimes \mathcal{H}_B$:

$$|\Phi_{AB}\rangle \overset{\mathcal{B}_{AB}}{=} \frac{1}{\sqrt{2}}(1,0,0,1)^\mathsf{T}, \quad |\Psi_C\rangle \overset{\mathcal{B}_C}{=} \frac{1}{\sqrt{2}}(1,0,0,1)^\mathsf{T}, \tag{7.7}$$

$$\mathcal{B}_{AB} = \{|0_A\rangle \otimes |0_B\rangle, |0_A\rangle \otimes |1_B\rangle, |1_A\rangle \otimes |0_B\rangle, |1_A\rangle \otimes |1_B\rangle\}, \tag{7.8}$$

$$\mathcal{B}_C = \{|0_C\rangle, |1_C\rangle, |2_C\rangle, |3_C\rangle\}, \tag{7.9}$$

where $\mathcal{B}_{AB}$ and $\mathcal{B}_C$ are the basis set for systems AB and C, and $\overset{\mathcal{B}_{AB/C}}{=}$ indecates the vector representaiton with the basis set $\mathcal{B}_{AB/C}$. Thus, any compression algorithm (even if it is ideal) returns the same complexity for both $|\Phi_{AB}\rangle$ and $|\Psi_C\rangle$, despite the fact that they are *physically distinct* as $|\Phi_{AB}\rangle$ has entanglement and $|\Psi_C\rangle$ does not. Other choices of basis sets can be made, and

the vector representations may differ. However, this does not fundamentally solve this problem since the information that the system AB is composite is not encoded in the representation anyway. Thus, it is essential to consider classical representations of quantum objects that, at least, encode system information.

7.4.1 Matrix product form

Let us now introduce the matrix product form of quantum states. Consider an N-qubit quantum state $|\psi\rangle \in \mathcal{H}_d$ in a d-dimensional Hilbert space with d being 2^N. A baisis set $\mathcal{B}$ for $\mathcal{H}_d$ is defined as $\mathcal{B} = \{|i\rangle\}_{i=0}^{d-1}$ where $|i\rangle = |i_0 i_1 \cdots i_{N-1}\rangle$ is a tensor product of the computational states $|0_j\rangle, |1_j\rangle$ of individual qubits, $j = 0, 1, \cdots, N-1$. Then, the quantum state $|\psi\rangle$ is now spanned with $\mathcal{B}$,

$$|\psi\rangle = \sum_{i=0}^{d-1} \langle i_0 \cdots i_{N-1}|\psi\rangle \, |i_0 \cdots i_{N-1}\rangle \, . \tag{7.10}$$

The matrix product form expresses those complex-valued coefficients $\langle i_0 \cdots i_{N-1}|\psi\rangle$ as a matrix product [141],

$$\begin{aligned} \langle i_0 \cdots i_{N-1}|\psi\rangle &= A^{i_0} A^{i_1} \cdots A^{i_{N-1}} \\ &= \sum_{\alpha_0, \cdots, \alpha_{N-2}} \left(A^{i_0}\right)_{\alpha_0} \left(A^{i_1}\right)_{\alpha_0 \alpha_1} \cdots \left(A^{i_{N-2}}\right)_{\alpha_{N-3} \alpha_{N-2}} \left(A^{i_{N-1}}\right)_{\alpha_{N-2}}, \end{aligned} \tag{7.11}$$

where A^{i_0} and $A^{i_{N-1}}$ are complex-valued vectors , and others A^{i_j} ($j = 1, \cdots, N-2$) are rectanguler matrices satisfying that $\sum_{i_j} (A^{i_j})^\dagger A^{i_j} = I$. This form explicitly encodes the information that the system is composed of N qubits through the following decomposition,

$$|\Phi_{AB}\rangle = \sum_{\alpha_0, \cdots, \alpha_{N-1}} \left(\sum_{i_0=0,1} \left(A^{i_0}\right)_{\alpha_0} |i_0\rangle \right) \left(\sum_{i_1=0,1} \left(A^{i_1}\right)_{\alpha_0 \alpha_1} |i_1\rangle \right) \tag{7.12}$$

$$\cdots \left(\sum_{i_{N-1}=0,1} \left(A^{i_{N-1}}\right)_{\alpha_{N-1}} |i_{N-1}\rangle \right). \tag{7.13}$$

The range of the matrix indices α_j determines the maximum amount of entanglement for the bipartition between the j-th and $(j+1)$-th qubits. One observes that each A^{i_j} expresses the local complexity, which may be increased by local operations such as H and T gates. Thus, by entanglement,

the number of entries to be compressed increases, and by T gates, it may become harder to compress those entries efficiently.

The matrix product form can be applied to quantum states of a system that is not a composite system with dimensionality 2^m for some $m \in \mathbb{N}$, but it is not preferable since the form yields more components to be stored compared to the vector representation. However, for a composite system, its quantum system should be expressed with this form to explicitly encode the system information, resulting in a proper characterization of the complexity in the system.

7.4.2 The complexity of matrix product states

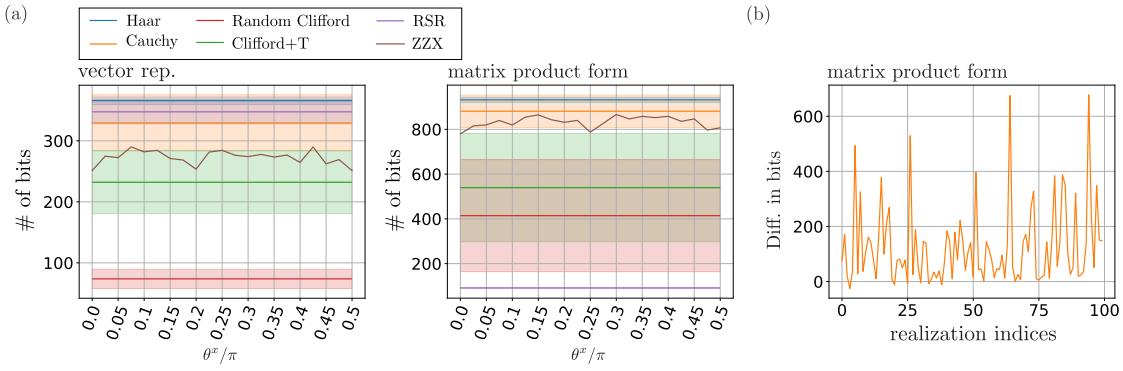

Figure 7.3: (a) Comparison of the compression between the vector representation and the matrix product form. The compression is carried out within inprecision $\epsilon = 0.01$, where it is estimated by the infidelity $\sqrt{1 - |\langle\psi|\tilde{\psi}\rangle|^2}$ between the quantum state $|\psi\rangle$ to be compressed and the reconstructed state $|\tilde{\psi}\rangle$. (b) The difference in the compression complexity between the random Clifford model and the Clifford+T model. For each realization of the random Clifford model, we obtain the Clifford+T model by applying the layer of T gates after the realization.

We now apply this form to quantum states obtained from the quantum reservoirs under consideration. We compress the vector representation and matrix product form of $U|0\rangle$ for the random models U and $U_{ZZX}|+\rangle$ for the ZZX model since $U_{ZZX}|0\rangle$ is trivial. Values to be compressed are now coefficients of the vector representaiton or entries of matrices A^{i_j}, and those values are compressed with arithmetic coding. In Fig. 7.3(a), the comparison is shown for five qubits. For the vector representation, the complexity of the RSR model is overestimated, while it is the lowest in the matrix product

form. The hierarchy of the complexities of the other models is essentially the same as what we observed with Method 2 in Fig. 7.2.

In Fig. 7.3(a), the complexity in the matrix product form of the random Clifford and Clifford models has a large standard deviation with respect to realization, and the ranges of the deviation overlap. In Fig. 7.3(b), we show how much complexity is increased by the layer of T gates in each realization for the matrix product form. While the deviation is large, the complexity increases for almost all realizations. In this sense, the Clifford+T model tends to be more complex than the random Clifford model.

The hierarchy and trend of the complexities of these models are similar to those of the accuracy rate, but these results are not directly connected. The compression results of Fig. 7.3 are obtained for quantum states, not for unitary matrices. Thus, it is necessary to further develop this approach.

7.5 Fundamental aspects of the compression complexity

In this section, we focus on more fundamental properties of quantum states that the compression complexity could capture. What the compression complexity in the matrix product form can potentially describe is the competition between entanglement and non-stabilizerness. Let us now examine three-qubit states, which have a rich structure of entanglement, by compressing their matrix product form.

Entanglement in a three-qubit system has been classified into various types [142]. For instance, let us consider the following parametrization of a three-qubit pure state,

$$|\psi(\theta_0,\theta_1)\rangle = \cos\theta_0\,|u_0v_0w_0\rangle + \sin\theta_0\,|u_1\rangle \otimes (\cos\theta_1\,|v_0w_0\rangle + \sin\theta_1\,|v_1w_1\rangle), \tag{7.14}$$

where $0 \leq \theta_0, \theta_1 \leq 2\pi$, and $|u_i\rangle$, $|v_i\rangle$, $|w_i\rangle$ are local basis states of each qubit. These basis states are set as $|u_0\rangle = |v_0\rangle = |w_0\rangle \doteq (1,0)^{\mathsf{T}}$ and $|u_1\rangle = |v_1\rangle = |w_1\rangle \doteq (0,1)^{\mathsf{T}}$. The quantum state falls into four entanglement types as shown in Fig. 7.4(a), and those types of entanglement are determined [142] as follows:

Type 0: Fully separable. No entanglement.

Type 1: The second and third qubits are entangled, but none of those qubits is entangled with the first one.

Type 2-0: The bipartite entanglement of any pair of the three qubits is zero, while the tripartite entanglement is nonzero.

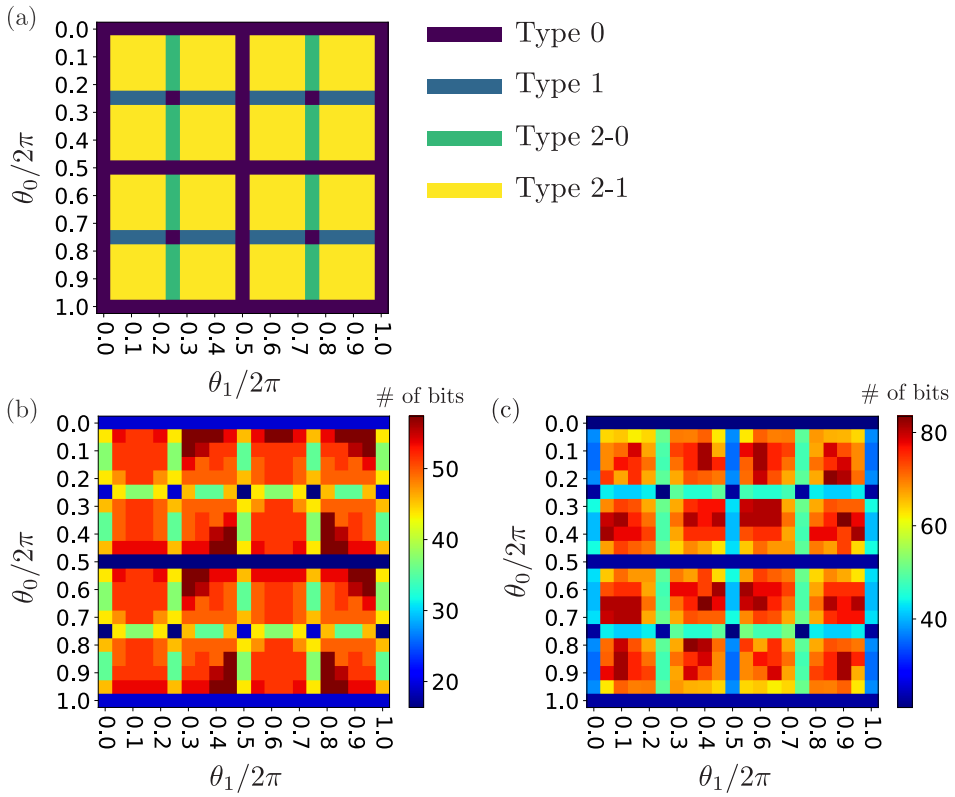

Figure 7.4: (a) A classification of three-qubit entanglement of Eq. 7.14 according to [142]. The compression complexity of the parametrized state in (b) the vector representation and (c) the matrix product form. The compression is carried out with $\epsilon = 10^{-3}$.

Type 2-1: The tripartite entanglement is nonzero, and the bipartite entanglement of the second and third qubits is nonzero, while the other bipartite entanglements are zero.

In Fig. 7.4(b) and (c), the compression complexity of Eq. 7.14 in the vector representation and the matrix product form is shown as a color matrix for different parameters θ_0, θ_1. The checkerboard structure is illustrated in both Fig. 7.4(b) and (c), but (c) seems richer. For instance, the states on the lines at $\theta_1 = 0$ and $\pi/2$ are distinguished in the classification (a), but in the vector representation (b), they have almost the same complexity. On the other hand, in the matrix product form (c), those states are distinguishable.

An interesting observation is that the quantum state of Eq. 7.14 with $\theta_1 = 0$ has a slightly higher complexity than one with $\theta_0 = 0$, although these are in the same type. This is because the state with $0 < \theta_0 < \pi/2$ and $\theta_1 = 0$ is fully

separable, and the first-qubit state is the superposition $\cos\theta_0 |u_0\rangle + \sin\theta_0 |u_1\rangle$. Thus, the local complexity increases the total compression complexity, while it is lower than that of entangled states, for example, those with $\theta_0 = \pi/2$ (Type 1). In this sense, entanglement dominates non-stabilizerness in the compression complexity of Type 1.

7.6 Summary and Discussion

In this chapter, we characterized unitary matrices and quantum states by compression, inspired by the ideal MDL principle, where the complexity of machine learning models is quantified by data compression. Compressing unitary matrices with Method 2 illustrated the complexity of various quantum reservoirs. The trend in QERC performance with those reservoirs is explained by the compression complexity, but the performance with the RSR reservoir was not captured reasonably.

Method 2 does not take system information into account as it compresses unitary matrices represented in the computational basis states. Compressing such classical information associated with a quantum object does not necessarily capture physical differences, as discussed with Eq. 7.7, motivating us to employ the matrix product form.

This approach may be conceptually viewed in the context of Kolmogorov complexity as follows. The Kolmogorov complexity of a bitstring can be understood as a sort of difficulty in an operation to output the bitstring in a universal Turing machine (UTM). Hence, compressing a classical representation of a quantum object provides an upper bound on the operational complexity in a *classical* UTM. Thus, the operational complexity to produce the quantum object in a *quantum* computer is not necessarily captured, suggesting that various constraints of operations in a quantum computer should be taken into account.

A possible approach to capturing the quantum operational complexity is to extend the Kolmogorov complexity to a quantum analog using quantum Turing machines (QTMs), as they explicitly allow for quantum operations. Several definitions of the quantum Kolmogorov complexity have been proposed [143–148], but a consensus has yet to be reached. Some definitions do not explicitly require QTMs [143, 145, 147], but they have successfully reproduced quantum characteristics such as the no-cloning theorem [143, 145] and entanglement [146–148]. The key idea common in those literatures seems to introduce appropriate constraints in UTMs.

This strongly suggests representing quantum objects as classical information with appropriate knowledge of the quantum system of interest and

compressing it. For instance, if the quantum system under consideration is a composite one, the operational complexity in the system cannot be accurately quantified, given no information on the system. In this sense, the matrix product form explicitly encodes the information that the quantum system under consideration is composite, thereby differentiating the operational complexity of local and nonlocal operations.

However, our approach should be further developed to more certainly characterize quantum reservoirs and capture their ML behavior. A remaining challenge is to examine the objective of the ideal MDL principle (Eq. 7.4). While the trend of the QERC performance can be explained with the current compression method in the matrix product formalism, the compression complexity is too high and dominates the cross-entropy loss, which is an upper bound of $K(Y|M,X)/D$, preventing us from the direct evaluation of the MDL objective.

A possible approach to this challenge is to compress crucial aspects of unitary matrices as a feature map, rather than compressing the unitary matrices themselves. The unitary matrix fully determines the relation between any input quantum states and the corresponding final states. However, the role of the quantum reservoir is restricted in QERC, as we have a fixed encoding and measurement part. Thus, it is essential to compress the quantum reservoir as a feature map by, for example, directly estimating the conditional complexity of the final state given the information of the initial state.

Another future direction is to explain the QERC behavior within the PAC-Bayes framework [130, 149], as its connection ot the Kolmogorov complexity has been discussed [137, 138]. This direction should provide a more ML-theoretic understanding of QERC, bridging the quantum information theoretic properties of quantum dynamics and their ML behaviors.

Chapter 8

Conclusion

In this thesis, I have investigated the computational capability of natural quantum dynamics in QERC to address the open questions raised in the Introduction: (i) how well natural quantum dynamics perform nontrivial ML tasks compared to engineered systems, and (ii) what kind of engineering in quantum systems is essential and appropriate for those tasks.

Through Chapters 4-7, it was shown that the extensive engineering required for random unitaries should not be necessary for performing mid-scale classification tasks of images within the QERC framework, as more natural quantum dynamics possess a comparable learning performance. Especially, in Chapter 5, the ZZ-Ising model with long-range interactions was shown to achieve high classification performance, suggesting that the intrinsic quantum dynamics in several quantum platforms, such as trapped ions and neutral atoms, are complex enough to solve these tasks. In Chapter 6, the XY model with nearest-neighbor interactions in a 1D chain also achieved decent performance. This also suggests the non-necessity of extensive engineering in different quantum platforms, such as superconducting qubits. These results strengthen the effectiveness of the quantum reservoir model as an alternative form of quantum information processing for ML.

Regarding (ii), understanding what properties of the quantum reservoir are beneficial for learning performance was not trivial, as several complexity measures standard in the quantum information field, such as entanglement, non-stabilizerness, and unitary designs, did not correlate clearly on their own. In this sense, engineering quantum systems to acquire those properties does not necessarily improve learning performance efficiently, suggesting that a different axis of complexity is essential. Thus, I explored complexity of quantum reservoirs in different aspects: the weight distribution in Chapter 4, the participation ratio in Chapter 5, spectral properties and symmetries in Chapter 6, and the compression complexity in Chapter 7.

In Chapters 4 and 6, various insights on the beneficial properties of the quantum reservoir were obtained. It was observed in Chapter 4 that the width of the weight distribution of the DTC model grows with the number of periods, and converges to the same width as that of the Haar-random model. This behavior of the complexity matched the growth and saturation of the accuracy rate. In Chapter 5, the participation ratio (PR) was extended based on an ML perspective. While the PR is widely used in condensed matter physics, the extended ones capture ML properties of the probability vector obtained from the quantum reservoir in classification tasks. In Chapter 6, it was found that the periodicity of the unitary operator induced by its spectral property prevents us from achieving decent learning performance in QERC. It was also suggested that certain symmetries behave as the regularizer for ML models. Although these results suggest a specific engineering approach that can enhance learning performance, they do not elucidate the underlying mechanism behind these observations.

For a deeper understanding, in Chapter 7, I quantified the complexity of the quantum reservoir using data compression, inspired by the ideal MDL principle and the Kolmogorov complexity. Although the objective function of the principle was not directly evaluated, the compression complexity of quantum reservoirs explained the trend in QERC performance. A key consideration in this demonstration was the classical representation of quantum objects. To properly evaluate the compression complexity of quantum objects, the necessity of including system information was discussed with a simple physical example. It was also noted that this consideration was also justified by the literature on the quantum Kolmogorov complexity. Discussing the operational interpretation of the compression complexity tied the ML theoretical concept to the corresponding quantum operations. Although there is still room for improvement, this approach has the potential to pave the definitive way for further exploring alternative forms of quantum information processing for ML.

Along with this direction, the following future works may be suggested. The first one is to develop a compression method that enables the direct evaluation of the MDL objective. For example, it may be crucial to estimate the conditional complexity of the final quantum state under quantum dynamics, given the initial state, to compress quantum dynamics as a feature map. Another future work is to apply the PAC-Bayes framework, as it has been tied to the Kolmogorov complexity in classical ML theory. The framework should provide a more theoretical ground for understanding the impacts of the quantum reservoir's properties on learning.

It is also worth exploring the compression complexity for a more fundamental contribution to quantum information theory, independent of QML.

For instance, it would be interesting to identify the types of quantum operations that contribute to the complexity and quantify them by data compression. It may elucidate the significance of natural analog quantum dynamics over digital quantum operations. Therefore, this direction paves the way for a novel fundamental characterization of quantum states and operations and its practical application to QML.

Bibliography

[1] R. P. Feynman, "Simulating physics with computers," *International Journal of Theoretical Physics*, vol. 21, pp. 467–488, 1981.

[2] J. Zhang, P. W. Hess, A. Kyprianidis, P. Becker, A. Lee, J. Smith, G. Pagano, I.-D. Potirniche, A. C. Potter, A. Vishwanath, N. Y. Yao, and C. Monroe, "Observation of a discrete time crystal," *Nature (London)*, vol. 543, p. 217, mar 2017.

[3] N. Friis, O. Marty, C. Maier, C. Hempel, M. Holzäpfel, P. Jurcevic, M. B. Plenio, M. Huber, C. Roos, R. Blatt, and B. Lanyon, "Observation of entangled states of a fully controlled 20-qubit system," *Phys. Rev. X*, vol. 8, p. 021012, Apr 2018.

[4] R. Harris, Y. Sato, A. J. Berkley, M. Reis, F. Altomare, M. H. Amin, K. Boothby, P. Bunyk, C. Deng, C. Enderud, S. Huang, E. Hoskinson, M. W. Johnson, E. Ladizinsky, N. Ladizinsky, T. Lanting, R. Li, T. Medina, R. Molavi, R. Neufeld, T. Oh, I. Pavlov, I. Perminov, G. Poulin-Lamarre, C. Rich, A. Smirnov, L. Swenson, N. Tsai, M. Volkmann, J. Whittaker, and J. Yao, "Phase transitions in a programmable quantum spin glass simulator," *Science*, vol. 361, no. 6398, pp. 162–165, 2018.

[5] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, pp. 505–510, Oct 2019.

[6] H.-S. Zhong, H. Wang, Y.-H. Deng, M.-C. Chen, L.-C. Peng, Y.-H. Luo, J. Qin, D. Wu, X. Ding, Y. Hu, *et al.*, "Quantum computational advantage using photons," *Science*, vol. 370, no. 6523, pp. 1460–1463, 2020.

[7] M. Gong, S. Wang, C. Zha, M.-C. Chen, H.-L. Huang, Y. Wu, Q. Zhu, Y. Zhao, S. Li, S. Guo, *et al.*, "Quantum walks on a programmable two-

dimensional 62-qubit superconducting processor," *Science*, vol. 372, no. 6545, pp. 948–952, 2021.

[8] P. Scholl, M. Schuler, H. J. Williams, A. A. Eberharter, D. Barredo, K.-N. Schymik, V. Lienhard, L.-P. Henry, T. C. Lang, T. Lahaye, A. M. Läuchli, and A. Browaeys, "Quantum simulation of 2d antiferromagnets with hundreds of rydberg atoms," *Nature*, vol. 595, pp. 233–238, Jul 2021.

[9] L. S. Madsen, F. Laudenbach, M. F. Askarani, F. Rortais, T. Vincent, J. F. F. Bulmer, F. M. Miatto, L. Neuhaus, L. G. Helt, M. J. Collins, *et al.*, "Quantum computational advantage with a programmable photonic processor," *Nature*, vol. 606, pp. 75–81, Jun 2022.

[10] P. Frey and S. Rachel, "Realization of a discrete time crystal on 57 qubits of a quantum computer," *Science Advances*, vol. 8, no. 9, p. eabm7652, 2022.

[11] A. L. Shaw, Z. Chen, J. Choi, D. K. Mark, P. Scholl, R. Finkelstein, A. Elben, S. Choi, and M. Endres, "Benchmarking highly entangled states on a 60-atom analogue quantum simulator," *Nature*, vol. 628, pp. 71–77, Apr 2024.

[12] A. D. King, A. Nocera, M. M. Rams, J. Dziarmaga, R. Wiersema, W. Bernoudy, J. Raymond, N. Kaushal, N. Heinsdorf, R. Harris, K. Boothby, F. Altomare, M. Asad, A. J. Berkley, M. Boschnak, K. Chern, H. Christiani, S. Cibere, J. Connor, M. H. Dehn, R. Deshpande, S. Ejtemaee, P. Farre, K. Hamer, E. Hoskinson, S. Huang, M. W. Johnson, S. Kortas, E. Ladizinsky, T. Lanting, T. Lai, R. Li, A. J. R. MacDonald, G. Marsden, C. C. McGeoch, R. Molavi, T. Oh, R. Neufeld, M. Norouzpour, J. Pasvolsky, P. Poitras, G. Poulin-Lamarre, T. Prescott, M. Reis, C. Rich, M. Samani, B. Sheldan, A. Smirnov, E. Sterpka, B. T. Clavera, N. Tsai, M. Volkmann, A. M. Whiticar, J. D. Whittaker, W. Wilkinson, J. Yao, T. J. Yi, A. W. Sandvik, G. Alvarez, R. G. Melko, J. Carrasquilla, M. Franz, and M. H. Amin, "Beyond-classical computation in quantum simulation," *Science*, vol. 388, no. 6743, pp. 199–204, 2025.

[13] IBMQuantum, "The era of quantum utility is here," *IBM Research Blog*, Dec. 4th, 2023 (visited Oct. 28th 2025).

[14] D. Castelvecchi, "Ibm releases first-ever 1,000-qubit quantum chip," *Nature*, vol. 624, p. 238, 2023.

[15] M. A. Norcia, H. Kim, W. B. Cairncross, M. Stone, A. Ryou, M. Jaffe, M. O. Brown, K. Barnes, P. Battaglino, T. C. Bohdanowicz, A. Brown, K. Cassella, C.-A. Chen, R. Coxe, D. Crow, J. Epstein, C. Griger, E. Halperin, F. Hummel, A. M. W. Jones, J. M. Kindem, J. King, K. Kotru, J. Lauigan, M. Li, M. Lu, E. Megidish, J. Marjanovic, M. McDonald, T. Mittiga, J. A. Muniz, S. Narayanaswami, C. Nishiguchi, T. Paule, K. A. Pawlak, L. S. Peng, K. L. Pudenz, D. Rodríguez Pérez, A. Smull, D. Stack, M. Urbanek, R. J. M. van de Veerdonk, Z. Vendeiro, L. Wadleigh, T. Wilkason, T.-Y. Wu, X. Xie, E. Zalys-Geller, X. Zhang, and B. J. Bloom, "Iterative assembly of ^{171}Yb atom arrays with cavity-enhanced optical lattices," *PRX Quantum*, vol. 5, p. 030316, Jul 2024.

[16] D. Deutsch, "Quantum theory, the church–turing principle and the universal quantum computer," *Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences*, vol. 400, no. 1818, pp. 97–117, 1985.

[17] D. E. Deutsch, "Quantum computational networks," *Proceedings of the royal society of London. A. mathematical and physical sciences*, vol. 425, no. 1868, pp. 73–90, 1989.

[18] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, vol. 439, no. 1907, pp. 553–558, 1992.

[19] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, 1994.

[20] A. Y. Kitaev, "Quantum measurements and the abelian stabilizer problem," *arXiv preprint quant-ph/9511026*, 1995.

[21] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219, 1996.

[22] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, p. 150502, Oct 2009.

[23] S. Aaronson, "Read the fine print," *Nature Physics*, vol. 11, pp. 291–293, Apr 2015.

[24] E. Campbell, A. Khurana, and A. Montanaro, "Applying quantum algorithms to constraint satisfaction problems," *Quantum*, vol. 3, p. 167, 2019.

[25] Y. R. Sanders, D. W. Berry, P. C. Costa, L. W. Tessler, N. Wiebe, C. Gidney, H. Neven, and R. Babbush, "Compilation of fault-tolerant quantum heuristics for combinatorial optimization," *PRX Quantum*, vol. 1, p. 020312, Nov 2020.

[26] R. Babbush, J. R. McClean, M. Newman, C. Gidney, S. Boixo, and H. Neven, "Focus beyond quadratic speedups for error-corrected quantum advantage," *PRX Quantum*, vol. 2, p. 010103, Mar 2021.

[27] R. Chrisley, "Quantum learning," in *New directions in cognitive science: Proceedings of the international symposium, Saariselka*, vol. 4, 1995.

[28] P. Berkhin, *A Survey of Clustering Data Mining Techniques*, pp. 25–71. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006.

[29] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, 2020.

[30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[31] C. Mead, "Neuromorphic electronic systems," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1629–1636, 1990.

[32] C. D. Schuman, T. E. Potok, R. M. Patton, J. D. Birdwell, M. E. Dean, G. S. Rose, and J. S. Plank, "A survey of neuromorphic computing and neural networks in hardware," *arXiv preprint arXiv:1705.06963*, 2017.

[33] D. Kudithipudi, C. Schuman, C. M. Vineyard, T. Pandit, C. Merkel, R. Kubendran, J. B. Aimone, G. Orchard, C. Mayr, R. Benosman, J. Hays, C. Young, C. Bartolozzi, A. Majumdar, S. G. Cardwell, M. Payvand, S. Buckley, S. Kulkarni, H. A. Gonzalez, G. Cauwenberghs, C. S. Thakur, A. Subramoney, and S. Furber, "Neuromorphic computing at scale," *Nature*, vol. 637, pp. 801–812, Jan 2025.

[34] S. C. Kak, "Quantum neural computing," vol. 94 of *Advances in Imaging and Electron Physics*, pp. 259–313, Elsevier, 1995.

[35] E. C. Behrman, J. Niemel, J. E. Steck, and S. R. Skinner, "A quantum dot neural network," in *Proceedings of the 4th Workshop on Physics of Computation*, pp. 22–24, 1996.

[36] M. Peruš, "Neuro-quantum parallelism in brain-mind and computers," *Informatica*, vol. 20, pp. 173–183, 1996.

[37] R. L. Chrisley, "Learning in non-superpositional quantum neurocomputers," *Brain, Mind and Physics. IOS Press, Amsterdam*, pp. 126–139, 1997.

[38] A. A. Ezhov and D. Ventura, *Quantum Neural Networks*, pp. 213–235. Heidelberg: Physica-Verlag HD, 2000.

[39] M. Altaisky, "Quantum neural network," *arXiv preprint quant-ph/0107012*, 2001.

[40] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Physical Review A*, vol. 98, no. 3, p. 032309, 2018.

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

[42] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature communications*, vol. 9, no. 1, p. 4812, 2018.

[43] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, "Connecting ansatz expressibility to gradient magnitudes and barren plateaus," *PRX quantum*, vol. 3, no. 1, p. 010313, 2022.

[44] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, "Barren plateaus in variational quantum computing," *Nature Reviews Physics*, pp. 1–16, 2025.

[45] K. Fujii and K. Nakajima, "Harnessing disordered-ensemble quantum dynamics for machine learning," *Physical Review Applied*, vol. 8, no. 2, p. 024030, 2017.

[46] M. Negoro, K. Mitarai, K. Fujii, K. Nakajima, and M. Kitagawa, "Machine learning with controllable quantum dynamics of a nuclear spin ensemble in a solid," *arXiv preprint arXiv:1806.10910*, 2018.

[47] K. Fujii and K. Nakajima, "Quantum reservoir computing: a reservoir approach toward quantum machine learning on near-term quantum devices," in *Reservoir Computing: Theory, Physical Implementations, and Applications*, pp. 423–450, Springer, 2021.

[48] S. Ghosh, A. Opala, M. Matuszewski, T. Paterek, and T. C. Liew, "Quantum reservoir processing," *npj Quantum Information*, vol. 5, no. 1, p. 35, 2019.

[49] S. Ghosh, T. Paterek, and T. C. H. Liew, "Quantum neuromorphic platform for quantum state preparation," *Phys. Rev. Lett.*, vol. 123, p. 260404, Dec 2019.

[50] A. Sakurai, M. P. Estarellas, W. J. Munro, and K. Nemoto, "Quantum extreme reservoir computation utilizing scale-free networks," *Phys. Rev. Applied*, vol. 17, p. 064044, Jun 2022.

[51] A. W. Harrow and R. A. Low, "Random quantum circuits are approximate 2-designs," *Communications in Mathematical Physics*, vol. 291, pp. 257–302, 2009.

[52] J. Haferkamp, "Random quantum circuits are approximate unitary t-designs in depth $o(nt^{5+o(1)})$," *Quantum*, vol. 6, p. 795, 2022.

[53] R. Martínez-Peña, G. L. Giorgi, J. Nokkala, M. C. Soriano, and R. Zambrini, "Dynamical phase transitions in quantum reservoir computing," *Phys. Rev. Lett.*, vol. 127, p. 100502, Aug 2021.

[54] R. Martínez-Peña, J. Nokkala, G. L. Giorgi, R. Zambrini, and M. C. Soriano, "Information processing capacity of spin-based quantum reservoir computing systems," *Cognitive Computation*, vol. 15, no. 5, pp. 1440–1451, 2023.

[55] A. Palacios, R. Martínez-Peña, M. C. Soriano, G. L. Giorgi, and R. Zambrini, "Role of coherence in many-body quantum reservoir computing," *Communications Physics*, vol. 7, no. 1, p. 369, 2024.

[56] M. Vetrano, G. Lo Monaco, L. Innocenti, S. Lorenzo, and G. M. Palma, "State estimation with quantum extreme learning machines beyond the scrambling time," *npj Quantum Information*, vol. 11, no. 1, p. 20, 2025.

[57] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, "Data re-uploading for a universal quantum classifier," *Quantum*, vol. 4, p. 226, 2020.

[58] M. Schuld, R. Sweke, and J. J. Meyer, "Effect of data encoding on the expressive power of variational quantum-machine-learning models," *Phys. Rev. A*, vol. 103, p. 032430, Mar 2021.

[59] W. Xiong, G. Facelli, M. Sahebi, O. Agnel, T. Chotibut, S. Thanasilp, and Z. Holmes, "On fundamental aspects of quantum extreme learning machines," *Quantum Machine Intelligence*, vol. 7, no. 1, p. 20, 2025.

[60] H. Georgi, *Lie algebras in particle physics: from isospin to unified theories*. Taylor & Francis, 2000.

[61] R. Achilles and A. Bonfiglioli, "The early proofs of the theorem of campbell, baker, hausdorff, and dynkin," *Archive for History of Exact Sciences*, vol. 66, pp. 295–358, May 2012.

[62] K. Nemoto, "Generalized coherent states for su(n) systems," *Journal of Physics A: Mathematical and General*, vol. 33, p. 3493, may 2000.

[63] G. Kimura, "The bloch vector for n-level systems," *Physics Letters A*, vol. 314, no. 5, pp. 339–349, 2003.

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

[65] A. Y. Kitaev, "Quantum computations: algorithms and error correction," *Russian Mathematical Surveys*, vol. 52, no. 6, p. 1191, 1997.

[66] J. S. Bell, "On the einstein podolsky rosen paradox," *Physics Physique Fizika*, vol. 1, pp. 195–200, Nov 1964.

[67] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt, "Proposed experiment to test local hidden-variable theories," *Phys. Rev. Lett.*, vol. 23, pp. 880–884, Oct 1969.

[68] N. Brunner, D. Cavalcanti, S. Pironio, V. Scarani, and S. Wehner, "Bell nonlocality," *Rev. Mod. Phys.*, vol. 86, pp. 419–478, Apr 2014.

[69] D. Gottesman, *Stabilizer codes and quantum error correction*. California Institute of Technology, 1997.

[70] S. Aaronson and D. Gottesman, "Improved simulation of stabilizer circuits," *Phys. Rev. A*, vol. 70, p. 052328, Nov 2004.

[71] V. Veitch, S. H. Mousavian, D. Gottesman, and J. Emerson, "The resource theory of stabilizer quantum computation," *New Journal of Physics*, vol. 16, no. 1, p. 013009, 2014.

[72] M. Howard and E. Campbell, "Application of a resource theory for magic states to fault-tolerant quantum computing," *Phys. Rev. Lett.*, vol. 118, p. 090501, Mar 2017.

[73] L. Leone, S. F. E. Oliviero, and A. Hamma, "Stabilizer rényi entropy," *Phys. Rev. Lett.*, vol. 128, p. 050402, Feb 2022.

[74] A. Hamma, S. Santra, and P. Zanardi, "Quantum entanglement in random physical states," *Phys. Rev. Lett.*, vol. 109, p. 040502, Jul 2012.

[75] X. Turkeshi, E. Tirrito, and P. Sierant, "Magic spreading in random quantum circuits," *Nature Communications*, vol. 16, p. 2575, Mar 2025.

[76] D. DiVincenzo, D. Leung, and B. Terhal, "Quantum data hiding," *IEEE Transactions on Information Theory*, vol. 48, no. 3, pp. 580–598, 2002.

[77] A. Ambainis, J. Bouda, and A. Winter, "Nonmalleable encryption of quantum information," *Journal of Mathematical Physics*, vol. 50, no. 4, 2009.

[78] E. Magesan, J. M. Gambetta, and J. Emerson, "Scalable and robust randomized benchmarking of quantum processes," *Phys. Rev. Lett.*, vol. 106, p. 180504, May 2011.

[79] M. Kliesch and I. Roth, "Theory of quantum system certification," *PRX Quantum*, vol. 2, p. 010201, Jan 2021.

[80] J. Helsen, I. Roth, E. Onorati, A. Werner, and J. Eisert, "General framework for randomized benchmarking," *PRX Quantum*, vol. 3, p. 020357, Jun 2022.

[81] D. Hangleiter and J. Eisert, "Computational advantage of quantum random sampling," *Rev. Mod. Phys.*, vol. 95, p. 035001, Jul 2023.

[82] D. A. Roberts and B. Yoshida, "Chaos and complexity by design," *Journal of High Energy Physics*, vol. 2017, no. 4, pp. 1–64, 2017.

[83] L. Leone, S. F. Oliviero, and A. Hamma, "Isospectral twirling and quantum chaos," *Entropy*, vol. 23, no. 8, p. 1073, 2021.

[84] M. A. Valdez, D. Jaschke, D. L. Vargas, and L. D. Carr, "Quantifying complexity in quantum phase transitions via mutual information complex networks," *Phys. Rev. Lett.*, vol. 119, p. 225301, Nov 2017.

[85] B. Sundar, M. A. Valdez, L. D. Carr, and K. R. A. Hazzard, "Complex-network description of thermal quantum states in the ising spin chain," *Phys. Rev. A*, vol. 97, p. 052320, May 2018.

[86] R. Mukherjee, A. E. Mirasola, J. Hollingsworth, I. G. White, and K. R. A. Hazzard, "Geometric representation of spin correlations and applications to ultracold systems," *Phys. Rev. A*, vol. 97, p. 043606, Apr 2018.

[87] S. Zaman and W.-C. Lee, "Real-space visualization of quantum phase transitions by network topology," *Phys. Rev. E*, vol. 100, p. 012304, Jul 2019.

[88] A. A. Bagrov, M. Danilov, S. Brener, M. Harland, A. I. Lichtenstein, and M. I. Katsnelson, "Detecting quantum critical points in the t-t' fermi-hubbard model via complex network theory," *Scientific Reports*, vol. 10, p. 20470, Nov 2020.

[89] J. Nokkala, J. Piilo, and G. Bianconi, "Complex quantum networks: a topical review," *Journal of Physics A: Mathematical and Theoretical*, vol. 57, p. 233001, may 2024.

[90] M. P. Estarellas, T. Osada, V. M. Bastidas, B. Renoust, K. Sanaka, W. J. Munro, and K. Nemoto, "Simulating complex quantum networks with time crystals," *Science Advances*, vol. 6, no. 42, p. eaay8892, 2020.

[91] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[92] C. F. G. D. Santos and J. a. P. Papa, "Avoiding overfitting: A survey on regularization methods for convolutional neural networks," *ACM Comput. Surv.*, vol. 54, Sept. 2022.

[93] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Phys. Rev. Lett.*, vol. 113, p. 130503, Sep 2014.

[94] P. Mujal, R. Martínez-Peña, J. Nokkala, J. García-Beni, G. L. Giorgi, M. C. Soriano, and R. Zambrini, "Opportunities in quantum reservoir computing and extreme learning machines," *Advanced Quantum Technologies*, vol. 4, no. 8, p. 2100027, 2021.

[95] N. Munikote, "Comparing quantum encoding techniques," *arXiv preprint arXiv:2410.09121*, 2024.

[96] Y. Wang, S. Crain, C. Fang, B. Zhang, S. Huang, Q. Liang, P. H. Leung, K. R. Brown, and J. Kim, "High-fidelity two-qubit gates using a microelectromechanical-system-based beam steering system for individual qubit addressing," *Physical Review Letters*, vol. 125, no. 15, p. 150505, 2020.

[97] I. N. Moskalenko, I. A. Simakov, N. N. Abramov, A. A. Grigorev, D. O. Moskalev, A. A. Pishchimova, N. S. Smirnov, E. V. Zikiy, I. A. Rodionov, and I. S. Besedin, "High fidelity two-qubit gates on fluxoniums using a tunable coupler," *npj Quantum Information*, vol. 8, no. 1, p. 130, 2022.

[98] M. Ragone, B. N. Bakalov, F. Sauvage, A. F. Kemper, C. Ortiz Marrero, M. Larocca, and M. Cerezo, "A lie algebraic theory of barren plateaus for deep parameterized quantum circuits," *Nature Communications*, vol. 15, no. 1, p. 7172, 2024.

[99] H. Kawai and Y. O. Nakagawa, "Predicting excited states from ground state wavefunction by supervised quantum machine learning," *Machine Learning: Science and Technology*, vol. 1, no. 4, p. 045027, 2020.

[100] Y. Suzuki, Q. Gao, K. C. Pradel, K. Yasuoka, and N. Yamamoto, "Natural quantum reservoir computing for temporal information processing," *Scientific reports*, vol. 12, no. 1, p. 1353, 2022.

[101] K. Kobayashi, K. Fujii, and N. Yamamoto, "Feedback-driven quantum reservoir computing for time-series analysis," *PRX Quantum*, vol. 5, p. 040325, Nov 2024.

[102] A. Sannia, R. Martínez-Peña, M. C. Soriano, G. L. Giorgi, and R. Zambrini, "Dissipation as a resource for quantum reservoir computing," *Quantum*, vol. 8, p. 1291, 2024.

[103] J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar, "Information processing capacity of dynamical systems," *Scientific reports*, vol. 2, no. 1, p. 514, 2012.

[104] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley interdisciplinary reviews: computational statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[105] Y. S. Weinstein, W. G. Brown, and L. Viola, "Parameters of pseudorandom quantum circuits," *Phys. Rev. A*, vol. 78, p. 052332, Nov 2008.

[106] F. Tacchino, C. Macchiavello, D. Gerace, and D. Bajoni, "An artificial neuron implemented on an actual quantum processor," *npj Quantum Information*, vol. 5, p. 26, Mar 2019.

[107] N. J. Higham, *Functions of Matrices: Theory and Computation*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2008.

[108] M. Mahoney and C. Martin, "Traditional and heavy tailed self regularization in neural network models," in *International Conference on Machine Learning*, pp. 4284–4293, PMLR, 2019.

[109] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[110] J. Emerson, E. Livine, and S. Lloyd, "Convergence conditions for random quantum circuits," *Phys. Rev. A*, vol. 72, p. 060302, Dec 2005.

[111] S. Boixo, S. V. Isakov, V. N. Smelyanskiy, R. Babbush, N. Ding, Z. Jiang, M. J. Bremner, J. M. Martinis, and H. Neven, "Characterizing quantum supremacy in near-term devices," *Nature Physics*, vol. 14, pp. 595–600, Jun 2018.

[112] S. Mullane, "Sampling random quantum circuits: a pedestrian's guide," 2020.

[113] F. Mezzadri, "How to generate random matrices from the classical compact groups," *Notices of the American Mathematical Society*, vol. 54, no. 5, pp. 592 – 604, 2007.

[114] L. Kuśmierz, S. Ogawa, and T. Toyoizumi, "Edge of chaos and avalanches in neural networks with heavy-tailed synaptic weight distribution," *Phys. Rev. Lett.*, vol. 125, p. 028101, Jul 2020.

[115] R. Caruana, S. Lawrence, and C. Giles, "Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping," *Advances in neural information processing systems*, vol. 13, 2000.

[116] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. 61, pp. 2121–2159, 2011.

[117] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," *arXiv preprint arXiv:1708.07747*, 2017.

[118] J. Edwards and D. Thouless, "Numerical studies of localization in disordered systems," *Journal of Physics C: Solid State Physics*, vol. 5, no. 8, p. 807, 1972.

[119] T. J. Osborne, "Statics and dynamics of quantum xy and heisenberg systems on graphs," *Phys. Rev. B*, vol. 74, p. 094411, Sep 2006.

[120] M. Christandl, N. Datta, A. Ekert, and A. J. Landahl, "Perfect state transfer in quantum spin networks," *Phys. Rev. Lett.*, vol. 92, p. 187902, May 2004.

[121] M. Christandl, N. Datta, T. C. Dorlas, A. Ekert, A. Kay, and A. J. Landahl, "Perfect transfer of arbitrary states in quantum spin networks," *Phys. Rev. A*, vol. 71, p. 032312, Mar 2005.

[122] A. Kay, "Perfect, efficient, state transfer and its application as a constructive tool," *International Journal of Quantum Information*, vol. 08, no. 04, pp. 641–676, 2010.

[123] C. Godsil, "Periodic graphs.," *The Electronic Journal of Combinatorics [electronic only]*, vol. 18, no. 1, pp. Research Paper P23, 15 p., electronic only–Research Paper P23, 15 p., electronic only, 2011.

[124] I. Niven, *Irrational numbers*. No. 11, Cambridge University Press, 2005.

[125] A. De Pasquale, G. Costantini, P. Facchi, G. Florio, S. Pascazio, and K. Yuasa, "Xx model on the circle," *The European Physical Journal Special Topics*, vol. 160, pp. 127–138, Jul 2008.

[126] R. Fabila-Monroy, D. Flores-Peñaloza, C. Huemer, F. Hurtado, J. Urrutia, and D. R. Wood, "Token graphs," *Graphs and Combinatorics*, vol. 28, pp. 365–380, May 2012.

[127] F. Evers and A. D. Mirlin, "Anderson transitions," *Rev. Mod. Phys.*, vol. 80, pp. 1355–1417, Oct 2008.

[128] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin of the American Mathematical Society*, vol. 43, no. 4, pp. 439–561, 2006.

[129] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural computation*, vol. 4, no. 1, pp. 1–58, 1992.

[130] D. A. McAllester, "Pac-bayesian stochastic model selection," *Machine Learning*, vol. 51, no. 1, pp. 5–21, 2003.

[131] P. D. Grünwald, *The minimum description length principle*. MIT press, 2007.

[132] M. Belkin, D. Hsu, S. Ma, and S. Mandal, "Reconciling modern machine-learning practice and the classical bias–variance trade-off," *Proceedings of the National Academy of Sciences*, vol. 116, no. 32, pp. 15849–15854, 2019.

[133] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[134] P. Grünwald and T. Roos, "Minimum description length revisited," *International journal of mathematics for industry*, vol. 11, no. 01, p. 1930001, 2019.

[135] P. M. Vitányi and M. Li, "Minimum description length induction, bayesianism, and kolmogorov complexity," *IEEE Transactions on information theory*, vol. 46, no. 2, pp. 446–464, 2000.

[136] U. V. Luxburg, O. Bousquet, and B. Schölkopf, "A compression approach to support vector model selection," *Journal of Machine Learning Research*, vol. 5, no. Apr, pp. 293–323, 2004.

[137] S. Lotfi, M. Finzi, S. Kapoor, A. Potapczynski, M. Goldblum, and A. G. Wilson, "Pac-bayes compression bounds so tight that they can explain generalization," *Advances in Neural Information Processing Systems*, vol. 35, pp. 31459–31473, 2022.

[138] M. Goldblum, M. Finzi, K. Rowan, and A. G. Wilson, "The no free lunch theorem, kolmogorov complexity, and the role of inductive biases in machine learning," *arXiv preprint arXiv:2304.05366*, 2023.

[139] M. Li, P. Vitányi, *et al.*, *An introduction to Kolmogorov complexity and its applications*, vol. 3. Springer, 2008.

[140] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[141] U. Schollwöck, "The density-matrix renormalization group," *Rev. Mod. Phys.*, vol. 77, pp. 259–315, Apr 2005.

[142] C. Sabín and G. García-Alcaine, "A classification of entanglement in three-qubit systems," *The european physical journal D*, vol. 48, no. 3, pp. 435–442, 2008.

[143] P. Gacs, "Quantum algorithmic entropy," in *Proceedings 16th Annual IEEE Conference on Computational Complexity*, pp. 274–283, IEEE, 2001.

[144] A. Berthiaume, W. Van Dam, and S. Laplante, "Quantum kolmogorov complexity," *Journal of Computer and System Sciences*, vol. 63, no. 2, pp. 201–221, 2001.

[145] P. M. Vitányi, "Quantum kolmogorov complexity based on classical descriptions," *IEEE Transactions on Information Theory*, vol. 47, no. 6, pp. 2464–2479, 2002.

[146] T. Yamakami, "Computational complexity measures of multipartite quantum entanglement: (extented abstract)," in *International Symposium on Algorithms and Computation*, pp. 117–128, Springer, 2003.

[147] C. E. Mora and H. J. Briegel, "Algorithmic complexity and entanglement of quantum states," *Physical review letters*, vol. 95, no. 20, p. 200503, 2005.

[148] M. Lemus, R. Faleiro, P. Mateus, N. Paunković, and A. Souto, "Quantum kolmogorov complexity and quantum correlations in deterministic-control quantum turing machines," *Quantum*, vol. 8, p. 1230, 2024.

[149] P. Alquier *et al.*, "User-friendly introduction to pac-bayes bounds," *Foundations and Trends® in Machine Learning*, vol. 17, no. 2, pp. 174–303, 2024.