On the Expressibility and Overfitting of Quantum Circuit Learning

CHIH-CHIEH CHEN, Grid Inc., 107-0061 Tokyo, Japan
MASAYA WATABE, Engineering Department, The University of Electro-Communications, 182-8585 Tokyo, Japan
KODAI SHIBA, Grid Inc., 107-0061 Tokyo, Japan and Engineering Department, The University of Electro-Communications, 182-8585 Tokyo, Japan
MASARU SOGABE, Grid Inc., 107-0061 Tokyo, Japan
KATSUYOSHI SAKAMOTO, Engineering Department, The University of Electro-Communications, 182-8585 Tokyo, Japan and i-PERC, The University of Electro-Communications, 182-8585 Tokyo, Japan
TOMAH SOGABE, Engineering Department, The University of Electro-Communications, 182-8585 Tokyo, Japan and i-PERC, The University of Electro-Communications, 182-8585 Tokyo, Japan and Grid Inc., 107-0061 Tokyo, Japan

Applying quantum processors to model a high-dimensional function approximator is a typical method in quantum machine learning with potential advantage. It is conjectured that the unitarity of quantum circuits provides possible regularization to avoid overfitting. However, it is not clear how the regularization interplays with the expressibility under the limitation of current Noisy-Intermediate Scale Quantum devices. In this article, we perform simulations and theoretical analysis of the quantum circuit learning problem with hardware-efficient ansatz. Thorough numerical simulations show that the expressibility and generalization error scaling of the ansatz saturate when the circuit depth increases, implying the automatic regularization to avoid the overfitting issue in the quantum circuit learning scenario. This observation is supported by the theory on PAC learnability, which proves that VC dimension is upper bounded due to the locality and unitarity of the hardware-efficient ansatz. Our study provides supporting evidence for automatic regularization by unitarity to suppress overfitting and guidelines for possible performance improvement under hardware constraints.

CCS Concepts: • **Computing methodologies** → **Machine learning**;

Additional Key Words and Phrases: Expressibility, overfitting, regularization, quantum circuit

This work is supported by the New Energy and Industrial Technology Development Organization (NEDO) and the Ministry of Economy, Trade and Industry (METI), Japan, under grant number 17H06293.
Authors' addresses: C.-C. Chen and M. Sogabe, Grid Inc., AO Bldg. 6F., 3-11-7, Kitaaoyama, Minato-ku, Tokyo, Japan 107-0061; email: chen.chih.chieh@gridsolar.jp; M. Watabe, Engineering Department, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo, Japan 182-8585; K. Shiba, Grid Inc., AO Bldg. 6F., 3-11-7, Kitaaoyama, Minato-ku, Tokyo, Japan 107-0061 and Engineering Department, The University of Electro-Communications, 182-8585 Tokyo, Japan; K. Sakamoto, Engineering Department, The University of Electro-Communications, 182-8585 Tokyo, Japan and i-PERC, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo, Japan 182-8585; T. Sogabe, Engineering Department, The University of Electro-Communications, 182-8585 Tokyo, Japan and i-PERC, The University of Electro-Communications, 1-5-1 Chofugaoka, Chofu, Tokyo, Japan 182-8585 and Grid Inc., AO Bldg. 6F., 3-11-7, Kitaaoyama, Minato-ku, Tokyo, Japan 107-0061; email: sogabe@uec.ac.jp.

2643-6817/2021/06-ART8 $15.00
https://doi.org/10.1145/3466797

ACM Reference format:
Chih-Chieh Chen, Masaya Watabe, Kodai Shiba, Masaru Sogabe, Katsuyoshi Sakamoto, and Tomah Sogabe. 2021. On the Expressibility and Overfitting of Quantum Circuit Learning. *ACM Trans. Storage* 2, 2, Article 8 (June 2021), 24 pages.
https://doi.org/10.1145/3466797

1 INTRODUCTION

1.1 Background

Engineering quantum states for data manipulations is suggested as an approach to efficiently tackle computational problems beyond the limitation of classical computers [1–3]. Fault-tolerant quantum computers are expected to be able to solve some problems that are difficult for classical computations to do [4, 5]. Recent hardware technology progresses enable the operations of coupled qubits at a scale beyond the simulation capability of most classical computers [6–9]. However, the current quantum devices are limited by coherence time and the error rates, which are still above the error-correction threshold. Hence, fault-tolerant quantum computation for practical applications cannot be performed yet [10, 11]. This situation is termed the Noisy-Intermediate-Scale Quantum (NISQ) era [12]. In the NISQ era, quantum-classical hybrid algorithms, quantum heuristics, and variational methods are expected to play an important role [13] for optimization, chemistry, and machine learning applications. The quantum circuit is employed as a high-dimensional function approximator to tackle the curse of dimensionality. For optimization problems, the Quantum Approximate Optimization Algorithm (QAOA) and Quantum Alternating Operator Ansatz are proposed [14, 15]. For chemistry applications [16, 17], various variational-based quantum chemistry algorithms are demonstrated [18–21]. For machine learning applications, various types of quantum circuits have also been proposed to model learning machines [22–28, 56].

While using a quantum circuit as a subroutine for the classical machine learning algorithms could provide possible quantum advantages, the issues that occur in classical machine learning should also be addressed in the hybrid quantum-classical circuit learning. In particular, one of the goals of learning theory is to identity the proper way to prevent overfitting [29, 53, 57, 58]. One cause of overfitting is that the learning network has too much expressibility; a method to avoid such overfitting is to decrease the expressibility by a regularization technique such as dropout regularization in deep learning [30, 31]. Importantly, it was observed and conjectured that the unitarity of quantum evolution provides possible automatic regularization to avoid overfitting [32]. This is considered one of the potential quantum advantages in machine learning applications. However, it is not clear whether this observation can survive in other settings, like different kinds of machine learning applications, or the limitation in the coupling between qubits in the hardware. Also, the theoretical reason of the conjecture is not well understood yet.

1.2 Our Results

In this work, we address these questions by performing simulations and theoretical studies in the scenario of quantum circuit learning (QCL). Rather than the Ising coupling used in [32], we focus on more realistic hardware-efficient ansatz (HEA), which is often used in variational quantum eigensolvers [33, 54, 55]. To understand the effect of increasing circuit depth on the learning performance associated with the overfitting issue, we study the expressibility of the circuit under various hyperparameters [36–39]. Note that previous studies focus on the comparison between different ansatzes [36]. However, these studies did not analyze the relation between the overfitting issue and the expressibility in the quantum machine learning scenario. In this work, for specific regression and classification problems, we observe that the expressibility saturates when the circuit depth

is increased; then, the generalization error also saturates. This fact supports the existing conjecture of automatic regularization under the hardware-efficient circuit implementation. Moreover, to strengthen this indirect answer to the question, which is limited to the small-size problems, we give a general theory for studying the Probably Approximately Correct (PAC) learnability of quantum circuit hypothesis set through the analysis of the Vapnik-Chervonenkis (VC) dimension [29, 40–42, 57, 58]. We then prove that the VC dimension of a particular yet general HEA saturates as a function of the circuit depth due to the unitarity and data locality of HEA. The combination of these numerical and theoretical investigations gives a basis for understanding the automatic regularization to avoid the overfitting issue in QCL.

1.3 Related Works

Mitarai et al. [32] propose the QCL ansatz in this work with a different entangler and observe the avoidance of overfitting for regression but do not study the overfitting for classification. Aaronson [59] studies the learnability of quantum states based on the fat-shattering dimension instead of the VC dimension. Arunachalam and de Wolf [60, 61] study the quantum sample complexity (i.e., the samples are coherent quantum states instead of classical data). The KL-expressibility used in this work is proposed by Sim et al. [36]. Sim et al. [36], Nakaji and Yamamoto [37], and Hubregtsen et al. [38] contain discussions of the expressibility of various ansatzes by using KL-expressibility but do not discuss learnability. Sim et al. [36] observe the saturation of KL-expressibility of deep variational quantum circuits and discuss its relation to the entangling capability. Sim et al. [36] also discuss the difference between controlled-X and controlled-Z entanglers. Nakaji and Yamamoto [37] study the trainability of quantum circuits and give analytical results for frame potential. Goto et al. [52], Schuld et al. [39], and Perez-Salinas et al. [66] study the expressive power and universality of quantum circuit approximators. Schuld et al. [39] also point out the expressibility limitation of parallel encoding due to the spectrum limitation; they further discuss the situation for sequential encoding [66]. Schuld et al. [39] also suggest regularization of QCL by limiting the Fourier frequency. Huang et al. [62] and Abbas et al. [67] provide data-dependent generalization error bounds, while our result based on VC-analysis is independent of input distribution. In the process of revising this article, Gyurik et al. [69] give upper bounds for the VC dimension and fat-shattering dimension. Their upper bound for the VC dimension is bounded by the rank of observables, while in this work we are interested in the upper bound due to limited circuit depth.

1.4 Organization

This article is organized as follows. Section 2 is devoted to preliminaries, describing the circuit ansatz, the regression and classification problems considered in this article, the definitions of the measure for expressibility, and the definition of the VC dimension. As a by-product, a simulator-based backpropagation method is developed for gradient calculation [43, 63], which can be applied to classical simulations of quantum learning algorithms [34, 35]. Section 3 is the main body of the article. First, we show the performance of the quantum circuit model, followed by the numerical observation on the saturation of expressibility. VC dimension analysis is provided in Section 3.3, giving the theoretical basis of the saturation of expressibility. Section 3.4 combines these results to provide an in-depth investigation focusing on the overfitting issue. Section 3.5 provides more numerical results for sensitivity studies. We present our conclusions in Section 4.

2 METHODS

2.1 Circuit Ansatz

Given training dataset $\{(\boldsymbol{x}_i, f(\boldsymbol{x}_i))\}$, a supervised learning algorithm provides a method to compute a model $g(\boldsymbol{x})$ from a hypothesis set $\{h_{\boldsymbol{\theta}}(\boldsymbol{x})\}$ for good prediction accuracy with respect

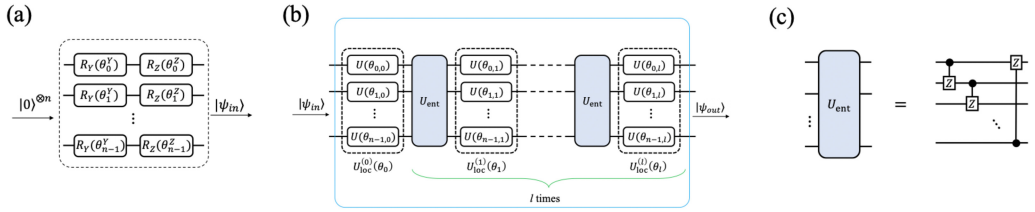

Fig. 1. Hardware-efficient ansatz for quantum learning circuit. (a) Preparation of input state by a unitary input gate $U_{\text{in}}(x)$ exemplified by a series of rotation gates. (b) Quantum circuit to present variational parameter state $W(\boldsymbol{\theta})$. l denotes the depth of quantum circuit. (c) Quantum entanglement circuit, where U_{ent} gate is composed of CZ gates with control qubit j and target qubit $(j+1)$ *mod* n, where $j \in \{0, \ldots, n-1\}$. Figure taken from [68].

to the usually unknown target function $f(\boldsymbol{x})$. In a quantum circuit learning algorithm [32], the hypothesis set is given by quantum circuits parametrized by $\boldsymbol{\theta}$. The measurement results of the output quantum state $|\psi_{\text{out}}(\boldsymbol{\theta})\rangle$ are then used to calculate some expectation values for model predictions. The hardware efficient quantum circuit used for learning for an n-qubits system is depicted in Figure 1. The quantum circuit consists of a unitary input layer $U_{\text{in}}(\boldsymbol{x})$ that creates an input state from classical input data $\boldsymbol{x}$ and a unitary transformation $W(\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta}$. The output state of the circuit is then given by $|\psi_{\text{out}}(\boldsymbol{\theta})\rangle = W(\boldsymbol{\theta})U_{\text{in}}(\boldsymbol{x})|0\rangle$. We use local unitaries $U_{\text{in}}(\boldsymbol{x}) = \otimes_{j=0}^{n-1} R_Z(\theta^Z_{j,(\text{in})}(\boldsymbol{x}))R_Y(\theta^Y_{j,(\text{in})}(\boldsymbol{x}))$ for the input layer. The transformation $W(\boldsymbol{\theta}) = U^{(l)}_{\text{loc}}(\theta_l)U_{\text{ent}} \cdots U^{(1)}_{\text{loc}}(\theta_1)U_{\text{ent}}U^{(0)}_{\text{loc}}(\theta_0)$ is formed by alternatively applying l-layers of local unitaries $U^{(k)}_{\text{loc}}(\theta_k)$ and entanglers U_{ent}, where l is the depth of the circuit. Throughout the article, we may use l or L to denote circuit depth. Each local unitary layer $U^{(k)}_{\text{loc}}(\theta_k) = \otimes_{j=0}^{n-1} R_Z\left(\theta^Z_{j,k}\right) R_Y\left(\theta^Y_{j,k}\right)$ is a product of single-qubit rotations. For hardware-efficient ansatz, we use nearest neighbor controlled-Z gates (CZ) for entanglers, $U_{\text{ent}} = \prod_{j=0}^{n-1} CZ\ (j, (j+1)\ mod\ n)$. This entangler is different from the Ising entangler used in [32].

2.2 Problem Formulation—Regression and Classification

For regression circuit, one-dimensional data x is input by setting circuit parameters as $\theta^Z_{j,(\text{in})}(x) = \cos^{-1}(x^2)$ and $\theta^Y_{j,(\text{in})}(x) = \sin^{-1}(x)$ for all $j \in \{0, 1, \ldots, n-1\}$. The expectation value of observable Pauli Z for the zero-th qubit was obtained from the output state $|\psi_{out}\rangle$ of the circuit. The regression prediction is then given by twice the Z expected value of the 0-th qubit. A conventional least square loss function is adopted as $\mathcal{L} = \sum_{i=1}^{m} \frac{1}{2}(2\langle Z_0(\boldsymbol{x}_i)\rangle - f(\boldsymbol{x}_i))^2$, where m is the number of training data. For classification, the input layer parameters for two components data $\boldsymbol{x}_i = (x_{i,0}, x_{i,1})$ are given by $\theta^Z_{j,(\text{in})}(\boldsymbol{x}_i) = \cos^{-1}((x_{i,j\ mod\ 2})^2)$ and $\theta^Y_{j,(\text{in})}(\boldsymbol{x}_i) = \sin^{-1}(x_{i,j\ mod\ 2})$ for $j \in \{0, 1, \ldots, n-1\}$. The loss function is the cross-entropy loss function $\mathcal{L} = \sum_{i=1}^{m}\sum_{j=0}^{1} f(\boldsymbol{x}_{i,j})\log(y_{i,j})$, where $y_{i,j} = e^{\langle Z_j(\boldsymbol{x}_i)\rangle} / (e^{\langle Z_0(\boldsymbol{x}_i)\rangle} + e^{\langle Z_1(\boldsymbol{x}_i)\rangle})$ is given by the softmax function for $j \in \{0, 1\}$.

2.3 Error Backpropagation Technique

To optimize the circuit parameters efficiently, we developed a simulator-based error backpropagation [43, 63, 68]. The details of the method are presented in Appendix A.1 and [68]. We use capital indices to denote binary string $I = (i_0, \ldots, i_{n-1})$. Circuit parameters $\theta_{p,l}$ denotes the rotation angle of local unitary at qubit p and layer l. The derivatives of loss functions with respect to circuit parameters $\theta_{p,l}$ are given by $\frac{\partial \mathcal{L}}{\partial \theta_{p,l}} = 2\text{Re}[\sum_I \frac{\partial \mathcal{L}}{\partial p^I_\theta} \frac{\partial p^I_\theta}{\partial C^I_\theta} \frac{\partial C^I_\theta}{\partial \theta_{p,l}}]$, where p^I_θ is the probability

density of the amplitude C_θ^I for output state $|\psi_{out}\rangle = \sum_I C_\theta^I |I\rangle$. The derivatives of the amplitude can be calculated using $\frac{\partial C_\theta^{I_L}}{\partial \theta_{p,l}} = C_\theta^{I_L}\ [U(\theta_{p,l}) \rightarrow \frac{\partial U(\theta_{p,l})}{\partial \theta_{p,l}}]$, which means that we substitute the local single-qubit rotation $U(\theta_{p,l})$ which depends on $\theta_{p,l}$ by its derivatives. The expression can be directly evaluated by a backpropagation-like series of unitary multiplications of the original circuits in a simulator. In these expressions, we have suppressed the Y/Z symbols for simplicity.

2.4 Two Measures: KL-Based Expressibility and VC Dimension

To analyze the expressibility of the quantum circuit ansatz, KL divergence between the fidelity distribution of random-sampled quantum circuits and the Haar distribution is calculated [37]. Since the Haar distribution describes the best possible output of random unitary matrices, the KL divergence provides a metric for the expressibility of a given quantum circuit ansatz. For a given circuit C, random circuit parameters $\theta, \phi \in [0, 2\pi]$ are drawn from uniform distribution. The fidelity is then calculated by $F = |\langle\psi_\theta|\psi_\phi\rangle|^2$. The KL divergence between the ansatz and the Haar distribution is $D_{KL}(P_C(F)|\ |P_{Haar}(F)) = \int_0^1 P_C(F)\log(\frac{P_C(F)}{P_{Haar}(F)})dF$, where $P_{Haar}\ (F) = \ (n-1)(1-F)^{n-2}$. The smaller $D_{KL}(P_C(F)||P_{Haar}(F))$ means that the distribution is closer to the Haar distribution, implying that the ansatz provides better expressibility.

The computational feasibility of learning can be theorized in the framework of PAC learning [29, 40–42]. If the VC dimension of a hypothesis set is finite, the generalization error can be bounded by the size of the training dataset with high probability, and the hypothesis set is said to be PAC learnable. The VC generalization error bound [29] used in this work is $E_{out} - E_{in} \leq \sqrt{\frac{8}{N}\ln(\frac{4(1+(2N)^{d_{VC}})}{\delta})}$, where N is the number of training data. $E_{in} = \frac{1}{N}\ \sum_{i=1}^{N}[\![h(\boldsymbol{x_i}) \neq f(\boldsymbol{x_i})]\!]$ is the in-sample error, where $[\![.]\!]$ is the Iverson bracket. $E_{out} = \mathbb{P}[h(\boldsymbol{x}) \neq f(\boldsymbol{x})]$ is the out-of-sample error, where $\mathbb{P}$ is the probability based on input data distribution. δ is the confidence parameter. The number d_{VC} is the VC dimension of the hypothesis set, where higher d_{VC} means higher model complexity and expressive power. The inequality holds with probability $\geq (1 - \delta)$ for any binary target function, any hypothesis set, and any input data probability distribution. The inequality and the VC dimension are the properties of the hypothesis set and are independent of the learning algorithm. Conceptually, the VC dimension is the maximum number of some points that can be shattered by the models in the hypothesis set. We note that other generalization error bounds such as the Rademacher bound exist; we choose the VC bound in this work for its generality and simplicity. The numerical computations for KL-expressibility and VC dimension are based on Qiskit [47].

3 RESULTS AND DISCUSSION

3.1 Performance of Quantum Circuit

In regression tasks, the circuit parameters are set to number of qubits $n = 3$ and circuit depth $l = 3$. We performed regression experiments for various functions to verify the effectiveness of the proposed approach. Figure 2 shows the regression results for two typical target functions. Figure 2(a) is for the target function $f_1\ (x) = x$, which represents a simple linear function; Figure 2(b) is for the target function $f_2\ (x) = e^{-(\frac{x}{0.5})^2} - 0.5$, which represents a Gaussian nonlinear problem. Note that Gaussian function has a convex-concave profile and infinite series expansions for both Taylor and Fourier expansions. Hence, it is more difficult to fit compared with simple polynomial, exponential, and sinusoidal functions. The uniformly distributed noise of strength 0.02 was also added into the target function to simulate realistic situations. The training dataset of size 200 is drawn from uniform distribution on the range [−1,1]. The root mean square errors = $RMSE = \sqrt{\sum_{i=1}^{m}\frac{1}{m}(2\langle Z_0(\boldsymbol{x}_i)\rangle - f(\boldsymbol{x}_i))^2}$ are $RMSE = 0.000665$ for linear function and

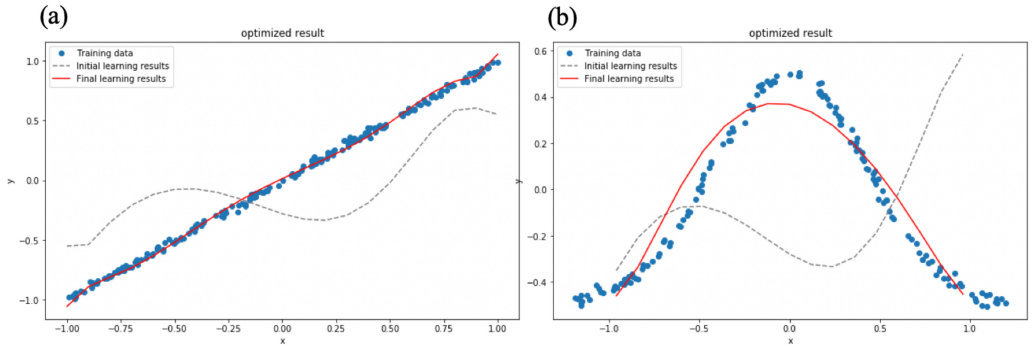

Fig. 2. Regression results using 3 qubits $l = 3$ circuit. Dashed black lines are initial guesses. Blue dots are training data. Red solid lines are learned results. (a) Regression of linear target function $f_1(x) = x$. (b) Regression results for Gaussian target function $f_2(x) = e^{-\left(\frac{x}{0.5}\right)^2} - 0.5$. For each function, a uniformly distributed random noise of strength 0.02 is added to the training data. The number of training data is 200.

$RMSE = 0.004319$ for Gaussian function. A larger error is generally observed at the boundaries for the Gaussian function. Thus, we use a method by drawing the training dataset from an extended range $[-1.1, 1.1]$ to cure this problem.

For the classification problems, we have modified the quantum circuit architecture to accommodate more variational parameters with a larger circuit width and depth. The parameters for the classification problem were $n = 4$ and $l = 6$. Here, we show the experimental results for two-dimensional nonlinear binary classification problems. The dataset was prepared by referring to datasets from Scikit-learn [45]. We consider two examples: make_circles and make_moons. The make_moons dataset possesses more complicated nonlinear features than make_circles. For the proof of concept, the number of training data was set to 200. Half of the data was labeled as "0"; the rest half of the data was labeled as "1." The results are depicted in Figure 3. Figure 3(a) and Figure 3(b) are the results for noiseless and noisy make_circles data, respectively. The Gaussian noise with standard deviation 0.02 is added to the noisy data. For both noiseless and noisy cases, the accuracy $\frac{1}{m}\sum_{i=1}^{m}[\![h(\boldsymbol{x_i}) = f(\boldsymbol{x_i})]\!]$ is estimated to be 1.0 with 200 random testing data, but the testing results on a 30 by 30 grid shows a few misclassified points at the corners. For the make_moons datasets, the situation becomes more complicated due to the increased nonlinearity in the training data. Figure 3(c) and Figure 3(d) are the results for noiseless and noisy make_moons datasets, respectively. The Gaussian noise with standard deviation 0.02 is added to the noisy data. The accuracies are 0.98 for noiseless data and 0.97 for noisy data, estimated by using 200 testing data. The classification mistakes usually occur near the terminal area where the label "0" and label "1" overlapped with each other.

3.2 Saturation of the Expressibility

Further investigation aimed at improving the test accuracy for the make_moons data was conducted. We study the effect by varying the depth of the quantum circuit. We consider that one of the reasons for misclassification occurring in Figure 3(d) would be attributed to the limited expressive power due to the limited depth of the quantum circuit. Therefore, we investigated the effect of quantum circuit depth on the learning accuracy. The results are shown in Figures 4(a) to 4(c). The depths of the quantum circuits are 2, 6, and 10, respectively. The accuracies are 0.915, 0.975, and 0.985, respectively. The number of qubits is n = 4, and the standard deviation of Gaussian noise is 0.02. It can be seen that the 2-layer result has a large misclassification area near the separation

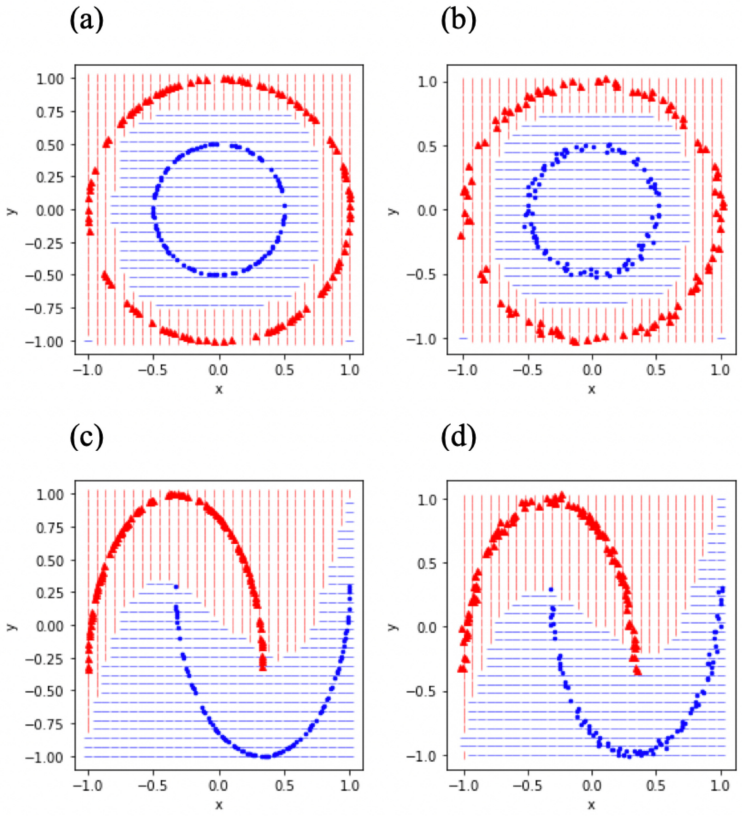

Fig. 3. Results for nonlinear binary classification problem with 4 qubits and depth $l = 6$. Red triangles are the training data for label "0" and blue dots are the data for label "1." Red vertical lines are points predicted to be "0" and blue horizontal lines are points predicted to be "1." (a) Training dataset and test results for make_circles. The testing accuracy = 1.0. (b) Training dataset and test results for noisy make_circles data. The strength of Gaussian noise is 0.02. The testing accuracy = 1.0. (c) Training dataset and test results for make_moons. The testing accuracy = 0.98. (d) Training dataset and test results for noisy make_moons data. The standard deviation of Gaussian noise is 0.02. The testing accuracy = 0.97.

boundary, indicating insufficient expressiveness for the nonlinear feature in the training data. However, with the increase of circuit depth, the classification boundary becomes more nonlinear as shown in Figure 4(b), where the depth of the quantum circuit is 6 layers. Figure 4(c) shows the results obtained at the 10-layer depth; the classification boundary seems to be almost identical to the 6-layer depth one shown in Figure 4(b). This observation indicates the existence of a critical depth, where the learning efficiency is saturated and no further improvement could be obtained for deeper circuits beyond the critical depth. For the current experimental condition with a 4-qubit system and training dataset of size 200, the critical depth is conjectured to be around 6 layers.

We note that increasing circuit depth seems to improve the classification results without overfitting, and saturation of improvements do occur at some critical depth. Similar observation for regression leads to the conjecture of regularization by unitarity in [32]. One question we could ask is: can the expressibility of the quantum circuit help us to explain the disappearance of overfitting? To address this question, we analyze the expressibility and generalization errors for these quantum circuits. Figure 5 shows the expressibility study for different hyperparameters for the

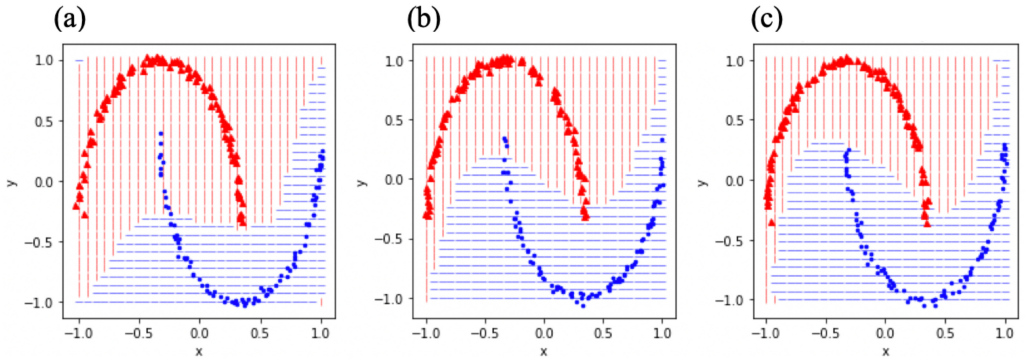

Fig. 4. Improvement of quantum learning classification accuracy on the make_moons data by increasing the circuit depth. Red triangles are the training data for label "0" and blue dots are the data for label "1." Red vertical lines are points predicted to be "0" and blue horizontal lines are points predicted to be "1." (a) Circuit depth is 2 layers and testing accuracy is 0.915. (b) Circuit depth is 6 layers and testing accuracy is 0.975. (c) Circuit depth is 10 layers and testing accuracy is 0.985. The number of qubits is n = 4, and the standard deviation of Gaussian noise is 0.02.

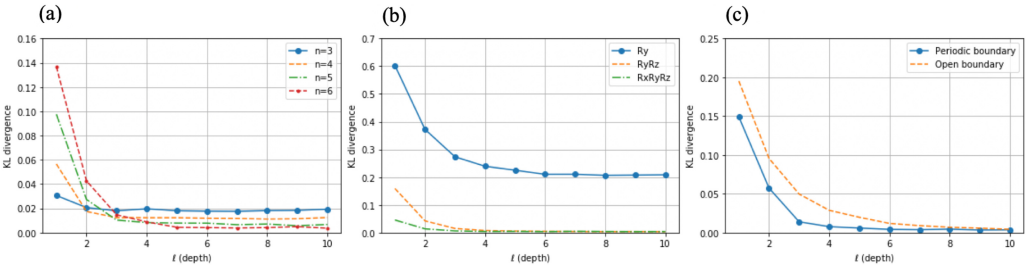

Fig. 5. Expressibility analysis for hardware efficient ansatz. KL divergence $D_{KL}(P_C(F)|\ |P_{Haar}(F)) = \int_0^1 P_C(F)\log(\frac{P_C(F)}{P_{Haar}(F)})dF$ as a function of circuit depth is calculated for different hyperparameters. The number of random circuits is 200. (a) Different numbers of qubits $n = 3, 4, 5, 6$ are denoted by blue solid line with marker, orange dashed line, green dashed dotted line, and red dashed line with marker, respectively. (b) Different local unitaries. Blue solid line with marker, orange dashed line, and green dashed dotted line denotes $U_{\text{loc}}^{(k)}(\theta_k) = \otimes_{j=0}^{n-1} R_Y(\theta_{j,k}^Y)$, $U_{\text{loc}}^{(k)}(\theta_k) = \otimes_{j=0}^{n-1} R_Z(\theta_{j,k}^Z)R_Y(\theta_{j,k}^Y)$, and $U_{\text{loc}}^{(k)}(\theta_k) = \otimes_{j=0}^{n-1} R_Z(\theta_{j,k}^Z)R_Y(\theta_{j,k}^Y)R_X(\theta_{j,k}^X)$, respectively. (c) Different boundary conditions for entangler U_{ent}. Blue solid line and orange dashed line are periodic boundary condition (PBC) and open boundary condition (OBC), respectively.

hardware-efficient ansatz. The KL divergence $D_{KL}(P_C(F)||P_{Haar}(F))$ between random learning circuit fidelity and Haar distribution is plotted as functions of circuit depth. Increasing the circuit depth leads to decreasing of the KL divergence, which suggests improvement of expressibility. Figure 5(a) shows the scaling of different numbers of qubits. We see that for all curves, the expressibility does not improve beyond the circuit depth $l = 8$. This observation supports the conjecture of automatic regularization of quantum circuits. For smaller numbers of qubits, such as $n = 3$ and $n = 4$, the saturation happens faster. For larger numbers of qubits, such as $n = 6$, there is more room for improvement. Figure 5(b) shows the effects on expressibility by changing the local unitary rotations U_{loc} for $n = 6$. We can increase the expressibility without increasing the circuit depth by including more rotation angles. We notice that if the local unitaries have y-rotation

Table 1. VC Dimension Bounds

Ansatz	Bounds
General	$2 \leq d_{VC} \leq \left(2\frac{n}{d}+1\right)^{2d}$
CZ-HEA 1D PBC	$2 \leq d_{VC} \leq \left(2\min\left(\frac{n}{d}, \left\lceil\frac{2L+1}{d}\right\rceil+1\right)+1\right)^{2d}$

Note: Assuming $n \geq d$ and $n\ (mod\ d) = 0$, where n is the number of qubits and d is the feature space dimension. L is the circuit depth. Assuming uniform input layer with alternating feature dimension encoding.

$R_Y$only, the expressibility is limited. The KL divergence reaches the saturation value $D_{KL} = 0.2$. On the other hand, if z-rotation R_Z and x-rotation R_X are included, we observe better expressibility. The gap between R_Y-only ansatz and the other two ansatz cannot be closed by simply increasing the circuit depth. The reason is due to the complex phase degrees of freedom in R_Z that provide extra expressibility. The observation suggests the importance of including complex rotations in the learning ansatz. Figure 5(c) shows the effects for different topologies of entanglers. The coupling of entanglers is limited by hardware constraint of the quantum devices. The KL divergence for the periodic boundary condition (PBC) $U_{\text{ent}} = \prod_{j=0}^{n-1} CZ\ (j, (j+1)\ mod\ n)$ and open boundary condition (OBC) $U_{\text{ent}} = \prod_{j=0}^{n-2} CZ\ (j, j+1)$ are plotted. The periodic boundary condition has better coupling between qubits; hence, the expressibility is generally better as expected. The difference, however, can be cured by increasing the circuit depth to $l = 9$. This shows the limitation on the expressibility of circuits imposed by hardware constraints.

3.3 Limitation of the VC Dimension

To understand the learnability and model complexity of QCL, we study the VC dimension of the model. For simplicity, we assume that $n \geq d$ and $n\ (mod\ d) = 0$, where n is the number of qubits and d is the feature space dimension. The input layer consists of Ry and Rz rotation. Each $R_Y(\theta^Y_{j,(\text{in})}(\boldsymbol{x}))\ R_Z(\theta^Z_{j,(\text{in})}(\boldsymbol{x}))$ encodes one feature dimension alternatively by $\theta^Y_{j,(\text{in})}(\boldsymbol{x}) = \sin^{-1}(x_{j\ (mod\ d)}) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and $\theta^Z_{j,(\text{in})}(\boldsymbol{x}) = \cos^{-1}((x_{j\ (mod\ d)})^2) \in [0, \frac{\pi}{2}]$ for j-th qubit. Each dimension is encoded by $\frac{n}{d}$ qubits. We also assume that binary classification is done by thresholding the linear combination $f\ (\langle Z_0\rangle, \langle Z_1\rangle, \ldots, \langle Z_{d-1}\rangle) = \sum_{i=0}^{d-1} \gamma_i \langle Z_i\rangle$ for real numbers γ_i instead of the softmax function $softmax(\langle Z_i\rangle)$ used in our experiment. The variational parameter circuit can be any general unitary. Since the resulting function $f(\langle Z_0\rangle, \langle Z_1\rangle, \ldots, \langle Z_{d-1}\rangle)$ is a real trigonometric polynomial [39], we can apply the Dudley-Cover theorem to obtain upper bounds for the VC dimension [48–50]. For hardware-efficient ansatz, we further obtain an upper bound as a function of circuit depth by considering the light-cone limitation of tensor networks [51]. The light-cone limitation is due to the unitarity and locality of the HEA variational quantum circuit. The details and proofs are explained in Appendix A.2. The results are summarized in Table 1. Some examples of the VC dimension upper bounds d^*_{VC} are plotted against circuit depth in Figure 6(a). d^*_{VC} provides a precise tool to understand the saturation of model complexity. The critical depth for the saturation of d^*_{VC} is exactly solvable by the equation $\frac{n}{d} \leq [\frac{2L^*+1}{d}] + 1$ for CZ-HEA 1D PBC. For example, $L^* = 1$ for $n = 4$ and $d = 2$. We may compare the scaling of the VC dimension upper bound d^*_{VC} to the KL-based expressibility. For comparison, we define the rescaled KL-expressibility by $D^*_{KL} = d^*_{VC}\ (\frac{D^{max}_{KL} - D_{KL}}{D^{max}_{KL} - D^{min}_{KL}})$. Note that a larger D^*_{KL} means higher expressibility, while the original D_{KL} is smaller for higher expressibility. The comparison is plotted in Figures 6(b), 6(c), and 6(d). For increasing circuit depth, we observe that the saturation of the VC dimension upper bound is similar to the saturation of the KL-expressibility. This observation explains the saturation of the KL expressibility.

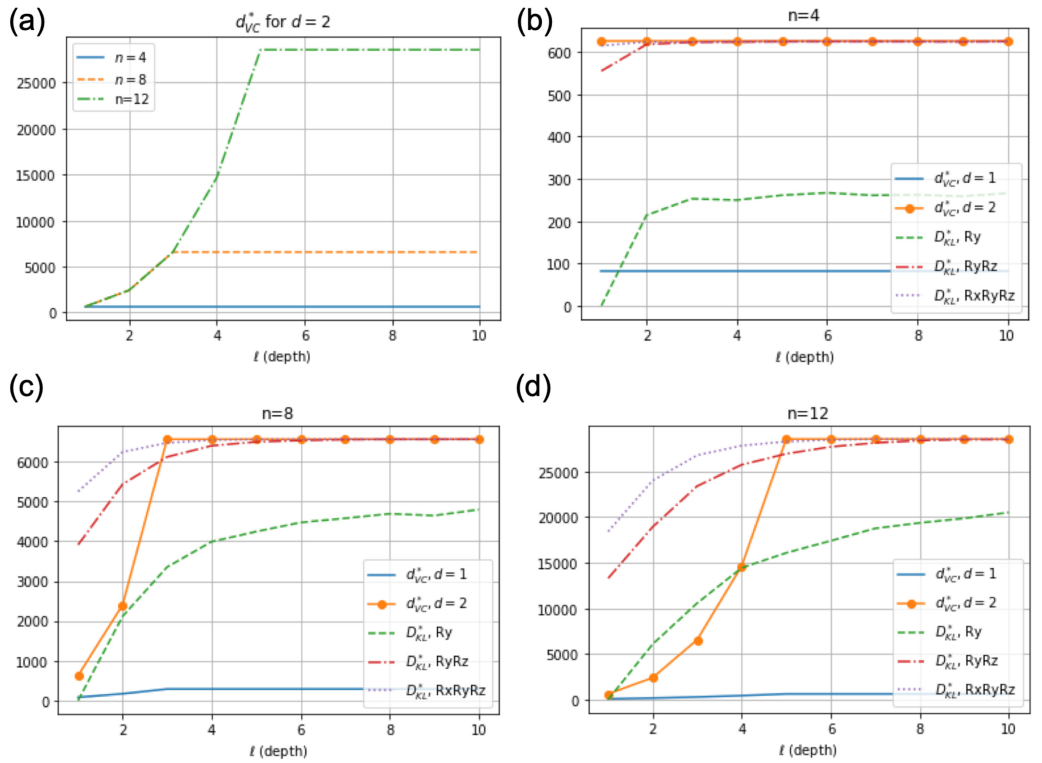

Fig. 6. Comparing the saturation of VC dimension upper bound and rescaled KL-expressibility with respect to circuit depth for CZ-HEA 1D PBC. The VC dimension upper bound is calculated from the formula in Table 1. The KL-expressibility is computed by sampling 400 random circuits. Larger D^*_{KL} means higher expressibility.

Note that, according to the derivations, the KL-expressibility is a property of the circuit and is independent of the feature space dimension and the input layer encoding, while our VC dimension bound depends on both feature space dimension and input layer encoding. For example, one can use a single qubit to encode $\sin(\theta x)$ with $\theta \in \mathbb{R}$, $x \in \mathbb{R}$ to achieve an infinite VC dimension hypothesis set [57, 58]. We may use KL-expressibility and VC dimension as two measures of model complexity depending on whether we want to take the input encoding into account. We further note that the assumption on the output function $f\ (\langle Z_0\rangle, \langle Z_1\rangle, \ldots, \langle Z_{d-1}\rangle) = \sum_{i=0}^{d-1} \gamma_i \langle Z_i\rangle$ can be relaxed to some real polynomials of $\langle Z_i\rangle$, which gives higher-degree polynomial upper bounds for the VC dimension. Finally, we have focused on the limitation of QCL in this work. The universality of QCL function approximation is discussed in [39, 52].

3.4 Overfitting Issue

We study the generalization error and overfitting issue through simulations in various parameter settings. All classification experiments in this section are using make_moons data. All regression experiments are using the linear target function.

First, we examine the scaling of training error and testing error with respect to the depth of circuits in Figure 7. If overfitting occurs, the training error would be a decreasing function of the

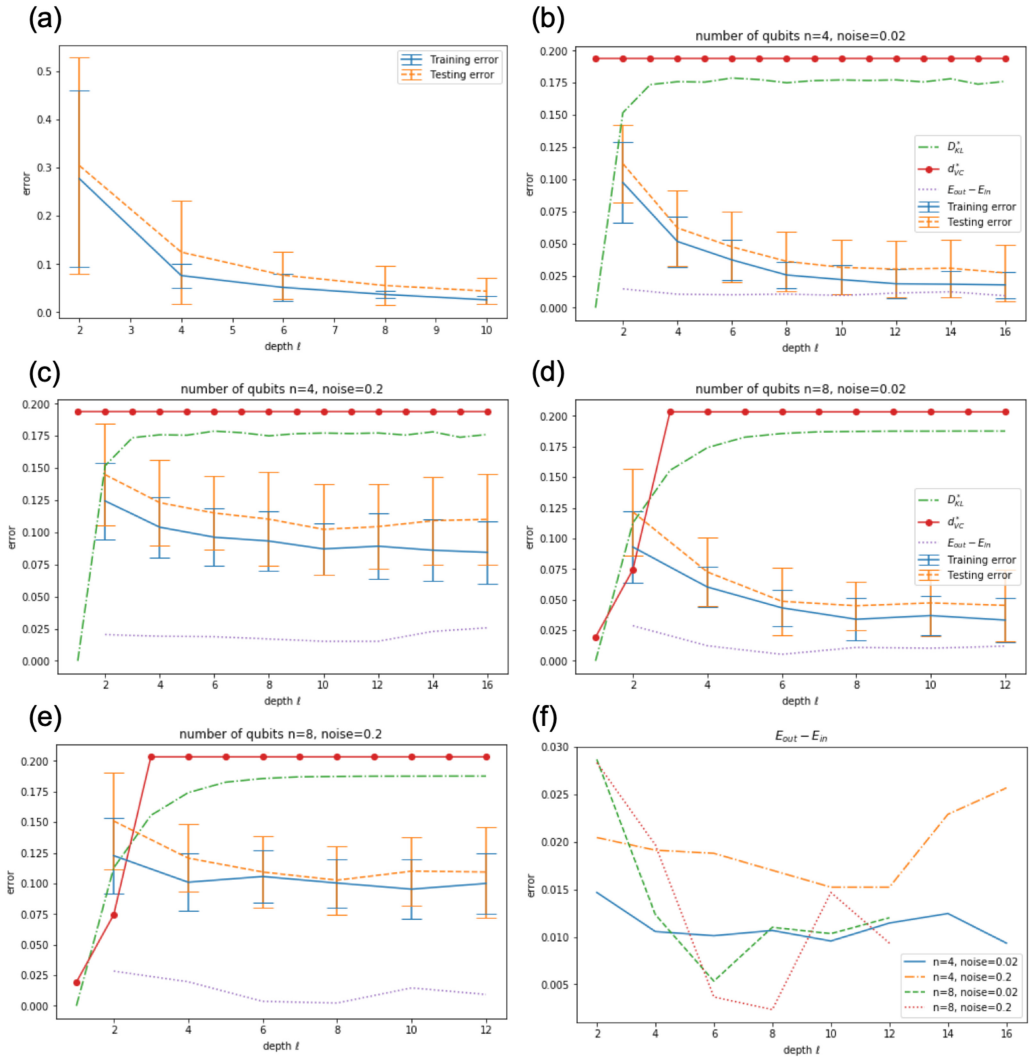

Fig. 7. Training, testing, and generalization error scaling with respect to circuit depth for (a) regression and (b) to (e) classification. (a) Regression: Blue solid line is training error. Orange dashed line is testing error. The number of training/classification data is 8/8, and the target function is linear function. The noise is 0.1 times uniform distribution. The number of qubits is $n = 3$. Error bars denote one-sigma fluctuation for 20 independent runs. (b) to (e) Classification: Blue solid lines are training errors. Orange dashed lines are testing errors. Purple dotted lines are the generalization errors $E_{out} - E_{in}$. The VC dimension upper bound (red solid lines with circle markers) and rescaled KL-expressibility (green dashed dotted lines) are arbitrarily rescaled as references for the model complexity. The number of training/classification data is 100/100, and the target dataset is make_moons. (b) $n = 4$ and noise = 0.02. Error bars denote one-sigma fluctuation for 90 independent runs. (c) $n = 4$ and noise = 0.2. Error bars denote one-sigma fluctuation for 90 independent runs. (d) $n = 8$ and noise = 0.02. Error bars denote one-sigma fluctuation for 30 independent runs. (e) $n = 8$ and noise = 0.2. Error bars denote one-sigma fluctuation for 30 independent runs. (f) Generalization errors for data in (b) to (e). Blue solid line is $n = 4$ and noise = 0.02. Orange dashed dotted line is $n = 4$ and noise = 0.2. Green dashed line is $n = 8$ and noise = 0.02. Red dotted line is $n = 8$ and noise = 0.2.

model complexity, while the testing error would be increasing. We do not observe significant overfitting for both regression (Figure 7(a)) and classification with small data noise 0.02 (Figures 7(b) and 7(d)) errors, even when the number of parameters is much larger than the number of training data. The error is defined to be $Err = \sum_{i=1}^{m} \frac{1}{m}(2\langle Z_0(\boldsymbol{x}_i)\rangle - f(\boldsymbol{x}_i))^2$ for regression and $Err = (1 - accuracy)$ for classification. For n qubits and l layers circuit, the number of parameters is $2n(l+1)$ in our ansatz. In regression experiments, the number of data is $m = 8$ and the number of qubits is $n = 3$; thus, we expect overfitting for circuit depth $l \geq 1$. In Figure 7(a), the testing error decreases up to $l = 10$. In Figure 7(b), the testing error decreases up to $l = 12$. This provides extra supporting evidence for regularization by unitarity of quantum circuits. Since d^*_{VC} saturation at critical depth $l^* = 1$ for $n = 4$, this behavior is consistent with our VC dimension analysis. The saturation of expressibility and model complexity serves as a reason for the absence of overfitting. Figures 7(c) and 7(e) show circuit depth scaling for strong input noise 0.2. Due to stronger input noise, we observe higher in-sample error as expected. We note a weak sign of overfitting at strong noise and deep circuit limit, but the overfitting is not significant. The generalization errors are plotted against circuit depth in Figure 7(f). While we are not able to identify significant overfitting in these experiments, signs of overfitting are observed in strong noise experiments and will be discussed further in Section 3.5. Finally, we note the following facts regarding the saturation: (1) We are using an upper bound for the VC dimension. The actual VC dimension might be smaller. (2) Without VC dimension analysis and without light-cone limitation, we know that the number of variational parameters $2n(l+1)$ will eventually hit the maximum real degrees of freedom of unitary group 2^{2n} for deeper circuits. However, the observed saturation cannot be explained by comparing the number of variational parameters with the maximum degrees of freedom of variational unitary, because the critical depth $\hat{l} = \frac{2^{2n-1}}{n} - 1$ would be expected to be much higher than the observed value. For example, for $n = 4$, the critical depth would be $\hat{l} = 31$.

For regression tasks, we cannot directly use the VC dimension as a model complexity measure, but KL-expressibility still can be applied. The generalization error for bounded regression can be upper bounded by using Pollard's pseudo-dimension [58, 65]. For a hypothesis set, which is a vector space of real-valued functions, the pseudo-dimension is equal to the dimension of the vector space, which is equal to its VC dimension [58]. Hence, our upper bound for the VC dimension could also be used to understand the interplay between model complexity and overfitting for regression in deep QCL. Another possible reason that overfitting can be avoided in regression is the compactness of unitary groups [32]. One sign of overfitting in regression is the large magnitude of weight coefficients [52]. The parameter space for quantum circuit learning is a compact manifold and the magnitude of matrix elements is upper bounded by $\sum_{j=0}^{2^n-1} |U_{ji}|^2 = 1$ and $\sum_{j=0}^{2^n-1} |U_{ij}|^2 = 1$ for n qubits. It would be interesting to see whether such a constraint could lead to automatic regularization effects in quantum-inspired classical learning algorithms [46]. Further investigation of regression tasks using a fat-shattering dimension or pseudo-dimension is beyond the scope of this work.

3.5 Sensitivity Study

We perform further sensitivity studies for varying dataset size, circuit widths, and noise strengths. All experiments in this section are classifications of make_moons data. We experimentally check the PAC learnability by looking at the data size scaling. Figure 8 shows the training, testing, and generalization errors for classification with respect to dataset size. For different circuit sizes and depths in Figures 8(a), 8(b), and 8(c), all generalization errors could be suppressed by the size of dataset. The scaling behaviors are consistent with PAC learnability. Figure 8(a) is the base case $n = 4$, $l = 4$. Figure 8(b) is the result for deeper circuit $n = 4$, $l = 8$. Figure 8(c) is the result

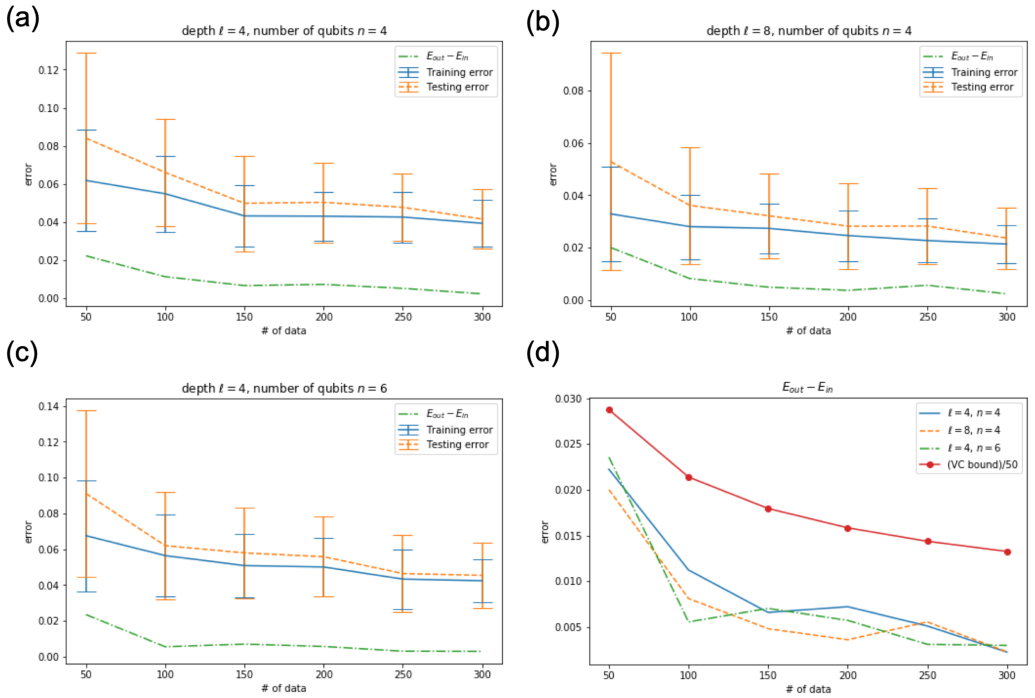

Fig. 8. Experimental evidence for PAC learnability. Training, testing, and generalization errors scaling are plotted with respect to size of dataset for different circuit depths and sizes. # of data = 100 means that there are 100 training data and 100 testing data. (a) $n = 4, l = 4$. (b) $n = 4, l = 8$. (c) $n = 6, l = 4$. Blue solid lines are training errors. Orange dashed lines are testing errors. Green dashed dotted lines are generalization errors. (d) Comparisons of the generalization error scaling for different circuit depths and sizes. Blue solid line is $n = 4, l = 4$. Orange dashed line is $n = 4, l = 8$. Green dashed dotted line is $n = 6, l = 4$. The VC generalization error bound divided by 50 for $\delta = 0.1$ and $d_{VC} = 2$ is plotted (red solid line with solid circle marker) for reference. The data is make_moons, and noise = 0.02 is added. Error bars denote one-sigma fluctuation for 90 independent runs.

for more qubits $n = 6$, $l = 4$. In Figure 8(d), the rescaled VC generalization error bound for $d_{VC} = 2$ is plotted together with all of the experimental generalization errors for comparison. With the sample size increased from 50 to 300, the averaged generalization errors are suppressed from $\gtrsim$0.02 to $\lesssim$0.005 for all cases. We note the following facts: (1) The VC generalization error bound is a loose upper bound; thus, it generally overestimates the generalization error. (2) The VC bound is for the difference $E_{out} - E_{in}$; thus, with large sample sizes, E_{out} and E_{in} are guaranteed to be close to each other, but the value E_{in} itself could be increased. In general, E_{in} depends on not only the model complexity but also other factors, such as the training algorithm and input data noise. In this work, we only show that overfitting could be suppressed because of limited expressibility in QCL. In practice, it could be useful to limit the model complexity if the available training dataset is small. On the other hand, to achieve high prediction accuracy, one should also consider the approximation-estimation trade-off and the algorithm used in the model selection procedure [29, 57, 58]. The study of the interplay between model complexity and in-sample error (i.e., the problem of underfitting) is beyond the scope of this work. (3) We fix the number of epochs in these experiments; thus, increasing sample size also implies increasing number of iterations.

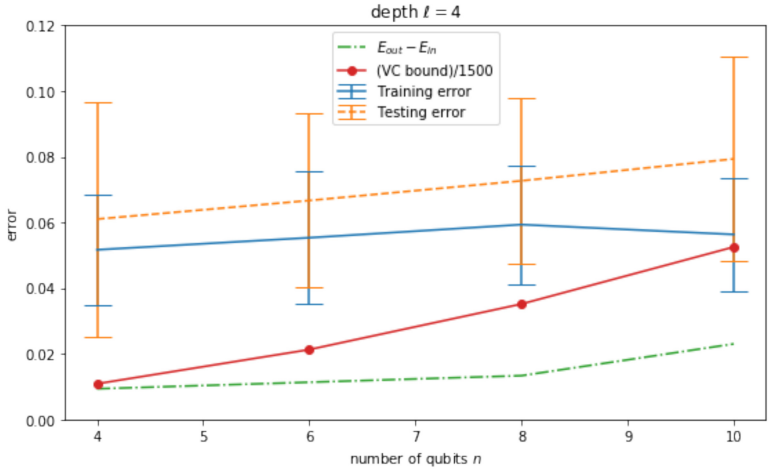

Fig. 9. Training, testing, and generalization errors as functions of number of qubits. Blue solid line is the training error. Orange dashed line is the testing error. Green dashed dotted line is the generalization error. The data is make_moons, and noise = 0.02 is added. Number of data = 100. Error bars denote one-sigma fluctuation for 30 independent runs. The VC generalization bound (red solid line with red circle marker) for d^*_{VC} is rescaled by a factor of 1500 for comparison. Confidence parameter is $\delta = 0.1$.

We further investigate the circuit size scaling of generalization error in Figure 9. For fixed depth $l = 4$, we expect the VC dimension upper bound to be $d^*_{VC} = (625, 2401, 6561, 14641)$ for $n = (4, 6, 8, 10)$. The corresponding VC generalization error bound is rescaled and plotted for reference. Since d^*_{VC} is large, the generalization error bound is evaluated by the approximation $E_{out} - E_{in} \leq \sqrt{\frac{8}{N} \ln(\frac{4(1+(2N)^{d_{VC}})}{\delta})} \approx \sqrt{\frac{8}{N} \ln(\frac{4((2N)^{d_{VC}})}{\delta})} = \sqrt{\frac{8}{N}(\ln(4) + d_{VC} \ln(2N) - \ln(\delta))}$ to avoid overflow. We observe the increase of generalization error for growing number of qubits. The observation is consistent with the VC dimension analysis. For higher model complexity with circuit width $n = 8$ and $n = 10$, we observe decreasing training error and increasing testing error, which is a sign of overfitting. The generalization error is increased from ≈ 0.01 for $n = 4$ to ≈ 0.02 for $n = 10$.

Data noise is considered as a catalyst of overfitting. Increasing noise would increase both training and testing error; for large enough model complexity, increasing noise leads to increasing generalization error. We study error scaling with respect to data noise strength in Figure 10. The training error, testing error, and generalization errors are plotted against data noise for $n = 4$, $l = 4$, $n = 4$, $l = 8$, and $n = 6$, $l = 4$, up to a strong noise strength 0.32. We observe an increase of E_{in} and E_{out} due to higher data noise. Furthermore, we observe an increase of generalization error $E_{out} - E_{in}$ in all cases, which is a sign of overfitting. All of the generalization errors grow to a similar value at noise = 0.32. The difference of model complexity seems to take a minor effect in the generalization error at the strong noise regime in these experiments.

4 CONCLUSIONS

In this work, we implemented and investigated the quantum learning circuit algorithm with hardware-efficient ansatz. Numerical simulations for regression and classification are performed, and the learning algorithm successfully captures the features of target functions without overfitting. We then analyze the expressibility and error scaling of the quantum circuits. For increasing

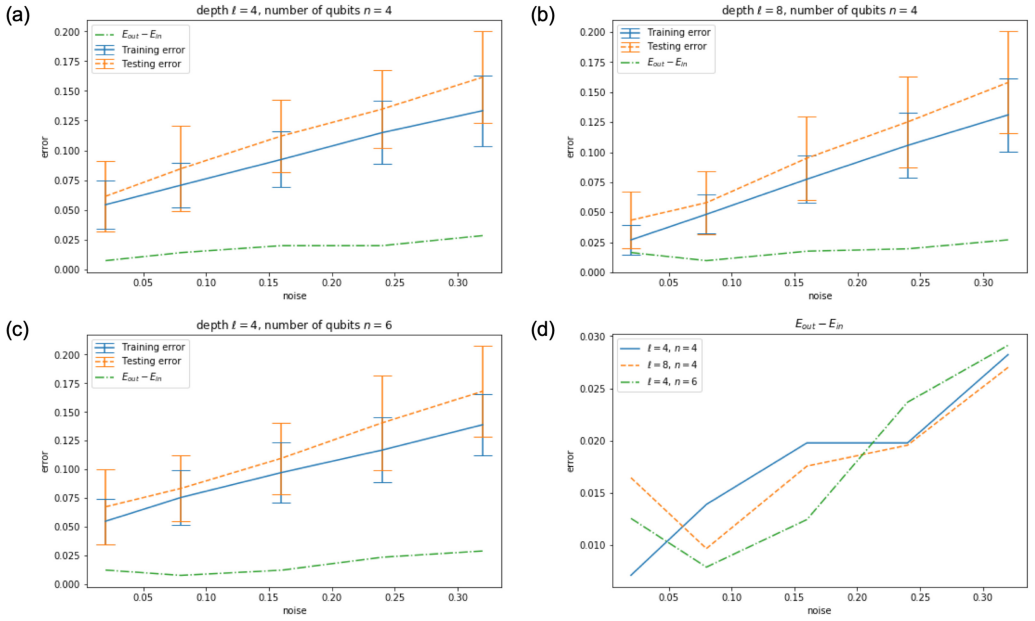

Fig. 10. Data noise scaling. Training, testing, and generalization errors scaling are plotted with respect to strength of data noise for different circuit depths and sizes. The noise strength is the standard deviation of Gaussian noise added to the data. (a) $n = 4, l = 4$. (b) $n = 4, l = 8$. (c) $n = 6, l = 4$. Blue solid lines are training errors. Orange dashed lines are testing errors. Green dashed dotted lines are generalization errors. (d) Comparison of the generalization error scaling for different circuit depths and sizes. Blue solid line is $n = 4, l = 4$. Orange dashed line is $n = 4, l = 8$. Green dashed dotted line is $n = 6, l = 4$. The data is make_moons. Error bars denote one-sigma fluctuation for 90 independent runs.

circuit depth, the expressibility saturates and the testing error does not increase. The saturation can be explained by the VC dimension upper bound, which is due to the unitarity and locality of the hardware-efficient ansatz variational quantum circuit. The saturation of expressibility is consistent with the absence of overfitting in learning results. These observations point to the conjecture of regularization by unitarity and the PAC learnability of the ansatz. The results of this work pave the way for the future development of the theory and applications of quantum circuit learning. Future research directions include scaling of the VC dimension lower bound, analysis of other ansatzes or encodings, and generalization error bound analysis for regression or multi-class classification.

A APPENDICES

A.1 Backpropagation Method and the Simulator Construction

A.1.1 Backpropagation Method. In this appendix, we introduce the backpropagation method for the gradient and the simulator construction [68]. We use the notation $I = (i_0, \ldots, i_{n-1})$ for a binary string and $U_{i,j} = \langle iUj \rangle$ for the matrix element of a local unitary U. Define functions

$$f_{I,J}\left(\overrightarrow{\theta_{(l)}}\right) = U_{i_0,j_0}\ (\theta_{0,l}) \ldots U_{i_{n-1},j_{n-1}}\ (\theta_{n-1,l}),$$

$$g\,(I)\ \ = i_0\ i_1 + \cdots + i_{n-1} i_0.$$

Then, the input quantum states, local unitary layer, entangler, and output quantum state can be expressed as

$$|\psi_{in}\rangle = \sum_I f_{I,0}\left(\overrightarrow{\theta_{in}}\right)|I\rangle,$$
$$U_{loc}^{(l)} = \sum_{I,J} f_{I,J}\left(\overrightarrow{\theta_{(l)}}\right)|I\rangle\langle J|,$$
$$U_{ent} = \sum_I (-1)^{g(I)}|I\rangle\langle I|,$$

$$|\psi_{out}\rangle = \left(\prod_{l=1}^{L} U_{loc}^{(l)} U_{ent}\right) U_{loc}^{(0)}|\psi_{in}\rangle = \sum_{I_L, I_{L-1}, \ldots, I_0, I_{-1}} (-1)^{g(I_{L-1})+\cdots+g(I_0)} \times f_{I_L, I_{L-1}}\left(\overrightarrow{\theta_L}\right)\ldots f_{I_1, I_0}\left(\overrightarrow{\theta_1}\right) f_{I_0, I_{-1}}\left(\overrightarrow{\theta_0}\right) f_{I_{-1},0}\left(\overrightarrow{\theta_{in}}\right)|I_L\rangle.$$

Hence, the amplitude of the output state and its derivatives can be obtained:

$$C_\theta^{I_L} = \sum_{I_{L-1}, \ldots, I_0, I_{-1}} (-1)^{g(I_{L-1})+\cdots+g(I_0)} f_{I_L, I_{L-1}}\left(\overrightarrow{\theta_L}\right)\ldots f_{I_1, I_0}\left(\overrightarrow{\theta_1}\right) f_{I_0, I_{-1}}\left(\overrightarrow{\theta_0}\right) f_{I_{-1},0}\left(\overrightarrow{\theta_{in}}\right)$$

$$\frac{\partial C_\theta^{I_L}}{\partial \theta_{p,l}} = \sum_{I_{L-1}, \ldots, I_0, I_{-1}} (-1)^{g(I_{L-1})+\cdots+g(I_0)} f_{I_L, I_{L-1}}\left(\overrightarrow{\theta_L}\right)\ldots f_{I_{l+1}, I_l}\left(\overrightarrow{\theta_{l+1}}\right) \times \frac{\partial f_{I_l, I_{l-1}}\left(\overrightarrow{\theta_l}\right)}{\partial \theta_{p,l}} f_{I_{l-1}, I_{l-2}}\left(\overrightarrow{\theta_{l-1}}\right)\ldots f_{I_0, I_{-1}}\left(\overrightarrow{\theta_0}\right) f_{I_{-1},0}\left(\overrightarrow{\theta_{in}}\right)$$

where

$$\frac{\partial f_{I,J}\left(\overrightarrow{\theta_l}\right)}{\partial \theta_{p,l}} = U_{i_0, j_0}\left(\theta_{0,l}\right)\ldots U_{i_{p-1}, j_{p-1}}\left(\theta_{p-1,l}\right)\frac{\partial U_{i_p, j_p}\left(\theta_{p,l}\right)}{\partial \theta_{p,l}} U_{i_{p+1}, j_{p+1}}\left(\theta_{p+1,l}\right)\ldots U_{i_{n-1}, j_{n-1}}\left(\theta_{n-1,l}\right).$$

This expression is just a substitution of local unitary $U(\theta_{p,l})$ with its derivatives $\frac{\partial U(\theta_{p,l})}{\partial \theta_{p,l}}$. Therefore, we can write,

$$\frac{\partial C_\theta^{I_L}}{\partial \theta_{p,l}} = C_\theta^{I_L}\left[U\left(\theta_{p,l}\right) \to \frac{\partial U\left(\theta_{p,l}\right)}{\partial \theta_{p,l}}\right].$$

The simulation is performed by the Kronecker products of the quantum gates matrix elements for each layer, and then compute the gradient state-vector through layer-by-layer matrix multiplication. Notice that the information from previous layers can be stored and reused to save computational resources, which gives an efficient backpropagation algorithm.

A.1.2 Simulator Construction The algorithm is implemented in a state vector–based simulator built in-house [63, 68]. The unitaries are applied to the state vectors by fast tensor contractions using NumPy arrays and Einstein summations [64]. The computations are done with CPU clusters without using a GPU.

One question is whether the vanishing gradient problem [44] occurs in our backpropagation method. Figure A1 shows a typical example of (a) loss curve and (b) gradient magnitude for

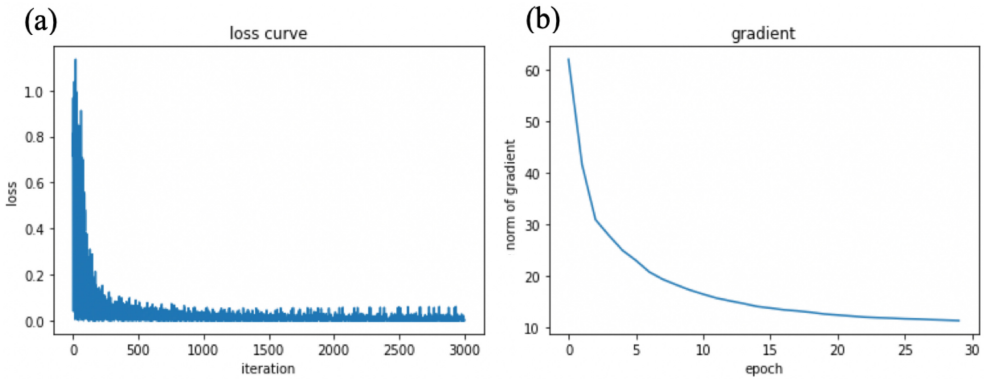

Fig. A1. Loss curve and gradient for a typical regression experiment. The circuit hyperparameters and number of training data are the same as in Figure 2. The target function is linear function. (a) Loss curve as a function of training iterations. (b) Gradient magnitude $\|\nabla\mathcal{L}(\theta)\|$ as a function of training epochs. The gradient does not vanish throughout the learning epochs in this example.

regression experiments. The target function is linear function. The hyperparameters are the same as those given in Figure 2. The number of training data is 100, and the number of epochs is 30; thus, the number of iterations is 3,000. In Figure 7(b), the sum of gradient norm $\|\nabla\mathcal{L}(\theta)\|$ for each epoch is plotted as a function of training epochs. We do not observe the vanishing of gradient in this case [44].

A.2 VC Dimension Bounds

In this appendix, we derive bounds for the VC dimension of QCL in Table 1. We use a theorem due to Dudley and Cover (Theorem 7.2 in Dudley 1978) [48–50]: For a hypothesis set formed by thresholding a real linear combination of real value basis functions, the VC dimension is equal to the dimension of the real vector space spanned by the basis functions. Let the number of qubits be n. For one-dimensional feature space and uniform input layer, the input state density matrix is $\rho_{in}(\theta_{in},\phi_{in}) = (\frac{1}{2}\left(\begin{smallmatrix} 1+\cos\theta_{in} & e^{-i\phi_{in}}\sin\theta_{in} \\ e^{i\phi_{in}}\sin\theta_{in} & 1-\cos\theta_{in}\end{smallmatrix}\right))^{\otimes n}$, where $\theta_{in} = \sin^{-1}(x) \in [-\frac{\pi}{2},\frac{\pi}{2}]$, $\phi_{in} = \cos^{-1}(x^2) \in [0,\frac{\pi}{2}]$. Then, the expectation value $\langle Z_0\rangle = Tr(Z_0 U\rho_{in}U^\dagger)$ as a function of $(\theta_{in},\phi_{in}) \in [-\pi,\pi]\times[-\pi,\pi]$ is a real trigonometric polynomial of two variables, where each variable has degree at most n. (Proof: The function $f(\theta_{in},\phi_{in}) = \langle Z_0\rangle = Tr\,(Z_0 U\rho_{in}U^\dagger) = Tr\,(U\rho_{in}^\dagger U^\dagger Z_0^\dagger) = \langle Z_0\rangle^\dagger$ has real values. Assume that $f(\theta_{in},\phi_{in}) = \sum_i a_i f_i(\theta_{in},\phi_{in})$, where $\{f_i(\theta_{in},\phi_{in})\}$ is the real trigonometric basis. Then, $a_i = \langle f_i f\rangle$ is real for all i.) Hence, $d_{VC} \le (2n+1)^2$.

For hardware-efficient ansatz learning circuits, light-cone restriction in tensor networks leads to a tighter upper bound, which scales with respect to circuit depth [51]. We consider a shallow circuit where shallow circuit $(L+1) < n$. Figure A2(a) shows the tensor network of $\langle Z_0\rangle$ expectation value for CZ-HEA 1D OBC. Figure A2(b) shows the tensor network with light-cone simplification. Outside of the light-cone, the local unitary contractions lead to identities since $U_{loc}^\dagger U_{loc} = 1$. The controlled-Z gate contractions also lead to identities since $(CZ)^\dagger (CZ) = 1\otimes 1$. Thus, any qubit not covered by the light-cone does not survive the tensor contraction. Hence, for L-layer circuits, only $L+1$ qubits are involved in $\langle Z_0\rangle = Tr(Z_0 U\rho_{in}U^\dagger)$. The effective density matrix is $\rho_{in}(\theta_{in}) = (\frac{1}{2}\left(\begin{smallmatrix} 1+\cos\theta_{in} & e^{-i\phi_{in}}\sin\theta_{in} \\ e^{i\phi_{in}}\sin\theta_{in} & 1-\cos\theta_{in}\end{smallmatrix}\right))^{\otimes(L+1)}$. Hence, $d_{VC} \le (2(L+1)+1)^2$. Including deep circuit cases, we obtain that $d_{VC} \le (2\min(n,L+1)+1)^2$. Note that the locality of short-range

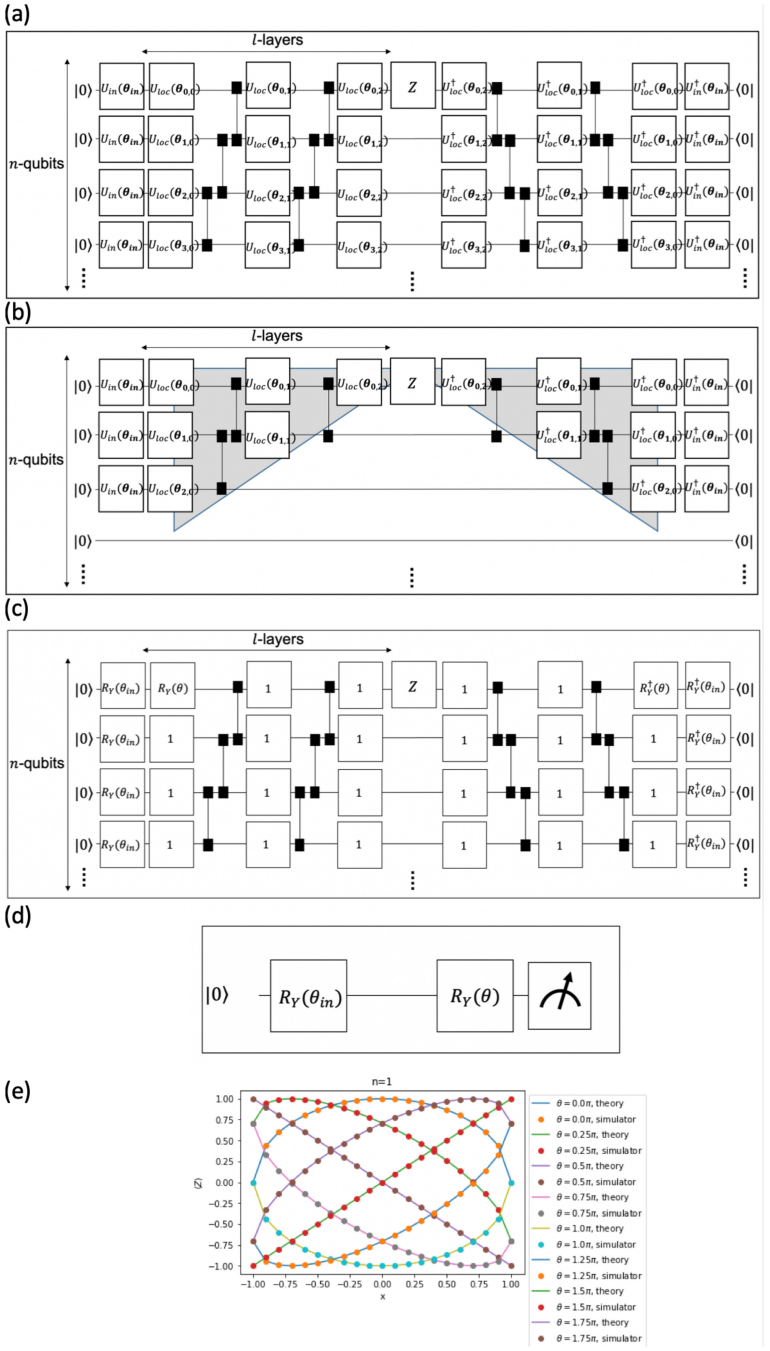

Fig. A2. (a) Tensor network representation of the expectation value $\langle Z_0 \rangle = Tr(Z_0 U \rho_{in} U^{\dagger})$ for the CZ-HEA 1D OPC. The CZ gate is denoted by two black squares connected by a solid line. (b) Light-cone limitation. The gray area is the light-cone. The operators outside of the light-cone are contracted to identities. (c) Reduction to a single-qubit zero layer case by choosing all other local unitaries to be identity, denoted by 1. (d) The single-qubit single-layer quantum circuit. (e) The single-qubit single-layer model function.

entanglers and the unitarity of quantum gates are both essential in the light-cone limitation. For a fully connected long-range entangler such as the Ising entanglers in [32], the light-cone limitation cannot be applied, and we have only the general upper bound $d_{VC} \leq (2n+1)^2$, which is independent of circuit depth. Hence, the saturation of the VC dimension upper bound is due to the unitarity and locality of HEA. For other HEA topology, we have $d_{VC} \leq (2\min(n, poly(L))+1)^2$, where the polynomial $poly(L)$ can be obtained by counting the qubits covered by the light-cone corresponding to a given entangler topology.

We then apply this theorem to obtain the VC dimension upper bound for d-dimensional feature space. With the assumptions made in Section 3.3, the model function is $f(\langle Z_0\rangle, \langle Z_1\rangle, \ldots, \langle Z_{d-1}\rangle) = \sum_{i=0}^{d-1} \gamma_i \langle Z_i\rangle$, where γ_is are real numbers. The input density matrix is $\rho_{in}(\boldsymbol{\theta_{in}}, \boldsymbol{\phi_{in}}) = \prod_{i=0}^{\frac{n}{d}-1} \prod_{j=0}^{d-1} (\frac{1}{2} \begin{pmatrix} 1+\cos\theta_{in}(x_j) & e^{-i\phi_{in}(x_j)}\sin\theta_{in}(x_j) \\ e^{i\phi_{in}(x_j)}\sin\theta_{in}(x_j) & 1-\cos\theta_{in}(x_j) \end{pmatrix}_{j+di})$, where the subscript $(j+di)$ means that the operator acts on the $(j+di)$-th qubit. The function $f(\langle Z_0\rangle, \langle Z_1\rangle, \ldots, \langle Z_{d-1}\rangle)$ is a $(2d)$-variable real trigonometric polynomial. Each variable has a degree at most $(\frac{n}{d})$. The total number of linearly independent basis function is at most $(2(\frac{n}{d})+1)^{2d}$. Hence, we get a VC dimension upper bound $d_{VC} \leq (2(\frac{n}{d})+1)^{2d}$. Applying the light-cone limitation to each $\langle Z_i\rangle$, we get that $d_{VC} \leq (2min(\frac{n}{d}, \frac{2L+1}{d})+1)^{2d}$ for 1D PBC.

The lower bound is obtained by reduction to a single-qubit case. The reduction for 1D OBC is depicted in Figure A2(c). All of the local unitaries are chosen to be identity, except for the 0-th qubit 0-th layer. Since the CZ gates are diagonal and commute with the Z operator, all of the CZ gates can be contracted to the identity by $(CZ)^\dagger\ (CZ) = 1 \otimes 1$. The resulting single-qubit hypothesis is depicted in Figure A2(d). The model function is $\langle Z_0\rangle = \cos(\theta+\theta_{in}) = \cos\theta\sqrt{1-x^2} - \sin\theta x$. Observe that this function can shatter two points in some cases. It has maximally two roots and can never shatter four points. Hence, its VC dimension is bounded by $2 \leq d_{VC} \leq 3$, and we obtain a general lower bound $2 \leq d_{VC}$ for $d = 1$. This lower bound remains true for other HEA entangling topology. The lower bound can be generalized to general feature space dimensions by considering data on the first feature axis.

A.3 Iris Dataset Result

As an example of real-world data, we test the Iris dataset from Scikit-learn [45]. This is a multi-class and multi-feature dataset. We perform full 3-class classification while the feature space is truncated to the first 2 features for visualization. The classification results are shown in Figure A3. The QCL hyperparameters are $n = 4$ and $l = 8$. We obtain 0.82 accuracy with the Scikit-learn support vector classifier and 0.807 for the QCL classifier.

A.4 Hardware Quantum Noise Simulation

We study the effect of hardware quantum noise on the prediction accuracy. The simulation noise model is based on Qiskit [47]. The single-qubit noise is applied to $\{U_1, U_2, U_3\}$ single-qubit gates defined in Qiskit. For simplicity, we assume perfect learning where the target function is exactly the same as the learned model. The circuit ansatz, models, and the results are plotted in Figure A4(a), A4(b), and A4(c), respectively. The expectation value for thresholding can be calculated to be $\langle Z\rangle = \cos(\theta+\theta_{in}) + \sin\theta\sin\theta_{in}(1-\cos\theta_{in}) = (\cos\theta - \sin\theta x)\sqrt{1-x^2}$. The predication accuracy is a decreasing function of hardware noise. We note that the hardware quantum noise has a stronger effect if the number of shots is smaller. The single-qubit rotation noise affects both the input layer and variational layer; thus, the single-qubit noise has a stronger effect than the two-qubit noise.

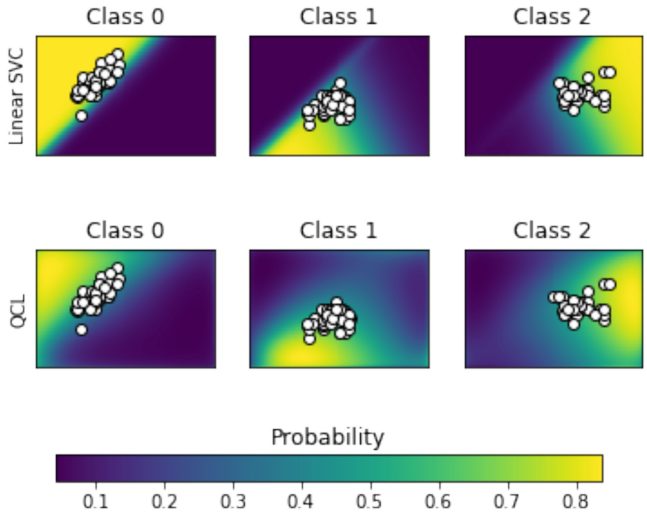

Fig. A3. Classification results for Iris dataset. The support vector classifier results are compared to the QCL results. The color bar shows the classification probability for each class.

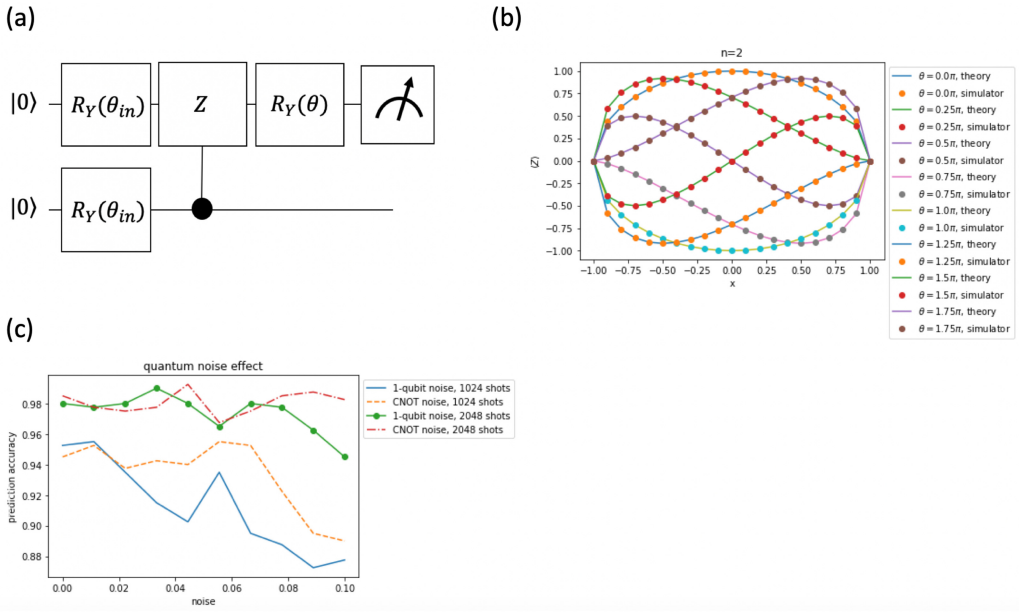

Fig. A4. Hardware quantum noise simulation. (a) Circuit ansatz. (b) The hypothesis models $\langle Z_0 \rangle(x)$ for different θ parameters. (c) Prediction accuracy as a function of quantum noise strength. The test data is located at $x = 0.05$. The model is $\theta = \frac{\pi}{2}$. For the 1-qubit noise scaling curves, the CNOT noise is fixed at 0.01. For the CNOT noise scaling curves the 1-qubit noise is fixed at 0.001. The accuracy is based on 400 independent predictions.

ACKNOWLEDGMENTS

We thank Naoki Yamamoto for valuable discussions and comments. CCC thanks Anthony Shao for discussions about learning theory.

REFERENCES

[1] David Deutsch. 1985. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proc R Soc Lond A* 400 (1985), 97–117. http://doi.org/10.1098/rspa.1985.0070

[2] Richard P. Feynman. 1982. Simulating physics with computers. *Int J Theor Phys* 21 (1982), 467–488. https://doi.org/10.1007/BF02650179

[3] Tadashi Kadowaki and Hidetoshi Nishimori. 1998. Quantum annealing in the transverse Ising model. *Phys Rev E* 58 (1998), 5355. https://doi.org/10.1103/PhysRevE.58.5355

[4] Michael A. Nielsen and Isaac L. Chuang. 2011. *Quantum computation and quantum information* (10th Anniversary Edition). Cambridge University Press, New York.

[5] Peter W. Shor. 1994. Algorithms for quantum computation: Discrete logarithms and factoring. *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, Santa Fe, NM, 1994, 124–134. https://doi.org/10.1109/SFCS.1994.365700

[6] Sergio Boixo, Sergei V. Isakov, Vadim N. Smelyanskiy, Ryan Babbush, Nan Ding, Zhang Jiang, Michael J. Bremner, John M. Martinis, and Hartmut Neven. 2018. Characterizing quantum supremacy in near-term devices. *Nature Phys* 14 (2018). 595–600. https://doi.org/10.1038/s41567-018-0124-x

[7] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. 2018. *Quantum supremacy and the complexity of random circuit sampling*. arXiv: 1803.04402. Retrieved from https://arxiv.org/abs/1803.04402.

[8] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandra, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574 (2019), 505–510. https://doi.org/10.1038/s41586-019-1666-5

[9] Michael Brooks. 2019. Beyond quantum supremacy: The hunt for useful quantum computers. *Nature* 574 (2019). 19–21.

[10] David P. DiVincenzo. 2000. The physical implementation of quantum computation. *Fortschr Phys* 48 (2000), 771–783.

[11] Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. 2012. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev A* 86 (2012) 032324. https://doi.org/10.1103/PhysRevA.86.032324

[12] John Preskill. 2018. Quantum computing in the NISQ era and beyond. *Quantum* 2 (2018), 79. https://doi.org/10.22331/q-2018-08-06-79

[13] Alexander McCaskey, Eugene Dumitrescu, Dmitry Liakh, and Travis Humble. 2018. Hybrid programming for near-term quantum computing systems. arXiv: 1805.09279. Retrieved from https://arxiv.org/abs/1805.09279.

[14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. arXiv: 1411.4028. Retrieved June 14, 2021 from https://arxiv.org/abs/1411.4028.

[15] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor G. Rieffel, Davide Venturelli, and Rupak Biswas. 2019. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* 12, 2 (2019), 34. https://doi.org/10.3390/a12020034

[16] Yudong Cao, Jonathan Romero, Jonathan P. Olson, Matthias Degroote, Peter D. Johnson, Marria Kieferova, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, Sukin Sim, Libor Veis, and Alan Aspuru-Guzik. 2019. Quantum chemistry in the age of quantum computing. *Chem Rev* 119, 19 (2019), 10856–10915. https://doi.org/10.1021/acs.chemrev.8b00803

[17] Kenji Sugisaki, Kazuo Toyota, Kazunobu Sato, Daisuke Shiomia, and Takeji Takui. 2020. A probabilistic spin annihilation method for quantum chemical calculations on quantum computers. *Phys Chem Chem Phys* 22 (2020), 20990–20994. https://doi.org/10.1039/D0CP03745A

[18] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alan Aspuru-Guzik, and Jeremy L. O'Brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nat Commun* 5 (2014), 4213. https://doi.org/10.1038/ncomms5213

[19] Jonathan Romero, Ryan Babbush, Jarrod R. McClean, Cornelius Hempel, Peter Love, and Alán Aspuru-Guzik. 2018. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. arXiv: 1701.02691. Retrieved June 14, 2021 from https://arxiv.org/abs/1701.02691.

[20] Robert M. Parrish, Edward G. Hohenstein, Peter L. McMahon, and Todd J. Martínez. 2019. Quantum computation of electronic transitions using a variational quantum eigensolver. *Phys Rev Lett* 122 (2019), 230401. https://doi.org/10.1103/PhysRevLett.122.230401

[21] Harper R. Grimsley, Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. 2019. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nat Commun* 10 (2019), 3007. https://doi.org/10.1038/s41467-019-10988-2

[22] Siddarth Srinivasan, Carlton Downey, and Byron Boots. 2018. Learning and inference in Hilbert space with quantum graphical models. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*. Curran Associates Inc., Red Hook, NY, 10359–10368.

[23] Maria Schuld and Nathan Killoran. 2019. Quantum machine learning in feature Hilbert spaces. *Phys Rev Lett* 122 (2019), 040504. https://doi.org/10.1103/PhysRevLett.122.040504

[24] Vojtech Havlicek, Antonio D. Corcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. 2019. Supervised learning with quantum-enhanced feature spaces. *Nature* 567 (2019), 209–212. https://doi.org/10.1038/s41586-019-0980-2

[25] Tongyang Li, Shouvanik Chakrabarti, and Xiaodi Wu. 2019. Sublinear quantum algorithms for training linear and kernel-based classifiers. In *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, PMLR 97:3815–3824.

[26] Carsten Blank, Daniel K. Park, June-Koo Kevin Rhee, and Francesco Petruccione. 2019. Quantum classifier with tailored quantum kernel. arXiv: 1909.02611. Retrieved June 14, 2021 from https://arxiv.org/abs/1909.02611.

[27] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. 2019. Evaluating analytic gradients on quantum hardware. *Phys Rev A* 99 (2019), 032331. https://doi.org/10.1103/PhysRevA.99.032331

[28] Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, M. Sohaib Alam, Shahnawaz Ahmed, Juan Miguel Arrazola, Carsten Blank, Alain Delgado, Soran Jahangiri, Keri McKiernan, Johannes Jakob Meyer, Zeyue Niu, Antal Száva, and Nathan Killoran. 2018. PennyLane: Automatic differentiation of hybrid quantum-classical computations. arXiv: 1811.04968. Retrieved June 14, 2021 from https://arxiv.org/abs/1811.04968.

[29] Yaser S. Abu-Mostafa, Malik Magdon-Ismail, and Hsuan-Tien Lin. 2012. *Learning from data*. AMLBook, New York.

[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521 (2015), 436–444. https://doi.org/10.1038/nature14539

[31] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15, 1 (2014), 1929–1958.

[32] Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. 2018. Quantum circuit learning. *Phys Rev A* 98 (2018), 032309. http://dx.doi.org/10.1103/PhysRevA.98.032309

[33] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. 2017. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* 549 (2017), 242–246. https://doi.org/10.1038/nature23879

[34] Zhao-Yun Chen, Qi Zhou, Cheng Xue, Xia Yang, Guang-Can Guo, and Guo-Ping Guo. 2018. 64-qubit quantum circuit simulation. *Science Bulletin* 63, 15 (2018), 964–971. http://dx.doi.org/10.1016/j.scib.2018.06.007

[35] Benjamin Villalonga, Sergio Boixo, Bron Nelson, Christopher Henze, Eleanor Rieffel, Rupak Biswas, and Salvatore Mandra. 2019. A flexible high-performance simulator for verifying and benchmarking quantum circuits implemented on real hardware. *npj Quantum Inf* 5 (2019), 86. https://doi.org/10.1038/s41534-019-0196-1

[36] Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. 2019. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Adv Quantum Technol* 2 (2019), 1900070. https://doi.org/10.1002/qute.201900070

[37] Kouhei Nakaji and Naoki Yamamoto. 2020. Expressibility of the alternating layered ansatz for quantum computation. arXiv: 2005.12537. Retrieved June 14, 2021 from https://arxiv.org/abs/2005.12537.

[38] Thomas Hubregtsen, Josef Pichlmeier, Patrick Stecher, and Koen Bertels. 2020. Evaluation of parameterized quantum circuits: On the relation between classification accuracy, expressibility and entangling capability. arXiv:2003.09887. Retrieved June 14, 2021 from https://arxiv.org/abs/2003.09887.

[39] Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. 2021. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys Rev A* 103, 032430. https://doi.org/10.1103/PhysRevA.103.032430

[40] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. 1989. Learnability and the Vapnik-Chervonenkis dimension. *J ACM* 36, 4 (1989), 929–965. https://doi.org/10.1145/76359.76371

[41] V. N. Vapnik and A. Ya. Chervonenkis. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications* 16, 2 (1971), 264–280. https://doi.org/10.1137/1116025

[42] L. G. Valiant. 1984. A theory of the learnable. *Commun ACM* 27, 11 (1984), 1134–1142. https://doi.org/10.1145/1968.1972

[43] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature* 323 (1986), 533–536. https://doi.org/10.1038/323533a0

[44] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. 2018. Barren plateaus in quantum neural network training landscapes. *Nat Commun* 9 (2018), 4812. https://doi.org/10.1038/s41467-018-07090-4

[45] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine learning in Python. *J Mach Learn Res* 12, null (2/1/2011), 2825–2830.

[46] Ewin Tang. 2019. A quantum-inspired classical algorithm for recommendation systems. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC'19)*. ACM, New York, NY, 217–228. https://doi.org/10.1145/3313276.3316310

[47] Gadi Aleksandrowicz, Thomas Alexander, Panagiotis Barkoutsos, Luciano Bello, Yael Ben-Haim, David Bucher, Francisco Jose Cabrera-Hernández, Jorge Carballo-Franquis, Adrian Chen, Chun-Fu Chen, Jerry M. Chow, Antonio D. Córcoles-Gonzales, Abigail J. Cross, Andrew Cross, Juan Cruz-Benito, Chris Culver, Salvador De La Puente González, Enrique De La Torre, Delton Ding, Eugene Dumitrescu, Ivan Duran, Pieter Eendebak, Mark Everitt, Ismael Faro Sertage, Albert Frisch, Andreas Fuhrer, Jay Gambetta, Borja Godoy Gago, Juan Gomez-Mosquera, Donny Greenberg, Ikko Hamamura, Vojtech Havlicek, Joe Hellmers, Łukasz Herok, Hiroshi Horii, Shaohan Hu, Takashi Imamichi, Toshinari Itoko, Ali Javadi-Abhari, Naoki Kanazawa, Anton Karazeev, Kevin Krsulich, Peng Liu, Yang Luh, Yunho Maeng, Manoel Marques, Francisco Jose Martín-Fernández, Douglas T. McClure, David McKay, Srujan Meesala, Antonio Mezzacapo, Nikolaj Moll, Diego Moreda Rodríguez, Giacomo Nannicini, Paul Nation, Pauline Ollitrault, Lee James O'Riordan, Hanhee Paik, Jesús Pérez, Anna Phan, Marco Pistoia, Viktor Prutyanov, Max Reuter, Julia Rice, Abdón Rodríguez Davila, Raymond Harry Putra Rudy, Mingi Ryu, Ninad Sathaye, Chris Schnabel, Eddie Schoute, Kanav Setia, Yunong Shi, Adenilton Silva, Yukio Siraichi, Seyon Sivarajah, John A. Smolin, Mathias Soeken, Hitomi Takahashi, Ivano Tavernelli, Charles Taylor, Pete Taylour, Kenso Trabing, Matthew Treinish, Wes Turner, Desiree Vogt-Lee, Christophe Vuillot, Jonathan A. Wildstrom, Jessica Wilson, Erick Winston, Christopher Wood, Stephen Wood, Stefan Wörner, Ismail Yunus Akhalwaya, and Christa Zoufal. 2019. Qiskit: An open-source framework for quantum computing (Version 0.7.2). *Zenodo*. https://doi.org/10.5281/zenodo.2562111

[48] Thomas. M. Cover. 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers EC*-14, 3 (1965), 326–334, https://doi.org/10.1109/PGEC.1965.264137

[49] Richard M. Dudley. 1978. Central limit theorems for empirical measures. *The Annals of Probability* 6, 6 (1978), 899–929. https://doi.org/10.1214/aop/1176995384

[50] Eduardo D. Sontag. 1998. VC dimension of neural networks. *NATO ASI Series F Computer and Systems Sciences* 168 (1998), 69–96.

[51] Ulrich Schollwoeck. 2011. The density-matrix renormalization group in the age of matrix product states. *Annals of Physics* 326, 96 (2011), 96–192. https://doi.org/10.1016/j.aop.2010.09.012

[52] Takahiro Goto, Quoc Hoan Tran, and Kohei Nakajima. 2020. Universal approximation property of quantum feature map. arXiv:2009.00298. Retrieved June 14, 2021 from https://arxiv.org/abs/2009.00298.

[53] Christopher M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin.

[54] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, Jeremy L. O'Brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nat Commun* 5, 4213 (2014). https://doi.org/10.1038/ncomms5213

[55] Jarrod R. McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. 2016. The theory of variational hybrid quantum-classical algorithms. *New J Phys.* 18 023023. https://doi.org/10.1088/1367-2630/18/2/023023

[56] Edward Farhi and Hartmut Neven. 2018. Classification with quantum neural networks on near term processors. arXiv:1802.06002. Retrieved June 14, 2021 from https://arxiv.org/abs/1802.06002.

[57] Shai Shalev-Shwartz and Shai Ben-David. 2014. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, New York.

[58] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. 2012. *Foundations of Machine Learning*. The MIT Press, Cambridge, MA.

[59] Scott Aaronson. 2007. The learnability of quantum states. *Proc R Soc A*.4633089–3114. http://doi.org/10.1098/rspa.2007.0113

[60] Srinivasan Arunachalam and Ronald de Wolf. 2018. Optimal quantum sample complexity of learning algorithms. *J Mach Learn Res* 19, 1 (2018), 2879–2878.

[61] Srinivasan Arunachalam and Ronald de Wolf. 2017. Guest column: A survey of quantum learning theory. *SIGACT News* 48, 2 (2017), 41–67. https://doi.org/10.1145/3106700.3106710

[62] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. arXiv:2011.01938. Retrieved June 14, 2021 from https://arxiv.org/abs/2011.01938.

[63] Kodai Shiba, Katsuyoshi Sakamoto, Koichi Yamaguchi. Dinesh Bahadur Malla, and Tomah Sogabe. Convolution filter embedded quantum gate autoencoder. arXiv:1906.01196. Retrieved June 14, 2021 from https://arxiv.org/abs/1906.01196.

[64] Charles R. Harris, K. Jarrod Millman, Stefan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernandez del Rio, Mark Wiebe, Pearu Peterson, Pierre Gerard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. 2020. Array programming with NumPy. *Nature* 585, (2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[65] David Pollard. 1984. *Convergence of Stochastic Processes*. Springer-Verlag New York.

[66] Adrian Perez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and Jose I. Latorre. 2020. Data re-uploading for a universal quantum classifier. *Quantum* 4, 226 (2020). https://doi.org/10.22331/q-2020-02-06-226

[67] Amira Abbas, David Sutter, Christa Zoufal, Aurélien Lucchi, Alessio Figalli, and Stefan Woerner. 2020. The power of quantum neural networks. Retrieved June 14, 2021 from https://arxiv.org/abs/2011.00027v1.

[68] Masaya Watabe, Kodai Shiba, Masaru Sogabe, Katsuyoshi Sakamoto, and Tomah Sogabe. 2019. Quantum circuit parameters learning with gradient descent using backpropagation. arXiv:1910.14266. Retrieved June 14, 2021 from https://arxiv.org/abs/1910.14266.

[69] Casper Gyurik, Dyon van Vreumingen, and Vedran Dunjko. 2021. Structural risk minimization for quantum linear classifiers. arXiv:2105.05566. Retrieved June 14, 2021 from https://arxiv.org/abs/2105.05566.

Received October 2020; revised May 2021; accepted May 2021