# Provable Advantage in Quantum Phase Learning via Quantum Kernel Alphatron

**Wu Yusen**
The University of Western Australia
**Bujiao Wu**
Peking University
**Jingbo Wang** ( jingbo.wang@uwa.edu.au )
The University of Western Australia https://orcid.org/0000-0001-7544-0084
**Xiao Yuan**
Peking University https://orcid.org/0000-0003-0205-6545

**Article**

**Keywords:**

**Posted Date:** January 5th, 2022

**DOI:** https://doi.org/10.21203/rs.3.rs-1179453/v1

# Provable Advantage in Quantum Phase Learning via Quantum Kernel Alphatron

**Yusen Wu**[1,*], **Bujiao Wu**[2,*], **Jingbo 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*
[*] *These two authors contributed equally*
[†] *jingbo.wang@uwa.edu.au,* [‡] *xiaoyuan@pku.edu.cn*

The use of quantum computation to speed-up machine learning algorithms is among the most exciting prospective applications in the NISQ era. Here, we focus on the quantum phase learning problem, which is crucially important in understanding many-particle quantum systems. We prove that, under widely believed complexity theory assumptions, quantum phase learning problem cannot be efficiently solved by machine learning algorithms using classical resources and classical data. Whereas using quantum data, we prove the universality of quantum kernel Alphatron in efficiently predicting quantum phases, indicating clear quantum advantages in such learning problems. We numerically benchmark the 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 efficient prediction of quantum phases of many-particle systems.

The complex nature of multipartite entanglement has stimulated different powerful classical techniques, including quantum Monte Carlo [1, 2, 3, 4], density functional theory [5], density matrix renormalization group [6, 7], etc, to study many-body quantum systems. Among them, classical machine learning techniques have been recently considered as a means of either representing the state or learning the quantum behaviour. From both theoretical and numerical perspectives, many works have shown that neural network quantum state ansätze have stronger representation powers than conventional tensor networks and may solve complex static and dynamical quantum problems [8, 9, 10, 11, 12, 13, 14]. Yet, recent quantum supremacy experiments [15, 16, 17] have indicated that the sole application of these classical learning algorithms might still be challenging for an arbitrary quantum system that possesses genuine and intricate entanglement. Therefore, two critical questions arise: (1) where is exactly the limitation of classical machine learning in quantum problems? and (2) do more advanced solutions exist?

Here we study these two questions. For a many-body quantum system described by a parameterized Hamiltonian $H(\boldsymbol{a})$, we focus on a general quantum phase learning (QPL) problem [18, 19, 20], aiming at detecting quantum phases determined by certain order parameters on an eigenstate of $H(\boldsymbol{a})$. While QPL plays the key role to investigate tremendous behaviours in condensed-matter physics [21], it is an inherently hard problem since the order parameter is generally unknown and estimating the order parameter is classically hard. Indeed, under two reasonable assumptions — (1) the polynomial hierarchy does not collapse in the computational complexity theory, and (2) the classical hardness for random circuit sampling holds, we rigorously prove that the QPL problem is hard for any classical machine learning methods if the training data set is generated by a classical Turing machine. We therefore answer the first question by showing the exact limitation of classical machine learning in QPL.

For the second question, we consider the solution of QPL using quantum computers. Among different applications of quantum computing [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32], a variety of quantum machine learning algorithms [20, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42] have been developed for different problems and demonstrated the potential for solving classically intractable problems, using parameterized circuits and classical optimization of noisy-intermediate-scale-quantum devices. Using a quantum computer, we propose the *quantum kernel Alphatron* algorithm to efficiently solve the QPL problem. We show convincing numerical results in detecting symmetry-protected topological phases of the Haldane chain and symmetry broken phases of the XXZ model.

Recently, Rebentrost, Santha and Yang [43] proposed a quantum alphatron in training multinomial kernel functions for fault-tolerant quantum devices, whereas our method focuses on the quantum kernel and

is designed for near-term quantum devices.

This paper is organized as follows. In Sec. 1 we review the supervised learning and quantum feature spaces, and give the definition of the quantum phase recognition learning problem. We give the hardness results for classical learning algorithms in Sec. 2. Sec. 3 gives a quantum learning algorithm for quantum phase recognition problem. Sec. 4 gives the numerical results for it, and we also provide the complexity class of the QPL problem in Sec. 5. We give a discussion in Sec. 6.

# 1 Preliminaries

In this section, we review the definitions of supervised learning and kernel methods, and introduce the quantum phase recognition problem.

## 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 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, with the goal of achieving low error or risk

$$R(h) = \Pr_{\boldsymbol{x}\sim\mathcal{D}}[h(\boldsymbol{x}) \neq y]. \tag{1}$$

Here, we assume that the datum $\boldsymbol{x}$ is sampled randomly according to $\mathcal{D}(\boldsymbol{x})$, in both 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 [44, 45, 46], 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 $\mathcal{K}(\boldsymbol{x}, \boldsymbol{x}') = \Psi(\boldsymbol{x})^T \Psi(\boldsymbol{x}')$, where $\Psi : \mathcal{X} \to \mathcal{H}$ 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 [46, 45] 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 [35, 39, 36, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61]. Here we leverage the *quantum kernel* as our kernel function, which is defined as $Q(\boldsymbol{x}, \boldsymbol{x}') = |\langle\phi(\boldsymbol{x})|\phi(\boldsymbol{x}')\rangle|^2$, where $|\phi(\boldsymbol{x})\rangle$ is a quantum state associated with $\boldsymbol{x}$.

## Quantum Phase Learning (QPL) problem

In this section, we introduce the *quantum phase learning* (QPL) problem, which is the key prerequisite to investigate a large number of behaviours in many condensed-matter systems [18, 19].

Given an $n$-qubit Hamiltonian $H(\boldsymbol{a})$ with interaction parameters $\boldsymbol{a}$ and an order parameter $\mathcal{M} \in \mathbb{C}^{2^n \times 2^n}$, the goal of *quantum phase computation* is to approximate the phase value $b = \langle\phi(\boldsymbol{a})|\mathcal{M}|\phi(\boldsymbol{a})\rangle$ to additive error $\epsilon = 1/\text{poly}(n)$, where $|\phi(\boldsymbol{a})\rangle$ is the ground state of $H(\boldsymbol{a})$. For example, considering the Ising Hamiltonian, the parameter and the order parameter could be the strength of the transverse magnetic field and the spin correlation respectively, and quantum phases include paramagnetic, ferromagnetic, and antiferromagnetic phases. 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 phases, whether we could learn or predict quantum phases for other cases. Therefore, it is natural to consider the learning version of the quantum phase recognition problem.

DEFINITION 1.1 (QPL Problem). *Given training data $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^{N}$ for which $\boldsymbol{a_i}, b_i$ indicate the classical coupling weight and phase value observed from the $i$-th experiment associated with Hamiltonian $H(\boldsymbol{a_i})$, the*

*target is to learn a prediction model $h(\boldsymbol{a})$ to minimize the risk*

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

*for some fixed distribution $\mathcal{D}(\boldsymbol{a})$ defined on the datum space $\mathcal{X}$.*

In the following, we consider solving the QPL problem using classical and quantum computing methods.

## 2 Classical hardness for QPL problem

In this section, we show the hardness for quantum phase computation and QPL problems using classical computers.

Here we assume that the order parameter $\mathcal{M}$ is a general $n$-fold tensor product of local Pauli operators. Therefore, the quantum phase computation 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 open, as mentioned in Ref [62]. Note that for any quantum state $|\phi\rangle$, there exists a quantum circuit $U$ associated with $|\phi\rangle$ such that $|\phi\rangle = U|0^n\rangle$. Bravyi et al. [62] proposed an upper bound on estimating $\langle\phi|\mathcal{M}|\phi\rangle$ in the case of a poly$(n)$-depth $U$ associated with $|\phi\rangle$. Here, we provide a lower bound of this problem based on the following conjecture raised by Bouland et al. [63].

CONJECTURE 1 (Ref. [63]). *There exists an $n$-qubit quantum circuit $U$ such that the following task is # P-hard: approximate $|\langle 0^n|U|0^n\rangle|^2$ to additive error $\epsilon_c/2^n$ with probability $\frac{3}{4} + \frac{1}{\text{poly}(n)}$.*

Here, a candidate of the worst-case $U \in \mathbb{C}^{2^n\times 2^n}$ is a size $m \leq \text{poly}(n)$ unitary where each basic gate is a single- or two-qubit gate drawn from Haar-measure [63], following some fixed gate position structure $\mathcal{A}$. We denote this distribution as $\mathcal{H}_\mathcal{A}$, see Appendix A for the details of this distribution. The hardness result for the quantum phase calculation problem can be stated as the following lemma.

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

We provide detailed proof in the Appendix B. This lemma also serves for the hardness of the QPL problem. Following the "worst-to-average-case" reduction [63], we can construct a testing set $\mathcal{T} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^M$ with associated feature states $\{|\phi(\boldsymbol{x_i})\rangle\}_{i=1}^M$ (ground states). Given classical training data $\mathcal{S}$ (from classical method), we prove that no classical learning algorithm can efficiently learn the hypothesis $h^*$ such that $R(h^*(\boldsymbol{x}))$ is close to zero for $\boldsymbol{x} \in \mathcal{T}$, as shown in the following theorem.

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 classical coupling weight and phase value associated with the Hamiltonian $H(\boldsymbol{a}_i)$, there exists a testing set $\mathcal{T} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^M$ and corresponding ground states $\{|\phi(\boldsymbol{x_i})\rangle\}_{i=1}^M$ such that learning a hypothesis $h$ enabling $\Pr[R(h) < \epsilon] > 1 - \delta$ on $\mathcal{T}$ is hard for any classical ML algorithm, with the assumption that Conjecture 1 holds and the PH does not collapse. The generalized error $\epsilon = \mathcal{O}(1/\sqrt{N})$, failure probability $\delta \in (0, 1)$, and the scale of testing data $M = \text{poly}(N)$.*

As we discuss in Sec 5, we divide the QPL problem into four different cases in terms of training data acquiring method and learning algorithms. Theorem 1 gives hardness result for the classical data acquiring method and classical learning algorithm, using the hardness result in Lemma 1.

We give a proof sketch in the following and defer the details in Appendix B.4.

*Proof sketch.* The fundamental idea of the proof combines the 'worst-to-average-case' reduction of the random circuit sampling problem and the learning complexity class of classical ML algorithms.

Firstly, we design a learning instance with the same format as in Lemma 1. Since the quantum adiabatic algorithm with 2-local Hamiltonians can implement universal quantum computational tasks [32, 64], any quantum state $|\phi(\boldsymbol{a})\rangle$ generated from the distribution $\mathcal{H}_\mathcal{A}$ (mentioned in Conjecture 1) can be encoded as

the ground state of a 2-local Hamiltonian $H(\boldsymbol{a})$. Therefore, constructing ground states in $\mathcal{T}$ is equivalent to constructing random quantum circuits. To give an average-case result, we replace $|\phi(\boldsymbol{a})\rangle = U|0^{\otimes n}\rangle$ into $|\phi(\boldsymbol{x})\rangle$. Here $|\phi(\boldsymbol{x})\rangle$ is obtained by performing a circuit $\widetilde{U}(\boldsymbol{x})$ drawn from a distribution $\widetilde{\mathcal{H}}_{\mathcal{A}}$ to the initial state $|0^{\otimes n}\rangle$. For a circuit $U = U_m \dots U_1 \sim \mathcal{H}_{\mathcal{A}}$, $\widetilde{U}(\boldsymbol{x})$ is obtained by multiplying $H_j^{1-\boldsymbol{x}}$ for each $U_j$ where $H_j$ is drawn from Haar-measure distribution with structure $\mathcal{A}$, $\boldsymbol{x} = \sum_j x_j 2^{-j}$ and $\boldsymbol{x} = (x_1, x_2, \dots)$. The new generated distribution is denoted as $\widetilde{\mathcal{H}}_{\mathcal{A}}$. Bouland et al. [63] proved that the distance between $\langle \boldsymbol{j}|\widetilde{U}(\boldsymbol{x})|0\rangle$ and $\langle \boldsymbol{j}|V(\boldsymbol{x})|0\rangle$ (a low-degree polynomial function of $x$) is bounded, where $V(\boldsymbol{x}) = V_m(\boldsymbol{x}) \dots V_1(\boldsymbol{x})$, and $V_j(\boldsymbol{x})$ is the approximation of $U_j H_j^{1-\boldsymbol{x}}$ by replacing $H_j^{1-\boldsymbol{x}}$ into the truncated Taylor expansion in $\boldsymbol{x}$. By Aaronson and Arkhipov [65], it is $\#P$-hard to compute $\langle \boldsymbol{j}|V(\boldsymbol{x})|0\rangle$ for most of $V(\boldsymbol{x})$, then it is also $\#P$-hard to compute $\langle \boldsymbol{j}|\widetilde{U}(\boldsymbol{x})|0\rangle$ for most of $\widetilde{U}(\boldsymbol{x})$. Combined with the result of Lemma 1, we have the average-hardness to calculate the label $y_i$ given $\boldsymbol{x}_i$.

Secondly, the power of classical ML is in BPP/samp $\subseteq$ P/poly class [40]. Hence with the assumption that PH does not collapse, learning with classical data is intractable even in the average-case. □

Therefore we have proved that the QPL problem cannot be efficiently solved only using classical computers. Specifically, if the training data is generated classically and the learning algorithm is also classical, efficiently solving QPL would lead to an unlikely collapse of complexity classes.

# 3 Quantum learning algorithm

In this section, we show the possibility of solving the QPL problem with quantum data (acquiring from quantum devices) leveraging the alphatron algorithm [45] combined with the quantum kernel method. From the learning theory perspective, training can be phrased as the empirical risk minimisation, and the associated learning model $h^*$ for the minimizer of the empirical risk is cast as follows.

THEOREM 2 (Representer theorem [44]). *Let $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^N$ and corresponding feature states $\{U(\boldsymbol{a_i})|0^n\rangle\}_{i=1}^N$ be the training data set. $Q : \mathcal{X} \times \mathcal{X} \mapsto \mathcal{R}$ be a quantum kernel with the kernel space $\mathcal{H}$. Consider a strictly monotonic increasing regularisation function $g : [0, \infty) \mapsto \mathcal{R}$, and regularised empirical risk*

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

*Then any minimiser 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 $\alpha_i \in \mathcal{R}$ for all $i \in \{1, 2, ..., N\}$, $\boldsymbol{x}$ and $\boldsymbol{a}$ are drawn from the same distribution.*

This representer theorem holds since the quantum kernel matrix is positive semi-definite. Using Theorem 2, the critical point for solving the QPL problem is the selection of the quantum kernel and the training algorithm. Since the label $b$ represents the phase value (mean value) by Definition 1.1, one of the options of the kernel is $Q(\boldsymbol{a_i}, \boldsymbol{x}) = |\langle \phi(\boldsymbol{a_i})|\phi(\boldsymbol{x})\rangle|^2$, and unknown measurement $\mathcal{M}$ thus can be represented as a linear combination of feature states, that is, $\mathcal{M} \approx \sum_i \alpha_i |\phi(\boldsymbol{a_i})\rangle\langle\phi(\boldsymbol{a_i})|$. Given the kernel matrix $\mathcal{Q} = [Q(\boldsymbol{a}_i, \boldsymbol{a}_j)]_{N\times N}$, the optimal weight parameters $\alpha_i$ in the expression of $\mathcal{M}$ has a closed-form solution by leveraging linear regression algorithms, and it requires $\mathcal{O}(N^w)(w \approx 2.373)$ time complexity for solving the above problem [66].

Here we give a learning approach for all of $\alpha_i$, which has a polynomial speedup to the above method. In particular, we introduce the quantum kernel into the Alphatron algorithm [45], as depicted in Alg. 1. We find that the quantum kernel perfectly fit into Alphatron algorithm and hence the QPL problem can be solved in $\mathcal{O}(N^{1.5})$ time if the kernel matrix $\mathcal{Q}$ is provided, as depicted in Theorem 3.

THEOREM 3. *Let quantum kernel $Q(\boldsymbol{a_i}, \boldsymbol{x}) = |\langle \phi(\boldsymbol{a_i})|\phi(\boldsymbol{x})\rangle|^2$, and $\{(\boldsymbol{a_i}, b_i)\}_{i=1}^N$ be the training set such that $\mathbb{E}[b_j|\boldsymbol{a_j}] = \sum_i \alpha_i \left|\langle \phi(\boldsymbol{a_i})|\phi(\boldsymbol{a_j})\rangle\right|^2 + g(\boldsymbol{a_j})$, $g : [0, \infty) \mapsto [-G, G]$ is a strictly monotonic increasing regularisation function such that $\mathbb{E}[g^2] \leq \varepsilon_g$ and $\sum_{ij} \alpha_i \alpha_j \left|\langle \phi\left(\boldsymbol{a_i}\right)|\phi\left(\boldsymbol{a_j}\right)\rangle\right|^2 < B$. 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*

**Algorithm 1:** Quantum Kernel Alphatron

**Input :** training set $\mathcal{S} = \{(\boldsymbol{a_i}, b_i)\}_{i=1}^{N} \in \mathcal{R}^d \times [0,1]$, Hamiltonian $\mathcal{H}(\boldsymbol{a})$ with coupling weight $\boldsymbol{a}$, parameterized quantum circuit $U(\boldsymbol{\theta})$, quantum circuit approximation $\hat{Q}(\boldsymbol{a_i}, \boldsymbol{x})$ for quantum kernel $Q(\boldsymbol{a_i}, \boldsymbol{x})$, learning rate $\lambda > 0$, number of iterations $T$, testing data $\mathcal{T} = \{(\boldsymbol{x_j}, y_j)\}_{j=1}^{M} \in \mathcal{R}^d \times [0,1]$

**Output:** $\hat{h}^r$

**for** $i = 1, 2, ..., N$ **do**

  *Prepare the ground state* $|\phi(\boldsymbol{a}_i^*)\rangle = U(\boldsymbol{\theta}(\boldsymbol{a}_i^*))|0^{\otimes n}\rangle$, *where* $\boldsymbol{\theta}(\boldsymbol{a}_i^*) := \arg\min_{\boldsymbol{\theta}(\boldsymbol{a}_i)} \langle 0^{\otimes n}|U^\dagger(\boldsymbol{\theta}(\boldsymbol{a}_i))|\mathcal{H}(\boldsymbol{a}_i)|U(\boldsymbol{\theta}(\boldsymbol{a}_i))|0^{\otimes n}\rangle$;

$\boldsymbol{\alpha^1} := 0 \in \mathcal{R}^N$;

**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** $\hat{h}^r$

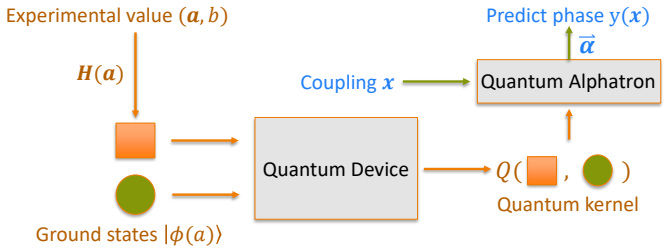


Figure 1: The procedure of the proposed quantum learning algorithm.

$\hat{R}_L(\hat{h}^*)$ *can be bounded by*

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

*by selecting* $\lambda = 1$, $T = \mathcal{O}(\sqrt{N/\log(1/\delta)})$ *and* $M = \mathcal{O}(N\log(T/\delta))$.

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

$$Q(\boldsymbol{a_i}, \boldsymbol{x}) = |\langle\phi(\boldsymbol{a_i})|\phi(\boldsymbol{x})\rangle|^2 = \langle\Psi(\boldsymbol{a_i})|\Psi(\boldsymbol{x})\rangle, \tag{5}$$

and

$$\sum_i \alpha_i \left|\langle\phi(\boldsymbol{a_i})|\phi(\boldsymbol{a_j})\rangle\right|^2 = \langle \boldsymbol{w}, \Psi(\boldsymbol{x})\rangle\,, \tag{6}$$

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

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

. Hence, this theorem followed by substituting the quantum kernel $Q$ and feature map $\Psi$ into Theorem 1 of Goel and Klivans [45] if we can implement $Q$ perfectly.

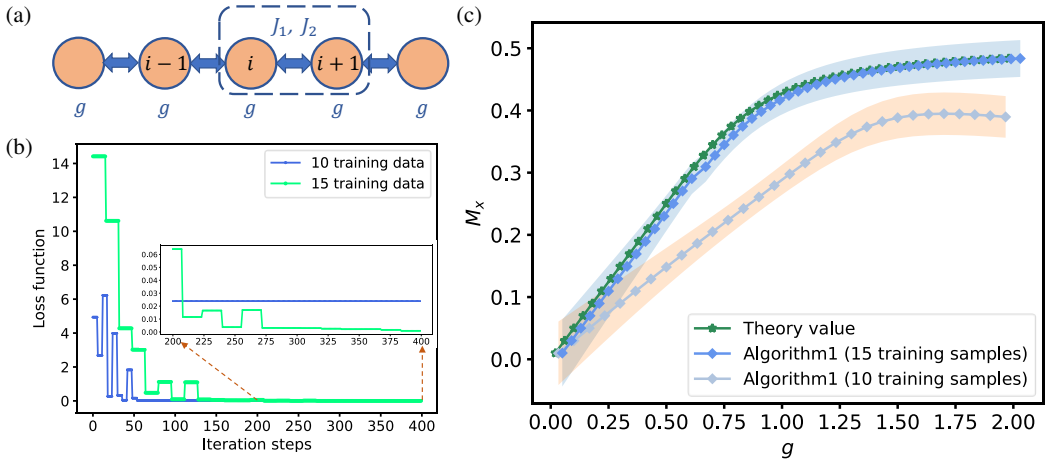


Figure 2: Numerical results for predicting ground-state properties in a $S = 1/2$ XXZ spin model with 16 qubits. (a) Illustration of the concerned spin model geometry. (b) The two curves depicts the tendency of $R_L(h)$ in the training procedure, where the green (blue) curve represents the number of training data $N = 15$ ($N = 10$). (c) For a 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 $M_x$.

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 [67] to $\mathcal{O}(\log(1/\delta)/\varepsilon^2)$ copies of $2n$-qubit state $|\phi(\boldsymbol{a_i})\rangle \otimes |\phi(\boldsymbol{x})\rangle$, with additive error $\epsilon_Q$ and failure probability $\delta$, see Appendix D for the details of the circuit implementation of quantum kernel.

The main idea to bound $\hat{R}_L(\hat{h}^*)$ in Theorem 3 is to prove that with polynomial copies of training states, we can bound

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

where $h^t$ is the ideal hypothesis with exact quantum kernel function $Q(\cdot,\cdot)$. We defer to Appendix B.2 for the tedious calculations. □

The regularisation function $g$ is used to avoid the overfitting problem, and a general selection is to bound the norm of $\|\boldsymbol{w}\|$, that is $g(\cdot) = 2L\|\boldsymbol{w}\|^2 - G$ which is determined by all $\boldsymbol{a_i} \in \mathcal{S}$ and the positive parameter $L \in [0, GB^{-1}]$. According to Theorem 3, the proposed quantum learning algorithm can output a hypothesis that minimizes the empirical risk in $\mathcal{O}(\sqrt{N\log(1/\delta)}t_Q)$ classical time, where $t_Q$ is the time required to compute kernel function $Q$. The complete learning procedure is illustrated in Fig. 1.

## 4 Numerical simulation results

Here we test the capability of the quantum kernel Alphatron algorithm for several instances of QPL 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 [69]. The Hamiltonian is defined as

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

where $S_i^\alpha$ is the $\alpha$-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. 2 (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

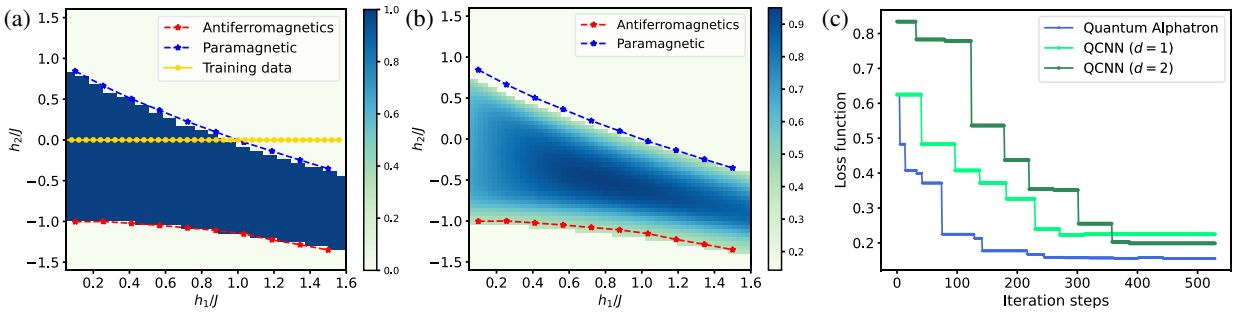


Figure 3: Numerical results for recognizing a $Z_2 \times Z_2$ Symmetry-Protected-Topological (SPT) phase of Haldane Chain. (a) The exact phase diagram of Haldane Chain (Eq. 8), where the phase boundary points (blue and red curves) are extracted from the literature [20], and the background shading represents 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 phase diagram of Haldane Chain generated by the Alg. 1 with the size of $n = 16$. (c) The three curves indicate the variation trend of the loss function $\hat{R}_L(h^*)$ in the training procedure. It shows that our method has a significant less convergence steps compared to the quantum convolution neural network (QCNN) method [20] with QCNN layer depth $d = 1, 2$ in recognizing the SPT phase problem.

the interval $[0, 2]$. The predictions proposed by Alg. 1 are illustrated in Fig. 2, which shows Alg. 1 yields a high accuracy prediction even for a very small training set ($N = 15$).

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 $\{|\Phi(h_1/J, h_2/J)\rangle\}$ belongs to a family of Hamiltonians

$$H_s = -J \sum_{i=1}^{n-2} Z_i X_{i+1} Z_{i+2} - h_1 \sum_{i=1}^{n} X_i - h_2 \sum_{i=1}^{n-1} X_i X_{i+1}, \tag{8}$$

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. 3 (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. 3 (a)). Our target is to identify whether a given, unknown ground state $|\Phi(\boldsymbol{x})\rangle$ belongs to $\mathcal{P}$. The simulation results for $n = 16$ are illustrated as Fig. 3 (b), which shows that Alg. 1 can reproduce the phase diagram with high accuracy on $M = 4096$ testing points.

Remarkably, since the training data is only on the line with $h_2 = 0$, we cannot classically learn the quantum phase only from the relationship between the phases and the $h_1, h_2$ parameters. However, the quantum kernel Alphatron algorithm works using more information of quantum kernels. We thus demonstrate that even if the training data are all from classically solvable cases, quantum kernel Alphatron still works. We also note that the quantum convolution neural network (QCNN) method [20] has been proposed to solve the same problem by applying a CNN quantum circuit to the quantum state. We numerically show a faster convergence of our quantum kernel Alphatron and leave a more detailed comparison to QCNN in future works.

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 $J_1, J_2, \delta$ are coupling parameters of $H_b$. The XXZ model has three different phases that can be detected by the topological invariant $Z_R(J_1/J_2, \delta)$ [68]. 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. 4 (a), we utilize Alg. 1 to generate the phase diagram as a function of $\boldsymbol{x} = (J_1/J_2, \delta)$,

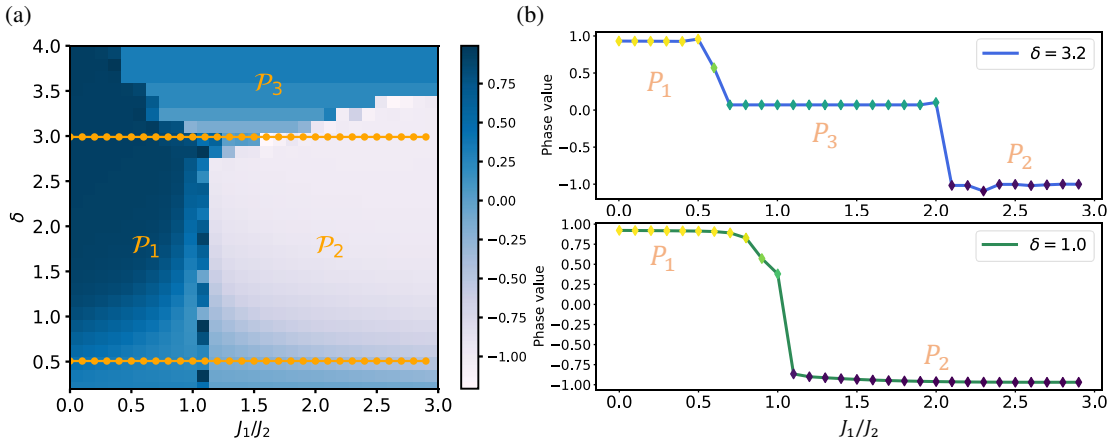


Figure 4: Numerical results for recognizing three distinct phases of XXZ model. (a) The system's three distinct phases are characterized by the topological invariant $Z_R$ discussed in the Ref [68]. The invariant $Z_R = +1$ marks the Trivial phase $\mathcal{P}_1$, $Z_R = -1$ marks the Topological phase $\mathcal{P}_2$ and $Z_R = 0$ marks the Symmetry broken phase $\mathcal{P}_3$. Here, the (white and blue) background shading is depicted by the Alg. 1, and the lines $\delta = 3.0$, $\delta = 0.5$ represents 60 training points. (b) The function $Z_R(J_1/J_2)$ at corss sections $\delta = 3.2$ and $\delta = 1.0$ of the phase diagram.

Table 1: QPL categories in terms of the training data acquiring method and learning algorithms.

| Training data acquiring method | Learning Alg. | Hardness |
|---|---|---|
| classical | classical | **NP**-hard |
| classical | quantum | open |
| quantum | classical | open |
| quantum | quantum | $\mathbf{BQP}^O$ |

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.

# 5 The complexity class of QPL problem

Before concluding the results, we discuss more on the difference between "classical" and "quantum" learning. In terms of the method that produces the training data and learning algorithm being classical or quantum, we consider four categories, as shown in Table 1. We discuss the relationship between the four categories respectively.

- For the QPL problem that satisfies Lemma 1, Theorem 1 indicates that this problem is outside the "C-Learning Alg. + C-Data" class, while Theorem 3 shows that it belongs to the "Q-Learning Alg. + Q-Data" class. These results thus imply "Q-Learning Alg. + Q-Data" is strictly stronger than "C-Learning Alg. + C-Data" with suitable complexity assumptions. In Table 1, we use $\mathbf{BQP}^O$ to denote the hardness of the QPL problem. This implies that the quantum kernel Alphatron with quantum data can efficiently solve the QPL problem if there exists an algorithm $O$ that provides the ground state of the concerned Hamiltonians.

- Our simulation results also indicate that some QPL problems are classically hard, yet they could be solved by a quantum learning algorithm with "C-Data". Therefore, "Q-Learning Alg. + C-Data" could also be strictly stronger than "C-Learning Alg. + C-Data".

- Another interesting class is "C-Learning Alg. + Q-Data". Whether it is stronger than "Q-Learning Alg. + C-Data" or strictly weaker than "Q-Learning Alg. + Q-Data" is an interesting problem. Recently,

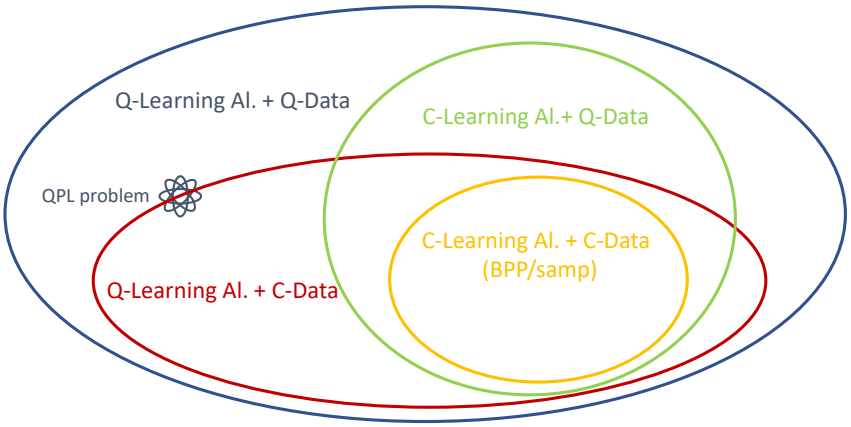


Figure 5: Visualization of the learning ability in terms of data acquiring method and learning algorithms, where "Q-Learning Alg." ("C-Learning Alg.") represents learning algorithms with quantum (classical) computer, "Q-Data" ("C-Data") represents the learning data that are directly observed from physical experiments (can be efficiently simulated by classical Turing machines).

Huang et al. [14] provided a protocol that takes classical shadow representations as the input of classical MLs, and utilizes the trained classical MLs to predict properties of many-body wave functions. Their method belongs to the "C-Learning Alg. + Q-Data" class. However, the efficiency of the classical shadow depends on the locality of the concerned order parameter, *whether QPL problems belong to "C-Learning Alg. + Q-Data" class* is left as an open problem.

We summarize the complexity relationship of these four categories for general problems in Fig. 5.

# 6 Discussions

In this paper, we study the quantum phase learning (QPL) problem using classical and quantum approaches. We prove that with a reasonable conjecture in Ref. [63] that approximate $|\langle 0^n|U|0^n\rangle|^2$ to additive error $\epsilon_c/2^n$ is $\#\ P$-hard, joint with the assumption that PH does not collapse, it is computationally hard to learn QPL problem with classical tractable training data set. On the other hand, we also prove that we can learn QPL problem efficiently if we have a quantum computer. We propose an effective algorithm to illustrate this quantum learning process. The quantum learning algorithm is a quantization of the Alphatron algorithm [45] by leveraging of the quantum kernel method.

Numerically, we apply the quantum learning algorithm to solving QPL problems, and numerical experiments corroborate our theoretical results in a variety of scenarios, including symmetry-protected topological phases and symmetry-broken phases. The numerical results show that our quantum kernel Alphatron algorithm has a good learning performance for quantum properties even with classical training data.

We leave an open problem on whether our quantum learning algorithm has a better performance with a more complicated quantum neural network instead of the quantum kernel. Since our numerical results hint the possibility to efficiently solve QPL problem, is there any rigorous proof to show that QPL problem belongs to "Q-Learning Alg.+C-Data" class? The capability of "C-Learning Alg.+Q-Data" is another interesting open question.

# Acknowledgments

Y. Wu thanks Y. Song for providing valuable discussions. We would like to thank P. Rebentrost to provide some suggestions for the manuscript.

# References


1. Federico Becca and Sandro Sorella. *Quantum Monte Carlo approaches for correlated systems*. Cambridge University Press, 2017.

2. David Ceperley and Berni Alder. Quantum monte carlo. *Science*, 231(4738):555–560, 1986.

3. WMC Foulkes, Lubos Mitas, RJ Needs, and Guna Rajagopal. Quantum monte carlo simulations of solids. *Reviews of Modern Physics*, 73(1):33, 2001.

4. 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(3):1067, 2015.

5. WALTER Kohn. Nobel lectures: Electronic structure of matter–wave functions and density functionals. *Review of Modern Physics*, 71:1253, 1999.

6. Steven R White. Density matrix formulation for quantum renormalization groups. *Physical Review Letters*, 69(19):2863, 1992.

7. Steven R White. Density-matrix algorithms for quantum renormalization groups. *Physical Review B*, 48(14):10345, 1993.

8. Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

9. Juan Carrasquilla and Roger G Melko. Machine learning phases of matter. *Nature Physics*, 13(5):431–434, 2017.

10. Xun Gao and Lu-Ming Duan. Efficient representation of quantum many-body states with deep neural networks. *Nature Communications*, 8(1):1–6, 2017.

11. 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(1):011006, 2018.

12. Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, 2018.

13. Javier Robledo Moreno, Giuseppe Carleo, and Antoine Georges. Deep learning the hohenberg-kohn maps of density functional theory. *Physical Review Letters*, 125(7):076402, 2020.

14. Hsin-Yuan Huang, Richard Kueng, Giacomo Torlai, Victor V Albert, and John Preskill. Provably efficient machine learning for quantum many-body problems. *arXiv:2106.12627*, 2021.

15. 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(6):595–600, 2018.

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

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

18. 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(15):1153, 1983.

19. Frank Pollmann and Ari M Turner. Detection of symmetry-protected topological phases in one dimension. *Physical Review B*, 86(12):125441, 2012.

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

21. Subir Sachdev. Quantum phase transitions. *Physics World*, 12(4):33, 1999.

22. Dominic W Berry, Andrew M Childs, Richard Cleve, Robin Kothari, and Rolando D Somma. Simulating hamiltonian dynamics with a truncated taylor series. *Physical Review Letters*, 114(9):090502, 2015.

23. Guang Hao Low and Isaac L Chuang. Optimal hamiltonian simulation by quantum signal processing. *Physical Review Letters*, 118(1):010501, 2017.

24. Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999.

25. Xiaogang Qiang, Thomas Loke, Ashley Montanaro, Kanin Aungskunsiri, Xiaoqi Zhou, Jeremy L O'Brien, Jingbo B Wang, and Jonathan CF Matthews. Efficient quantum walk on a quantum processor. *Nature Communications*, 7(1):1–6, 2016.

26. Xiaogang Qiang, Yizhi Wang, Shichuan Xue, Renyou Ge, Lifeng Chen, Yingwen Liu, Anqi Huang, Xiang Fu, Ping Xu, Teng Yi, and Jingbo B Wang. Implementing graph-theoretic quantum algorithms on a silicon photonic quantum walk processor. *Science Advances*, 7(9):8375, 2021.

27. Mario Szegedy. Quantum speed-up of markov chain based algorithms. In *45th Annual IEEE symposium on foundations of computer science*, pages 32–41. IEEE, 2004.

28. Sergey Bravyi, David Gosset, and Robert König. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, 2018.

29. Dmitri Maslov, Jin-Sung Kim, Sergey Bravyi, Theodore J Yoder, and Sarah Sheldon. Quantum advantage for computations with limited space. *Nature Physics*, 17:894–897, 2021.

30. Bujiao Wu, Maharshi Ray, Liming Zhao, Xiaoming Sun, and Patrick Rebentrost. Quantum-classical algorithms for skewed linear systems with an optimized hadamard test. *Physical Review A*, 103(4):042422, 2021.

31. Bujiao Wu, Jinzhao Sun, Qi Huang, and Xiao Yuan. Overlapped grouping measurement: A unified framework for measuring quantum states. *arXiv:2105.13091*, 2021.

32. Yusen Wu, Chunyan Wei, Sujuan Qin, Qiaoyan Wen, and Fei Gao. Quantum restricted boltzmann machine universal for quantum computation. *arXiv:2005.11970*, 2020.

33. Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

34. Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.

35. Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical Review Letters*, 122(4):040504, 2019.

36. Maria Schuld. Quantum machine learning models are kernel methods. *arXiv2101.11020*, 2021.

37. Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17:1013–1017, 2021.

38. Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit born machines. *Physical Review A*, 98(6):062324, 2018.

39. Carsten Blank, Daniel K Park, June-Koo Kevin Rhee, and Francesco Petruccione. Quantum classifier with tailored quantum kernel. *npj Quantum Information*, 6(1):41, 2020.

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

41. Hsin-Yuan Huang, Richard Kueng, and John Preskill. Information-theoretic bounds on quantum advantage in machine learning. *Physical Review Letters*, 126(19):190505, 2021.

42. Tobias Haug, Chris N Self, and MS Kim. Large-scale quantum machine learning. *arXiv:2108.01039*, 2021.

43. Patrick Rebentrost, Miklos Santha, and Siyi Yang. Quantum alphatron. *arXiv:2108.11670*, 2021.

44. Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

45. 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.

46. Adam Tauman Kalai and Ravi Sastry. The isotron algorithm: High-dimensional isotonic regression. In *COLT*. Citeseer, 2009.

47. Ruslan Shaydulin and Stefan M Wild. Importance of kernel bandwidth in quantum machine learning. *arXiv:2111.05451*, 2021.

48. Junyu Liu, Francesco Tacchino, Jennifer R Glick, Liang Jiang, and Antonio Mezzacapo. Representation learning via quantum neural tangent kernels. *arXiv:2111.04225*, 2021.

49. Norihito Shirai, Kenji Kubo, Kosuke Mitarai, and Keisuke Fujii. Quantum tangent kernel. *arXiv:2111.02951*, 2021.

50. Kouhei Nakaji, Hiroyuki Tezuka, and Naoki Yamamoto. Quantum-enhanced neural networks in the neural tangent kernel framework. *arXiv:2109.03786*, 2021.

51. Sofiene Jerbi, Lukas J Fiderer, Hendrik Poulsen Nautrup, Jonas M Kübler, Hans J Briegel, and Vedran Dunjko. Quantum machine learning beyond kernel methods. *arXiv preprint arXiv:2110.13162*, 2021.

52. Teresa Sancho-Lorente, Juan Román-Roche, and David Zueco. Quantum kernels to learn the phases of quantum matter. *arXiv:2109.02686*, 2021.

53. Long Hin Li, Dan-Bo Zhang, and ZD Wang. Quantum kernels with squeezed-state encoding for machine learning. *arXiv:2108.11114*, 2021.

54. 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(3):032416, 2021.

55. Jonas M Kübler, Simon Buchholz, and Bernhard Schölkopf. The inductive bias of quantum kernels. *arXiv:2106.03747*, 2021.

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

57. 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. *arXiv:2105.02276*, 2021.

58. Stefan Klus, Patrick Gelß, Feliks Nüske, and Frank Noé. Symmetric and antisymmetric kernels for machine learning problems in quantum physics and chemistry. *arXiv:2103.17233*, 2021.

59. Xinbiao Wang, Yuxuan Du, Yong Luo, and Dacheng Tao. Towards understanding the power of quantum kernels in the nisq era. *arXiv:2103.16774*, 2021.

60. Riikka Huusari and Hachem Kadri. Entangled kernels–beyond separability. *arXiv:2101.05514*, 2021.

61. Takeru Kusumoto, Kosuke Mitarai, Keisuke Fujii, Masahiro Kitagawa, and Makoto Negoro. Experimental quantum kernel machine learning with nuclear spins in a solid. *arXiv preprint arXiv:1911.12021*, 2019.

62. Sergey Bravyi, David Gosset, and Ramis Movassagh. Classical algorithms for quantum mean values. *Nature Physics*, 17(3):337–341, 2021.

63. Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. On the complexity and verification of quantum random circuit sampling. *Nature Physics*, 15(2):159–163, 2019.

64. Julia Kempe, Alexei Kitaev, and Oded Regev. The complexity of the local hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

65. Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing*, STOC '11, page 333–342, New York, NY, USA, 2011. Association for Computing Machinery.

66. 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.

67. Juan Carlos Garcia-Escartin and Pedro Chamorro-Posada. Swap test and hong-ou-mandel effect are equivalent. *Physical Review A*, 87(5):052330, 2013.

68. 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(15):3666, 2020.

69. 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(22):224422, 2001.

70. Larry Stockmeyer. On approximation algorithms for# p. *SIAM Journal on Computing*, 14(4):849–861, 1985.

71. Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

72. L Welch. Error correction for algebraic block codes. In *IEEE Int. Symp. Inform. Theory*, 1983.

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

74. 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. *arxiv:2109.08062*, 2021.

75. Changsu Cao, Jiaqi Hu, Wengang Zhang, Xusheng Xu, Dechin Chen, Fan Yu, Jun Li, Hanshi Hu, Dingshun Lv, and Man-Hong Yung. Towards a larger molecular simulation on the quantum computer: Up to 28 qubits systems accelerated by point group symmetry. *arxiv:2109.02110*, 2021.

76. Dave Wecker, Matthew B Hastings, Nathan Wiebe, Bryan K Clark, Chetan Nayak, and Matthias Troyer. Solving strongly correlated electron models on a quantum computer. *Physical Review A*, 92(6):062318, 2015.

77. 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):1–12, 2021.

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

79. 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):1–6, 2018.

# A Architecture for Haar random circuit distribution

In this section, we give the explicit definition for the architecture of Haar random circuit distribution.

DEFINITION A.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 A.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.*

# B Proof of theorems

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

## B.1 Proof of Lemma 1

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

LEMMA 2 (Stockmeyer [70]). *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 $\alpha$ such that*

$$(1-\epsilon)p \leq \alpha \leq (1+\epsilon)p \tag{10}$$

*for the value*

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

*if the function $f$ can be computed efficiently given $x$.*

**Conjecture 1.** (Ref. [63]) *There exists an $n$-qubit quantum circuit $U$ such that the following task is # P-hard: approximate $|\langle 0^n|U|0^n\rangle|^2$ to additive error $\epsilon_c/2^n$ with probability $\frac{3}{4} + \frac{1}{\mathrm{poly}(n)}$.*

*Proof of Lemma 1.* For a ground state $|\phi\rangle = U|0^n\rangle$ of $H(\boldsymbol{a})$ satisfies Conjecture 1, we can project it to any computational basis $|\boldsymbol{j}\rangle$ with probability $p(\boldsymbol{j}) = |\langle \boldsymbol{j}|\phi\rangle|^2$. The *hiding argument* shows that if one can approximate the probability $p(\boldsymbol{j})$, then one can approximate $p(0^n) = |\langle 0^n|\phi\rangle|^2$. Therefore Conjecture 1 suggests that approximating the $p(\boldsymbol{j})$ to additive error $2^{-\mathrm{poly}(n)}$ is # P-hard.

Here, we can construct a series of observable $\mathcal{M}$ enabling Lemma 1 holds. Let the observable set $\{\mathcal{M}(\boldsymbol{s})|\mathcal{M}(\boldsymbol{s}) = Z_1^{s_1} \otimes Z_2^{s_2} \otimes \cdots \otimes Z_n^{s_n}\}$, 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

$$o_{\boldsymbol{s}} = \langle\phi|\mathcal{M}(\boldsymbol{s})|\phi\rangle = \sum_{\boldsymbol{j}} p(\boldsymbol{j})(-1)^{\boldsymbol{j}\cdot\boldsymbol{s}}, \tag{11}$$

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

$$p(\boldsymbol{j}) = \sum_{\boldsymbol{s}} o_{\boldsymbol{s}}(-1)^{\boldsymbol{j}\cdot\boldsymbol{s}}/2^n. \tag{12}$$

According to Conjecture 1, there exists a quantum circuit $U$ and state $|\phi\rangle = U|0^n\rangle$ such that it is $\#P$-hard to approximate $p(\boldsymbol{j}) = |\langle \boldsymbol{j}|\phi\rangle|^2 := |\langle \boldsymbol{j}|U|0^n\rangle|^2$ with additive error $\frac{1}{2^n \mathrm{poly}(n)}$. On the other hand, 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 [70]. Considering $\mathrm{BPP} \subseteq \mathrm{P/poly}$, this yields $\mathrm{P}^{\#\mathrm{P}} \subseteq \mathrm{BPP}^{\mathrm{NP}^{\mathrm{BPP}}} \subseteq \mathrm{BPP}^{\mathrm{NP}}/\mathrm{poly}$. 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 [71].

Therefore, there does not exist a classical algorithm that can efficiently calculate $o_{\boldsymbol{s}}$ based on the assumption that PH does not collapse and Conjecture 1 holds. □

## B.2 Proof of Theorem 3

*Proof of Theorem 3.* Notice that if the quantum kernel $Q$ can be exactly calculated, then by Goel and Klivans [45], Alg. 1 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

$$R(h^*) \leq \hat{R}\left(h^{t*}\right) + \mathcal{O}\left(B\sqrt{\frac{1}{N}} + \sqrt{\frac{\log(1/\delta)}{N}}\right), \tag{13}$$

and

$$\hat{R}\left(h^{t*}\right) \leq \mathcal{O}\left(\sqrt{\varepsilon_g} + G\sqrt[4]{\frac{\log(1/\delta)}{N}} + B\sqrt{\frac{\log(1/\delta)}{N}}\right). \tag{14}$$

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. (14) should be replaced with

$$\hat{R}\left(\hat{h}^{t*}\right) \leq \mathcal{O}\left(\sqrt{\varepsilon_g} + G\sqrt[4]{\frac{\log(1/\delta)}{N}} + B\sqrt{\frac{\log(1/\delta)}{N}}\right). \tag{15}$$

where $\hat{h}^t(\boldsymbol{x}) = \sum_{i=1}^m \alpha_i^t \hat{Q}(\boldsymbol{a_i}, \boldsymbol{x})$, 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 $R(h^*)$ is bounded by combining Eq. (13), (14) and (15). By Theorem 3, $\hat{Q}(\boldsymbol{a_i}, \boldsymbol{x})$ is an $\epsilon_Q$ approximation of $Q(\boldsymbol{a_i}, \boldsymbol{x})$, *i.e.*,

$$\left|\hat{Q}(\boldsymbol{a_i}, \boldsymbol{x}) - Q(\boldsymbol{a_i}, \boldsymbol{x})\right| \leq \epsilon_Q \tag{16}$$

with high probability.

For convenience, in the later proof we require for all $i$, $\delta_Q^i = \hat{Q}(\boldsymbol{a_i}, \boldsymbol{x}) - Q(\boldsymbol{a_i}, \boldsymbol{x})$ 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 $\delta_Q^i$ 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 $R(h^*)$ without the assumptions and with a more tedious proof. Then for any $i$, we have

$$\begin{aligned}
-\delta_\alpha^t &= \alpha_i^t - \hat{\alpha}_i^t \\
&= \alpha_i^1 - \hat{\alpha}_i^1 + \frac{1}{N}\sum_{k=1}^{t-1}\left(\hat{h}^k(\boldsymbol{a_i}) - h^k(\boldsymbol{a_i})\right) \\
&= \frac{1}{N}\sum_{k=1}^{t-1}\sum_{j=1}^{N}\left(\hat{\alpha}_j^k \hat{Q}(\boldsymbol{a_j}, \boldsymbol{a_i}) - \alpha_j^k Q(\boldsymbol{a_j}, \boldsymbol{a_i})\right) \\
&= \sum_{k=1}^{t-1}\left(A^k \delta_Q + \bar{Q}_i \delta_\alpha^k + \delta_\alpha^k \delta_Q\right)
\end{aligned} \tag{17}$$

where $A^k = \frac{1}{N}\sum_{j=1}^N \alpha_i^k$, and $\bar{Q}_i = \frac{1}{N}\sum_{j=1}^N Q(\boldsymbol{a_j}, \boldsymbol{a_i})$. We can also obtain the value of $-\delta_\alpha^{t-1}$ by leveraging of Eq. (17) and the recurrence relationship. The following equations follows by subtracting $-\delta_\alpha^{t-1}$ by $-\delta_\alpha^t$,

$$\delta_\alpha^t = \left(\bar{Q}_i - 1\right)\delta_\alpha^{t-1} + A^{t-1}\delta_Q + \delta_\alpha^{t-1}\delta_Q, \tag{18}$$

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

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

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

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

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

Therefore,

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

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

The additive error $\epsilon_Q$ for the quantum kernel $Q\left(\boldsymbol{a}_i,\boldsymbol{x}\right)$ can be bounded to $O\left(\frac{\sqrt{\log(1/\delta)}}{N^{5/4}}\right)$ 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 $t=\mathcal{O}\left(N/\log(1/\delta)\right)$ and $G=\Omega\left(1\right)$. □

### B.3 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. [40]. 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 have $y_i=1$, else $y_i=0$. Specifically, one require:
(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.

## B.4 Proof of Theorem 1

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

LEMMA 3 (Ref. [72]). *Let $q$ be a degree $d$ univariate polynomial over any field $\mathbb{F}$. Suppose $k$ pairs elements $\{(x_i, y_i)\}_{i=1}^k$ in $\mathbb{F}$ are provided, where all $x_i$ distinct with the promise that $y_i = q(x_i)$ for at least $\min(d+1, (k+d)/2)$ points. Then, one can recover $q$ exactly in $\mathrm{poly}(k, d)$ deterministic time.*

LEMMA 4 (Ref. [63]). *There exists an architecture $\mathcal{A}$ so that the following task is #P-hard: Approximate $p(\boldsymbol{j}) = |\langle \boldsymbol{j}|U|0^n\rangle|^2$ to additive error $\pm\epsilon/2^n$ with probability $\frac{3}{4} + \frac{1}{\mathrm{poly}(n)}$ over the choice of $U \sim \mathcal{H}_{\mathcal{A}}$ in time $\mathrm{poly}(n, 1/\epsilon)$.*

*Proof of Theorem 1.* Firstly, we introduce how to construct a testing set $\mathcal{T} = \{(\boldsymbol{x_i}, y_i)\}_{i=1}^M$. Suppose we take a worst-case quantum state $|\phi\rangle = U|0^n\rangle$ generated by a circuit $U$ so that computing $p(\boldsymbol{j}) = |\langle \boldsymbol{j}|\phi\rangle|^2$ to within additive error $2^{-\mathrm{poly}(n)}$ is #$P$-hard. Here, the single- and two- qubit gate structures of $U = U_m U_{m-1}...U_1$ is provided, where $U_k$ denotes the $k$-th single- or two- qubit gate for $k \in [m]$. Then it can multiply each gate $U_k$ by a Haar-random matrix $H_k^{1-x}$, that is, $\widetilde{U}_k(\boldsymbol{x}) = U_k H_k^{1-x}$, where $H_k$ drawn from $\mathcal{H}_{\mathcal{A}}$, and real value $\boldsymbol{x} \in [0, 1]$ can represent a $d$-length bit string $\boldsymbol{x} = x_1 x_2 ... x_d$, that is $\boldsymbol{x} = \sum_{j=1}^d x_j 2^{-j}$. Since $H_k$ is a Haar random matrix, the quantum gate $\widetilde{U}_k(\boldsymbol{x})$ is completely random. In this way, one can construct a bridge between the datum $\boldsymbol{x}$ and its corresponding feature state $|\phi(\boldsymbol{x})\rangle$, that is

$$|\phi(\boldsymbol{x})\rangle = \widetilde{U}(\boldsymbol{x})|0^n\rangle = \prod_{k=1}^{m} \left(U_k H_k^{1-x}\right)|0^n\rangle. \tag{19}$$

Here, if $\boldsymbol{x} = 1$, this gives us back the state $|\phi\rangle$, and if $\boldsymbol{x} \in (0, 1/\mathrm{poly}(n))$ the resulting state $|\phi(\boldsymbol{x})\rangle$ looks almost uniformly random. The 'worst-to-average-case' reduction can be achieved by *proof of contradiction*: Using Taylor series, the matrix $\widetilde{U}_k(\boldsymbol{x}) = U_k H_k^{1-x}$ can be approximated by a polynomial function $V_k(\boldsymbol{x})$ with $K = \mathrm{poly}(n)$ degree with precision $\epsilon = \mathcal{O}(1/K!)$ that is

$$\widetilde{U}_k(\boldsymbol{x}) \approx V_k(\boldsymbol{x}) = U_k H_k \sum_{s=0}^{K} \frac{(-x \log H_k)^s}{s!}. \tag{20}$$

According to the Feynman path integral,

$$\langle \boldsymbol{j}|V_m(\boldsymbol{x})...V_1(\boldsymbol{x})|0^n\rangle = \sum_{s_1,...,s_{m-1} \in \{0,1\}^n} \prod_{k=1}^{m} \langle s_k|V_k(\boldsymbol{x})|s_{k-1}\rangle \tag{21}$$

is a $Km$ degree polynomial function in $\boldsymbol{x}$, where $s_m = \boldsymbol{j}$ and $s_0 = 0^n$. Then

$$p(\boldsymbol{j}, \boldsymbol{x}) = |\langle \boldsymbol{j}|\widetilde{U}(\boldsymbol{x})|0^n\rangle|^2 \tag{22}$$

can be approximated by a $2Km$-degree polynomial function in $\boldsymbol{x}$ with the truncated error $\mathcal{O}(2^{mn}/(K!)^m)$. By Aaronson and Arkhipov [65], it is #$P$-hard to compute $\langle \boldsymbol{j}|V(\boldsymbol{x})|0\rangle$ for most of $V(\boldsymbol{x})$, then it is also #$P$-hard to compute $\langle \boldsymbol{j}|\widetilde{U}(\boldsymbol{x})|0\rangle$ for most of $\widetilde{U}(\boldsymbol{x})$. Then, a testing set $\mathcal{T} = \{\boldsymbol{x_i}, y_i\}_{i=1}^M$ is obtained, where $\boldsymbol{x_i} \in [0, 1/\mathrm{poly}(n))$, feature states $|\phi(\boldsymbol{x_i})\rangle = U(\boldsymbol{x_i})|0^n\rangle$ , $y_i = \langle \mathcal{M}\rangle_{\phi(\boldsymbol{x_i})}$ and the scale of testing set $M = \mathrm{poly}(N)$.

Secondly, we prove that there does not exist efficient classical ML algorithm that can predict $y_i$ for $\boldsymbol{x_i} \in \mathcal{T}$. Given the training set $\mathcal{S}$, the power of classical ML can be characterized as the BPP/samp class [40]. If $y_{\boldsymbol{s},\boldsymbol{x}} = \langle \phi(\boldsymbol{x})|\mathcal{M}(\boldsymbol{s})|\phi(\boldsymbol{x})\rangle$ can be predicted by a classical ML algorithm given $(\boldsymbol{s}, \boldsymbol{x})$, based on Stockmeyer's theorem [70], there exists a $\mathrm{BPP}^{\mathrm{NP}}$ algorithm with a BPP/poly oracle, which can approximate $p(\boldsymbol{j}, x)$ with a multiplicative error $1/\mathrm{poly}(n)$. Combining BPP/poly $\subseteq$ P/poly, we directly have $\mathrm{P}^{\#\mathrm{P}} \subseteq \mathrm{BPP}^{\mathrm{NP}}/\mathrm{poly}$, and this thus yields PH collapses to the second level [71]. □

# C QPL problem and quantum random circuit

In the main text, the proof of Theorem 1 is established on the classical hardness for random circuit sampling problem, and the feature states are thus generated by random circuit states. On other hand, the QPL problem has a similar structure to the ground state problem.

In the field of quantum computation, the Variational Quantum Eigensolver (VQE) is a popular method in approximating the ground state of $\mathcal{H}(\boldsymbol{a})$ [73, 74, 75]. The key idea of VQE is that the parameterized quantum state $|\Psi(\boldsymbol{\theta})\rangle$ is prepared and measured on a quantum computer, and the classical optimizer updates the parameter $\boldsymbol{\theta}$ according to the measurement information. The ground state can be obtained by minimizing the energy

$$E(\boldsymbol{\theta}, \boldsymbol{a}) = \langle \Psi(\boldsymbol{\theta}) | \mathcal{H}(\boldsymbol{a}) | \Psi(\boldsymbol{\theta}) \rangle \tag{23}$$

following the variational principle. Basically, the selection of the ansatz $|\Psi(\boldsymbol{\theta})\rangle$ is flexible, which includes the unitary coupled cluster ansatz [76], alternating layered ansatz [77] and hardware efficient ansatz [78].

The hardware efficient ansatz is composed of single- and two-qubit gates in each repeated layer and it is experimental friendly on near term quantum devices. Following the notation in the Ref. [78], a $D$-depth hardware efficient ansatz is formalized as

$$U_h(\boldsymbol{\theta}) = \prod_{d=1}^{D} U_d(\boldsymbol{\theta}_d) W_d, \tag{24}$$

in which $U_d(\boldsymbol{\theta}_d)$ is the tensor product of $n$ single-qubit rotations and $W_d$ is the entanglement gate. Generally, the construction of $U_h(\boldsymbol{\theta})$ promises it will be close to a random unitary with the increasing of $D$ [79, 77]. From the above discussion, it is reasonable to assume that the QPL problem involves the quantum random circuit, and this is consistent to the condition in Conjecture 1.

# 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 [67] 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 $\epsilon_Q$ and failure probability $\delta$. The expectation of the measurement results of the Destructive-Swap-Test is

$$\langle \phi(\boldsymbol{a_i}) \otimes \phi(\boldsymbol{x}) | \mathbf{SWAP} | \phi(\boldsymbol{a_i}) \otimes \phi(\boldsymbol{x}) \rangle = Q(\boldsymbol{a_i}, \boldsymbol{x}), \tag{25}$$

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.

# Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- JingboWangEPCflat.pdf