Quantum Phase Recognition via Quantum Kernel Methods

Yusen Wu[1], Bujiao Wu[2], Jingbo B. Wang[1], and Xiao Yuan[2]

[1]Department of Physics, The University of Western Australia, Perth, WA 6009, Australia

[2]Center on Frontiers of Computing Studies, Peking University, Beijing 100871, China

The application of quantum computation to accelerate machine learning algorithms is one of the most promising areas of research in quantum algorithms. In this paper, we explore the power of quantum learning algorithms in solving an important class of Quantum Phase Recognition (QPR) problems, which are crucially important in understanding many-particle quantum systems. We prove that, under widely believed complexity theory assumptions, there exists a wide range of QPR problems that cannot be efficiently solved by classical learning algorithms with classical resources. Whereas using a quantum computer, we prove the efficiency and robustness of quantum kernel methods in solving QPR problems through Linear order parameter Observables. We numerically benchmark our algorithm for a variety of problems, including recognizing symmetry-protected topological phases and symmetry-broken phases. Our results highlight the capability of quantum machine learning in predicting such quantum phase transitions in many-particle systems.

1 Introduction

The complex nature of multi-particle entanglement has stimulated various powerful classical techniques to study many-particle quantum systems, including density functional theory [1], density matrix renormalization group [2, 3], quantum Monte Carlo [4–7] etc. Classical machine learning techniques have recently been considered as an alternative means to study and understand many-particle quantum systems and the associated quantum processes [8–16]. From numerical perspectives, these works have shown that neural network state ansätzes have stronger representation power than conventional tensor networks and may help to solve complex static and dynamical quantum problems. Nevertheless, most of these methods lack of rigorous theoretical guarantees, and whether a learning procedure outperforms original method in solving quantum many-body problems is unknown. To answer this question, Huang et al. [17] explored the power of classical machine learning in classifying some quantum phases of matter and proposed a provable efficient classical method in learning through shadow tomography [18].

However, for quantum many-body systems that possess intricate and long-range entanglement, and if the target quantum phase is determined by a non-local order parameter observable, the required sampling complexity is expected to increase exponentially with respect to the system size. Quantum machine learning has been intensely studied in terms of its expressive ability [19–21], optimization [22–25], provable quantum advantages [26–28], as well as potential limitations [29]. Meanwhile, recent pioneering experiments on quantum computer processors [26, 30–32] have demonstrated significant quantum computing advantages in random state sampling [30–32] and density matrix learning problems [26]. In detail, Huang et al. [26, 27] proved that the entangled Bell-Measurement protocol can efficiently extract information from an unknown density matrix and predict its linear properties, meanwhile, it is classically hard in the *worst-case* scenario. Therefore, two critical questions are still open: (1) what is the limitation of classical machine learning in solving quantum many-body problems? and (2) whether a near-term quantum computer can enhance the power of classical learning algorithms in solving practical problems?

In this paper, we provide a novel approach to address the above two questions. For a many-body quantum system described by an n-qubit parameterized Hamiltonian $H(\boldsymbol{a}) = \sum_{j=1}^{m} \boldsymbol{a}_j P_j$, where $\boldsymbol{a} \in \mathbb{R}^m$ represents external parameters and P_j are n-qubit Pauli operators, we focus on a class of Quantum Phase Recognition problems that can be distinguished by a Linear order parameter Observable, termed as the LO-QPR problem. We aim at learning about detailed phase transitions of many-particle quantum systems using a quantum kernel method. In the learning phase, a classical training data set $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^{N}$ is used, where $\boldsymbol{a_i}$ and b_i are respectively the external parameters and ground state property observed from experiments. In the prediction phase, the learning algorithm succeeds if it correctly predicts the property b of the ground state $|\psi(\boldsymbol{a})\rangle$. For example, considering the Ising Hamiltonian, the external parameter $\boldsymbol{a}$ could be the strength of the transverse magnetic field, and b represents quantum phases such as the paramagnetic, ferromagnetic, and antiferromagnetic phases. Phase transitions occur when the external parame-

Jingbo B. Wang: jingbo.wang@uwa.edu.au
Xiao Yuan: xiaoyuan@pku.edu.cn

ters varies [33], and the ability to correctly predict the quantum phase transition boundary can help us understand many strong-correlated systems, even for canonical microscopic physical models [34].

Under two widely accepted assumptions: (1) the polynomial hierarchy does not collapse in the computational complexity theory, and (2) the classical hardness for ground state sampling holds, we prove that certain LO-QPR problems are hard for any classical machine learning methods with classical resources. We demonstrate that if these LO-QPR problems could be solved by a classical learner with classical resources (even up to a general additive error tolerance), then the infinite tower of the polynomial hierarchy would collapse to its second level. While this does not imply that P = NP, such a collapse is also widely regarded as being implausible. We therefore answer the first question by showing the exact limitation of classical machine learning in LO-QPR.

The rapid advancement of realistic quantum devices provides an opportunity to answer the second question in a fundamentally different and more powerful way compared with classical ML. Instead of classically simulating ground states and then infer quantum phase transition, we utilize quantum machine learning to extract high-level abstractions from observed data and directly process quantum ground states information by a quantum computer. Here, the ground state $|\psi(\boldsymbol{a})\rangle$ of $H(\boldsymbol{a})$ embeds classical external parameter $\boldsymbol{a}$ onto a specific quantum-enhanced feature space, where inner products of such quantum feature states give rise to a quantum kernel, a metric to characterize distances in the feature space. As a result, predicting the ground state property can be transformed into quantum state overlap computation, and thus bypasses the required exponential sample complexity in Ref. [17]. We prove that the proposed Quantum Kernel Alphatron (QKA) algorithm can efficiently learn from quantum data and solve LO-QPR problems with a promisingly small learning error. We benchmark the proposed QKA in detecting symmetry-protected topological phases and symmetry broken phases, and simulation results show better performances compared with previous QMLs [35] and classical MLs [17]

This paper is organized as follows. In Sec. 2 we review some related works and give the definition of the LO-QPR problem, then introduce supervised learning with quantum feature spaces. We prove the hardness of classical learning algorithms in Sec. 3. Sec. 4 details our quantum learning algorithm for LO-QPR problem. Sec. 5 presents our numerical simulations. Sec 6 classifies various complexity classes of learning algorithms. Finally, Sec. 7 concludes the paper.

2 Preliminaries

To clearly demonstrate our contributions in this paper, we first review previous related works and define the learning and computation tasks of interest.

Task 1 (Density Matrix Learning [26, 27])**.** Given an artificial n-qubit density matrix $\rho = (I + 0.9P)/2^n$ where $P \in \mathcal{P} = \{I, X, Y, Z\}^{\otimes n} \setminus I^{\otimes n}$. The learning algorithms learn about ρ through conventional or quantum-enhanced measurement strategies. The learning algorithm succeeds if it can correctly predict the expectation value $\mathrm{Tr}\,(\rho Q)$ within an ϵ additive error with 3/4 probability for $Q \in \mathcal{P}$.

In [26, 27], the authors proved that the *entangled Bell-Measurement* protocol can use $\mathcal{O}(n/\epsilon^4)$ copies of ρ to solve this learning problem, meanwhile, it is classically hard in the *worst-case* scenario in estimating $\mathrm{Tr}\,(\rho Q)$ for some $Q \in \mathcal{P}$.

Task 2 (Quantum Phase Learning [17])**.** *Given the shadow tomography data set $\mathcal{D} = \{\Phi_{\text{shadow}}(\boldsymbol{a}_i)\}_{i=1}^N$, where $\Phi_{\text{shadow}}(\boldsymbol{a}_i)$ is a classical representation of the ground state of a Hamiltonian $H(\boldsymbol{a}_i)$, the task is to determine the quantum phase of matter for each $\Phi_{\text{shadow}}(\boldsymbol{a}_i) \in \mathcal{D}$.*

In [17], the authors developed a "classical" learning algorithm based on classical data, which are however obtained by shadow tomography of the target ground state generated by a quantum computer. As they discussed in [36], when the quantum phase transition can only be determined by a non-local order parameter observable, the sample complexity is expected to increase exponentially with respect to the system size. In this case, a better way is using a quantum computer to learn directly from the quantum phase value $b \in \{0, 1\}$ observed from the experiment, which will be defined as Task 4 below.

Task 3 (Ground state Linear Property (GLP) problem)**.** *Given an n-qubit Hamiltonian $H(\boldsymbol{a})$ with external parameters $\boldsymbol{a}$ and an observable $\mathcal{M} \in \mathbb{C}^{2^n \times 2^n}$, the goal of GLP is to approximate the ground state property $b = \langle\psi(\boldsymbol{a})|\mathcal{M}|\psi(\boldsymbol{a})\rangle$ to additive error $\epsilon = 1/\mathrm{poly}(n)$, that is $|\hat{b} - b| \le \epsilon$, where $\hat{b}$ represents the estimated property and $|\psi(\boldsymbol{a})\rangle$ is the ground state of $H(\boldsymbol{a})$.*

Noting that the GLP problem could characterize a class of quantum phase problem that is determined by a linear order parameter, which is one of the most significant elements in understanding quantum fluctuation phenomenons in condensed-matter systems [41, 42]. For example, considering the Ising Hamiltonian, the external parameter $\boldsymbol{a}$ and the order parameter observable $\mathcal{M}$ could be the strength of the transverse magnetic field and the spin correlation respectively, and quantum phases b include paramagnetic, ferromagnetic, and antiferromagnetic phases.

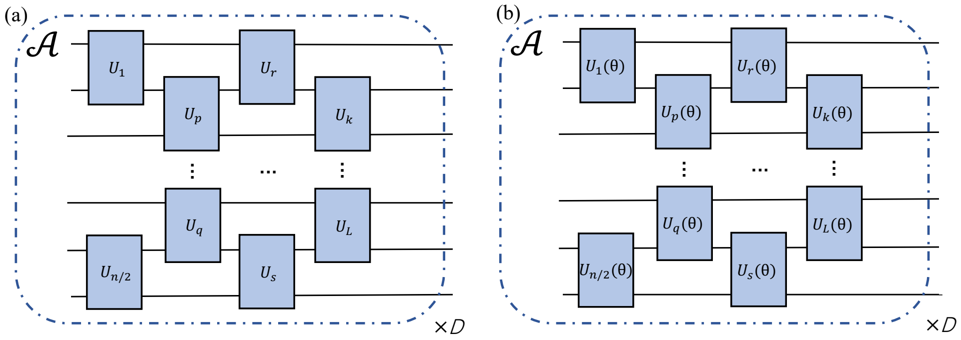


Figure 1: The comparison between quantum random circuits and variational quantum circuits: (a) The quantum random circuit $U = U_1U_2...U_L...U_{LD}$ is generated according to a brickwork architecture $\mathcal{A}$, and each U_i is randomly sampled from $SU(4)$ group based on the Haar measure [37]. (b) The variational quantum circuit $U(\boldsymbol{\theta}) = U_1(\theta_1)U_2(\theta_2)...U_L(\theta_L)...U_{LD}(\theta_{LD})$ can be generated by the same architecture $\mathcal{A}$. Each parameterized quantum gate $U_i(\theta_i)$ is determined by θ_i. Basically, the selection of the $U_i(\theta_i)$ is flexible, including the alternating layered ansatz [38], hardware efficient ansatz [39] and particle preserved ansatz [40].

In general, it would be hard to recognize quantum phases of an arbitrary many-body quantum system, owing to the hardness of obtaining the ground state and the fact that the order parameter is generally unknown. Nevertheless, there may also exist cases where the problem is exactly efficiently solvable for very specific choices of parameters. Then, a natural question is, based on the solvable or known quantum phases from experiments, whether we could learn and predict GLP for other external parameter domains. Therefore, it is reasonable to consider the learning version of the GLP problem.

Task 4 (LO-QPR)**.** *Given the training data $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^{N}$ for which $\boldsymbol{a_i}$ indicates the external parameter, and b_i represents its quantum phase value characterized by some unknown observable, associated with an n-qubit Hamiltonian $H(\boldsymbol{a_i}) = \sum_k \boldsymbol{a}_i^{(k)} P_k$, our aim is to efficiently learn a model $h(\boldsymbol{a})$ enabling the risk*

$$R(h) = \sum_{\boldsymbol{a}\sim\mathcal{X}} \mathcal{D}(\boldsymbol{a})\left(h(\boldsymbol{a}) - b\right)^2 \tag{1}$$

is upper bounded by $\mathcal{O}\left(\sqrt[4]{\frac{\log(1/\delta)}{N}}\right)$ with $1-\delta$ success probability for some fixed distribution $\mathcal{D}(\boldsymbol{a})$ defined on the external parameter space $\mathcal{X}$.

The crux of the matter on solving LO-QPR problems can be summarized into two levels: how to generate high-quality training data and how to efficiently learn from these data. Solving LO-QPR problems inevitably involves high dimensional Hilbert space, and classical learning algorithms thus are challenging in both generating and operating training data with quantum entanglement.

3 Classical hardness for LO-QPR

3.1 Express ground state by quantum circuit

The *brickwork* architecture $\mathcal{A}$ is a general approach to construct ansatz for ground state computation [37], whose structure is formed as follows: perform a string of two-qubit gates $U_1 \otimes U_2 \otimes ... \otimes U_{n/2}$ as the first layer, then perform a staggering string of gates, as illustrated as Fig. 1 (a). Meanwhile, the brickwork architecture can induce structured variational quantum circuit, which is a convinced method in approximating the ground state of many-body Hamiltonians $H(\boldsymbol{x})$ [39, 43–47].

The key idea of using variational quantum circuit is that the parameterized quantum state is prepared and measured on a quantum computer, and the classical optimizer updates the parameter $\boldsymbol{\theta}$ according to the measurement information. With the brickwork architecture $\mathcal{A}$, the variational ansatz can be prepared by $|\Psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0^n\rangle = \prod_{d=1}^{D} U_d(\boldsymbol{\theta}_d)|0^n\rangle$, where $U(\boldsymbol{\theta})$ is composed of D unitaries $U_d(\boldsymbol{\theta}_d)$ whose structure is shown in Fig. 1 (b). After several classical optimisation steps, the classical optimizer can provide a parameter $\boldsymbol{\theta_x}$ enabling $|\Psi(\boldsymbol{\theta_x})\rangle$ be an approximation of $|\psi(\boldsymbol{x})\rangle$. We provide a deterministic method in finding a $\boldsymbol{\theta_x}$ to approximate $|\psi(\boldsymbol{x})\rangle$ by using the quantum imaginary time evolution in the Appendix B. Since the variational quantum circuit $U(\boldsymbol{\theta})$ has the same architecture $\mathcal{A}$ to that of random circuit U, and two-qubit gates $U_i(\theta_i)$ are sampled from a subset of SU(4). Then the relationship $\mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta}) \subseteq \mathcal{U}_{\mathcal{A}}$ holds, where $\mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta})$ and $\mathcal{U}_{\mathcal{A}}$ denote the set of $U(\boldsymbol{\theta})$ and random circuit set, respectively.

3.2 Classical hardness results

The GLP problem is an instance of the mean-value problem which is the central part of the variational quantum algorithms, and "*Is the quantum computer necessary for the mean value problem?*" is still an open problem as mentioned in [48]. Note that for a quantum state $|\psi(\boldsymbol{x})\rangle = U|0^n\rangle$, Ref [48] proposed an upper bound on estimating $\langle\psi(\boldsymbol{x})|\mathcal{M}|\psi(\boldsymbol{x})\rangle$ in the case of a poly(n)-depth U associated with $|\psi(\boldsymbol{x})\rangle$. Here, we show the reduction from quantum circuit sampling problem to the GLP problem (see the proof in Lemma 1), and prove that the classical hardness of quantum sampling will imply the hardness of the GLP problem in the worst-case scenario, and thus provide a lower bound on estimating $\langle\psi(\boldsymbol{x})|\mathcal{M}|\psi(\boldsymbol{x})\rangle$.

The proof of Lemma 1 is inspired by the classical hardness conjecture of the random quantum state sampling problem in [49]. The random quantum state sampling is an artificial problem that samples from the output distribution of some experimentally feasible quantum algorithms. Sample one particular bitstring $\boldsymbol{j}$ from a random quantum circuit U with the exact probability $p_U(\boldsymbol{j})$ is believed classically hard under standard complexity assumptions, which is also recognized as the worst-case hardness.

However, a convincing quantum advantage must be established in the average-case scenario, that is, the classical hardness should be held across the entire distribution, rather than concentrated in a single quantum process and output. Bouland et al. [49] introduced the Feynman path integral to connect a bridge between a fixed outcome $\boldsymbol{j}$ and a low-degree multivariate polynomial described by quantum gates, and thus proved the worst-to-average reduction. We utilize a similar Feynman path integral method to prove the classical learning hardness result for LO-QPR problem (see Theorem 1).

Conjecture 1 (Ref. [49])**.** *There exists an n-qubit ground state $|\psi(\boldsymbol{x})\rangle = U(\boldsymbol{\theta_x})|0^n\rangle$ such that the following task is #P-hard: approximate $p_U(\boldsymbol{j}) = |\langle\boldsymbol{j}|\psi(\boldsymbol{x})\rangle|^2$ to additive error $\epsilon_c/2^n$ with probability $\frac{3}{4}+\frac{1}{\mathrm{poly}(n)}$, where $\boldsymbol{j}$ is a $\{0,1\}^n$ bit string, $U(\boldsymbol{\theta_x})$ is an n-qubit quantum circuit and $\epsilon_c = 1/\mathrm{poly}(n)$* [1]*.*

The probability distribution of a truly random quantum state $|\psi\rangle$ possesses the Porter-Thomas (PT) distribution $\Pr(|\langle\boldsymbol{j}|\psi\rangle|^2) = 2^n e^{-2^n|\langle\boldsymbol{j}|\psi\rangle|^2}$, which is known to be classically hard to sample [30, 50]. Whether ground states $|\psi(\boldsymbol{x})\rangle$ of a family of Hamiltonian $H(\boldsymbol{x})$ satisfy conjecture 1 can be verified by comparing probability distribution of $|\psi(\boldsymbol{x})\rangle$ to the PT distribution. We show that if the probability distribution of a ground state $|\psi(\boldsymbol{x})\rangle$ is $\mathcal{O}(n^{-1})$-close to PT distribution by means of trace distance, then sample from $|\psi(\boldsymbol{x})\rangle$ is classically hard. Details refer to Theorem 4 in the Appendix E.

Using this result and the relationship between ground states and variational quantum states, the hardness result for the GLP problem can be stated as the following lemma.

Lemma 1. *With the assumption that Conjecture 1 holds, and the Polynomial Hierarchy (PH) in the computational complexity theory does not collapse, there exists an n-qubit Hamiltonian $H(\boldsymbol{a})$ and an observable $\mathcal{M}$, such that their corresponding GLP problem cannot be efficiently calculated by any classical algorithm.*

We provide the detailed proof in the Appendix C.1. This lemma also serves for the hardness of the LO-QPR problem. Let $|\psi(\boldsymbol{a})\rangle = U(\boldsymbol{\theta_a})|0^n\rangle$ be the ground state of Hamiltonian $H(\boldsymbol{a})$ satisfying Lemma 1, where $U(\boldsymbol{\theta_a}) \in \mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta})$. For a general family of Hamiltonian $H(\boldsymbol{a})$, the explicitly mathematical expression of its ground state is hard to be characterized, and variational quantum circuit is a natural expression which characterizes the circuit complexity of these concerned ground states, but not change their properties [39, 43–45]. As a starting point of worst case scenario $|\psi(\boldsymbol{a})\rangle = U(\boldsymbol{\theta_a})|0^n\rangle$, we provide a method on constructing ground states $\mathcal{T} = \{|\psi(\boldsymbol{a}_i)\rangle\}$ by using variational circuit $U(\boldsymbol{\theta})$. Detail refers to the Appendix C.3. Then we have the following theorem for the average case hardness of the LO-QPR problem on $\mathcal{T}$. Given classical training data $\mathcal{S}$ (given $\boldsymbol{a}_i$, its label b_i can be computed by some classical Turing machines), we prove that no classical learning algorithm can efficiently learn the hypothesis h^* such that $R(h^*(\boldsymbol{x}))$ is upper bounded by $1/\mathrm{poly}(n)$ for $\boldsymbol{x} \in \mathcal{T}$.

Theorem 1. *Given training data $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^N$ (acquired from classical methods) for which $(\boldsymbol{a_i}, b_i)$ indicate the external parameter and phase value associated with the n-qubit Hamiltonian $H(\boldsymbol{a_i})$, there exists a testing set $\mathcal{T} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^M$, such that predicting $8/9$ of $y_i \in \mathcal{T}$ with additive error $1/\mathrm{poly}(n)$ is hard for any classical ML algorithm, with the assumption that Conjecture 1 holds and the PH does not collapse, where the scale of testing data $M = \mathrm{poly}(N)$ and $N = \mathrm{poly}(n)$.*

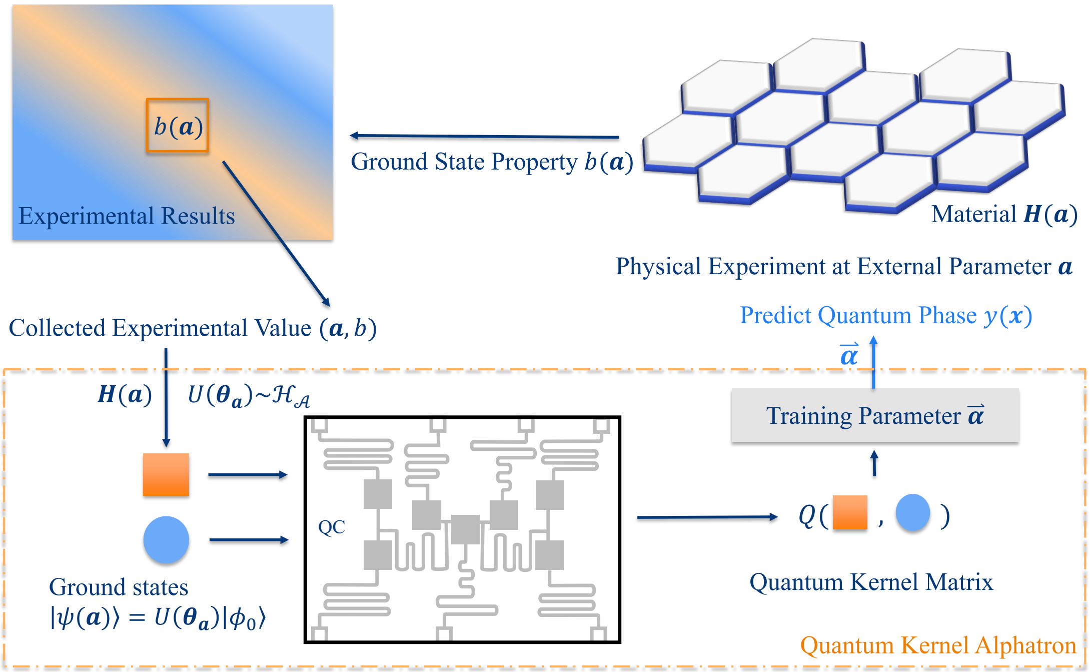
Theorem 1 gives average-case hardness results for the C-Learning Alg.+ C-Data on the testing set $\mathcal{T}$. The meaning of C-Learning Alg.+C-Data refers to Sec. 6, and proof details are provided in the Appendix C.3.

4 Quantum learning algorithm

4.1 Supervised learning with quantum feature space

Here, we denote $(\boldsymbol{a}, b)$ (or $(\boldsymbol{x}, y)$) as a pair of the datum $\boldsymbol{a}$ $(\boldsymbol{x})$ and the corresponding label b (y) in the

[1] Since any quantum state can be encoded as a ground state of some Hamiltonian, the presented conjecture uses the 'ground state' to substitute 'random quantum state' mentioned in Ref. [49]



$b(\boldsymbol{a})$
Experimental Results
Ground State Property $b(\boldsymbol{a})$
Material $\boldsymbol{H}(\boldsymbol{a})$
Physical Experiment at External Parameter $\boldsymbol{a}$
Collected Experimental Value $(\boldsymbol{a}, b)$
Predict Quantum Phase $y(\boldsymbol{x})$
$\vec{\boldsymbol{\alpha}}$
$\boldsymbol{H}(\boldsymbol{a})$
$U(\boldsymbol{\theta_a})\sim\mathcal{H}_{\mathcal{A}}$
QC
Training Parameter $\vec{\boldsymbol{\alpha}}$
$Q($ ■ , ● $)$
Ground states
$|\psi(\boldsymbol{a})\rangle = U(\boldsymbol{\theta_a})|\phi_0\rangle$
Quantum Kernel Matrix
Quantum Kernel Alphatron

Figure 2: The procedure of the proposed quantum learning algorithm. Here, $|\phi_0\rangle$ denotes the initial state of the quantum system.

training set $\mathcal{S}$ (testing set $\mathcal{T}$). Generally, the task of supervised learning is to learn a label y of the testing datum $\boldsymbol{x} \in \mathcal{T} \subset \mathcal{X}$ from a distribution $\mathcal{D}(\boldsymbol{x})$ defined on the space $\mathcal{X}$ according to some decision rule h. The decision rule h is assigned by a selected machine learning model from the training set $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^N$, where $\boldsymbol{a_i} \in \mathcal{X}$ follows distribution $\mathcal{D}(\boldsymbol{a_i})$, the label $b_i = h(\boldsymbol{a_i})$, and N is the size of the training set. Given the training set $\mathcal{S}$, an efficient learner needs to generate a classifier h in poly(N) time, to minimize the error

$$R(h) = \Pr_{\boldsymbol{x}\sim\mathcal{D}}[h(\boldsymbol{x}) \neq y]. \tag{2}$$

The datum $\boldsymbol{x}$ is sampled randomly according to $\mathcal{D}(\boldsymbol{x})$, in both the training and testing procedure, and the size N of the training set is polynomial in the data dimension.

The kernel method has played a crucial role in the development of supervised learning [51–53], which provides an approach to increase the expressivity and trainability of the original training set. We can describe a kernel function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ as $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \widetilde{\Psi}(\boldsymbol{x})^T \widetilde{\Psi}(\boldsymbol{x}')$, where $\widetilde{\Psi} : \mathcal{X} \to \mathcal{H}$ is the feature map which maps the datum $\boldsymbol{x} \in \mathcal{X}$ to a higher-dimensional space $\mathcal{H}$ (feature space). Tremendous classical kernel methods [52, 53] have been proposed to learn the non-linear functions or decision boundaries. With the rapid development of quantum computers, there is a growing interest in exploring whether the quantum kernel method can surpass the classical kernel [54–71]. Here we leverage the *quantum kernel* as our kernel function–$Q(\boldsymbol{x}, \boldsymbol{x}') = |\langle\psi(\boldsymbol{x})|\psi(\boldsymbol{x}')\rangle|^2$, where $|\psi(\boldsymbol{x})\rangle$ is the ground state of $H(\boldsymbol{x})$.

4.2 Quantum Kernel Alphatron

Here, we show the possibility of solving the LO-QPR problem with quantum data by leveraging the quantum kernel method combined with the Alphatron algorithm [52]. From the learning theory perspective, training can be phrased as the empirical risk minimization, and the associated learning model h^* for the minimise of the empirical risk follows the representer theorem.

Theorem 2 (Representer theorem [51]). *Let $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^N$ and corresponding feature states $\{|\psi(\boldsymbol{a}_i)\rangle\}_{i=1}^N$ be the training data set. $Q : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$ be a quantum kernel with the kernel space $\mathcal{H}$. Consider a strictly monotonic increasing regularisation function $g : [0, \infty) \mapsto \mathbb{R}$, and regularised empirical risk*

$$\hat{R}_L(h^*) = \frac{1}{N}\sum_{i=1}^N (h^*(\boldsymbol{a_i}) - b_i)^2 + g(\|h^*\|_{\mathcal{H}}). \tag{3}$$

Then any minimizer of the empirical risk $\hat{R}_L(h^*)$ admits a representation of the form $h^*(\boldsymbol{x}) = \sum_{i=1}^N \alpha_i Q(\boldsymbol{a_i}, \boldsymbol{x})$, where $\alpha_i \in \mathbb{R}$ for all $i \in \{1, 2, ..., N\}$, $\boldsymbol{x}$ and $\boldsymbol{a}$ are drawn from the same distribution.

According to the above theorem, one of the options to the quantum kernel is $Q(\boldsymbol{a_i}, \boldsymbol{x}) = |\langle\psi(\boldsymbol{a_i})|\psi(\boldsymbol{x})\rangle|^2$

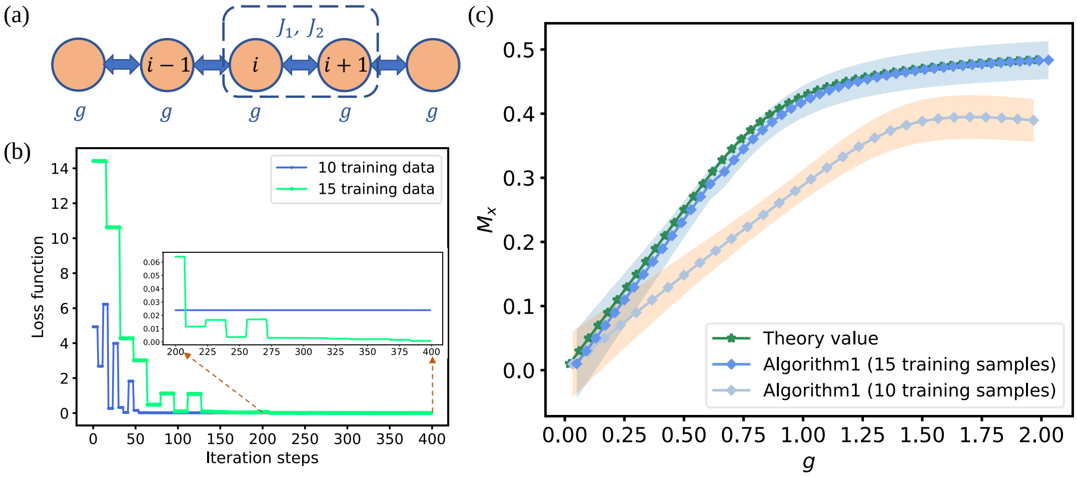


Figure 3: Numerical results for predicting ground-state properties in a $S = 1/2$ XXZ spin model with $n = 16$ qubits. (a) Illustration of the concerned spin model geometry. (b) The two curves depict the tendency of $R_L(h)$ in the training procedure, where the green (blue) curve represents the number of training data $N = 15$ ($N = 10$). (c) For fixed iterations (400), we randomly select the training data ($N = 10$ or $N = 15$) over 10 trials and plot the average performance by Alg. 1 in predicting the magnetization M_x.

Algorithm 1: Quantum Kernel Alphatron (QKA)

```
Input : training set
        $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^N \in \mathbb{R}^d \times [0,1]$,
        variational quantum circuit
        $U(\boldsymbol{\theta}) \in \mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta})$, quantum kernel
        function $\hat{Q}(\boldsymbol{a_i}.\boldsymbol{x}) = |\langle\psi(\boldsymbol{a_i})|\psi(\boldsymbol{x})\rangle|^2$,
        learning rate $\lambda > 0$, number of
        iterations $T$, testing data
        $\mathcal{T} = \{(\boldsymbol{x_j}, y_j)\}_{j=1}^M \in \mathbb{R}^d \times [0,1]$
Output: $\hat{h}^r$
for $i = 1, 2, ..., N$ do
    Prepare quantum state
      $|\psi(\boldsymbol{a_i})\rangle = U(\boldsymbol{\theta}_{\boldsymbol{a}_i})|0^{\otimes n}\rangle$, where $\boldsymbol{\theta}_{\boldsymbol{a}_i} :=$
      $\arg\min_{\boldsymbol{\theta}} \langle 0^{\otimes n}|U^\dagger(\boldsymbol{\theta})|H(\boldsymbol{a_i})|U(\boldsymbol{\theta})|0^{\otimes n}\rangle$;
$\boldsymbol{\alpha}^1 := 0 \in \mathbb{R}^N$;
for $t = 1, 2, ..., T$ do
    $\hat{h}^t(\boldsymbol{x}) := \sum_{i=1}^N \alpha_i^t \hat{Q}(\boldsymbol{a_i}, \boldsymbol{x})$;
    for $i = 1, 2, ..., N$ do
        $\alpha_i^{t+1} = \alpha_i^t + \frac{\lambda}{N}(b_i - \hat{h}^t(\boldsymbol{a_i}))$;
Let $r = \arg\min_{t\in\{1,...,T\}} \sum_{j=1}^M (\hat{h}^t(\boldsymbol{x_j}) - y_j)^2$;
return $\hat{h}^r$
```

where $|\psi(\boldsymbol{a}_i)\rangle$ represents the ground state of $H(\boldsymbol{a}_i)$, and unknown order parameter observable $\mathcal{M}$ thus can be represented as a linear combination of feature states, that is, $\mathcal{M} \approx \sum_i \alpha_i |\psi(\boldsymbol{a_i})\rangle\langle\psi(\boldsymbol{a_i})|$. Given the kernel matrix $\mathcal{Q} = [Q(\boldsymbol{a_i}, \boldsymbol{a}_j)]_{N\times N}$, the optimal weight parameters α_i in the expression of $\mathcal{M}$ has a closed-form solution by leveraging linear regression algorithms, and it requires $\mathcal{O}(N^{2.373})$ time complexity for solving such problem [73].

Here, we provide a more advanced method in learning α_i. From the learning theory perspective, training can be phrased as the empirical risk minimization, and the associated learning model $\hat{h}^*$ for the minimizer of the empirical risk can be learned as Alg. 1. As a result, given the quantum kernel matrix $\mathcal{Q}$, the LO-QPR problem can be solved in $T \times N = \mathcal{O}(N^{1.5})$ running time, as shown in Theorem 3. The outline of quantum learning algorithm is shown in Fig. 2.

Theorem 3. *Let quantum kernel* $Q(\boldsymbol{a_i}, \boldsymbol{x}) = |\langle\psi(\boldsymbol{a_i})|\psi(\boldsymbol{x})\rangle|^2$*, and* $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^N$ *be the training set such that*

$$\mathbb{E}[b_j|\boldsymbol{a_j}] = \sum_i \alpha_i |\langle\psi(\boldsymbol{a_i})|\psi(\boldsymbol{a_j})\rangle|^2 + g(\boldsymbol{a_j}), \tag{4}$$

$g : [0,\infty) \mapsto [-G, G]$ *is a strictly monotonic increasing regularisation function such that* $\mathbb{E}[g^2] \le \varepsilon_g$ *and* $\sum_{ij} \alpha_i\alpha_j |\langle\psi(\boldsymbol{a_i})|\psi(\boldsymbol{a_j})\rangle|^2 < B$*. Then for failure probability* $\delta \in (0,1)$*,* $\mathcal{O}\left(N^{5/2}\right)$ *copies of quantum states to estimate* $Q(\boldsymbol{a_i}, \boldsymbol{x})$*, Alg. 1 outputs a hypothesis* $\hat{h}^*$ *such that the training error* $\hat{R}_L(\hat{h}^*)$ *can be bounded by*

$$\mathcal{O}\left(\sqrt{\varepsilon_g} + G\sqrt[4]{\frac{\log(1/\delta)}{N}} + B\sqrt{\frac{\log(1/\delta)}{N}}\right) \tag{5}$$

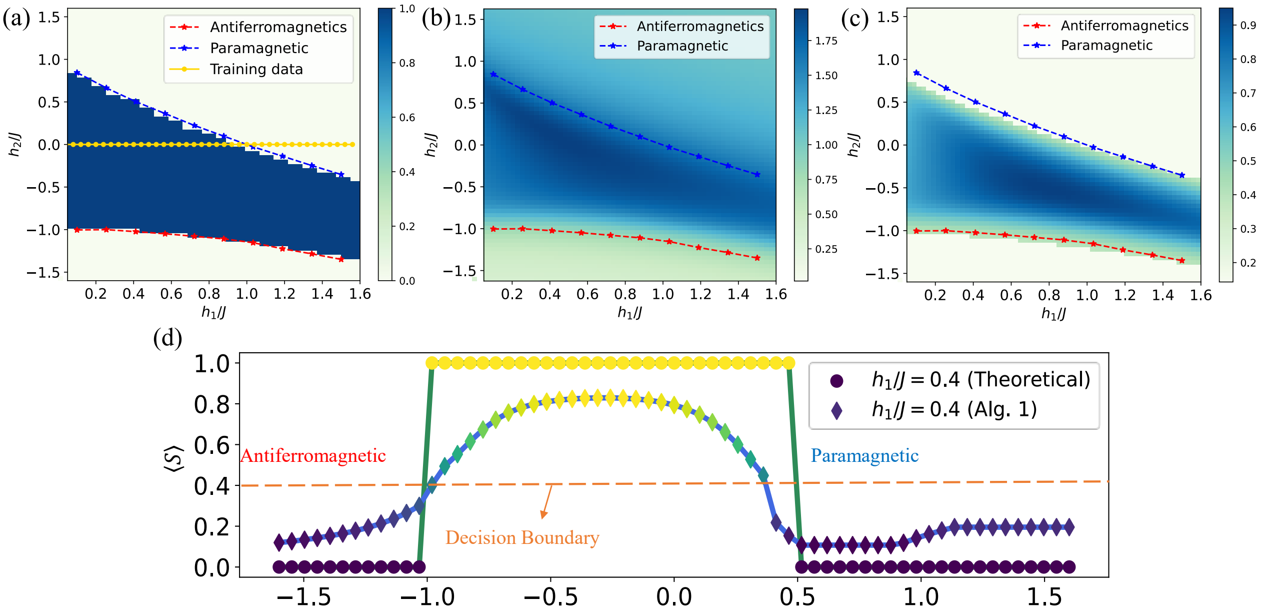


Figure 4: Numerical results for recognizing a $Z_2 \times Z_2$ Symmetry-Protected-Topological (SPT) phase of the Haldane Chain. (a) The exact phase diagram of the Haldane Chain (Eq. 8), where the phase boundary points (blue and red curves) are extracted from the Ref [35], and the background shading represents the phase function of $(h_1/J, h_2/J)$. The yellow line indicates 40 training points on the line $h_2 = 0$. (b) The classification result by using $N = 15$ training data. (c) The quantum phase classification result of Haldane Chain by using Alg. 1 with $N = 40$ training data. Here, the classification accuracy $v_s = 0.985$ which has a better performance compared to the 2-layer QCNN method [35] ($v_s = 0.971$, detail refers to SI material). (d) The quantum phase function at cross-section $h_1/J = 0.4$ of the Haldane Chain proposed by Alg. 1.

by selecting $\lambda = 1$, $T = \mathcal{O}(\sqrt{N/\log(1/\delta)})$ *and* $M = \mathcal{O}(N \log(T/\delta))$, *and* $R(\hat{h}^*)$ *(Eq. 2) can be upper bounded by*

$$R(\hat{h}^*) \leq \hat{R}_L(\hat{h}^*) + \mathcal{O}\left(\sqrt{\frac{\log(2/\delta)}{N}}\right). \qquad (6)$$

The essential difference between Alg. 1 and the original Alphatron algorithm is that we substitute the quantum kernel $Q(\boldsymbol{a_i}, \boldsymbol{x})$ into the classical kernel function. Although estimating the quantum kernel will introduce an additive error ϵ_q, we rigorously prove the robustness of Alg. 1 when encountering measurement errors. Specifically, utilizing $\mathcal{O}\left(N^{5/2}\right)$ copies of quantum states to estimate the quantum kernel function, the estimation error ϵ_q of $Q(\boldsymbol{a_i}, \boldsymbol{x})$ can be upper bounded by

$$\epsilon_q = \mathcal{O}\left(N^{-5/4}\sqrt{\log(1/\delta)}\right),$$

which provides the quantum learner $\mathcal{O}\left(N^{5/2}\right)$ quantum joint measurement overhead. This gives an upper bound of $\|h^t(\boldsymbol{x}) - \hat{h}^t(\boldsymbol{x})\|$ at the t-th iteration step, and the quantum empirical error $R(\hat{h})$ will saturate the upper bound as indicated in Eq. 5.

Remark (properties of Alg. 1) With the increase of training data scale N, Alg. 1 enables $\hat{R}(\hat{h^*})$ of QPR convergence to a low-level empirical risk, which is promised by Theorem 3. Specifically, if the quantum kernel matrix Q can be exactly calculated, then by Goel and Klivans [52], QKA will output a hypothesis h^* with an $\mathcal{O}(\sqrt[4]{\log(1/\delta)/N})$ empirical risk, where δ represents faliure probability. However, quantum kernel $Q(\boldsymbol{a_i}, \boldsymbol{x})$ is actually obtained by performing Destructive-Swap-Test algorithm [74] finite rounds, and the estimated $\hat{Q}$ has an additive error ϵ_q to the exact Q. In this quantum scenario, utilizing $\mathcal{O}\left(N^{5/2}\right)$ copies of quantum feature states to estimate the quantum kernel function suffices to provide an estimation of $Q(\boldsymbol{a_i}, \boldsymbol{x})$ with $\epsilon_q = \mathcal{O}\left(N^{-5/4}\sqrt{\log(1/\delta)}\right)$ additive error, and this leads the quantum empirical error $\hat{R}(\hat{h^*})$ to saturate $\mathcal{O}(\sqrt[4]{\log(1/\delta)/N})$ risk in $\mathcal{O}(\sqrt{N}t_Q)$ time complexity, where t_Q is the time required to compute kernel function $\hat{Q}$. The above procedure introduces $\mathcal{O}\left(N^{5/2}\right)$ quantum joint measurements overhead to the quantum learner, nevertheless, it provides a convincing performance for QKA. Further details are explained in the Appendix C.4.

5 Numerical Simulations

Given a general parameterized Hamiltonian $H(\boldsymbol{a})$, there only exists specific choices of parameters $\boldsymbol{a}$ for which the ground state $|\psi(\boldsymbol{a})\rangle$ can be classically solved. Thus the number of collected training data is often limited. In this paper, we explore the capability

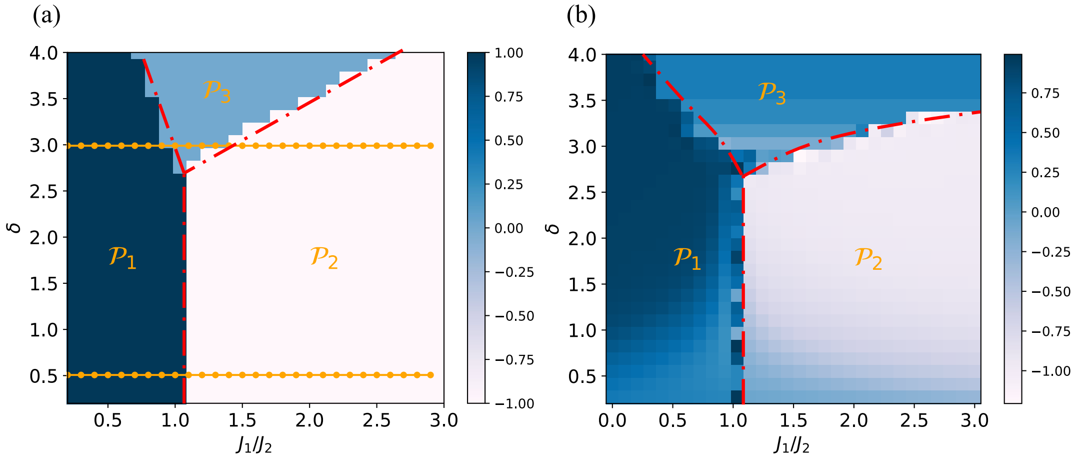


Figure 5: Numerical results for recognizing three distinct phases of bond-alternating XXZ model. (a) The system's three distinct phases are characterized by the topological invariant Z_R discussed in the Ref [72]. The invariant $Z_R = +1$ marks the Trivial phase $\mathcal{P}_1$, $Z_R = -1$ marks the Topological phase $\mathcal{P}_2$ and $Z_R = 0$ marks the Symmetry broken phase $\mathcal{P}_3$. Here, the (white and blue) background shading indicates the theoretical quantum phase value, and the red curves depict the quantum phase transition boundary. The two orange lines $\delta = 3.0$ and $\delta = 0.5$ represent 60 training points. (b) The quantum phase diagram predicted by the Alg. 1.

of QKA algorithm with small-scale of training data in recognizing quantum phases for several instances of LO-QPR tasks.

Firstly, we consider a warm-up case that detects the appearance of the staggered magnetization for the $S = \frac{1}{2}$XXZ spin chain in the Ising limit [75]. The Hamiltonian $H_w(g)$ is defined as

$$\sum_{i=1}^{n} J_1\left(S_i^x S_{i+1}^x + S_i^y S_{i+1}^y\right) + J_2 S_i^z S_{i+1}^z - g\sum_{i=1}^{n} S_i^x \tag{7}$$

where S_i^α is the α-component of the $S = 1/2$ spin operator at the i-th site, and g is the strength of the transverse field. The exchange coupling constant in xy plane is denoted by J_1 and that of the z-axis direction by J_2. Here, we set $J_1 = 0.2$, $J_2 = 1$ and depict the phase diagram $M_x = \langle X\rangle$ as a function of g (see the green curve in Fig. 3 (c)), where the expectation is under the ground state of Hamiltonian H_w. In this case, the number of qubits $n = 16$, the training data $\mathcal{S} = \{(g_i, M_x(g_i))\}_{i=1}^N$ where g_i is randomly sampled from the interval $[0, 2]$, and the testing data contains all the point from $\{g_t = 0.067t\}$ for $0 \leq t \leq 30$. The predictions proposed by Alg. 1 are illustrated in Fig. 3, which shows Alg. 1 yields higher accuracy prediction for the training set with more training data. This provides a simulation support for the theoretical bound in Eq. 5

Secondly, we consider a $Z_2 \times Z_2$ symmetry-protected topological (SPT) phase $\mathcal{P}$ which contains the $S = 1$ Haldane chain. The ground states $\{|\psi(h_1/J, h_2/J)\rangle\}$ belongs to a family of Hamiltonians

$$\begin{aligned} H_s(h_1/J, h_2/J) = &-J\sum_{i=1}^{n-2} Z_i X_{i+1} Z_{i+2} \\ &- h_1\sum_{i=1}^{n} X_i - h_2\sum_{i=1}^{n-1} X_i X_{i+1}, \end{aligned} \tag{8}$$

where X_i, Z_i are Pauli operators for the spin at site i, n is the number of spins, and h_1, h_2 and J are parameters of H_s. In Fig. 4 (a), the blue and red curves show the phase boundary points, and the background shading (colored tape) represents the phase diagram as a function of $\boldsymbol{x} = (h_1/J, h_2/J)$. When the parameter $h_2 = 0$, the ground states of H_s can be exactly solvable via the Jordan-Wigner transformation, and it can be efficiently detected by global order parameters whether these ground states belong to the SPT phase $\mathcal{P}$. Here, we utilize $N = 40$ data pairs $\{\boldsymbol{a} = (h_1/J, h_2/J), b\}$ as the training data, in which $h_2 = 0$ and b indicates phase value on $\boldsymbol{a}$ (see yellow points in Fig. 4 (a)). Our target is to identify whether a given, unknown ground state $|\psi(\boldsymbol{x})\rangle$ belongs to $\mathcal{P}$. In principle, the SPT phase $\mathcal{P}$ can be detected by measuring a non-local order parameter [42, 76] $S_{ij} = Z_i X_{i+1} X_{i+3} \ldots X_{j-3} X_{j-1} Z_j$, where X, Z are Pauli operators, indices $i < j$ and $i, j \in [n]$, n denotes the number of qubits in concerned Hamiltonian. Here, we choose $S = Z_1 X_1 \ldots X_{15} Z_{16}$ and utilize $N^{5/2} = 40^{2.5} \approx 10^5$ quantum measurement to estimate the quantum kernel function. The classification results for $n = 16$ and $N = 15, 40$ are illustrated as Fig. 4 (b) and (c) respectively, which show the performance on the approximation of the order parameters can be systematically improved by increasing the number of training samples. Fig. 4 (c) shows

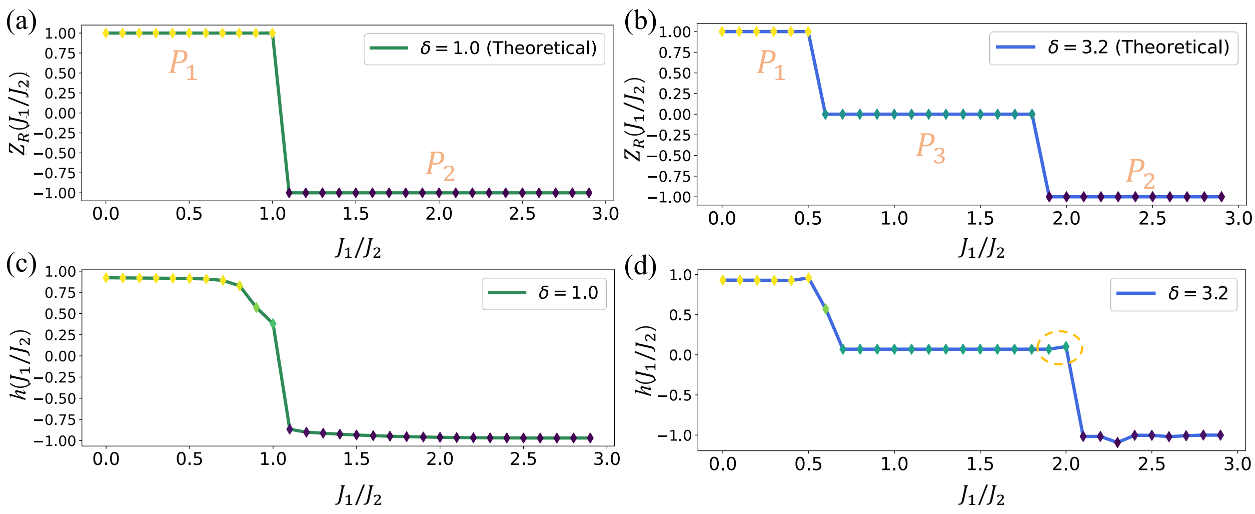


Figure 6: The function $Z_R(J_1/J_2)$ at cross sections $\delta = 3.2$ and $\delta = 1.0$ of the bond-alternating XXZ model phase diagram. Figures (a) and (b) are extracted from Fig. 5 (a) represent the theoretical value on the two lines, and Figures (c) and (d) are extracted from Fig. 5 (b) represent the predictive results. The dotted circle marks the mis-classification ground states.

that Alg. 1 can reproduce the phase diagram with high accuracy on $M = 4096$ testing points, where 61 points are mis-classified in the vicinity of paramagnetic boundary, and the classification accuracy $v_s = 0.985$ in this case.

Although the training data is only on the line with $h_2 = 0$, which can be classically simulated [35], a classical learner cannot learn from these data to predict the target quantum phase if the testing data satisfies Conjecture 1. However, we demonstrate that even if the training data can be classically simulated, quantum kernel Alphatron still works. Here, we provide more discussions on this result that shows training in classical data provides a predictive model for quantum points. The reason relies on that the order parameter observable is approximated by a linear combination of feature states in the training set, that is, $\mathcal{M} \approx \sum_i \alpha_i |\psi(\boldsymbol{a_i})\rangle\langle\psi(\boldsymbol{a_i})|$, and the prediction of the order parameter is significantly determined by the quality of training data. Although the training data are classically simulated, 40 ground states on line $h_2 = 0$ suffice to approximate an observable that accurately classifies the quantum phase transition boundary. A quantum learner then can utilize SWAP-test technique to efficiently estimate the quantum kernel $|\langle\psi(\boldsymbol{x})|\psi(\boldsymbol{a}_i)\rangle|^2$ and predict the quantum phase of $|\psi(\boldsymbol{x})\rangle$. In our model, we introduced a regularised term $g(\|h\|)$ to avoid over-fitting and to enhance its generalization ability. In other words, this model will not necessarily match all training data $\mathcal{S}$, but it has more generalized ability on the testing data $\mathcal{T}$. As a result, the qualitative domain walls are correctly mapped, but there exists some error in the vicinity of training data.

We also note that the quantum convolution neural network (QCNN) method [35] has been proposed to solve the same problem by applying a CNN quantum circuit to the quantum state. Given n-qubit ground states $|\psi(\boldsymbol{x})\rangle$, the QCNN method requires additionally $\mathcal{O}(\frac{7n}{2}(1-3^{1-d}) + n3^{1-d})$ multi-qubit operations and $4d$ single-qubit rotations to provide an output at depth d, and the proposed quantum kernel Alphatron requires additionally $\mathcal{O}(n)$ two-qubit operations to estimate the quantum kernel matrix. The measurement complexity of QCNN is determined by the number of iteration steps which is hard to be theoretically analyzed, however the quantum kernel Alphatron only requires $\mathcal{O}(N^2/\epsilon^2)$ quantum measurement in the whole training procedure. The comparison of the computational resources is summarized in Table 1.

Finally, we consider the bond-alternating XXZ model

$$H_b = \sum_{i:\text{even}} J_1 H_i + \sum_{i:\text{odd}} J_2 H_i, \tag{9}$$

where $H_i = X_i X_{i+1} + Y_i Y_{i+1} + \delta Z_i Z_{i+1}$, and J_1, J_2, δ are coupling parameters of H_b. The XXZ model has three different phases that can be detected by the topological invariant $Z_R(J_1/J_2, \delta)$ [72]. Here, we select totally $N = 60$ pairs $\{\boldsymbol{a} = (J_1/J_2, \delta), \boldsymbol{b} = Z_R(J_1/J_2, \delta)\}$ as the training data on the $\delta = 0.5, \delta = 3.0$ horizontal lines. In Fig. 5 (b) and Fig. 6, we utilize Alg. 1 to generate the phase diagram as a function of $\boldsymbol{x} = (J_1/J_2, \delta)$, where the colored shading background represents the phase classification results on a 16-qubit system. The data in phase diagram $\mathcal{P}_3$ is post-processed by the averaging scheme. The testing data contains 900 ground states, where 59 points are mis-classified in the vicinity of quantum phase transition boundary, and the classification accuracy $v_s = 0.934$.

Quantum Resources	QKA	QCNN
Quantum Gate Complexity	$\mathcal{O}(n)$	$\mathcal{O}(\frac{7n}{2}(1-3^{1-d})+n3^{1-d})$
Quantum Measurement Complexity	$\mathcal{O}(N^2/\epsilon^2)$, where $\epsilon = N^{-5/4}$	$\mathcal{O}(dNT/\epsilon^2)$
The number of iteration Steps T	$\mathcal{O}(\sqrt{N}\log(1/\delta))$	hard to analyze

Table 1: Quantum computational resources comparison between Quantum Kernel Alphatron (QKA) and Quantum Convolutional Neural Networks (QCNN).

6 The Power of Learning Algorithms

In this paper, relationships between four different machine learning classes in terms of the method that produces the training data and the learning algorithm are discussed.

- Here, "Q-Learning Alg." ("C-Learning Alg.") represents learning algorithms with quantum (classical) computer.
- "Q-Data" point $(\boldsymbol{a}, b)$ represents the property value b can be observed from physical experiments associated with the system $H(\boldsymbol{a})$, and "C-Data" point $(\boldsymbol{a}, b)$ represents the property b can be efficiently computed by some classical Turing machines given $\boldsymbol{a}$.

In this paper, a separation is proved between (C-Learning Alg. + C-Data) and (Q-Learning Alg. + Q-Data). Noting that the definition of C-Learning Alg. + C-Data is different to the 'classical' machine learning in Ref. [17], and the proposed theoretical result does not contradict to their statement.

(1) As shown in the Ref [77] (also see Appendix C.2), the power of C-Learning Alg. + C-Data will gradually enhance with the accumulating of training (advice) data, and the set of problems can be solved by classical learning algorithms is defined as the BPP/poly class. With the increase of the training data set, the learner will obtain more and more advicing data, and BPP/poly class will convergence to the P/poly class. It has been proved that the relationship BPP $\subseteq$ BPP/poly $\subseteq$ P/poly holds. Hence, a machine learning task where some data (even generated classically) is provided can be considerably different than commonly studied computational tasks. Classical learning algorithm with classical data (C-Learning Alg. + C-Data) has recently proven successful for many practical applications [78–82]. However, the direct application of these learning algorithms is challenging for intrinsically quantum problems. This is because the extremely large Hilbert space hinders the efficient translation of many-body problems into a classical learning framework. A natural question is thus raised–where is the limitation of such learning algorithms in quantum problems? Our first result rigorously proved that there exists a LO-QPR problem that cannot be solved by such classical learning algorithms under standard complexity assumptions (see Lemma 1 and Theorem 1).

(2) Quantum learning algorithm with quantum data (Q-Learning Alg. + Q-Data) is another significant theme in this paper. The Q-Learning Alg. can utilize quantum-enhanced feature spaces in the learning process, and ground states are provided by quantum circuits with polynomial circuit size. Here we select variational quantum circuits to provide the quantum-enhanced feature state. The proposed quantum learner first learns the order parameter from the training data $(\boldsymbol{a}, b)$, then utilizes this approximated order parameter to predict the quantum phase transition. Our second result claims that the LO-QPR problem can be efficiently solved by Q-Learning Alg. + Q-Data (see Alg. 1 and Theorem 3). These two results thus imply that "Q-Learning Alg. + Q-Data" is strictly stronger than "C-Learning Alg. + C-Data" with suitable complexity assumptions. This implies that the quantum kernel Alphatron with quantum data can efficiently solve the LO-QPR problem if there exists a quantum algorithm (or a quantum circuit) that provides the ground state of the concerned Hamiltonians.

(3) Another interesting class is classical learning algorithm with quantum data (C-Learning Alg. + Q-Data). Here, we briefly discuss a special learning scenario for the LO-QPR problem. Suppose the training data is provided by a quantum computer, and the quantum linear property is described by a nonlocal order parameter $\mathcal{M}$. In this setting, the training data has several choices, such as the external parameter and quantum phase $(\boldsymbol{a}, b)$ results from a quantum computer or the classical shadow representation used in [17]. Given an unknown ground state $|\psi(\boldsymbol{x})\rangle$, here we discuss whether a classical learner can predict its quantum phase. Similar to the quantum learning process, a possible follow-up classical learning steps are: (a) approximate the order parameter $\widetilde{\mathcal{M}} = \sum_i O_i$ from the quantum training data where O_i is the tensor product of Pauli operators; (b) Given the learned $\widetilde{\mathcal{M}}$, estimate the quantum phase value by using pre-obtained random computational basis measurements ($\{0,1\}^n$ bit-string samples) from the quantum state $|\psi(\boldsymbol{x})\rangle$. Under these two steps, can a classical learner solve the LO-QPR efficiently? The answer is yet unknown, but we are pessimistic about it. According to Theorem 2 in Ref. [18], at least $\Omega(3^{L(\widetilde{\mathcal{M}})}\|\widetilde{\mathcal{M}}\|_\infty/\epsilon^2)$ samples are needed to provide an approximation of $\langle\psi(\boldsymbol{x})|\widetilde{\mathcal{M}}|\psi(\boldsymbol{x})\rangle$ with additive error ϵ, where $\|\cdot\|_\infty$ denotes the spectral norm, and $L(\widetilde{\mathcal{M}})$ is the locality

of $\mathcal{M}$. Then in the worst case such that $L(\widetilde{\mathcal{M}}) = n$, it needs an exponential number of samples to obtain the order parameter. This example implies that a classical learner (without a quantum computer) might not efficiently predict a quantum phase transition phenomenon described by non-local order parameters. However, for this non-local order parameter $\mathcal{M}$, we can successfully learn it with Q-Learning Alg. + Q-Data.

(4) The last category is the quantum learning algorithm with classical data (Q-Learning Alg. + C-Data). Since the relationship BPP $\subseteq$ BQP holds[2], "Q-Learning Alg. + C-Data" could also be strictly stronger than "C-Learning Alg. + C-Data". In this paper, we provide simulation results on this scenario (see Fig. 4): quantum algorithm learns knowledge from classical data and can predict a classical-hardness quantum phase with high accuracy. Combined with the classical hardness results in Theorem 1, we conjecture that even if the training data are classically simulated, quantum learning algorithm might still have quantum advantages. We leave the rigours proof of this scenario to a future work.

We summarize the complexity relationship of these four categories in Fig. 7, where "Q-Learning Alg." refers to the use of a quantum computer, while "C-Learning Alg." relies only on a classical computer; "Q-Data" represents learning data directly observed from physical quantum experiments, while "C-Data" are efficiently producible by classical Turing machines.

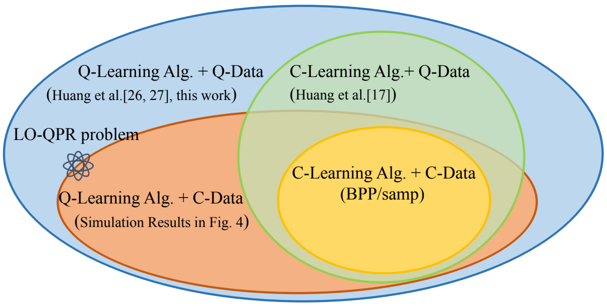


Figure 7: Visualization of the learning ability in terms of learning algorithms and data acquiring methods.

7 Conclusion

In this paper, we study the power of classical and quantum learning algorithms in solving LO-QPR problems. Specifically, we prove that under widely accepted assumptions, there exist some LO-QPR problems that cannot be efficiently solved by classical machine learning with classical data. We then prove that LO-QPR problems can be efficiently solved by leveraging the QKA algorithm with quantum data. Furthermore, we provided strong numerical evidence showing that the LO-QPR problems can be solved by the QKA algorithm with quantum data. In some cases, the QKA algorithm succeeded even with only classical data. Based on the above-mentioned theoretical and simulation results, we discussed the complexity relationships of four different machine learning classes in terms of the training data resources and the learning algorithm. We believe the proposed complexity classification helps us understand the power and limitation of classical and quantum learning algorithms.

This work leaves room for further research. For example, our numerical results witnessed the possibility of efficiently solving some LO-QPR problems by QML with classical data, then whether theoretical guarantees exist in showing that the LO-QPR problem belongs to the "Q-Learning Alg.+C-Data" class deserves to be further investigated. Finally, exploring the influences of noisy quantum channels on the effectiveness of quantum learning algorithms in solving LO-QPR would be important in practice.

Author Contributions

Y. Wu and B. Wu contributed equally to this work. All authors contributed to the discussion of results and writing of the manuscript.

Acknowledgement

We would like to thank Y. Song and P. Rebentrost to provide helpful suggestions for the manuscript. This work is supported by the China Scholarship Council (Grant No. 202006470011), the High-performance Computing Platform of Peking University, the National Natural Science Foundation of China (Grants No. 12175003, No. 12147133), and Zhejiang Lab's International Talent Fund for Young Professionals.

References

[1] WALTER Kohn. "Nobel lectures: Electronic structure of matter–wave functions and density functionals". Review of Modern Physics **71**, 1253 (1999).

[2] Steven R White. "Density matrix formulation for quantum renormalization groups". Physical Review Letters **69**, 2863 (1992).

[3] Steven R White. "Density-matrix algorithms for quantum renormalization groups". Physical Review B **48**, 10345 (1993).

[4] Federico Becca and Sandro Sorella. "Quantum monte carlo approaches for correlated systems". Cambridge University Press. (2017).

[2] Bounded error Probabilistic Polynomial time (BPP): the class of decision problems solvable by an NP oracle such that: if the answer is 'yes' then accept it with at least 2/3 probability, if the answer is 'no' then accept it with at most 1/3 probability.

[5] David Ceperley and Berni Alder. “Quantum monte carlo”. Science **231**, 555–560 (1986).

[6] WMC Foulkes, Lubos Mitas, RJ Needs, and Guna Rajagopal. “Quantum monte carlo simulations of solids”. Reviews of Modern Physics **73**, 33 (2001).

[7] J Carlson, Stefano Gandolfi, Francesco Pederiva, Steven C Pieper, Rocco Schiavilla, KE Schmidt, and Robert B Wiringa. “Quantum monte carlo methods for nuclear physics”. Reviews of Modern Physics **87**, 1067 (2015).

[8] Giuseppe Carleo and Matthias Troyer. “Solving the quantum many-body problem with artificial neural networks”. Science **355**, 602–606 (2017).

[9] Juan Carrasquilla and Roger G Melko. “Machine learning phases of matter”. Nature Physics **13**, 431–434 (2017).

[10] Ivan Glasser, Nicola Pancotti, Moritz August, Ivan D Rodriguez, and J Ignacio Cirac. “Neural-network quantum states, string-bond states, and chiral topological states”. Physical Review X **8**, 011006 (2018).

[11] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. “Neural-network quantum state tomography”. Nature Physics **14**, 447–450 (2018).

[12] Javier Robledo Moreno, Giuseppe Carleo, and Antoine Georges. “Deep learning the hohenberg-kohn maps of density functional theory”. Physical Review Letters **125**, 076402 (2020).

[13] Giacomo Torlai and Roger G Melko. “Learning thermodynamics with boltzmann machines”. Physical Review B **94**, 165134 (2016).

[14] Frank Schindler, Nicolas Regnault, and Titus Neupert. “Probing many-body localization with neural networks”. Physical Review B **95**, 245134 (2017).

[15] Eliska Greplova, Agnes Valenti, Gregor Boschung, Frank Schäfer, Niels Lörch, and Sebastian D Huber. “Unsupervised identification of topological phase transitions using predictive models”. New Journal of Physics **22**, 045003 (2020).

[16] Sebastian J Wetzel. “Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders”. Physical Review E **96**, 022140 (2017).

[17] Hsin-Yuan Huang, Richard Kueng, Giacomo Torlai, Victor V Albert, and John Preskill. “Provably efficient machine learning for quantum many-body problems”. Science **377**, eabk3333 (2022).

[18] Hsin-Yuan Huang, Richard Kueng, and John Preskill. “Predicting many properties of a quantum system from very few measurements”. Nature Physics **16**, 1050–1057 (2020).

[19] Yuxuan Du, Zhuozhuo Tu, Xiao Yuan, and Dacheng Tao. “Efficient measure for the expressivity of variational quantum algorithms”. Physical Review Letters **128**, 080506 (2022).

[20] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. “The power of quantum neural networks”. Nature Computational Science **1**, 403–409 (2021).

[21] Zoë Holmes, Kunal Sharma, Marco Cerezo, and Patrick J Coles. “Connecting ansatz expressibility to gradient magnitudes and barren plateaus”. PRX Quantum **3**, 010313 (2022).

[22] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. “Barren plateaus in quantum neural network training landscapes”. Nature Communications **9**, 1–6 (2018).

[23] Samson Wang, Enrico Fontana, Marco Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J Coles. “Noise-induced barren plateaus in variational quantum algorithms”. Nature communications **12**, 1–11 (2021).

[24] Kunal Sharma, Marco Cerezo, Lukasz Cincio, and Patrick J Coles. “Trainability of dissipative perceptron-based quantum neural networks”. Physical Review Letters **128**, 180505 (2022).

[25] Martin Larocca, Piotr Czarnik, Kunal Sharma, Gopikrishnan Muraleedharan, Patrick J Coles, and M Cerezo. “Diagnosing barren plateaus with tools from quantum optimal control”. Quantum **6**, 824 (2022).

[26] Hsin-Yuan Huang, Michael Broughton, Jordan Cotler, Sitan Chen, Jerry Li, Masoud Mohseni, Hartmut Neven, Ryan Babbush, Richard Kueng, John Preskill, et al. “Quantum advantage in learning from experiments”. Science **376**, 1182–1186 (2022).

[27] Hsin-Yuan Huang, Richard Kueng, and John Preskill. “Information-theoretic bounds on quantum advantage in machine learning”. Physical Review Letters **126**, 190505 (2021).

[28] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. “A rigorous and robust quantum speed-up in supervised machine learning”. Nature Physics **17**, 1013–1017 (2021).

[29] Daniel Stilck França and Raul Garcia-Patron. “Limitations of optimization algorithms on noisy quantum devices”. Nature Physics **17**, 1221–1227 (2021).

[30] Sergio Boixo, Sergei V Isakov, Vadim N Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J Bremner, John M Martinis, and Hartmut Neven. “Characterizing quantum supremacy in near-term devices”. Nature Physics **14**, 595–600 (2018).

[31] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak

Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. "Quantum supremacy using a programmable superconducting processor". Nature **574**, 505–510 (2019).

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

[33] Subir Sachdev. "Quantum phase transitions". Physics World **12**, 33 (1999).

[34] Bo-Xiao Zheng, Chia-Min Chung, Philippe Corboz, Georg Ehlers, Ming-Pu Qin, Reinhard M Noack, Hao Shi, Steven R White, Shiwei Zhang, and Garnet Kin-Lic Chan. "Stripe order in the underdoped region of the two-dimensional hubbard model". Science **358**, 1155–1160 (2017).

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

[36] Hsin-Yuan Huang, Richard Kueng, Giacomo Torlai, Victor V Albert, and John Preskill. "Supplementary materials for provably efficient machine learning for quantum many-body problems". Science **377**, eabk3333 (2022).

[37] Jonas Haferkamp, Philippe Faist, Naga BT Kothakonda, Jens Eisert, and Nicole Yunger Halpern. "Linear growth of quantum circuit complexity". Nature Physics **18**, 528–532 (2022).

[38] Marco Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. "Cost function dependent barren plateaus in shallow parametrized quantum circuits". Nature Communications **12**, 1–12 (2021).

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

[40] Bryan T Gard, Linghua Zhu, George S Barron, Nicholas J Mayhall, Sophia E Economou, and Edwin Barnes. "Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm". npj Quantum Information **6**, 1–9 (2020).

[41] F Duncan M Haldane. "Nonlinear field theory of large-spin heisenberg antiferromagnets: semiclassically quantized solitons of the one-dimensional easy-axis néel state". Physical Review Letters **50**, 1153 (1983).

[42] Frank Pollmann and Ari M Turner. "Detection of symmetry-protected topological phases in one dimension". Physical Review B **86**, 125441 (2012).

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

[44] Chris Cade, Lana Mineh, Ashley Montanaro, and Stasja Stanisic. "Strategies for solving the fermi-hubbard model on near-term quantum computers". Physical Review B **102**, 235122 (2020).

[45] Sam McArdle, Tyson Jones, Suguru Endo, Ying Li, Simon C Benjamin, and Xiao Yuan. "Variational ansatz-based quantum simulation of imaginary time evolution". npj Quantum Information **5**, 1–6 (2019).

[46] Weitang Li, Zigeng Huang, Changsu Cao, Yifei Huang, Zhigang Shuai, Xiaoming Sun, Jinzhao Sun, Xiao Yuan, and Dingshun Lv. "Toward practical quantum embedding simulation of realistic chemical systems on near-term quantum computers". Chemical Science (2022).

[47] Changsu Cao, Jiaqi Hu, Wengang Zhang, Xusheng Xu, Dechin Chen, Fan Yu, Jun Li, Han-Shi Hu, Dingshun Lv, and Man-Hong Yung. "Progress toward larger molecular simulation on a quantum computer: Simulating a system with up to 28 qubits accelerated by point-group symmetry". Physical Review A **105**, 062452 (2022).

[48] Sergey Bravyi, David Gosset, and Ramis Movassagh. "Classical algorithms for quantum mean values". Nature Physics **17**, 337–341 (2021).

[49] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. "On the complexity and verification of quantum random circuit sampling". Nature Physics **15**, 159–163 (2019).

[50] Tómas A Brody, Jorge Flores, J Bruce French, PA Mello, A Pandey, and Samuel SM Wong. "Random-matrix physics: spectrum and strength fluctuations". Reviews of Modern Physics **53**, 385 (1981).

[51] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. "Foundations of machine learning". MIT press. (2018). url: mitpress.mit.edu/9780262039406/foundations-of-machine-learning/.

[52] Surbhi Goel and Adam R Klivans. "Learning neural networks with two nonlinear layers in polynomial time". In Conference on Learning Theory. Pages 1470–1499. PMLR (2019).

[53] Adam Tauman Kalai and Ravi Sastry. "The isotron algorithm: High-dimensional isotonic regression.". In COLT. Citeseer (2009). url: www.microsoft.com/en-us/research/publication/isotron-algorithm-high-dimensional-isotonic-regression/.

[54] Maria Schuld and Nathan Killoran. "Quantum machine learning in feature hilbert spaces". Physical Review Letters **122**, 040504 (2019).

[55] Carsten Blank, Daniel K Park, June-Koo Kevin Rhee, and Francesco Petruccione. "Quantum

classifier with tailored quantum kernel". npj Quantum Information **6**, 41 (2020).

[56] Maria Schuld. "Supervised quantum machine learning models are kernel methods" (2021). arXiv:2101.11020.

[57] Ruslan Shaydulin and Stefan M Wild. "Importance of kernel bandwidth in quantum machine learning" (2021). arXiv:2111.05451.

[58] Junyu Liu, Francesco Tacchino, Jennifer R Glick, Liang Jiang, and Antonio Mezzacapo. "Representation learning via quantum neural tangent kernels". PRX Quantum **3**, 030323 (2022).

[59] Norihito Shirai, Kenji Kubo, Kosuke Mitarai, and Keisuke Fujii. "Quantum tangent kernel" (2021). arXiv:2111.02951.

[60] Kouhei Nakaji, Hiroyuki Tezuka, and Naoki Yamamoto. "Quantum-enhanced neural networks in the neural tangent kernel framework" (2021). arXiv:2109.03786.

[61] Sofiene Jerbi, Lukas J Fiderer, Hendrik Poulsen Nautrup, Jonas M Kübler, Hans J Briegel, and Vedran Dunjko. "Quantum machine learning beyond kernel methods". Nature Communications **14**, 517 (2023).

[62] Teresa Sancho-Lorente, Juan Román-Roche, and David Zueco. "Quantum kernels to learn the phases of quantum matter". Physical Review A **105**, 042432 (2022).

[63] Long Hin Li, Dan-Bo Zhang, and ZD Wang. "Quantum kernels with squeezed-state encoding for machine learning" (2021). arXiv:2108.11114.

[64] Louis-Paul Henry, Slimane Thabet, Constantin Dalyac, and Loïc Henriet. "Quantum evolution kernel: Machine learning on graphs with programmable arrays of qubits". Physical Review A **104**, 032416 (2021).

[65] Jonas Kübler, Simon Buchholz, and Bernhard Schölkopf. "The inductive bias of quantum kernels". Advances in Neural Information Processing Systems **34**, 12661–12673 (2021).

[66] Jennifer R Glick, Tanvi P Gujarati, Antonio D Corcoles, Youngseok Kim, Abhinav Kandala, Jay M Gambetta, and Kristan Temme. "Covariant quantum kernels for data with group structure" (2021). arXiv:2105.03406.

[67] Thomas Hubregtsen, David Wierichs, Elies Gil-Fuster, Peter-Jan HS Derks, Paul K Faehrmann, and Johannes Jakob Meyer. "Training quantum embedding kernels on near-term quantum computers". Physical Review A **106**, 042431 (2022).

[68] Stefan Klus, Patrick Gelß, Feliks Nüske, and Frank Noé. "Symmetric and antisymmetric kernels for machine learning problems in quantum physics and chemistry". Machine Learning: Science and Technology **2**, 045016 (2021).

[69] Xinbiao Wang, Yuxuan Du, Yong Luo, and Dacheng Tao. "Towards understanding the power of quantum kernels in the nisq era". Quantum **5**, 531 (2021).

[70] Riikka Huusari and Hachem Kadri. "Entangled kernels-beyond separability". The Journal of Machine Learning Research **22**, 1105–1144 (2021).

[71] Takeru Kusumoto, Kosuke Mitarai, Keisuke Fujii, Masahiro Kitagawa, and Makoto Negoro. "Experimental quantum kernel machine learning with nuclear spins in a solid" (2019). arXiv:1911.12021.

[72] Andreas Elben, Jinlong Yu, Guanyu Zhu, Mohammad Hafezi, Frank Pollmann, Peter Zoller, and Benoît Vermersch. "Many-body topological invariants from randomized measurements in synthetic quantum matter". Science Advances **6**, 3666 (2020).

[73] François Le Gall. "Powers of tensors and fast matrix multiplication". In Proceedings of the 39th international symposium on symbolic and algebraic computation. Pages 296–303. (2014).

[74] Juan Carlos Garcia-Escartin and Pedro Chamorro-Posada. "Swap test and hong-ou-mandel effect are equivalent". Physical Review A **87**, 052330 (2013).

[75] Yasuhiro Hieida, Kouichi Okunishi, and Yasuhiro Akutsu. "Anisotropic antiferromagnetic spin chains in a transverse field: Reentrant behavior of the staggered magnetization". Physical Review B **64**, 224422 (2001).

[76] Jutho Haegeman, David Pérez-García, Ignacio Cirac, and Norbert Schuch. "Order parameter for symmetry-protected phases in one dimension". Physical Review Letters **109**, 050402 (2012).

[77] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. "Power of data in quantum machine learning". Nature Communications **12**, 1–9 (2021).

[78] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. "Deep learning". Nature **521**, 436–444 (2015).

[79] Roxana Zeraati, Tatiana A Engel, and Anna Levina. "A flexible bayesian framework for unbiased estimation of timescales". Nature Computational Science **2**, 193–204 (2022).

[80] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need". Advances in neural information processing systems**30** (2017).

[81] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. "Mastering the game of go without human knowledge". Nature **550**, 354–359 (2017).

[82] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ron-

neberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. “Highly accurate protein structure prediction with alphafold”. Nature **596**, 583–589 (2021).

[83] Larry Stockmeyer. “On approximation algorithms for# p”. SIAM Journal on Computing **14**, 849–861 (1985).

[84] Sanjeev Arora and Boaz Barak. “Computational complexity: a modern approach”. Cambridge University Press. (2009).

[85] Larry Stockmeyer. “The complexity of approximate counting”. In Proceedings of the fifteenth annual ACM symposium on Theory of computing. Pages 118–126. (1983).

[86] Oded Goldreich. “Computational complexity: a conceptual perspective”. ACM Sigact News **39**, 35–39 (2008).

[87] Heiko Hoffmann. “Kernel pca for novelty detection”. Pattern Recognition **40**, 863–874 (2007).

[88] Patrick Rebentrost, Miklos Santha, and Siyi Yang. “Quantum alphatron” (2021). arXiv:2108.11670.

A Comparison to related works

Refs. [26, 27] focused on designing efficient measurement protocols to learn knowledge from an unknown density matrix, then predict its linear property by using accumulated measurement results, which is a learning analogue of shadow tomography problem as shown in Task 1.

Authors in [26, 27] proved that the entangled Bell measurement protocol can efficiently solve Task 1, meanwhile, it is classically hard in the *worst-case* scenario in estimating $\mathrm{Tr}\,(\rho Q)$ for some $Q \in \mathcal{P}$. Noting that the claimed quantum advantages might disappear without entangled measurement when Q represents a global observable. In contrast, the proposed quantum advantage in this paper does not depend on the entanglement of multiple copies of quantum states, while the power of quantum-enhanced feature space plays an essential role.

In Task 4, the training data set $\mathcal{S}$ only contains external parameters $\boldsymbol{a}$ and corresponding phase values b rather than the artificially designed density matrix. Furthermore, the order parameter observable $\mathcal{M}$ is unknown, and the learning protocol utilizes Quantum Kernel Alphatron (QKA) to approximate $\mathcal{M}$ by extracting abstract patterns from the data set $\mathcal{S} = \{(\boldsymbol{a}_{\boldsymbol{i}}, b_i)\}_{i=1}^{N}$, then using the approximated $\mathcal{M}$ to construct a prediction model $h(\boldsymbol{a})$. Then we rigorously proved that there exits a testing set $\mathcal{T} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$ such that $8/9$ of $y_i \in \mathcal{T}$ cannot be efficiently predicted by any classical ML algorithm under some standard complexity assumptions. In the table 2, we summarize the mainly differences between previous works and this paper.

Key Properties	This paper	Refs. [26, 27]
Training Data	External parameter $\boldsymbol{a}$ and phase value b	Artificial density matrix ρ
Prediction Task	Given new $\boldsymbol{a}^* \in \mathcal{X}$, predict b^*	Predict $\mathrm{Tr}\,(Q\rho)$
Related Observable	Unknown $\mathcal{M}$	A provided $Q \in \mathcal{P}$
Classical Hardness Result	Average-case hardness on $\mathcal{T}$	Worst-case hardness on Q
Quantum Learning Complexity	$\mathcal{O}(\sqrt{N}\log(1/\delta)/\epsilon^2)$	$\mathcal{O}(1/\epsilon^4)$
Source of Advantage	Quantum-enhanced feature space	Bell measurement

Table 2: Comparison between this paper and Refs. [26, 27].

From the above comparison, it is clear that previous approaches focus on learning from a single density matrix, but our paper learns patterns from a series of external parameters and their corresponding quantum phases. Then the Bell measurement methods in [26, 27] might not be directly applied to the problem studied in this paper.

Recently, Huang et al. [17] utilized an unsupervised learning method to learn samples from provided ground states, which can be summarized as Task 2. Then we show that the required sample complexity T by shadow-tomography based classical ML is expected to increase exponentially with respect to the system size, when the order parameter $\mathcal{M}$ performs on $\mathcal{O}(n)$ qubits. As discussed in [36], only a few LO-QPR problems determined by a global observable have a few-body observable approximation. Therefore, it is reasonable to consider a scenario where the shadow-tomography-formed training data is provided by a quantum computer, and the quantum phase transition can only be determined by a non-local order parameter $\mathcal{M}$. Given an unknown ground state $|\psi(\boldsymbol{x})\rangle$, here we discuss whether a classical learner can predict its quantum phase. Similar to the quantum learning process, possible follow-up classical learning steps are:

(a) Approximate the order parameter $\widetilde{\mathcal{M}} = \sum_i O_i$ from the quantum training data where O_i is the tensor product of Pauli operators;

(b) Given the learned $\widetilde{\mathcal{M}}$, estimate the quantum phase value by using pre-obtained random computational basis measurements ($\{0,1\}^n$ bit-strings) from the quantum state $|\psi(\boldsymbol{x})\rangle$.

Under these two steps, can a classical learner solve this LO-QPR efficiently? We are pessimistic about it. According to Theorem 2 in [18], at least $\Omega(3^{L(\widetilde{\mathcal{M}})}\|\widetilde{\mathcal{M}}\|_\infty/\epsilon^2)$ samples are needed to provide an approximation of $\langle\psi(\boldsymbol{x})|\widetilde{\mathcal{M}}|\psi(\boldsymbol{x})\rangle$ with additive error ϵ, where $\|\cdot\|_\infty$ denotes the spectral norm, and $L(\widetilde{\mathcal{M}})$ is the locality of $\mathcal{M}$. Then in the worst case such that $L(\widetilde{\mathcal{M}}) = n$, it is expected to require exponential number of samples to solve the LO-QPR problem. This example implies that a classical learner (without a quantum computer) might not efficiently predict a quantum phase transition phenomenon that only can be determined by non-local order parameters.

While our work utilizes quantum machine learning to extract high-level abstractions from observed data and directly process quantum ground states information by a quantum computer. Here, the ground state $|\psi(\boldsymbol{a})\rangle$ of $H(\boldsymbol{a})$ embeds classical external parameter $\boldsymbol{a}$ onto a specific quantum-enhanced feature space, where inner products of such quantum feature states give rise to a quantum kernel, a metric to characterize distances in the feature space. As a result, predicting the ground state property can be transformed into quantum state overlap computation, and thus bypasses the required exponential sample complexity.

Then we emphasize that the definition of “classical ML” is different to that in [17]. In our paper, we discuss the complexity relationship of four categories in terms of the method that produces the training data and the learning algorithm. Here, “Q-Learning Alg.” refers to the use of a quantum computer, while “C-Learning Alg.” relies only on a classical computer; “Q-Data” represents learning data directly observed from physical quantum experiments, while “C-Data” are efficiently producible by classical Turing machines. In this paper, a separation is proved between C-Learning Alg. + C-Data and Q-Learning Alg. + Q-Data.

Finally, we point that C-Learning Alg. + C-Data represents an nontrivial class. As shown in [77], the power of classical learning algorithms will gradually enhance with the accumulating of training (advice) data, and the set of problems can be solved by classical learning algorithms is defined as the BPP/poly class. With the increase of the training data set, the learner will obtain more and more advicing data, and BPP/poly class will be convergence to the P/poly class. It has been proved that the relationship BPP $\subseteq$ BPP/poly $\subseteq$ P/poly holds. Hence, a machine learning task where some data (even generated classically) is provided can be considerably different than commonly studied computational tasks. In our manuscript, we want to demonstrate quantum advantages by introducing quantum computational resources into learning algorithms, and our main contribution is to prove that there exists some LO-QPR problems cannot be efficiently solved by any ‘C-Learning Alg. + C-Data’, however, the ‘Q-Learning Alg. + Q-Data’ can efficiently solve this learning problem which thus illustrates quantum advantages.

B Construct Ground States with variational quantum circuit

Definition 1 (Architecture)**.** *An architecture $\mathcal{A}$ is a collection of directed acyclic graphs, one for each integer n. Each graph consists of $m < \mathrm{poly}(n)$ vertices, and the degree of each vertex v satisfies $\deg_{in}(v) = \deg_{out}(v) \in \{1, 2\}$.*

Definition 2 (Haar random circuit distribution)**.** *Let $\mathcal{A}$ be an architecture over circuits and let the gates in the architecture be $\{G_i\}_{i=1,\ldots,m}$. Define the distribution $\mathcal{H}_{\mathcal{A}}$ over circuits in $\mathcal{A}$ by drawing each gate G_i independently from the Haar measure.*

We first review the method on constructing a quantum random circuit. As presented in Ref [30], the quantum random circuit can be constructed in an iterative method in the realistic physical experiment. The construction starts with an initial layer of Hadamard gates to rotate the X basis, and the next D layers alternately insert controlled-Z (CZ) configurations. And one-qubit gates are also randomly sampled from the set $\{\mathrm{X}^{1/2}, \mathrm{Y}^{1/2}, \mathrm{T}\}$ and are placed between two CZ configurations. Theoretically, the *brickwork* architecture is also can be used to generate quantum random circuits. The brickwork is a kind of structure formed as follows: Perform a string of 2-qubit gates $U_1 \otimes U_2 \otimes \ldots \otimes U_{n/2}$ as the first layer, then perform a staggered string of gates, as illustrated in Fig. 2 (a) of the main file.

Definition 3 (Haar random quantum circuit)**.** *Let $\mathcal{A}$ be an architecture over circuits and let the gates in the architecture be $\{U_i\}_{i=1,..,R}$. Define the distribution $\mathcal{H}_{\mathcal{A}}$ over circuits in $\mathcal{A}$ by drawing each 2-qubit gate U_i independently from the Haar measure. Then construct the unitaries along the edges of $\mathcal{A}$, and each constructed circuit is defined as a Haar random quantum circuit.*

In the field of quantum computation, the variational quantum circuit is a popular method for approximating the ground state of $H(\boldsymbol{x})$. The key idea of using variational quantum circuit is that the parameterized quantum state $|\Psi(\boldsymbol{\theta})\rangle$ is prepared and measured on a quantum computer, and the classical optimizer updates the parameters $\boldsymbol{\theta}$ according to the measurement information. The quantum state $|\Psi(\boldsymbol{\theta})\rangle$ can be prepared by

$$|\Psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0^n\rangle = \prod_{d=1}^{D} U_d(\boldsymbol{\theta}_d)|0^n\rangle, \tag{10}$$

where $U(\boldsymbol{\theta})$ is composed of D unitaries $U_d(\boldsymbol{\theta}_d)$. Noting that the variational quantum circuit $U(\boldsymbol{\theta})$ has the same architecture to that of random circuit U, and two-qubit gates $U_i(\theta_i)$ are sampled from a subset of $SU(4)$. Then the relationship $\mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta}) \subseteq \mathcal{U}_{\mathcal{A}}$ holds, where $\mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta})$ and $\mathcal{U}_{\mathcal{A}}$ denote the set of $U(\boldsymbol{\theta})$ and random quantum circuit U based on $\mathcal{A}$, respectively.

Then we will show how to utilize one of the instances $U(\boldsymbol{\theta}) \in \mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta}) \subseteq \mathcal{U}_{\mathcal{A}}$ to generate ground states of a family of Hamiltonians $H(\boldsymbol{x})$ [45]. The ground state $|\psi(\boldsymbol{x})\rangle$ of $H(\boldsymbol{x})$ can be obtained from the imaginary time evolution, that is

$$|\psi(\boldsymbol{x})\rangle = \lim_{\beta\to\infty}|\eta(\beta, \boldsymbol{x})\rangle = \lim_{\beta\to\infty} A(\beta)e^{-\beta H(\boldsymbol{x})}|0^n\rangle, \tag{11}$$

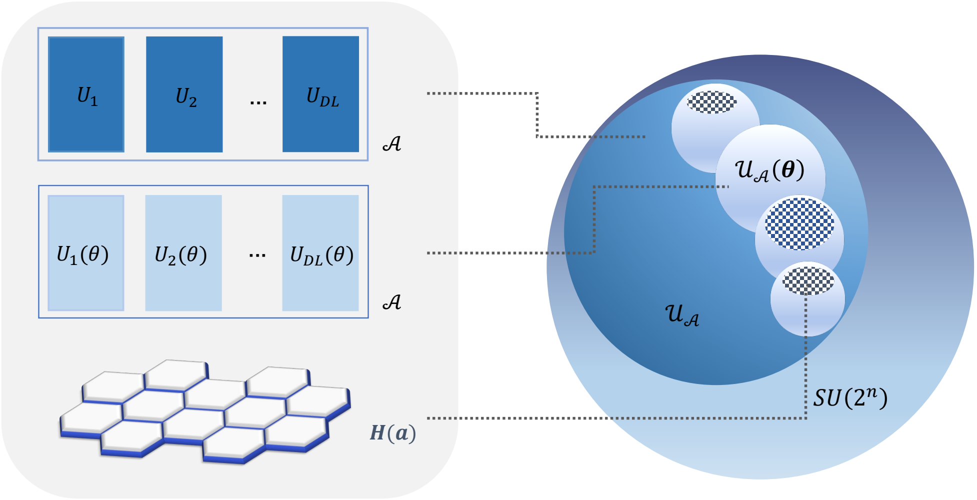


Figure 8: Visualization of the relationships between random quantum states set $\mathcal{U}_{\mathcal{A}}$, variational quantum states set $\mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta})$ and ground states from a family of Hamiltonian $H(\boldsymbol{x})$.

where β indicates the inverse temperature, $A(\beta) = 1/\sqrt{\langle\phi_0|e^{-2\beta H(\boldsymbol{x})}|0^n\rangle}$. If we consider the imaginary time evolution of the Schrödinger equation on the variational circuit state space, the parameter dynamics is governed by

$$\sum_i \frac{\partial|\Psi(\boldsymbol{\theta}(\beta))\rangle}{\partial\theta_i}\dot{\theta}_i = -\left(H(\boldsymbol{x}) - E_\beta(\boldsymbol{x})\right)|\Psi(\boldsymbol{\theta}(\beta))\rangle, \tag{12}$$

where the term $E_\beta(\boldsymbol{x}) = \langle\eta(\beta,\boldsymbol{x})|H(\boldsymbol{x})|\eta(\beta,\boldsymbol{x})\rangle$ and $\boldsymbol{\theta}(\beta)$ denotes the varational parameter in the circuit $U(\boldsymbol{\theta})$. Applying the McLachlan's variational principle to minimize the distance between the evolution of variational quantum state $\frac{\partial|\Psi(\boldsymbol{\theta}(\beta))\rangle}{\partial\beta}$ and $-(H(\boldsymbol{x}) - E_\beta(\boldsymbol{x}))|\Psi(\boldsymbol{\theta}(\beta))\rangle$, we have

$$\delta\left\|\left(\frac{\partial}{\partial\beta} + H(\boldsymbol{x}) - E_\beta(\boldsymbol{x})\right)|\Psi(\boldsymbol{\theta}(\beta))\rangle\right\| = 0, \tag{13}$$

and the evolution of parameters $\boldsymbol{\theta}(\beta)$ is obtained from the function

$$\sum_j A_{i,j}(\beta)\dot{\theta}_j = -C_i(\beta), \tag{14}$$

in which

$$\begin{aligned} A_{i,j}(\beta) &= \mathrm{Re}\left(\frac{\partial\langle\Psi(\boldsymbol{\theta}(\beta))|}{\partial\theta_i}\frac{\partial|\Psi(\boldsymbol{\theta}(\beta))\rangle}{\partial\theta_j}\right), \\ C_i(\beta) &= \mathrm{Re}\left(\frac{\partial\langle\Psi(\boldsymbol{\theta}(\beta))|}{\partial\theta_i}H(\boldsymbol{x})|\Psi(\boldsymbol{\theta}(\beta))\rangle\right). \end{aligned} \tag{15}$$

With the matrix $A(\beta)$ and vector $C(\beta)$, the quantum imaginary time evolution over a small interval $\delta\beta$ can be approximated by solving a linear system $\dot{\boldsymbol{\theta}}(\beta) = A^{-1}(\beta)C(\beta)$, and the variational parameter can be updated by

$$\boldsymbol{\theta}(\beta + \delta\beta) = \boldsymbol{\theta}(\beta) + A^{-1}(\beta)C(\beta)\delta\beta. \tag{16}$$

Given a large enough β, and repeat this procedure $N = \beta/\delta(\beta)$ times, the varational quantum state $|\Psi(\boldsymbol{\theta}(\beta)) = U(\boldsymbol{\theta}(\beta))|0^n\rangle$ will be an approximation of the ground state $|\psi(\boldsymbol{x})\rangle$, that is $|\psi(\boldsymbol{x})\rangle \approx U(\boldsymbol{\theta}(\beta))|0^n\rangle$. The relationships between unitary sets $\mathcal{U}_{\mathcal{A}}$, $\mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta})$ and ground states of $H(\boldsymbol{x})$ are visualized in Fig. 8.

C Proof of theorems

Here, we provide technical details for the proof of theorems in the main text.

C.1 Proof of Lemma 1

We first review several lemmas and assumptions which are closely related to our proof.

Lemma 2 (Stockmeyer Theorem [83])**.** *Given as input a function $f : \{0,1\}^n \mapsto \{0,1\}^m$ and any $y \in \{0,1\}^m$ there is a procedure that runs in randomized time $\mathrm{poly}(n, 1/\epsilon)$ with access to an* NP *oracle that outputs an α such that*

$$(1-\epsilon)p \leq \alpha \leq (1+\epsilon)p \tag{17}$$

for the value

$$p = \frac{1}{2^n}\sum_x f(x)$$

if the function f can be computed efficiently given x.

Conjecture 2 (Ref. [49])**.** *There exists an n-qubit quantum circuit U such that the following task is # P-hard: approximate $p_U(\boldsymbol{j}) = |\langle \boldsymbol{j}|U|0^n\rangle|^2$ to additive error $\epsilon_c/2^n$ with probability $\frac{3}{4} + \frac{1}{\mathrm{poly}(n)}$, where $\boldsymbol{j}$ is a $\{0,1\}^n$ bit string and $\epsilon_c = 1/\mathrm{poly}(n)$.*

Here, a candidate of the worst-case $U \in \mathbb{C}^{2^n \times 2^n}$ is a size $m \leq \mathrm{poly}(n)$ unitary where each basic gate is a two-qubit gate following some fixed gate position architecture $\mathcal{A}$. We denote this distribution as $\mathcal{H}_{\mathcal{A}}$. Note that the presented conjecture assets that it is $\#P$-hard to compute anything in an interval of radius $1/(2^n\mathrm{poly}(n))$ around the point $p_U(\boldsymbol{j})$ on the choice of U, however, Bouland et al. proved that it is $\#P$-hard to compute a truncated property $p_{U'}(\boldsymbol{j})$ which is close to $p_U(\boldsymbol{j})$ with an exponentially small error. Since this hardness interval is completely contained within the domain of conjectured hardness, their result is necessary for the conjecture. Therefore, if this conjecture holds, it implies computing most of $p_{U'}(\boldsymbol{j})$ is $\#P$-hard, and the worst and average-case quantum circuit instances share the same property in the architecture $\mathcal{A}$.

Proof of Lemma 1. Consider a family of Hamiltonian $\mathcal{H} = \{H(\boldsymbol{x})\}_{\boldsymbol{x}}$ that is invariant under the Clifford gate, that is $CHC^\dagger \in \mathcal{H}$ for any $H \in \mathcal{H}$. For a ground state $|\psi(\boldsymbol{x})\rangle$ of $H(\boldsymbol{x}) \in \mathcal{H}$ satisfies conjecture 1, we can project it to any computational basis $|\boldsymbol{j}\rangle$ with probability $p(\boldsymbol{j}) = |\langle \boldsymbol{j}|\psi(\boldsymbol{x})\rangle|^2$. The *hiding argument* shows that if one can approximate the probability $p(\boldsymbol{j})$, then one can approximate $p(0^n) = |\langle 0^n|\psi(\boldsymbol{x})\rangle|^2$. Therefore Conjecture 1 suggests that approximating the $p(\boldsymbol{j})$ to additive error $2^{-\mathrm{poly}(n)}$ is # P-hard.

Then we prove that for $\mathcal{M} \in \mathcal{P} = \{I, X, Y, Z\}^{\otimes n} \setminus I^{\otimes n}$, there exists a ground state $|\psi(\boldsymbol{y})\rangle$ of $H(\boldsymbol{y}) \in \mathcal{H}$ such that computing $\langle \psi(\boldsymbol{y})|\mathcal{M}|\psi(\boldsymbol{y})\rangle$ is classically hard. Consider the observable set $\{\mathcal{M}(\boldsymbol{s})|\mathcal{M}(\boldsymbol{s}) = Z_1^{s_1} \otimes \cdots \otimes Z_n^{s_n}\}$, where Z_k denotes Pauli-Z operator acts on the k-th qubit, and $\boldsymbol{s} = s_1 s_2 ... s_n \in \{0,1\}^n$. Then we have

$$o_{\boldsymbol{s}} = \langle \psi(\boldsymbol{x})|\mathcal{M}(\boldsymbol{s})|\psi(\boldsymbol{x})\rangle = \sum_{\boldsymbol{j}} p(\boldsymbol{j})(-1)^{\boldsymbol{j}\cdot\boldsymbol{s}}, \tag{18}$$

and $o_{\boldsymbol{s}}/2^n$ is the Fourier transformation of $p(\boldsymbol{j})$. Based on the algebra symmetry between $p(\boldsymbol{j})$ and $o_{\boldsymbol{s}}/2^n$, we have

$$p(\boldsymbol{j}) = \sum_{\boldsymbol{s}} o_{\boldsymbol{s}}(-1)^{\boldsymbol{j}\cdot\boldsymbol{s}}/2^n. \tag{19}$$

If $o_{\boldsymbol{s}}$ can be efficiently approximated by a classical computer given $\boldsymbol{s}$, there exists a $\mathrm{BPP}^{\mathrm{NP}^{\mathrm{BPP}}}$ algorithm that can approximate $p(\boldsymbol{j})$ with the multiplicative error $1/\mathrm{poly}(n)$ based on a theorem by Stockmeyer [83]. Considering $\mathrm{BPP} \subseteq \mathrm{P/poly}$ and approximating $p(\boldsymbol{j})$ is $\#P$-hard, these yield $\mathrm{P}^{\#\mathrm{P}} \subseteq \mathrm{BPP}^{\mathrm{NP}^{\mathrm{BPP}}} \subseteq \mathrm{BPP}^{\mathrm{NP}}/\mathrm{poly}$. Since $\mathrm{NP}^{\mathrm{NP}} \subseteq \mathrm{P}^{\#\mathrm{P}}$, one has $\mathrm{NP}^{\mathrm{NP}} \subseteq \mathrm{BPP}^{\mathrm{NP}}/\mathrm{poly}$, which implies PH collapses to the second level [84].

Therefore, there does not exist a classical algorithm that can efficiently calculate $o_{\boldsymbol{s}^*}$ for some $\boldsymbol{s}^* \in \{0,1\}^n$ based on the assumption that PH does not collapse and Conjecture 1 holds. Without loss of generality, let $\mathcal{M}(s^*) = Z_1^{s_1^*} \otimes Z_2^{s_2^*} \otimes \cdots \otimes Z_n^{s_n^*}$, and the interested physical order parameter observable $\mathcal{M}_t = P_1 \otimes \cdots \otimes P_n$ (for example, ferromagnetic parameter X or SPT parameter $Z_i X_{i+1} X_{i+3} ... X_{j-3} X_{j-1} Z_j$). Since a Clifford gate C maps a Pauli operator to another Pauli operator, then the target order parameter observable $\mathcal{M}_t$ can be expressed as $\mathcal{M}_t = (C)^\dagger\, Z_1^{s_1^*} \otimes Z_2^{s_2^*} \otimes \cdots \otimes Z_n^{s_n^*}\, (C)$, where C represents a Clifford gate. Therefore, for any target n-qubit Pauli observable $\mathcal{M}_t$, the expectation value

$$\langle \psi(\boldsymbol{x})|\mathcal{M}(\boldsymbol{s}^*)|\psi(\boldsymbol{x})\rangle = \langle \psi(\boldsymbol{x})|C\left(C^\dagger Z_1^{s_1^*} \otimes \cdots \otimes Z_n^{s_n^*} C\right)C^\dagger|\psi(\boldsymbol{x})\rangle = \langle \psi(\boldsymbol{y})|\mathcal{M}_t|\psi(\boldsymbol{y})\rangle. \tag{20}$$

The last equality is valid because $CH(\boldsymbol{x})C^{\dagger}$ belongs to the Hamiltonian family $\mathcal{H}$. If $|\psi(\boldsymbol{x})\rangle$ represents the ground state of $H(\boldsymbol{x})$, then $C|\psi(\boldsymbol{x})\rangle$ will be the ground state of $CH(\boldsymbol{x})C^{\dagger}$. Therefore, for any interested order parameter observable $\mathcal{M}_t \in \mathcal{P}$, there exists a ground state $|\psi(\boldsymbol{y})\rangle \in \mathcal{H}$ such that their corresponding quantum phase is classically hard. □

C.2 Complexity argument for the power of data

Here, we review the power of classical ML algorithms that can learn from data by means of a complexity class, which is defined as BPP/poly in Ref. [77]. A language L of bit strings is in BPP/poly if and only if the following holds. Suppose M and D are two probabilistic Turing machines, where D generates samples $\boldsymbol{x}$ with $|\boldsymbol{x}| = n$ in polynomial time for any size n and D defines a sequence of input distributions $\{D_n\}$. M takes an input $\boldsymbol{x}$ of size n along with a set $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{\mathrm{poly}(n)}$, where $\boldsymbol{x}_i$ is sampled from D_n using D and y_i indicates the corresponding label. If $\boldsymbol{x}_i \in L$, one has $y_i = 1$, else $y_i = 0$. Specifically, one requires:
(1) The probabilistic Turing machine M processes all inputs $\boldsymbol{x}$ in polynomial time.
(2) For all $\boldsymbol{x} \in L$, M outputs 1 with probability greater than 2/3.
(3) For all $\boldsymbol{x} \notin L$, M outputs 0 with probability less than 1/3.

From the above definition, we know that BPP is contained in this complexity class. Now we provide details on the separation between classical ML algorithms with classical data and BPP. Consider an undecidable language $L_h = \{1^n | n \in A\}$, where A is a subset of the natural numbers set, and consider a classically easy language $L_e \in$ BPP. Assuming that for any input size n, there exists an input $a_n \in L_e$ and an input $b_n \notin L_e$. Then a new language can be defined:

$$L = \bigcup_{n=1}^{\infty} \{x | \forall x \in L_e, 1^n \in L_h, |x| = n\} \cup \{x | \forall x \notin L_e, 1^n \notin L_h, |x| = n\}. \tag{21}$$

For each size n, if $1^n \in L_h$, the language L would include all $x \in L_e$ with $|x| = n$, otherwise, the language L would include all $x \notin L_e$ with $|x| = n$. That is to say, if one can decide whether a problem $x \in L$ for an input x using a classical algorithm, we can output whether $1^n \in L_h$ by checking whether $x \in L_e$. This is impossible since the language L_h is undicidable. Hence the language L is not in BPP class. On the other hand, if the training data $\{x_i, y_i\}$ are provided, where the label y_i represents whether x_i belongs to L, and we thus can decide whether 1^n belongs to L_h.

Based on the above discussion, we know that the power of classical learning algorithms will gradually enhance with the accumulating of training (advice) data, and the set of problems can be solved by classical learning algorithms is defined as the BPP/poly class. With the increase of the training data set, the learner will obtain more and more advicing data, and BPP/poly class will be convergence to the P/poly class. Hence, a machine learning task where some data is provided can be considerably different than commonly studied computational tasks. In our manuscript, we want to demonstrate quantum advantages by introducing quantum computational resources into learning algorithms, and our main contribution is to rigorously prove that any 'C-Learning Algorithm + C-Data' cannot solve the quantum phase learning problem. However, the 'Q-Learning Algorithm + Q-Data' can efficiently solve this learning problem which thus illustrates quantum advantages.

C.3 Proof of Theorem 1

The following lemma gives an average-case hardness for the quantum phase computation problem.

Lemma 3. *With the assumption that Conjecture 1 holds, and the PH in the computational complexity theory does not collapse, it is classically hard to approximate 8/9 of the quantum phase computation problem given a certain n-qubit Hamiltonian $H(\boldsymbol{a})$ with additive error $\epsilon = 1/(\mathrm{poly}(n))$, where its ground state $|\psi(\boldsymbol{a})\rangle = U(\vec{\boldsymbol{\theta}}(\boldsymbol{a}))|0^n\rangle$ and $U(\vec{\boldsymbol{\theta}}(\boldsymbol{a})) \in \mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta})$.*

Proof. Suppose we take a worst-case ground state $|\Psi\rangle = U(\vec{\boldsymbol{\theta}})|0^n\rangle$ of $H(\boldsymbol{a})$ generated by a variational quantum circuit $U(\vec{\boldsymbol{\theta}}) \in \mathcal{U}_{\mathcal{A}}(\boldsymbol{\theta})$, such that computing $p(\boldsymbol{j}) = |\langle \boldsymbol{j}|\Psi\rangle|^2$ to within additive error $2^{-\mathrm{poly}(n)}$ is $\#P$-hard (based on conjecture 1 in the main file). Since the two-qubit gate structures of $U(\vec{\boldsymbol{\theta}}) = U_{DL}(\boldsymbol{\theta}_{DL}) \cdots U_1(\boldsymbol{\theta}_1)$ is provided in $\mathcal{A}$, where $U_r(\boldsymbol{\theta}_r)$ denotes the r-th two-qubit gate for $r \in [DL]$, $\vec{\boldsymbol{\theta}} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{DL})$ and $\boldsymbol{\theta}_r \in \mathbb{R}^{15}$. Denote $R = DL$, each two-qubit gate

$$U(\boldsymbol{\theta}_r) = \exp\left(-i \sum_{j_1, j_2 = 0}^{4} \theta_r(j_1, j_2)\left(P_{j_1} \otimes P_{j_2}\right)\right) = \exp\left(-i\langle \boldsymbol{\theta}_r, \boldsymbol{P}_r\rangle\right), \tag{22}$$

where $P_j \in \{I, X, Y, Z\}$ and each $\theta_r(j_1, j_2) \in [0, 2\pi]$. Using Taylor series, one obtains

$$U(\vec{\boldsymbol{\theta}}) = \prod_{r=1}^{R} \sum_{k=0}^{\infty} \frac{(-i\langle \boldsymbol{\theta}_r, \boldsymbol{P}_r \rangle)^k}{k!}. \tag{23}$$

Denote

$$U(\boldsymbol{\theta}_r)_{\mathrm{tr}} = \sum_{k=0}^{K} \frac{(-i\langle \boldsymbol{\theta}_r, \boldsymbol{P}_r \rangle)^k}{k!}, \tag{24}$$

therefore $U(\boldsymbol{\theta}_r) - U(\boldsymbol{\theta}_r)_{\mathrm{tr}} = \sum_{k=K+1}^{\infty} \frac{(-i\langle \boldsymbol{\theta}_r, \boldsymbol{P}_r \rangle)^k}{k!}$. For arbitrary bit-string x, y, we can apply standard bound on Taylor series to bound $\|\langle x|U(\boldsymbol{\theta}_r) - U(\boldsymbol{\theta}_r)_{\mathrm{tr}}|y\rangle\|_1 \leq \kappa/K!$ for some constant κ. Therefore for an arbitrary observable $\mathcal{M}$

$$\begin{aligned}
&\langle 0^n|U^\dagger(\vec{\boldsymbol{\theta}})\mathcal{M}U(\vec{\boldsymbol{\theta}})|0^n\rangle = \sum_{i,j=0}^{2^n-1} \mathcal{M}_{ij} \langle 0^n|U^\dagger(\vec{\boldsymbol{\theta}})|i\rangle\langle j|U(\vec{\boldsymbol{\theta}})|0^n\rangle \\
&= \sum_{i,j=0}^{2^n-1} \mathcal{M}_{ij} \left(\sum_{\substack{y_1,y_2,\ldots y_{R-1}\in\{0,1\}^n \\ y_R=i}} \prod_{r=1}^{R} \langle 0^n|U(\boldsymbol{\theta}_r)|y_r\rangle \right) \left(\sum_{\substack{y_1,y_2,\ldots y_{R-1}\in\{0,1\}^n \\ y_R=j}} \prod_{r=1}^{R} \langle y_r|U(\boldsymbol{\theta}_r)|0^n\rangle \right) \\
&= \sum_{i,j=0}^{2^n-1} \mathcal{M}_{ij} \left(\sum_{\substack{y_1,y_2,\ldots y_{R-1}\in\{0,1\}^n \\ y_R=i}} \prod_{r=1}^{R} \langle 0^n| \sum_{k=0}^{\infty} \frac{(-i\langle \boldsymbol{\theta}_r, \boldsymbol{P}_r \rangle)^k}{k!} |y_r\rangle \right) \left(\sum_{\substack{y_1,y_2,\ldots y_{R-1}\in\{0,1\}^n \\ y_R=j}} | \prod_{r=1}^{R} \langle y_r| \sum_{k=0}^{\infty} \frac{(-i\langle \boldsymbol{\theta}_r, \boldsymbol{P}_r \rangle)^k}{k!} |0^n\rangle \right),
\end{aligned} \tag{25}$$

where the $y_1, .y_2, ...$ represent Feymann integration path. Since $\langle y_r|U(\boldsymbol{\theta}_r)|0^n\rangle$ can be approximated by a polynomial of degree K based on Taylor truncated method, the above expression can be rewritten by

$$\sum_{r,s=0}^{2^n-1} \mathcal{M}_{rs} \left(f_r(\boldsymbol{\theta}_1, ...\boldsymbol{\theta}_R) + \mathcal{O}\left(\frac{2^{Rn}}{(K!)^R}\right)\right) \left(f_s(\boldsymbol{\theta}_1, ...\boldsymbol{\theta}_R) + \mathcal{O}\left(\frac{2^{Rn}}{(K!)^R}\right)\right) \tag{26}$$

where f_r represents a polynomial of degree RK. Furthermore, $f_r(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_R)$ can be approximated by a low-degree function with at most $C_R^q(K)^q$ terms, where $q = \mathcal{O}(1)$ and C_R^q represents the combination number. Let $f_r(\vec{\boldsymbol{\theta}}) = \sum_{\vec{\boldsymbol{i}}} \boldsymbol{\alpha}_{\vec{\boldsymbol{i}}} \boldsymbol{\theta}_1^{i_1} \cdots \boldsymbol{\theta}_R^{i_R}$, where $\vec{\boldsymbol{i}} = (i_1, ..., i_R)$ and each $i_l \in [K]$, $l \in [R]$. For every term $\boldsymbol{\alpha}_{\vec{\boldsymbol{i}}} \boldsymbol{\theta}_1^{i_1} \cdots \boldsymbol{\theta}_v^{i_v}$ with $i_1 = \cdots = i_v = K$ and $v > q$, its corresponding parameter $\left|\boldsymbol{\alpha}_{\vec{\boldsymbol{i}}}\right| \leq 1/(K!)^q$ (based on Taylor series). Therefore, the relationship

$$\Delta f_r(\vec{\boldsymbol{\theta}}) = \left|f_r - \tilde{f}_r\right| = 2^{Rn} \left| \sum_{\vec{\boldsymbol{i}}} \boldsymbol{\alpha}_{\vec{\boldsymbol{i}}} \boldsymbol{\theta}_1^{i_1} \cdots \boldsymbol{\theta}_R^{i_R} - \sum_{j_1,\ldots,j_q \leq K-1} \boldsymbol{\alpha}_{\vec{\boldsymbol{j}}} \boldsymbol{\theta}_{j_1}^{j_1} \cdots \boldsymbol{\theta}_{j_q}^{j_q} \right| \leq 2^{Rn} \left(\frac{K^R}{(K!)^q}\right) \tag{27}$$

holds. Then let $q = \mathcal{O}(1)$, $R = \mathcal{O}(n^2)$, $K = \mathrm{poly}(n)$ and $K \gg R$, $\tilde{f}_r$ can provide an estimation to f_r within $2^{-\mathrm{poly}(n)}$ additive error, and $\tilde{f}_r$ only has $C_R^q(K)^q = \mathrm{poly}(n)$ terms. Then Eq. 26 can be represented by a muti-variable polynomial function $f(\vec{\boldsymbol{\theta}}, \mathcal{M})$ with R variables and at most $\mathrm{poly}(n)$ terms, and the relationship

$$\|\langle 0^n|U^\dagger(\vec{\boldsymbol{\theta}})\mathcal{M}U(\vec{\boldsymbol{\theta}})|0^n\rangle - f(\vec{\boldsymbol{\theta}}, \mathcal{M})\| \leq 2^{-\mathrm{poly}(n)} \tag{28}$$

holds.

Suppose the variational quantum circuit $U(\vec{\boldsymbol{\theta}})$ is powerful enough, such that it can calculate some ground states $\mathcal{G} = \{|\psi(\boldsymbol{a}_i)\rangle = U(\vec{\boldsymbol{\theta}}(\boldsymbol{a}_i))|0^n\rangle\}_{i=1}^{M}$ of the Hamiltonian $H(\boldsymbol{a})$. The 'worst-to-average-case' reduction can be achieved by *proof of contradiction*:

Since the variational quantum circuit can generate a ground state set $\mathcal{G}$ for a family of Hamiltonian $H(\boldsymbol{a})$, suppose there exists a classical algorithm $\mathcal{O}$, which can efficiently approximate $8/9$ of $\{\langle 0^n|U^\dagger(\vec{\boldsymbol{\theta}}(\boldsymbol{a}_i))\mathcal{M}U(\vec{\boldsymbol{\theta}}(\boldsymbol{a}_i))|0^n\rangle\}_{i=1}^{M}$, where $M = 100C_R^q(K)^q$. It implies that for at least $2/3$ choices of $\{U(\vec{\boldsymbol{\theta}}(\boldsymbol{a}_i))\}$, $\mathcal{O}$ correctly approximate $\{b_i = \langle 0^n|U^\dagger(\vec{\boldsymbol{\theta}}(\boldsymbol{a}_i))\mathcal{M}U(\vec{\boldsymbol{\theta}}(\boldsymbol{a}_i))|0^n\rangle\}$. According to the assumption in $\mathcal{G}$, the variational quantum circuit provides a map between $\boldsymbol{a}_i \mapsto \vec{\boldsymbol{\theta}}(\boldsymbol{a}_i)$. From Eq. 28, one can fit a polynomial function in $\boldsymbol{\theta}$ that recovers the value of $\langle 0^n|U^\dagger(\boldsymbol{\theta})\mathcal{M}U(\boldsymbol{\theta})|0^n\rangle$ by using $\{\vec{\boldsymbol{\theta}}(\boldsymbol{a}_i)\}$. However, according to the Lemma 1, successful approximating $\langle 0^n|U^\dagger(\boldsymbol{\theta})\mathcal{M}U(\boldsymbol{\theta})|0^n\rangle$ (worst-case scenario) by a BPP algorithm will yield PH collapse. Then, it is hard to approximate 8/9 of the $\{b_i\}$. Then the above $(\vec{\boldsymbol{\theta}}(\boldsymbol{a}_i), b_i)$ can be used in constructing a testing set

$$\mathcal{T} = \{(\boldsymbol{x}_i, y_i) || \psi(\boldsymbol{x}_i)\rangle \in \mathcal{G}, y_i = \langle \psi(\boldsymbol{x}_i)|\mathcal{M}|\psi(\boldsymbol{x}_i)\rangle\}, \tag{29}$$

where $|\psi(\boldsymbol{x}_i)\rangle = U(\vec{\boldsymbol{\theta}}(\boldsymbol{a}_i))|0^n\rangle$ and $y_i = b_i$.

□

Note. One might think that the above procedure could inspire a classical learning algorithm in predicting a hard quantum phase by using quantum data, however this cannot be directly used in solving LO-QPR problems. The reason is that the above procedure fits a polynomial function in $\boldsymbol{\theta}$ rather than the external parameter $\boldsymbol{a}$ in the training data set, then the proof is not sufficient in proving the efficiency of C-Learning Alg.+Q-Data on LO-QPRs determined by a global linear observable.

Proof of Theorem 1. Our proof depends on the quantum circuit representation of concerned ground states. We firstly provide the method on constructing a test set $\mathcal{T} = \{\boldsymbol{x_i}, y_i\}_{i=1}^M$. According to the construction in Sec. B, the variational quantum circuit state $|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))\rangle$ can approximate a ground state $|\psi(\boldsymbol{x})\rangle$ of the Hamiltonian $H(\boldsymbol{x})$. Starting from $|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))\rangle$, we want to approximate the ground state $|\psi(\boldsymbol{x}_i + \delta\boldsymbol{x})\rangle$ of $H(\boldsymbol{x}_i + \delta\boldsymbol{x})$ at external parameter $\boldsymbol{x}_i + \delta\boldsymbol{x}$ by $|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}) + \delta\boldsymbol{\theta})\rangle$. The value of $\delta\boldsymbol{\theta} = (\delta\boldsymbol{\theta}_1, \delta\boldsymbol{\theta}_2, ..., \delta\boldsymbol{\theta}_{DL})$ can be determined by minimizing the distance

$$\mathcal{L}(\delta\boldsymbol{\theta}) = \|\mathbf{d}|\psi(\boldsymbol{x}_i + \delta\boldsymbol{x})\rangle - \mathbf{d}|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}) + \delta\boldsymbol{\theta})\rangle\|, \tag{30}$$

where

$$\mathbf{d}|\psi(\boldsymbol{x}_i + \delta\boldsymbol{x})\rangle = |\psi(\boldsymbol{x}_i + \delta\boldsymbol{x})\rangle - |\psi(\boldsymbol{x}_i)\rangle, \tag{31}$$

and

$$\mathbf{d}|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}) + \delta\boldsymbol{\theta})\rangle = \sum_{d=1}^{DL} \frac{\partial|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))\rangle}{\partial\boldsymbol{\theta}_d}\delta\boldsymbol{\theta}_d, \tag{32}$$

and the notation $\|\cdot\|$ represents the fidelity norm. Then the function $\mathcal{L}^2(\delta\boldsymbol{\theta})$ can be further computed as

$$\begin{aligned} &\mathbf{d}\langle\psi(\boldsymbol{x}_i + \delta\boldsymbol{x})|\mathbf{d}|\psi(\boldsymbol{x}_i + \delta\boldsymbol{x})\rangle - \sum_{d=1}^{DL}\langle\psi(\boldsymbol{x}_i + \delta\boldsymbol{x})|\frac{\partial|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))\rangle}{\partial\boldsymbol{\theta}_d}\delta\boldsymbol{\theta}_d \\ &- \sum_{d=1}^{DL}\frac{\partial\langle\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))|}{\partial\boldsymbol{\theta}_d}|\psi(\boldsymbol{x}_i + \delta\boldsymbol{x})\rangle\delta\boldsymbol{\theta}_d + \sum_{m,s}\frac{\partial\langle\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))|}{\partial\boldsymbol{\theta}_m}\frac{\partial|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))\rangle}{\partial\boldsymbol{\theta}_s}\delta\boldsymbol{\theta}_m\delta\boldsymbol{\theta}_s. \end{aligned} \tag{33}$$

If we focus on the m-th variable $\delta\boldsymbol{\theta}_m$, the minimum of $\mathcal{L}^2(\delta\boldsymbol{\theta})$ obtains at

$$\sum_{s=1}^{DL} B_{s,m}\delta\boldsymbol{\theta}_m = E_m, \tag{34}$$

in which the parameter

$$B_{s,m} = \mathrm{Re}\left(\frac{\partial\langle\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))|}{\partial\boldsymbol{\theta}_s}\frac{\partial|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))\rangle}{\partial\boldsymbol{\theta}_m}\right), \tag{35}$$

and

$$E_m = \mathrm{Re}\left(\frac{\partial\langle\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))|}{\partial\boldsymbol{\theta}_m}\mathbf{d}|\psi(\boldsymbol{x}_i + \delta\boldsymbol{x})\rangle\right). \tag{36}$$

Once each elements are estimated, the variation of parameters $\delta\boldsymbol{\theta}$ can be efficiently computed by solving the linear system

$$B(\boldsymbol{\theta})\delta\boldsymbol{\theta} = E(\boldsymbol{\theta}), \tag{37}$$

where the matrix $B(\vec{\boldsymbol{\theta}}(\boldsymbol{x})) = (B_{s,m})_{DL\times DL}$ and $E(\vec{\boldsymbol{\theta}}(\boldsymbol{x})) = (E_1, ..., E_{DL})^T$. Since the matrix B is a real-valued symmetry matrix, the inverse of B must exist. And $\vec{\boldsymbol{\theta}}(\boldsymbol{x})$ can be updated by

$$\vec{\boldsymbol{\theta}}(\boldsymbol{x}) + \delta\boldsymbol{\theta} = \vec{\boldsymbol{\theta}}(\boldsymbol{x}) + B^{-1}(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))E(\vec{\boldsymbol{\theta}}(\boldsymbol{x})). \tag{38}$$

Then, the ground state $|\psi(\boldsymbol{x}_i + \delta\boldsymbol{x})\rangle$ can be approximated by $|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}) + \delta\boldsymbol{\theta})\rangle$. In this iterative method, one can construct a series of $(|\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))\rangle, |\Psi(\vec{\boldsymbol{\theta}}(\boldsymbol{x}) + \delta\boldsymbol{\theta})\rangle, ...)$ to represent the ground state $|\psi(\boldsymbol{x})\rangle, |\psi(\boldsymbol{x} + \delta\boldsymbol{x})\rangle, ...$ from a family of Hamiltonian $H(\boldsymbol{x})$, and this thus constructs a testing set $\mathcal{T}$ based on the Hamiltonian $H(\boldsymbol{x})$ and the architecture $\mathcal{A}$.

Now we only need to prove that there does not exist efficient classical ML algorithm that can predict y_i for $\boldsymbol{x_i} \in \mathcal{T}$ (constructed in Lemma 3) with probability 8/9. The basic idea relies on: if the classical ML can predict all $y_i \in \mathcal{T}$, then we can design an efficient classical algorithm to solve the worst-case hardness GLP problem, which results in a contradiction. Given the classical training set $\mathcal{S}$ (C-Data), the power of classical ML can be characterized as the BPP/samp class [77]. Suppose there exists a classical ML, which can efficiently predict 8/9 of $\{y_i = \langle\psi(\boldsymbol{x}_i)|\mathcal{M}|\psi(\boldsymbol{x}_i)\rangle\}_{i=1}^{M}$, where $M = 100KR$. Then given $\{(\vec{\boldsymbol{\theta}}(\boldsymbol{x}_i))\}_{i=1}^{M}$ and the vector $\vec{\boldsymbol{\theta}}(\boldsymbol{x})$ (parameter in the worst-case scenario), one can fit a polynomial function in $\vec{\boldsymbol{\theta}}$ that recovers the value of $\langle 0^n|U^\dagger(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))\mathcal{M}U(\vec{\boldsymbol{\theta}}(\boldsymbol{x}))|0^n\rangle$ which is the worst-case scenario. According to Lemma 1, an algorithm O that can approximate the worst-case scenario $\langle 0^n|U^\dagger(\vec{\boldsymbol{\theta}})\mathcal{M}U(\vec{\boldsymbol{\theta}})|0^n\rangle$ with $1/\text{poly}(n)$ additive error implies a $\text{BPP}^{\text{NP}^{\text{O}}}$ algorithm can approximate $p(\boldsymbol{j}) = |\langle\boldsymbol{j}|\Psi\rangle|^2 = |\langle\boldsymbol{j}|U(\vec{\boldsymbol{\theta}})|0^n\rangle|^2$ with the multiplicative error $1/\text{poly}(n)$ based on a theorem by Stockmeyer [83]. Therefore, if there exists a classical ML with classical data can efficiently predict 8/9 of $y_i \in \mathcal{T}$, this implies a $\text{BPP}^{\text{NP}^{\text{BPP/samp}}}$ algorithm that can approximate $p(\boldsymbol{j})$ with the multiplicative error. Considering BPP/samp $\subseteq$ P/poly and approximating $p(\boldsymbol{j})$ is $\#P$-hard, these yield

$$\text{P}^{\#\text{P}} \subseteq \text{BPP}^{\text{NP}^{\text{BPP/samp}}} \subseteq \text{BPP}^{\text{NP}}/\text{poly}.$$

Since $\text{NP}^{\text{NP}} \subseteq \text{P}^{\#\text{P}}$, one has $\text{NP}^{\text{NP}} \subseteq \text{BPP}^{\text{NP}}/\text{poly}$, which implies PH collapses to the second level [84]. Hence with the assumption that PH does not collapse, classical machine learning with classical resources cannot solve LO-QPRs even in the average-case scenario on $\mathcal{T}$. □

C.4 Proof of Theorem 3

Proof sketch of Theorem 3. Let $\boldsymbol{w} = \sum_{i=1}^{N}\alpha_i|\psi(\boldsymbol{a_i})\rangle \otimes |\psi(\boldsymbol{a_i})\rangle^*$, where $|\psi(\cdot)\rangle^*$ is the conjugate of $|\psi(\cdot)\rangle$, and the reproduced-kernel-feature-vector $\widetilde{\Psi}(\boldsymbol{a_i}) = |\psi(\boldsymbol{a_i})\rangle \otimes |\psi(\boldsymbol{a_i})\rangle^*$. Then

$$Q(\boldsymbol{a_i}, \boldsymbol{x}) = |\langle\psi(\boldsymbol{a_i})|\psi(\boldsymbol{x})\rangle|^2 = \langle\widetilde{\Psi}(\boldsymbol{a_i})|\widetilde{\Psi}(\boldsymbol{x})\rangle, \tag{39}$$

and

$$\sum_i \alpha_i |\langle\psi(\boldsymbol{a_i})|\psi(\boldsymbol{x})\rangle|^2 = \left\langle \boldsymbol{w}, \widetilde{\Psi}(\boldsymbol{x})\right\rangle, \tag{40}$$

by the definition of $\boldsymbol{w}$ and $|\Psi(\boldsymbol{a_i})\rangle$. Therefore, $\mathbb{E}\left[b_j|\boldsymbol{a_j}\right] = \langle\boldsymbol{w}, \Psi(\boldsymbol{x})\rangle + g\left(\boldsymbol{a_j}\right)$ and

$$\|\boldsymbol{w}\|^2 = \sum_{ij}\alpha_i\alpha_j|\langle\psi(\boldsymbol{a_i})|\psi(\boldsymbol{a_j})\rangle|^2 < B.$$

Hence, this theorem is followed by substituting the quantum kernel Q and feature map $\widetilde{\Psi}$ into Theorem 1 of Goel and Klivans [52], if we can implement Q perfectly. Nevertheless, we can only approximate it with small additive error via quantum circuit. Specifically, the quantum kernel can be approximated by independently performing the Destructive-Swap-Test [74] to $\mathcal{O}(\log(1/\delta)/\varepsilon^2)$ copies of $2n$-qubit state $|\psi\left(\boldsymbol{a_i}\right)\rangle \otimes |\psi\left(\boldsymbol{x}\right)\rangle$, with additive error ϵ_Q and failure probability δ, see SI material for the details of the circuit implementation of quantum kernel. If the quantum kernel $Q(\boldsymbol{a}_i, \boldsymbol{x})$ is estimated by performing Destructive-Swap-Test algorithm $\mathcal{O}\left(N^{5/2}\right)$ times, then

$$\left|h^t\left(\boldsymbol{x}\right) - \hat{h}^t\left(\boldsymbol{x}\right)\right| \leq \mathcal{O}\left(t^2\sqrt{\log\left(1/\delta\right)}/N^{5/4}\right) \tag{41}$$

holds with $1-\delta$ probability, where $h^t\left(\boldsymbol{x}\right)$, $\hat{h}^t\left(\boldsymbol{x}\right)$ represent the ideal QKA model and estimated QKA model at the t-th iteration step. □

The anticipated range of Eq. 41 matches perfectly to our simulation results, which can be checked from the generalized risk $\hat{R}_L(\hat{h}^*)$ at $t = T$, as demonstrated in Fig. 4 (b) and Supplementary material. For example, the theoretical upper bound in detecting SPT is $\hat{R}_L(\hat{h}^*) \leq 0.483$ $(N = 40, \delta = 0.1)$, and the numerical risk is lower than 0.483 after 15 iteration steps which is consistent with the anticipated bound.

Lemma 4 ([51])**.** *Fix a data distribution $(\boldsymbol{x}, y) \sim \mathcal{D}$, kernel function Q and training data size N. Then, with probability at least $1 - \delta$,*

$$R(h) \leq R_N(h) + \mathcal{O}\left(\sqrt{\frac{\log(2/\delta)}{N}}\right) \tag{42}$$

holds.

Proof. Notice that if the quantum kernel Q can be exactly calculated, then by Goel and Klivans [52], Quantum kernel Alphatron in the main file outputs a hypothesis h^* such that

$$R(h^*) \leq \mathcal{O}\left(\sqrt{\varepsilon_g} + G\sqrt[4]{\frac{\log(1/\delta)}{N}} + B\sqrt{\frac{\log(1/\delta)}{N}}\right).$$

This inequality is obtained by leveraging of

$$R(h^*) \leq \hat{R}\left(h^{t*}\right) + \mathcal{O}\left(B\sqrt{\frac{1}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right), \tag{43}$$

and

$$\hat{R}\left(h^{t*}\right) \leq \mathcal{O}\left(\sqrt{\varepsilon_g} + G\sqrt[4]{\frac{\log(1/\delta)}{N}} + B\sqrt{\frac{\log(1/\delta)}{N}}\right). \tag{44}$$

for some $t^* \leq T = \mathcal{O}\left(N/\log(1/\delta)\right)$. Nevertheless, if the quantum kernel Q is approximated via performing quantum circuits, Eq. (44) should be replaced with

$$\hat{R}\left(\hat{h}^{t*}\right) \leq \mathcal{O}\left(\sqrt{\varepsilon_g} + G\sqrt[4]{\frac{\log(1/\delta)}{N}} + B\sqrt{\frac{\log(1/\delta)}{N}}\right). \tag{45}$$

where $\hat{h}^t\left(\boldsymbol{x}\right) = \sum_{i=1}^m \alpha_i^t \hat{Q}\left(\boldsymbol{a_i}, \boldsymbol{x}\right)$, and $\hat{Q}$ is the approximation of Q.

In the following, we will prove that $\left|\hat{R}\left(\hat{h}^{t*}\right) - \hat{R}\left(h^{t*}\right)\right|$ is bounded, and hence $R\left(h^*\right)$ is bounded by combining Eq. (43), (44) and (45). By Theorem 3 in the main file, $\hat{Q}\left(\boldsymbol{a_i}, \boldsymbol{x}\right)$ is an ϵ_Q approximation of $Q\left(\boldsymbol{a_i}, \boldsymbol{x}\right)$, *i.e.*,

$$\left|\hat{Q}\left(\boldsymbol{a_i}, \boldsymbol{x}\right) - Q\left(\boldsymbol{a_i}, \boldsymbol{x}\right)\right| \leq \epsilon_Q \tag{46}$$

with high probability.

For convenience, in the later proof we require for all i, $\delta_Q^i = \hat{Q}\left(\boldsymbol{a_i}, \boldsymbol{x}\right) - Q\left(\boldsymbol{a_i}, \boldsymbol{x}\right)$ are the same, and $\delta_{\alpha_i}^t = \hat{\alpha}_i^t - \alpha_i^t$ are also the same, denoted them as $\delta_Q, \delta_\alpha^t$ respectively. Since δ_Q^i are in the same order for $i \in [N]$ ($\delta_{\alpha_i}^t$ similarly), hence it is reasonable for the assumptions. We will have the same upper bound for $R\left(h^*\right)$ without the assumptions and with a more tedious proof. Then for any i, we have

$$\begin{aligned}
&-\delta_\alpha^t = \alpha_i^t - \hat{\alpha}_i^t \\
&= \alpha_i^1 - \hat{\alpha}_i^1 + \frac{1}{N}\sum_{k=1}^{t-1}\left(\hat{h}^k\left(\boldsymbol{a_i}\right) - h^k\left(\boldsymbol{a_i}\right)\right) \\
&= \frac{1}{N}\sum_{k=1}^{t-1}\sum_{j=1}^{N}\left(\hat{\alpha}_j^k \hat{Q}\left(\boldsymbol{a_j}, \boldsymbol{a_i}\right) - \alpha_j^k Q\left(\boldsymbol{a_j}, \boldsymbol{a_i}\right)\right) \\
&= \sum_{k=1}^{t-1}\left(A^k \delta_Q + \bar{Q}_i \delta_\alpha^k + \delta_\alpha^k \delta_Q\right)
\end{aligned} \tag{47}$$

where $A^k = \frac{1}{N}\sum_{j=1}^N \alpha_i^k$, and $\bar{Q}_i = \frac{1}{N}\sum_{j=1}^N Q\left(\boldsymbol{a_j}, \boldsymbol{a_i}\right)$. We can also obtain the value of $-\delta_\alpha^{t-1}$ by leveraging of Eq. (47) and the recurrence relationship. The following equations follows by subtracting $-\delta_\alpha^{t-1}$ by $-\delta_\alpha^t$,

$$\delta_\alpha^t = \left(\bar{Q}_i - 1\right)\delta_\alpha^{t-1} + A^{t-1}\delta_Q + \delta_\alpha^{t-1}\delta_Q, \tag{48}$$

hence with the fact that $0 \leq \bar{Q}_i \leq 1$ and $A^k \leq \frac{k-1}{N}$, the absolute value of δ_α^t satisfies the inequality

$$\begin{aligned}\left|\delta_\alpha^t\right| &\leq (1+|\delta_Q|)\left|\delta_\alpha^{t-1}\right| + \frac{t-2}{N}|\delta_Q| \\ &= \left|\delta_\alpha^{t-1}\right| + \frac{3(t-2)}{N}\epsilon_Q\end{aligned}$$

By the recurrence of $|\delta_\alpha^t|$, we have

$$\begin{aligned}\left|\delta_\alpha^t\right| &\leq \frac{3\epsilon_Q}{N}\sum_{k=1}^{t-2} k \\ &\leq \frac{3t^2\epsilon_Q}{2N},\end{aligned}$$

then we have

$$\begin{aligned}\left|h^t(\boldsymbol{x}) - \hat{h}^t(\boldsymbol{x})\right| &= \left|\sum_{i=1}^N \left(\hat{\alpha}_i^t \hat{Q}(\boldsymbol{a_i},\boldsymbol{x}) - \alpha_i^t Q(\boldsymbol{a_i},\boldsymbol{x})\right)\right| \\ &\leq \sum_{i=1}^N \left(2\left|\alpha_i^t\right|\epsilon_Q + 2Q(\boldsymbol{a_i},\boldsymbol{x})\left|\delta_\alpha^t\right|\right) \\ &\leq 2N\left(\frac{t-1}{N}\epsilon_Q + \left|\delta_\alpha^t\right|\right) \\ &\leq 4t^2\epsilon_Q,\end{aligned}$$

for large where the second inequality holds since $|\alpha_i^t| \leq \frac{t-1}{N}$.

Therefore,

$$\begin{aligned}\left|\hat{R}\left(h^{t*}\right) - \hat{R}\left(\hat{h}^{t*}\right)\right| &\leq \frac{1}{N}\left|\sum_{i=1}^N \left(h^t(\boldsymbol{a_i}) - \hat{h}^t(\boldsymbol{a_i})\right)\right| \\ &\leq 4t^2\epsilon_Q,\end{aligned}$$

where the firstly inequality holds by the definition of $\hat{R}\left(h^{t*}\right)$ and $\hat{R}\left(\hat{h}^{t*}\right)$ (Recall that

$$\hat{R}\left(h^{t*}\right) = \frac{1}{N}\left\|\sum_{i=1}^N \left(\mathbb{E}\left[b_i|\boldsymbol{a_i}\right] - h^t(\boldsymbol{a_i})\right)|\phi(\boldsymbol{a_i})\rangle\right\|$$

). The additive error ϵ_Q for the quantum kernel $Q(\boldsymbol{a_i},\boldsymbol{x})$ can be bounded to $\mathcal{O}\left(\frac{\sqrt{\log(1/\delta)}}{N^{5/4}}\right)$ with $\mathcal{O}\left(N^{5/2}\right)$ copies of the quantum states.

Hence,

$$\begin{aligned}R(h^*) &\leq \mathcal{O}\left(\sqrt{\varepsilon_g} + G\sqrt[4]{\frac{\log(1/\delta)}{N}} + B\sqrt{\frac{\log(1/\delta)}{N}} + \frac{\log(1/\delta)}{N^{1/4}}\right) \\ &\leq \mathcal{O}\left(\sqrt{\varepsilon_g} + G\sqrt[4]{\frac{\log(1/\delta)}{N}} + B\sqrt{\frac{\log(1/\delta)}{N}}\right),\end{aligned}$$

where the last inequality holds since $t = \mathcal{O}(N/\log(1/\delta))$ and $G = \Omega(1)$. Combine the above inequality to Lemma 4, the upper bound of generalize error is obtained. □

D Implementation of quantum kernel with SWAP test

By leveraging of Chernoff bound, the quantum kernel can be approximated by independently performing the Destructive-Swap-Test [74] to $\mathcal{O}(\log(1/\delta)/\epsilon_Q^2)$ copies of $2n$-qubit state $|\phi(\boldsymbol{a_i})\rangle \otimes |\phi(\boldsymbol{x})\rangle$, with additive error ϵ_Q and failure probability δ. The expectation of the measurement results of the Destructive-Swap-Test is

$$\langle\phi(\boldsymbol{a_i}) \otimes \phi(\boldsymbol{x})|\mathbf{SWAP}|\phi(\boldsymbol{a_i}) \otimes \phi(\boldsymbol{x})\rangle = Q(\boldsymbol{a_i},\boldsymbol{x}), \tag{49}$$

where $\mathbf{SWAP}|\phi(\boldsymbol{a_i}) \otimes \phi(\boldsymbol{x})\rangle = |\phi(\boldsymbol{x}) \otimes \phi(\boldsymbol{a_i})\rangle$ denotes the $2n$-qubit swap operator. For QPL problem, $|\phi(\boldsymbol{a_i})\rangle$ and $|\phi(\boldsymbol{x})\rangle$ can all be generated with polynomial-size circuit, hence the Destructive-Swap-Test can be performed efficiently.

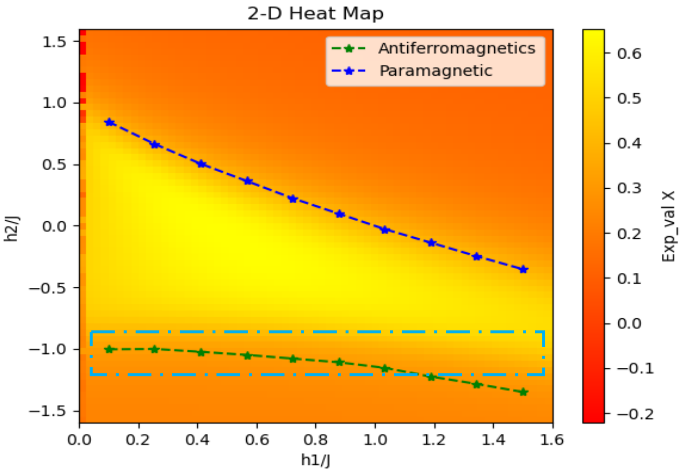


Figure 9: Numerical results for QCNN to recognize a $Z_2 \times Z_2$ Symmetry-Protected-Topological (SPT) phase of Haldane Chain by using code [?].

E Discussions on sample complexity of ground states

Here, we carried out further theoretical analysis and numerical calculation on the distribution probabilities of ground states $|\psi_g(\boldsymbol{x})\rangle$ of parameterized Hamiltonian $H(\boldsymbol{x})$. We aim to provide a numerical window of $\boldsymbol{x}$, for which the corresponding Hamiltonian simulation is expected to be classically hard.

The probability distribution of a truly random quantum state $|\psi\rangle$ possesses the Porter-Thomas (PT) distribution $\Pr(|\langle \boldsymbol{j}|\psi\rangle|^2) = 2^n e^{-2^n|\langle \boldsymbol{j}|\psi\rangle|^2}$, which is known to be classically hard to sample [30, 50]. In the following we compare the distribution probabilities of ground states $|\psi_g(\boldsymbol{x})\rangle$ of parameterized Hamiltonian $H(\boldsymbol{x})$ with the Porter-Thomas distribution.

Theorem 4. *Suppose $p_{\boldsymbol{x}}(\boldsymbol{j}) = |\langle \boldsymbol{j}|\psi_g(\boldsymbol{x})\rangle|^2$ and $p(\boldsymbol{j}) = |\langle \boldsymbol{j}|\psi\rangle|^2$ represent probability distributions of ground state $|\psi_g(\boldsymbol{x})\rangle$ and some random state $|\psi\rangle$, and their trace distance satisfies*

$$\mathrm{Tr}\left(\Pr(p_{\boldsymbol{x}}(\boldsymbol{j})), \Pr(p(\boldsymbol{j}))\right) = \frac{1}{2}\sum_{\boldsymbol{j}=0}^{2^n-1} |\Pr(p_{\boldsymbol{x}}(\boldsymbol{j})) - \Pr(p(\boldsymbol{j}))| < \epsilon. \tag{50}$$

If the trace distance $\epsilon \le n^{-1}$, then it is classically hard to sample from the ground state $|\psi_g(\boldsymbol{x})\rangle$.

Proof Sketch: Here, we provide a proof by contradiction. On one hand, it is known $\#P$-hard to approximate $p(\boldsymbol{j})$ to additive error $\mathcal{O}(1/(n2^n))$ with a constant probability [49]. On the other hand, if $\epsilon \le n^{-1}$ and assume that there exists a classical sample algorithm $\mathcal{A}$ that can efficiently sample from $|\psi_g(\boldsymbol{x})\rangle$, then $\mathcal{A}$ can be used to efficiently estimate $p(\boldsymbol{j})$ to additive error $\mathcal{O}(1/(n2^n))$ with probability $\frac{1}{4}$. This leads to a contradiction. Therefore, if $\epsilon \le n^{-1}$, such classical sample algorithm $\mathcal{A}$ does not exist.

Proof. Assume that there exists a classical sample algorithm $\mathcal{A}$ that can efficiently sample from the ground state $|\psi_g(\boldsymbol{x})\rangle$. Let $(\boldsymbol{j}_1, \boldsymbol{j}_2, ...)$ be samples generated by the classical algorithm $\mathcal{A}$, then an approximation of $\Pr(p_{\boldsymbol{x}}(j))$ can be obtained by using Stockmeyer Counting theorem [30, 85, 86], which is denoted as $\Pr(\hat{p}_{\boldsymbol{x}}(\boldsymbol{j}))$. Then we have

$$\begin{aligned} |\Pr(\hat{p}_{\boldsymbol{x}}(\boldsymbol{j})) - \Pr(p(\boldsymbol{j}))| &\le |\Pr(\hat{p}_{\boldsymbol{x}}(\boldsymbol{j})) - \Pr(p_{\boldsymbol{x}}(\boldsymbol{j}))| + |\Pr(p_{\boldsymbol{x}}(\boldsymbol{j})) - \Pr(p(\boldsymbol{j}))| \\ &\le \frac{\Pr(p_{\boldsymbol{x}}(\boldsymbol{j}))}{\mathrm{poly}(n)} + |\Pr(p_{\boldsymbol{x}}(\boldsymbol{j})) - \Pr(p(\boldsymbol{j}))| \\ &\le \left(1 + \frac{1}{\mathrm{poly}(n)}\right)|\Pr(p_{\boldsymbol{x}}(\boldsymbol{j})) - \Pr(p(\boldsymbol{j}))| + \frac{\Pr(p(\boldsymbol{j}))}{\mathrm{poly}(n)} \end{aligned} \tag{51}$$

where the second inequality comes from the Stockmeyer Counting theorem and the third inequality comes from the triangle inequality. According to Markov's inequality, one has

$$\Pr\left(|\Pr(p_{\boldsymbol{x}}(\boldsymbol{j})) - \Pr(p(\boldsymbol{j}))| \le \frac{2\epsilon}{2^n\delta}\right) \ge 1 - \delta, \tag{52}$$

where $\delta \in [0,1]$. Setting $\delta = \alpha\epsilon$, where α is a positive constant value, the relationship

$$|\Pr(\hat{p}_{\boldsymbol{x}}(\boldsymbol{j})) - \Pr(p(\boldsymbol{j}))| \leq \frac{1 + 1/\text{poly}(n)}{\alpha 2^{n-1}} + \frac{\Pr(p(\boldsymbol{j}))}{\text{poly}(n)} \tag{53}$$

is valid with the probability $1 - \alpha\epsilon$. If $\epsilon < \alpha^{-1}$ (which means the above estimation is valid), an estimation of $\Pr(p(\boldsymbol{j}))$ is obtained, that is

$$\frac{\Pr(\hat{p}_{\boldsymbol{x}}(\boldsymbol{j})) - a}{1+b} \leq \Pr(p(\boldsymbol{j})) \leq \frac{\Pr(\hat{p}_{\boldsymbol{x}}(\boldsymbol{j})) - a}{1+b} + \epsilon_1 \tag{54}$$

where $a = (1 + 1/\text{poly}(n))/(\alpha 2^{n-1})$, $b = 1/\text{poly}(n)$ and $\epsilon_1 = 2a/(1-b^2)$. Since $p(\boldsymbol{j})$ satisfies PT distribution, then we have

$$\frac{n - \log(l(\boldsymbol{x}) + \epsilon_1)}{2^n} \leq p(\boldsymbol{j}) \leq \frac{n - \log(l(\boldsymbol{x}) + \epsilon_1)}{2^n} + \frac{\epsilon_2}{2^n}, \tag{55}$$

where $l(\boldsymbol{x}) = \frac{\Pr(\hat{p}_{\boldsymbol{x}}(\boldsymbol{j}))-a}{1+b}$ and $\epsilon_2 = \log(1+\epsilon_1/l(\boldsymbol{x})) = \log(1 + 2a/(1+b)(\Pr(\hat{p}_{\boldsymbol{x}}(\boldsymbol{j})) - a))$. For the random quantum state $|\psi\rangle$, the expectation value of $\Pr(p(\boldsymbol{j}))$ is upper bounded by $2^n \int_0^1 pe^{-2^n p} dp \leq 2^{-n}$ [30]. Then using Markov's inequality, we have $\Pr(p(\boldsymbol{j})) \geq 1/2^n$ with probability $1/e$. Therefore $\Pr(\hat{p}_{\boldsymbol{x}}(\boldsymbol{j})) \geq \Pr(p(\boldsymbol{j})) - 1/(\alpha 2^n) \geq 1/2^n - 1/(\alpha 2^n)$ with probability $1/e$.

Taking everything together, the classical algorithm $\mathcal{A}$ combined with classical post-processing can provide an estimation to $p(\boldsymbol{j})$ within an additive error

$$\frac{\epsilon_2}{2^n} \leq \frac{1}{2^n} \log\left(1 + \frac{1}{\alpha 2^{n-2}(\Pr(\hat{p}_{\boldsymbol{x}}(\boldsymbol{j})) - a)}\right) \leq \frac{1}{2^n} \log\left(1 + \frac{4}{\alpha}\right) \leq \frac{1}{\alpha 2^{n-2}} \tag{56}$$

with probability $(1-\alpha\epsilon)e^{-1}$. Let $\alpha = (1 - e/4)\epsilon^{-1}$ ($(1-\alpha\epsilon)e^{-1} = \frac{1}{4}$), the above estimation is valid with probability $1/4$. Therefore, under the conditions $\epsilon \leq n^{-1}$, and assuming $\mathcal{A}$ can efficiently sample from the ground state $|\psi_g(\boldsymbol{x})\rangle$, then there exists a classical algorithm which can estimate $p(\boldsymbol{j})$ to a $\mathcal{O}(1/(n2^n))$ additive error with a constant probability.

However, we already know that it is $\#P$-hard to estimate $p(\boldsymbol{j})$ to a $\mathcal{O}(1/(n2^n))$ additive error with a constant probability. Then, if $\epsilon < n^{-1}$, the existence of such classical algorithm $\mathcal{A}$ would lead to Polynomial Hierarchy collapse to its second level [84]. Therefore, no classical algorithm can efficiently sample from $|\psi_g(\boldsymbol{x})\rangle$ if $\epsilon < n^{-1}$. □

To illustrate the usefulness of theorem 4, we show numerical calculations for (1) the lattice Transverse-field Ising model and (2) the Fermi-Hubbard model. The lattice Transverse-field Ising model is given by $H_I(W, J.F) = W\sum_{i=1}^n Z_i + J\sum_{(i,j)} Z_i Z_j - \frac{F}{2}\sum_{i=1}^n X_i$, where X_i and Z_i are Pauli operators on the i-th qubit, and $\boldsymbol{x} = (W, J, F)$ determines the relative strength of the Hamiltonian terms. And the Fermi-Hubbard model is given by $H_H(t,U) = -t\sum_{\langle i,j\rangle,s}(a_{i,s}^\dagger a_{j,s} + a_{j,s}^\dagger a_{i,s}) + U\sum_i n_{i\uparrow}n_{i\downarrow}$, where $\boldsymbol{x} = (t, U)$ determines the relative strength of the Hamiltonian terms, $a_{i,s}^\dagger$ and $a_{i,s}$ are fermionic creation and annihilation operators, $n_{i\uparrow} = a_{i\uparrow}^\dagger a_{i\uparrow}$ and similarly for $n_{i\downarrow}$. The notation $\langle i,j\rangle$ in the first sum associates sites that are adjacent in a $(n_a \times n_b)$ lattice, and $s \in \{\uparrow, \downarrow\}$. Here, we denote the above two Hamiltonians as $H(\boldsymbol{x})$, and analyze the probability distribution of $|\psi_g(\boldsymbol{x})\rangle$, namely the ground state of $H(\boldsymbol{x})$.

We use the trace distance $\text{Tr}(p,q)$ as a measure between two distributions, where $\text{Tr}(p,q) \in [0,1]$; $\text{Tr}(p,q) = 0$ holds if and only if distribution $p = q$. In Figure 10(a), we plot our results for lattice Transverse-field Ising model with $n = 12$, $W = 1$, and a range of J and F values. When the nearest neighbour coupling strength $J = 0$, the corresponding ground states are classically solvable. sample from these ground states is easy for classical algorithms, as indicated by large values of trace distance in Figure 10(a). On the other hand, $J \neq 0$ induces complex ground states $|\psi(\boldsymbol{x})\rangle$, resulting much smaller values of trace distances. However, the ground states of 2-dimensional lattice models donot saturate into the $\epsilon \leq n^{-1}$ domain which has been proved to be classically hard.

In Figure 10(b), we plot ground states of Hubbard model for $U = 1$, and a range of $n = (n_a \times n_b)$ and t values. We observe that the change of t increases the sample complexity of $|\psi_g(\boldsymbol{x})\rangle$ for different size of the Hubbard models. The ground states of (2×5) Fermi-Hubbard model saturate into the $\epsilon = n^{-1} = 0.1$ error. It also demonstrates Fermi-Hubbard ground states are more difficult compared with (a) lattice Ising model ground states, which is consistent with physical intuition.

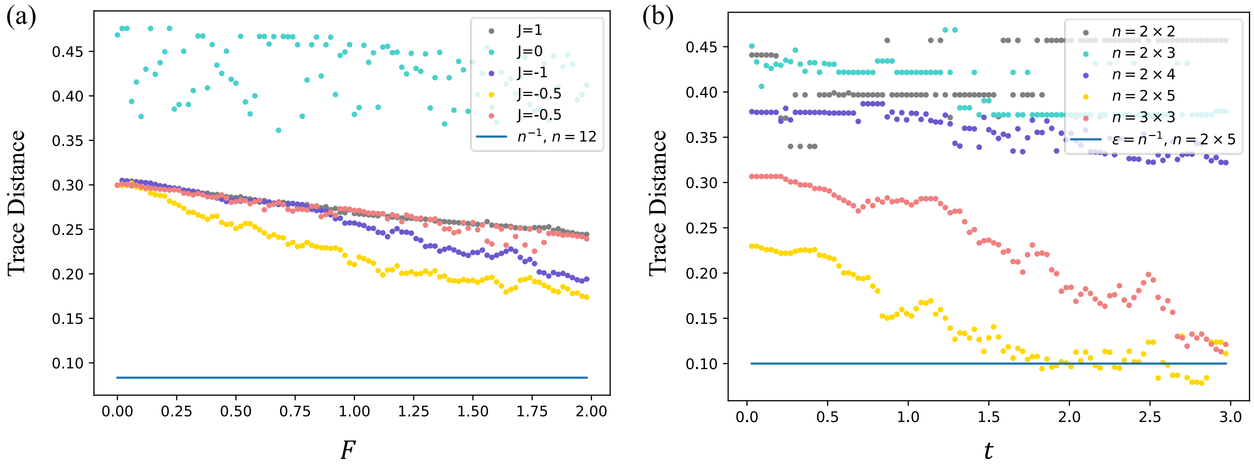


Figure 10: Numerical results to illustrate the sample complexity of $|\psi_g(\boldsymbol{x})\rangle$. Each dot represents the KLD between the probability distribution $\Pr(p_x(\boldsymbol{j}))$ and PT distribution for (a) 2-dimensional lattice Hamiltonian and (b) 2-dimensional Fermionic Hubbard Hamiltonian.

F Numerical comparison to related works

F.1 Comparison to QCNN

To provide a fairly comparison, we first elaborately provide the computational overhead of both method in each iteration step. For the proposed Algorithm 1, the quantum learner should first calculate an $N\times N$ quantum kernel method with $\mathcal{O}\left(N^2/\epsilon^2\right)$ sample complexity and a quantum circuit with $\mathcal{O}(n)$ controlled Z gates. After that, the proposed Algorithm 1 does not need any quantum resource in each iteration step. The QCNN loss function is

$$L(U,V,F)=\frac{1}{N}\sum_{i=1}^{N}\left(y_i-f_{U,V,F}(|\psi(\boldsymbol{x}_i)\rangle)\right)^2,$$

where $f_{U,V,F}$ denotes the output of QCNN and U,V,F represent the variational quantum circuits in QCNN. At depth d, the QCNN method requires $\mathcal{O}(\frac{7n}{2}(1-3^{1-d})+n3^{1-d})$ multi-qubit operations and $4d$ single-qubit rotations to provide an output. Repeat this quantum circuit N times, the QCNN obtains a loss function $L(U,V,F)$ in a single iteration step.

Consider a classification task on a set $C=\{c_1,c_2\}$ of 2 classes in a supervised learning scenario. In such settings, a training set $\mathcal{S}$ and a testing set $\mathcal{T}$ both are assumed to be labeled by a map $m:\mathcal{S}\cup\mathcal{T}\mapsto C$, and both $\mathcal{S}$ and $\mathcal{T}$ are provided to the learner, where only the training set $\mathcal{S}$ has the label. Formally, the learner has only access to a restriction $\widetilde{m}(\boldsymbol{x})$ of the indexing map:

$$\widetilde{m}:\mathcal{S}\mapsto C. \tag{57}$$

Suppose the learner outputs a model $h(\cdot)$ to predict the label of data $\boldsymbol{x}\in\mathcal{T}$, and the classification result can be defined as:

$$m(\boldsymbol{x})=\begin{cases}c_1, & h(\boldsymbol{x})>t_1,\\ c_2, & h(\boldsymbol{x})<t_2,\end{cases} \tag{58}$$

where t_1 and t_2 are selected thresholds. Then the accuracy of the model is quantified by a classification success rate, proportional to the number of collisions:

$$v_s=\frac{|\{x\in\mathcal{T}|m(\boldsymbol{x})=\widetilde{m}(\boldsymbol{x})\}|}{|\mathcal{T}|}. \tag{59}$$

In the classification of SPT cases, the training data set $\mathcal{S}$ contains 40 points on the line of $h_2=0$, and the testing data set $\mathcal{T}$ contains 4096 points uniformly distributed on the domain $(h_1/J,h_2/J)\in[-1.5,1.5]\times[0,1.6]$.

We test the proposed Algorithm 1 on $\mathcal{T}=\{\boldsymbol{x}_i,y_i\}$ where 61 points are mis-classified in the vicinity of paramagnetic boundary, and the classification accuracy $v_s=0.985$ in this case. Since the Ref [35] does not provide the exactly accuracy of QCNN, we thus simulate the QCNN based on the code in the link (https://github.com/Jaybsoni/Quantum-Convolutional-Neural-Networks). The simulation results show that its output cannot perfectly fit the Antiferromagnetic boundary which is consistent with Fig. 4 in ref [35]. Using the same testing set with 4096 data points, there are 116 points are mis-classified, and its classification accuracy $v_s = 0.971$. The predicted phase diagrams is illustrated as Fig 9.

We also numerically show our method takes less steps to a relatively stable landscape compared to the QCNN method (see Fig. 11).

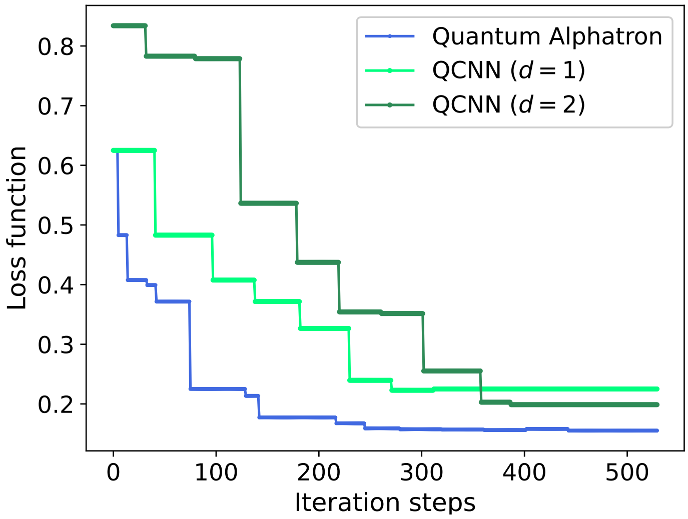


Figure 11: The three curves indicate the variation trend of the loss function $\hat{R}_L(h^*)$ in the training procedure.

F.2 Comparison to Ref. [17]

Here, we utilize shadow-tomography based kernel-PCA method [17] and common kernel-PCA method [87] in recognizing $Z_2 \times Z_2$ SPT phase of Haldane chain and three distinct phases of bond-alternating XXZ model.

We first consider the method proposed in [17] that designed a special kernel function for classical shadows. Given an unknown density matrix ρ, implementing the randomrized single-qubit Pauli measurements, a classical shadow representation is obtained, where each shadow raw data corresponds to a two-dimensional array

$$S_T(\rho) = \left\{|s_i^t\rangle : i \in \{1, ..., n\}, t \in \{1, ..., T\}\right\} \in \{|0\rangle, |1\rangle, |+\rangle, |-\rangle, |i+\rangle, |i-\rangle\}^{n\times T}. \tag{60}$$

And a classical representation of the density matrix ρ is

$$\sigma_T(\rho) = \frac{1}{T}\sum_{t=1}^{T} \sigma_1^{(t)} \otimes \sigma_2^{(t)} \otimes ... \otimes \sigma_n^{(t)}, \tag{61}$$

where $\sigma_i^{(t)} = 3|s_i^t\rangle\langle s_i^t| - I$. Then the kernel function for shadow tomography can be computed by

$$k^{\text{shadow}}(\rho_1, \rho_2) = \exp\left(\frac{\tau}{T^2}\sum_{t_1,t_2=1}^{T} \exp\left(\frac{\gamma}{n}\sum_{i=1}^{n} \text{Tr}\left(\sigma_i^{t_1}\sigma_i^{t_2}\right)\right)\right), \tag{62}$$

which is determined by hyper-parameters τ and γ. Then perform kernel-PCA method on the kernel matrix k^{shadow}, the classification result of quantum phases is obtained. Another strategy is directly applying kernel-PCA method on the provided shadow tomography, where the corresponding kernel matrix

$$k(\rho_1, \rho_2) = \exp\left(\tau \text{Tr}\left(\rho_1\rho_2\right)\right) \tag{63}$$

Here, we test their performances in classifying $Z_2 \times Z_2$ symmetry-protected-topological (SPT) phase of Haldane chain and three distinct phases of bond-alternating XXZ model. Let the sample complexity $T = 500$, and simulation results are shown in Fig. 12 and Fig. 13. Noting that the shadow tomography kernel-PCA can recognize different quantum phases of matter, including SPT phase, symmetry-broken phase and trivial phase, while it is hard for common kernel-PCA method. Meanwhile, it is observed that shadow tomography kernel-PCA cannot provide a comparable classification result with QKA method (see Fig. 12 (c) and Fig. 13 (c)), where some strange quantum phase assignments appear. For example, Fig. 12 (a) appears two mis-classified ground states in the top right corner which do not represent any quantum phase, and a similar phenomenon is observed on the left side in Fig. 13 (a).

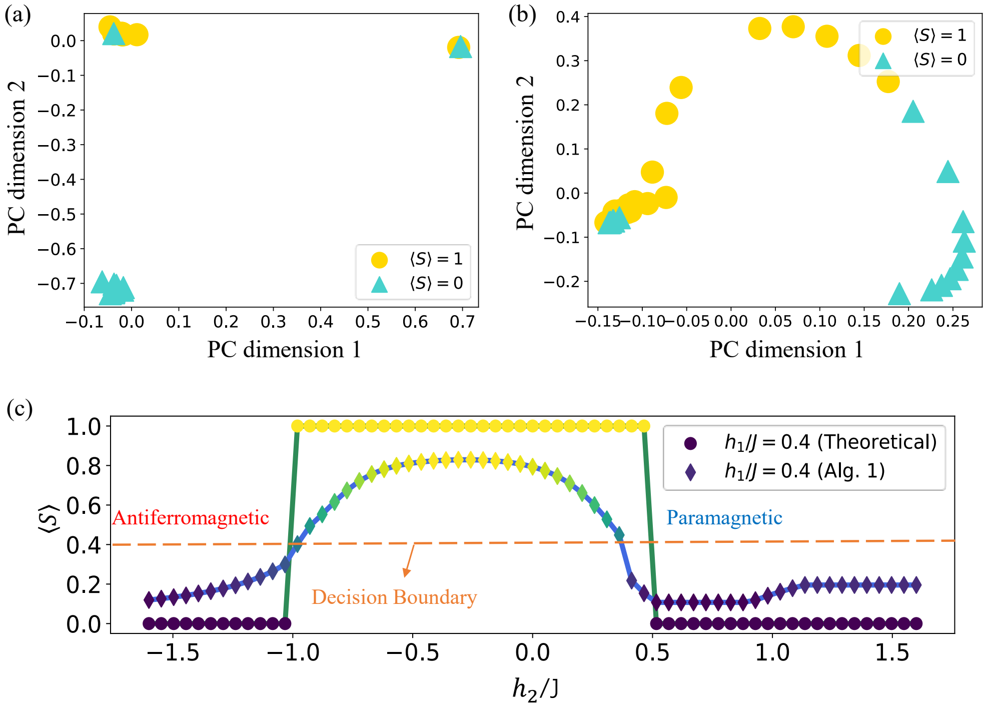


Figure 12: Numerical results for recognizing $Z_2 \times Z_2$ Symmetry-Protected-Topological (SPT) phases of Haldane Chain at the cross-section $h_1/J = 0.4$. (a) Classification results by using shadow tomography-based kernel-PCA [17] method by using code in (https://github.com/hsinyuan-huang/provable-ml-quantum). (b) Classification results by using kernel-PCA. (c) Prediction results by using quantum kernel method.

G Review of Alphatron algorithm

In this section, we review the Alphatron algorithm [52], and give the comparison for Alphatron, Quantum Alphatron [88], and the Quantum kernel Alphatron algorithm (this paper).

Algorithm 2: Alphatron algorithm

```
Input : training set $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^N \in \mathcal{R}^d \times [0,1]$, non-decreasing $L$-Lipschitz function $u : \mathcal{R} \to [0,1]$,
        kernel function $\mathcal{K}$, learning rate $\lambda > 0$, number of iterations $T$, testing data
        $\mathcal{S} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^M \in \mathcal{R}^d \times [0,1]$
$\boldsymbol{\alpha}^{\mathbf{1}} := 0 \in \mathcal{R}^N$;
for $t = 1, 2, ..., T$ do
    $\hat{h}^t(\boldsymbol{x}) := \sum_{i=1}^N \alpha_i^t \mathcal{K}(\boldsymbol{a_i}, \boldsymbol{x})$;
    for $i = 1, 2, ..., N$ do
        $\alpha_i^{t+1} = \alpha_i^t + \frac{\lambda}{N}(b_i - h^t(\boldsymbol{a_i}))$;
Let $r = \arg\min_{t \in \{1,...,T\}} \sum_{j=1}^M \left(h^t(\boldsymbol{x}_j) - y_j\right)^2$;
return $h^r$
```

Theorem 5 ([52])**.** *Let $\mathcal{K}$ be a kernel function correpsonding to feature map ψ such that $\forall \boldsymbol{x} \, \|\psi(\boldsymbol{x})\| \leq 1$. Consider samples $(\boldsymbol{a}_i, b_i)_{i=1}^N$ drawn iid from distribution $\mathcal{D}$ on $\mathcal{X} \times [0,1]$ such that $E[y|\boldsymbol{x}] = u\left(\langle \boldsymbol{v}, \psi(\boldsymbol{x}) \rangle\right) + \xi(\boldsymbol{x})$ where $u : \mathcal{R} \to [0,1]$ is a known L-Lipschitz non-decreasing function, $\xi : \mathcal{R}^d \to [-G, G]$ for $M > 0$ such that $\mathbb{E}\left[\xi(\boldsymbol{x})^2\right] \leq \epsilon$ and $\|\boldsymbol{v}\| \leq B$. Then for $\delta \in (0,1)$, with probability $1 - \delta$, Alphatron with $\lambda = 1/L, T =$*

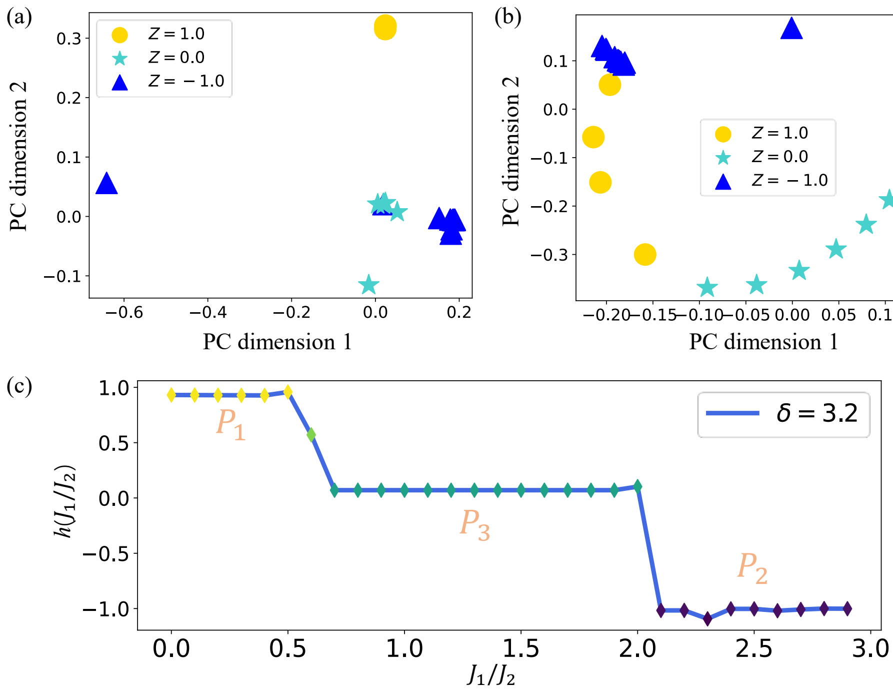


Figure 13: Numerical results for recognizing three distinct phases of bond-alternating XXZ model at the function $\delta = 3.2$. The invariant $Z = 1$ marks the trivial phase, $Z = 0.5$ marks symmetry broken phase and $Z = 0$ marks the topological phase. (a) Classification results by using shadow tomography kernel PCA method. (b) Classification results by using kernel-PCA. (c) Prediction results by using quantum kernel method.

$CBL\sqrt{N/log(1/\delta)}$ *and* $M = C'N\log(T/\delta)$ *for large enough constants* $C, C' > 0$ *outputs a hypothesis* h *such that,*

$$\varepsilon(h) \leq \mathcal{O}\left(L\sqrt{\epsilon} + LG\sqrt[4]{log(1/\delta)/N} + BL\sqrt{\log(1/\delta)/N}\right). \tag{64}$$

Alphatron algorithm requires $\mathrm{poly}(N, d, \log(1/\delta), t_{\mathcal{K}})$ running time to train a kernel model, where $t_{\mathcal{K}}$ is the running time on computing the kernel function. In Quantum Kernel Alphatron algorithm, we let the kernel be quantum kernel $Q(\boldsymbol{x}, \boldsymbol{a_i}) = \left|\langle 0^n| U(\boldsymbol{x})^\dagger U(\boldsymbol{a_i}) |0^n\rangle\right|^2$. Since $Q(\boldsymbol{x}, \boldsymbol{a_i})$ is approximated via SWAP test, the risk in Eq. (64) does not hold. Nevertheless, we prove that via $\mathcal{O}\left(N^{5/2}\right)$ copies of quantum states for each training data, the risk of the Quantum kernel Alphatron is also bounded.

As a comparison, Quantum Alphatron algorithm [88] quantizes the Alphatron algorithm to provide a quantum implementation of the well-known polynomial kernel function, which accelerate the running time of the original Alphatron algorithm in the scenario that the dimension d of the data is much larger than other parameters.