


# Optimal Quantum Dataset for Learning a Unitary Transformation

Zhan Yu<sup>1</sup>, Xuanqiang Zhao<sup>1</sup>, Benchi Zhao<sup>1</sup>, and Xin Wang<sup>1\*</sup>  
*Institute for Quantum Computing, Baidu Research, Beijing 100193, China*

 (Received 18 May 2022; revised 22 November 2022; accepted 27 January 2023; published 6 March 2023)

Unitary transformations formulate the time evolution of quantum states. How to learn a unitary transformation efficiently is a fundamental problem in quantum machine learning. The most natural and leading strategy is to train a quantum machine learning model based on a quantum dataset. Although the presence of more training data results in better models, using too much data reduces the efficiency of training. In this work, we solve the problem on the minimum size of sufficient quantum datasets for learning a unitary transformation exactly, which reveals the power and limitation of quantum data. First, we prove that the minimum size of a dataset with pure states is  $2^n$  for learning an  $n$ -qubit unitary transformation. To fully explore the capability of quantum data, we introduce a practical quantum dataset consisting of  $n + 1$  elementary tensor product states that are sufficient for exact training. The main idea is to simplify the structure utilizing decoupling, which leads to an exponential improvement in the size of the datasets with pure states. Furthermore, we show that the size of the quantum dataset with mixed states can be reduced to a constant, which yields an optimal quantum dataset for learning a unitary. We showcase the applications of our results in oracle compiling and Hamiltonian simulation. Notably, to accurately simulate a three-qubit one-dimensional nearest-neighbor Heisenberg model, our circuit only uses 96 elementary quantum gates, which is significantly less than 4080 gates in the circuit constructed by the Trotter-Suzuki product formula.

DOI: [10.1103/PhysRevApplied.19.034017](https://doi.org/10.1103/PhysRevApplied.19.034017)

## I. INTRODUCTION

Machine learning is a task that builds a model to learn an unknown function based on a training dataset. The training dataset is a set of example input-output pairs repeatedly used during the learning process and is used to fit the parameters of the model. Machine learning has been used in a wide variety of applications [1], such as computer vision, natural language processing, and speech recognition. At the same time, quantum computing, a technology that harnesses the laws of quantum mechanics to solve problems too complicated for classical computers, has been rapidly advancing. Inspired by the powerful capacity of machine learning [1], it is natural to develop their quantum counterparts and try to gain more benefits, which has given rise to an emerging research area, i.e., *quantum machine learning* (QML) [2–4].

QML is built on two components: models and data, both of which could be in the quantum version. QML models could be quantum analogs of classical machine learning models, such as quantum neural networks [5–8], quantum autoencoders [9–11], and quantum kernel methods [12–14]. Quantum data are quantum states that are

generated by quantum processes. Quantum data could be sampled from a natural process like a chemical reaction, or an artificial quantum system, e.g., a quantum computer.

In quantum mechanics, the time evolution of a quantum state according to the Schrödinger equation is mathematically represented by a unitary operator. A fundamental problem in quantum computing, *unitary learning*, consists in reproducing an unknown unitary transformation. Unitary learning has been studied intensively in classical approaches [15,16], quantum settings [17,18], and hybrid quantum-classical schemes [19–22]. One of the most natural and leading methods is to train a QML model based on a quantum dataset [23–26]. It raises a fundamental question: *what is the minimum size of the quantum training dataset that is sufficient for learning a unitary?*

The power and limitation for learning an unknown unitary based on datasets with pure states are investigated in Ref. [27], which suggests that the size of the dataset grows exponentially with the system size. The exponential scaling on the size of training dataset leads to an exponential overhead of the training process, which limits the efficiency of QML. With a sufficient ancillary system, it is not difficult to see that one Choi-Jamiołkowski state is sufficient for learning the unitary [25,28,29] by the Choi-Jamiołkowski isomorphism [30,31]. However, the costs of

\*wangxin73@baidu.com

using the ancillary quantum system and computation in a space with doubled dimensions are also expensive. Can we reduce and find the optimal size of the training dataset for an ancilla-free setting?

In this work, we resolve this problem by establishing optimal quantum datasets for learning a unitary transformation. We first introduce a formal definition of the minimum training dataset problem for unitary learning. We analyze the minimum size of training datasets with pure states, and show that using  $2^n$  nonorthogonal but linearly independent pure states as the training dataset is optimal for learning an  $n$ -qubit unitary. In order to exploit the power of the quantum dataset and reduce the exponential size, we introduce datasets with mixed states for learning a unitary transformation. In particular, we leverage the idea of decoupling to construct an efficient and practical dataset with  $n + 1$  elementary tensor product states that is sufficient to learn an  $n$ -qubit unitary operator. This implies that quantum datasets with certain structures could be extremely efficient for learning an arbitrary unitary transformation. We further reduce the size of the dataset to two and prove that this is optimal for learning a unitary with ancilla-free systems. More generally, we prove that a quantum dataset consisting of two randomly generated mixed states is sufficient for learning an unknown unitary.

We showcase practical applications of our results under the framework of hybrid quantum-classical algorithms. We apply our results to do Hamiltonian simulations and show that unitary learning can significantly reduce the depth of quantum circuits, compared to the traditional method using product formulas [32]. Another application is oracle compiling, which helps in transforming a high-level quantum algorithm (e.g., Grover's algorithm [33]) into a sequence of elementary quantum gates that could run on near-term quantum devices.

Our results on the optimal quantum dataset for unitary learning notably reduce the computational cost of training processes, from an exponential overhead to a constant. On the other hand, since the quantum data are generated by sampling from quantum processes, the optimal size of the dataset also leads to the optimal cost of data sampling. Our results also bring alternative insights regarding the power of quantum data. Quantum data used in QML can be considerably different than their classical counterparts, since quantum mechanics permits quantum states to be mixed, entangled, measured, distilled, concentrated, diluted, and manipulated. By fully leveraging the laws of quantum mechanics, one could significantly reduce the size of a quantum dataset, which is infeasible in the classical scenario.

We now begin the more technical part of our paper by giving some background and defining the minimum training dataset problem. We note here that detailed mathematical proofs are given in the Appendix.

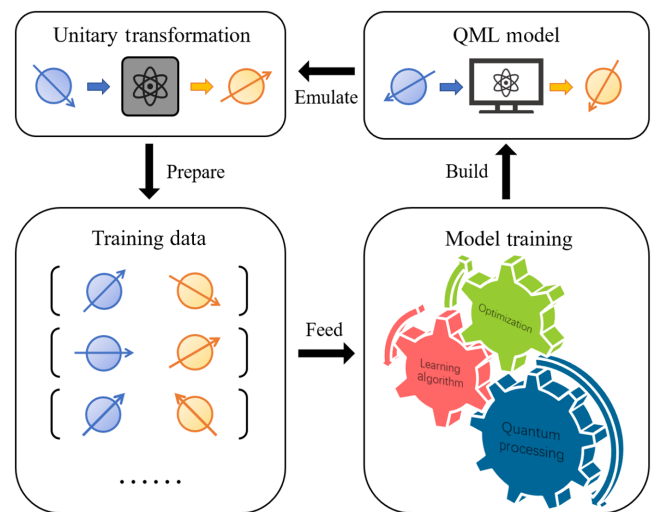


FIG. 1. The main scheme of using QML to learn an unknown unitary transformation. For the target unitary transformation, the first step is to prepare the training dataset consisting of input and output quantum states. Next, we train a QML model with the data using quantum processing, machine learning algorithms, and optimization methods. After the training process, the built QML model is able to emulate the action of the target unitary transformation.

## II. MAIN RESULTS ON UNITARY LEARNING

### A. The minimum training dataset problem

The strategy of learning an unknown unitary  $U$  is to train a QML model  $V$  with the training dataset  $\mathcal{D} = \{(\rho_j, U\rho_j U^\dagger)\}_{j=1}^t$  of size  $t$  so that  $V$  acts similarly to  $U$  on the training set  $\mathcal{D}$ . The goal is that the QML model  $V$  can emulate the action of the unitary  $U$  on any quantum state. The scheme is illustrated in Fig. 1. Notice that one needs to prepare many copies of each state in the dataset and repeatedly access them in the process of unitary learning. Throughout this paper, we refer to input states  $\rho_j$  as the training data, and target output states  $U\rho_j U^\dagger$  are the corresponding labels.

To effectively learn unitary operators, one must choose a suitable loss function for performing optimization and quantifying how well a model has learned the target unitary. Note that a standard method of directly evaluating the gate fidelity between the target unitary  $U$  and the QML model  $V$  relies on quantum process tomography [34], which is infeasible due to the exponential growth of required resource. Instead of using the gate fidelity, we choose the loss function that compares how well the output states of  $U$  and  $V$  are matched, which is arguably easier to evaluate on a quantum computer. A common loss function is the squared trace distance between  $U\rho_j U^\dagger$  and  $V\rho_j V^\dagger$ , i.e.,

$$\ell(U\rho_j U^\dagger, V\rho_j V^\dagger) = \|U\rho_j U^\dagger - V\rho_j V^\dagger\|_1^2, \quad (1)$$

where  $\|\cdot\|_1$  indicates the trace norm. The trace distance could be evaluated by a variational quantum algorithm [35]. There are many other distance measures that could also serve as the loss function, e.g., the Hilbert-Schmidt distance (or Frobenius distance), which could be evaluated by the SWAP test [36]. The *quantum empirical risk* is defined by averaging the loss function on the training dataset,

$$\hat{R}_U(V) = \frac{1}{t} \sum_{j=1}^t \ell(U\rho_j U^\dagger, V\rho_j V^\dagger). \quad (2)$$

We quantify how well the model  $V$  performs in simulating the unitary  $U$  by the *quantum risk*, defined as the squared trace distance between the outputs of  $U$  and  $V$  applied to the same input, averaged over all pure states induced by the uniform Haar measure,

$$R_U(V) = \int \ell(U|\psi\rangle\langle\psi|U^\dagger, V|\psi\rangle\langle\psi|V^\dagger) d|\psi\rangle \quad (3)$$

$$= \int d|\psi\rangle \|U|\psi\rangle\langle\psi|U^\dagger - V|\psi\rangle\langle\psi|V^\dagger\|_1^2 \quad (4)$$

$$= 1 - \int d|\psi\rangle |\langle\psi|U^\dagger V|\psi\rangle|^2 \quad (5)$$

$$= 1 - \frac{N + |\text{Tr}[U^\dagger V]|^2}{N(N+1)}, \quad (6)$$

where  $N = 2^n$  is the dimension of the corresponding Hilbert space.

Since our study focuses on the sufficiency of training datasets, for simplicity we assume the training is always *perfect*, i.e., the quantum empirical risk  $\hat{R}_U(V) = 0$ . Under the assumption of perfect training, a training dataset is sufficient for the model  $V$  to learn the unitary  $U$  if the quantum risk  $R_U(V) = 0$ , otherwise it is insufficient.

The minimum training dataset problem for unitary learning is to find the minimum size  $t_{\min}$  such that there exists a sufficient training dataset of size  $t_{\min}$  to learn an unknown unitary  $U$ , and any dataset of size less than  $t_{\min}$  is not sufficient.

## B. Minimum size of training datasets with pure states

The lower bound of expected quantum risk when applying datasets with pure states is given in Ref. [27], which shows that it is not sufficient to learn an  $n$ -qubit unitary using less than  $2^n$  pairs of Haar random pure states as the training data. For example, it could be inferred that the expected quantum risk of training  $V$  with  $O(n)$  pairs of pure states is  $\mathbb{E}_U[\mathbb{E}_{\mathcal{D}}[R_U(\theta)]] = 1 - O(1/2^n)$ .

However, the lower bound of the expected quantum risk does not imply  $t_{\min} = 2^n$  for training datasets with pure states. Consider using  $2^n$  pairs of pure states as the training data. A natural way is to pick an orthonormal basis.

Surprisingly, we find that using  $2^n$  orthogonal states as the training data is not sufficient for learning a unitary, as the worst case of quantum risk is  $R_U(V) = 1 - 1/2^n + 1$ . For more details, we refer to Lemma 1 in the Appendix.

Sharma *et al.* [28] give a discussion on the case that pure states in the training dataset are nonorthogonal but linearly independent, and conclude that this case leads to the same lower bound of average quantum risk as in Ref. [27]. Here we prove an upper bound of quantum risk when pure states in the training dataset are nonorthogonal but linearly independent.

**Proposition 1.** (Upper bound of quantum risk). *Consider learning an  $n$ -qubit  $U$  using the QML model  $V$  with a training set  $\mathcal{D} = \{(|x_j\rangle, U|x_j\rangle)\}_{j=1}^t$ , where  $\{|x_j\rangle\}_{j=1}^t$  is a set of nonorthogonal but linearly independent vectors. Assume that  $N/2 \leq t \leq N$ , then we have  $R_U(V) \leq 1 - N + (2t - N)^2/N(N+1)$ .*

The proof of this upper bound is based on the fact that  $U^\dagger V = \mathcal{I}$  in the  $t$ -dimensional subspace  $\mathcal{H}_{\mathcal{D}}$  spanned by  $\{|x_j\rangle\}_{j=1}^t$ . In the other  $(N-t)$ -dimensional subspace  $\mathcal{H}_{\mathcal{D}}^\perp$ ,  $U^\dagger V$  could be any unitary matrix. Considering the worst case,  $U^\dagger V = -\mathcal{I}$  in  $\mathcal{H}_{\mathcal{D}}^\perp$ , then  $\text{Tr}[U^\dagger V] = t - (N-t) = 2t - N$ .

The upper bound of quantum risk implies that using  $t = N$  nonorthogonal but linear-independent pure states is sufficient for learning a unitary. While  $t < N$ , it is not sufficient to learn a unitary since the worst case of quantum risk is greater than 0. From the upper bound of quantum risk, we conclude that  $t_{\min} = 2^n$  for training datasets with pure states.

The empirical risk is computed at each training iteration under the empirical risk minimization principle, which means the trace distance between  $U\rho_j U^\dagger$  and  $V\rho_j V^\dagger$  is evaluated for each state  $\rho_j$  in the training set. When using  $2^n$  pure states as the training data to learn a unitary, the overhead of this exponential scaling limits the efficiency of the learning process, and thus places a cutoff on the size of the target unitary in practical scenarios.

## C. Efficient training dataset with mixed states

As we show in the previous section, using pure states as the training dataset to learn a unitary is not efficient because of the overhead exponentially scaling with the number of qubits. However, quantum data are not limited to pure states but also involve mixed states. In practice, quantum systems are open and interact with environments. Such systems are in mixed states as we have only incomplete information about the systems. Hence, it is natural and sensible to consider the training dataset with mixed states.

A mixed state  $\rho$  is a mixture of pure states, i.e.,

$$\rho \equiv \sum_j p_j |\psi_j\rangle\langle\psi_j|, \quad (7)$$

where each  $|\psi_j\rangle$  is a pure state and each  $p_j$  is a positive real number such that  $\sum_j p_j = 1$ . The rank of a mixed state  $\rho$  is the rank of its density matrix in Eq. (7). For an  $n$ -qubit mixed state, the maximum rank is  $2^n$ , which is also called full rank, and the minimum rank is 1, in which case it is a pure state. Obviously, if a mixed state has a larger rank, it contains more information. Intuitively, fewer quantum states are needed to learn an unknown unitary when using mixed states as data.

To learn the target unitary  $U$ , we construct a training dataset  $\mathcal{D}_1$  that contains  $n + 1$  quantum states. Define

$$\rho_0 = (|+\rangle\langle+|)^{\otimes n} \quad (8)$$

and

$$\rho_j = \tau^{\otimes(j-1)} \otimes |0\rangle\langle 0| \otimes \tau^{\otimes(n-j)} \quad (9)$$

for  $1 \leq j \leq n$ , where  $\tau$  is the maximally mixed state. For an  $n$ -qubit unitary  $U$ , let  $\mathcal{D}_1 = \{(\rho_j, U\rho_j U^\dagger)\}_{j=0}^n$  be the training dataset.

To show that the training dataset  $\mathcal{D}_1$  is sufficient for the QML model  $V$  to learn  $U$ , we first prove that training with the dataset  $\{(\rho_i, U\rho_i U^\dagger)\}_{i=1}^n$  makes  $U^\dagger V$  a diagonal matrix. This is because training with each state decouples the space spanned by half of the computational basis states from the other half. Training with all these states together decouples the space spanned by each computational basis state from each other, which makes  $U^\dagger V$  diagonal in the computational basis. Then training with  $(\rho_0, U\rho_0 U^\dagger)$  unifies all diagonal entries in  $U^\dagger V$ , which ensures that  $U = V$ .

**Proposition 2.** *For an  $n$ -qubit unitary  $U$  and a QML model  $V$ , if it satisfies that*

$$U\rho_i U^\dagger = V\rho_i V^\dagger \quad (10)$$

for  $0 \leq i \leq n$ , then we have  $U = V$ .

Proposition 2 shows that we can learn a unitary using a QML model based on the training dataset  $\mathcal{D}_1$ , which consists of  $n + 1$  pairs of quantum states. The size of the training dataset  $\mathcal{D}_1$  grows linearly with the number of qubits, which is more efficient than the case of using pure states. Furthermore, each state in  $\mathcal{D}_1$  is in a form of tensor product without entanglement, which is arguably easy to construct.

## D. Minimum size of datasets with mixed states

Now that using mixed states could reduce the size of the training dataset, it is of interest to study the ultimate limits on the case of using mixed states to learn a unitary. In this section, we further reduce the size of training sets from linear to constant, which yields an optimal quantum dataset for unitary learning. Moreover, we study the more general cases where the data could be randomly generated instead of constructed.

We first consider learning a unitary  $U$  using the QML model  $V$  with one mixed state  $\rho$  as the training data. Suppose  $U\rho U^\dagger = V\rho V^\dagger$ , i.e.,  $U^\dagger V$  and  $\rho$  commute, then  $U^\dagger V$  and  $\rho$  share a common set of eigenvectors [37, Lemma 5F]. Since the eigenvalues of  $U^\dagger V$  are not fixed, the worst case of quantum risk is  $R_U(V) = 1 - 1/2^n + 1$ , we hence conclude that using one mixed state as the training data is not sufficient for learning a unitary.

We then prove that using two mixed states as the training data is sufficient to learn an  $n$ -qubit unitary.

**Proposition 3.** *Suppose a unitary  $W$  satisfies  $W\rho W^\dagger = \rho$  and  $W\sigma W^\dagger = \sigma$ , where  $\rho$  and  $\sigma$  are full-ranked mixed states that have nondegenerate eigenvalues, and eigenvectors of  $\rho$  and  $\sigma$  are nonorthogonal. Then  $W = \mathcal{I}$ .*

In order to prove Proposition 3, we show that  $U^\dagger V$  is simultaneously diagonal with respect to the eigenbasis of  $\rho$  and  $\sigma$  if  $U^\dagger V$  commute with the state  $\rho$  and  $\sigma$ , respectively. Since eigenvectors of  $\rho$  and  $\sigma$  are mutually nonorthogonal, it implies that  $U^\dagger V$  must be identity, i.e.,  $U = V$ .

More generally, any two random full-ranked mixed states  $\rho_r$  and  $\sigma_r$  that are Hilbert-Schmidt distributed satisfy the condition in Proposition 3, thus a random dataset  $\mathcal{D}_2 = \{(\rho_r, U\rho_r U^\dagger), (\sigma_r, U\sigma_r U^\dagger)\}$  is sufficient for learning an  $n$ -qubit unitary.

Our analytical results indicate that at least two quantum states are required for training a QML model to learn an unknown unitary. Thus we can conclude that  $t_{\min} = 2$  for the quantum training datasets, which solves the minimum training dataset problem for unitary learning.

**Remark 1:** For the task of characterizing a quantum gate, which consists of distinguishing two unitaries as well as assessing the unitarity of a time evolution, Ref. [38] showed that the minimum number of input states required is two. On the one hand, we note that perfectly distinguishing any two unitaries reduces to the same mathematical problem as unitary learning: find a set of quantum states  $\{\rho_j\}$  such that a unitary  $U$  satisfies  $U\rho_j U^\dagger = \rho_j$  for each  $\rho_j$  if and only if  $U$  is the identity matrix up to a global phase. Hence, utilizing the main tools and results in Ref. [38], one can conclude that two input states are sufficient for unitary learning.

On the other hand, since we assume that the target time evolution is unitary, the task of learning a unitary does not

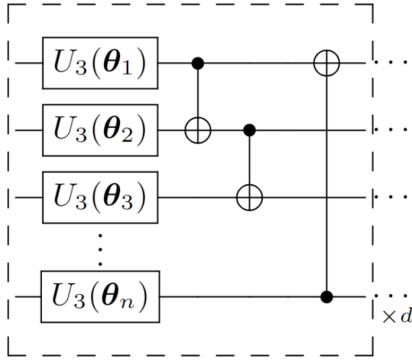


FIG. 2. The parameterized quantum circuit, one layer of which has  $n$  generic one-qubit rotation gates with three Euler angles  $\theta = (\theta, \phi, \lambda)$  and  $n$  CNOT gates. The CNOT gates in the circuit are only between neighboring qubits on a 1D chain with periodic boundary condition.

involve unitarity assessing. Thus, our results imply that two states are also necessary for distinguishing a pair of unitaries even if we do not need to assess the unitarity of time evolution. Moreover, by observing the equivalence between unitary learning and unitary distinguishing, Proposition 1 implies that the minimal set of pure states sufficient for distinguishing any two unitaries has a size of  $d$ , while the minimum number of pure states required for gate characterization is  $d + 1$  as shown in Ref. [38].

**Remark 2:** Notice that numerically generating density matrices in  $\mathcal{D}_2$  that satisfy the conditions in Proposition 3 is simple, following the method in Ref. [39]. However, preparing such quantum states (also known as totally mixed thermal states) on a quantum computer is very difficult. As a comparison, the dataset  $\mathcal{D}_1$  contains  $n + 1$  elementary tensor product states as proposed in Eqs. (8) and (9) that are easy to prepare. We remark that the dataset  $\mathcal{D}_1$  combines the properties of efficiency and feasibility, making it a particularly good choice for the task of unitary learning in practice.

We next demonstrate applications of unitary learning to investigate the performance of optimal quantum datasets with two mixed states. We adopt variational quantum algorithms in the numerical experiments, which use a classical optimizer to train a parameterized quantum circuit (PQC) as shown in Fig. 2. Note that there are many reasonable choices of PQCs that differ from ours, such as alternating layered ansatzes [40] and variable structure ansatzes [41], and the optimal choice may depend on the specifics of learning problem. Our numerical experiments were carried out with the Paddle Quantum toolkit [42] on the PaddlePaddle Deep Learning Platform [43], using a desktop with an 8-core i7 CPU and 32GB RAM.

### III. APPLICATIONS

#### A. Applications to Hamiltonian simulation

One of the first proposed and the most natural applications of quantum computers is simulating other quantum systems [44]. The dynamics of a closed quantum system are determined by a Hamiltonian  $H$  (a Hermitian matrix), and the state of the system at time  $t$  is represented by a state vector  $|\psi(t)\rangle$ . The system evolves over time according to the Schrödinger equation:

$$|\psi(t)\rangle = e^{-iHt}|\psi(0)\rangle. \quad (11)$$

The goal of quantum simulation is to produce the final state  $|\psi(t)\rangle$  within some error tolerance for a given Hamiltonian  $H$ , an evolution time  $t$ , and an initial state  $|\psi(0)\rangle$ .

The time evolution of a quantum system shown in Eq. (11) corresponds to a unitary operator  $U = e^{-iHt}$ . Simulating such time evolution on a quantum device requires the implementation of this unitary operator. If we can have access to the quantum system, it is possible to train a QML model to simulate  $U$  with a training dataset sampled from the quantum process. On the other hand, if the physical evolution  $U$  is not directly accessible but has been implemented as a quantum circuit  $V_0$  through methods like product formula [32], we can query  $V_0$  to prepare training data and learn  $V_0$  using another quantum circuit with a greatly reduced number of gates.

Here we consider learning a one-dimensional (1D) nearest-neighbor Heisenberg model [45] with  $n$  qubits, which is described by the Hamiltonian

$$H = \sum_{j=1}^n (\vec{\sigma}_j \cdot \vec{\sigma}_{j+1} + h_j \sigma_j^z), \quad (12)$$

where  $\vec{\sigma}_j = (\sigma_j^x, \sigma_j^y, \sigma_j^z)$  denotes a vector of Pauli  $x$ ,  $y$ , and  $z$  matrices on qubit  $j$ . Here a periodic boundary condition is imposed, i.e.,  $\vec{\sigma}_{n+1} = \vec{\sigma}_1$ . The coefficient  $h_j$  is chosen uniformly at random in  $[-h, h]$ , where the parameter  $h$  characterizes the strength of the disorder.

In numerical experiments, we first generate an  $n$ -qubit one-dimensional nearest-neighbor Heisenberg model with random  $h_j \in [0, 1]$ . We then construct the time-evolution circuit  $V_0$  using the second-order Suzuki product formula [32] with evolution time  $t = n$  and gate fidelity at least  $1 - 10^{-3}$ . We train a PQC  $V(\theta)$  with datasets  $\mathcal{D}_1$  and  $\mathcal{D}_2$ , respectively, to learn  $V_0$ . The two mixed states in  $\mathcal{D}_2$  are randomly generated according to the Hilbert-Schmidt measure [39]. Note that for the dataset  $\mathcal{D}_2$ , we independently generate ten samples of datasets and average the experimental results over ten independent training instances in order to reduce the effect of randomness. The number of layers of PQC is chosen empirically, depending on the size of the quantum system. We select the number of layers

$d = 8, 25$ , and  $100$  for  $n = 3, 4$ , and  $5$ , respectively. All parameters of trainable gates in PQCs are initialized following the uniform distribution on  $[0, 2\pi]$ . We choose the trace distance as the loss function and the empirical risk defined in Eq. (2) serves as the cost function in the training process. We set the training epoch to be 400 with a batch size equal to the size of datasets for all experiments. The Adam optimizer is used to minimize the cost function. The learning rate is 0.05 and 0.1 for training with dataset  $\mathcal{D}_1$  and  $\mathcal{D}_2$ , respectively. For  $n = 5$ , the learning rate decays by 0.5 every 100 epochs in order to achieve a better convergence.

As the numerical results shown in Fig. 3, PQC models trained with the dataset  $\mathcal{D}_1$  and randomly generated datasets  $\mathcal{D}_2$  could simulate the Hamiltonian with high accuracy (the average quantum risk achieves 0.01). What is more, the number of gates in PQC models is significantly fewer compared to circuits constructed by the second-order Suzuki product formula.

### B. Applications to oracle compiling

A *quantum oracle* is a “black-box” operator that is used extensively in quantum algorithms. Most known quantum algorithms, such as Shor’s algorithm [46], Grover’s algorithm [33], quantum walks [47], and quantum singular value transformation [48], are constructed using quantum oracles.

A quantum oracle is usually defined using a classical Boolean function  $f : \{0, 1\}^n \rightarrow \{0, 1\}$ , which maps an  $n$ -bit binary input to a binary output. The most common definition of quantum oracles is

$$U_f(|x\rangle \otimes |y\rangle) = |x\rangle \otimes |y \oplus f(x)\rangle, \quad (13)$$

where “ $\oplus$ ” is the XOR operation. Notice that the oracle  $U_f$  is unitary by the definition. In practice, such binary oracles are often used in quantum algorithms to do a phase rotation. This means that it is simpler to consider an alternative oracle, which is called *phase oracle*, defined as follows:

$$U_p|x\rangle = (-1)^{f(x)}|x\rangle. \quad (14)$$

As a “black-box” operator, the structure of a quantum oracle is typically unknown, thus we may not able to directly implement it on a quantum device. Based on the idea of unitary learning, we could train a QML model to learn an unknown quantum oracle, which permits us to compile and run an oracle-based quantum algorithm on quantum devices.

Here we showcase an example of compiling a phase oracle and then run the famous Grover’s algorithm based on this oracle. Suppose the search space consists of  $N = 2^n$  elements, which we label with  $n$ -bit strings. Let  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  be the Boolean function telling which elements are marked: the element labeled with  $x \in \{0, 1\}^n$

TABLE I. Performance of compiling Grover’s algorithm with datasets  $\mathcal{D}_1$  and  $\mathcal{D}_2$ . The success probability of compiled Grover’s algorithm with dataset  $\mathcal{D}_2$  is averaged over ten independent training instances.

No. of qubits	Success probability		
	Grover	Compiled with $\mathcal{D}_1$	Compiled with $\mathcal{D}_2$
3	0.94531250	0.94446218	0.94862192
4	0.96131897	0.96496582	0.95465205
5	0.99918232	0.99569303	0.99097359

is marked if  $f(x) = 1$  and unmarked otherwise. Let  $U_p$  be the phase oracle that encodes the Boolean function  $f$ , and we assume there is a unique marked element. Let  $D = 2|\Psi\rangle\langle\Psi| - \mathcal{I}$  be the Grover diffusion operator, where  $|\Psi\rangle = |+\rangle^{\otimes n}$ . The operator  $G = D \cdot U_p$  is known as the Grover iterate. Grover’s algorithm repeatedly applies the Grover iterate  $G$  for  $\lceil 4/\pi\sqrt{N} \rceil$  times on the initial state  $|\Psi\rangle$  and measures the resulting state. The success probability of Grover’s algorithm is the probability of obtaining the marked element when making the measurement.

In numerical experiments, we train a PQC  $V(\theta)$  with dataset  $\mathcal{D}_1$  and random datasets  $\mathcal{D}_2$ , respectively, to learn the phase oracle  $U_p$ . Except for different target unitaries, the setting of training PQCs is the same as what was used in the previous application. Then we append the circuit of Grover diffusion operator  $D$  to  $V(\theta)$  as the Grover iterate. By repeatedly applying the entire circuit of Grover iterate, we are able to run the compiled version of Grover’s algorithm. The experimental results with a comparison to the theoretical success probability of Grover’s algorithm are listed in Table I. From the results, we could see that, by training a PQC with our proposed datasets  $\mathcal{D}_1$  and  $\mathcal{D}_2$ , our method compiles the oracle into a sequence of single-qubit and two-qubit gates with high accuracy, which enables us to run an oracle-based algorithm on near-term quantum devices.

### IV. CONCLUDING REMARKS

Quantum computers could be used to simulate the dynamics of quantum systems, which is known to be computationally difficult for classical computers. One promising method is to train a quantum machine model based on quantum data, so that the model is able to mimic a unitary transformation. The performance of any machine learning model highly relies on the training data. Typical methods for learning unitaries often employ exponentially large training datasets [23–25,27]. In this work, we identify mixed states as the key to reduce the size of quantum datasets that are sufficient for learning an unknown unitary operator. We introduce efficient quantum datasets with mixed states that are exponentially smaller than datasets consisting of pure states. Explicitly, we show that the

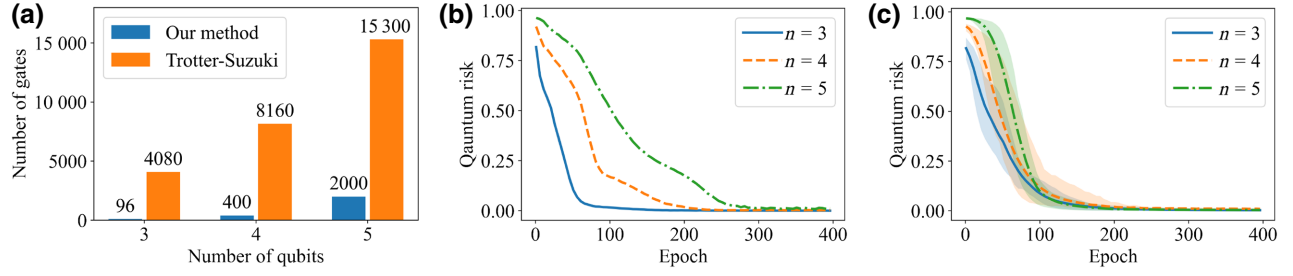


FIG. 3. Panel (a) displays the number of required gates to simulate Hamiltonian of different quantum system sizes by our method and Trotter-Suzuki method. Note that each generic  $U_3$  gates in the PQC is counted as three single-qubit rotation gates. Panel (b) shows the quantum risk in training with the dataset  $\mathcal{D}_1$ . The blue, orange, and green lines correspond to the results on the quantum system with size  $n = 3, 4$ , and  $5$ , respectively. Panel (c) shows the quantum risk in training with randomly generated dataset  $\mathcal{D}_2$ . The blue, orange, and green lines are averaged over ten independent training instances for the three system sizes. The corresponding shaded region is the min-max threshold in different runs.

optimal size of quantum datasets is  $t_{\min} = 2$  and we also obtain a practical dataset consisting of elementary tensor product states. Our analytical results are completely general and thus are not restricted to some particular quantum machine learning models. We showcase the applications of our results in Hamiltonian simulation and oracle compiling under the framework of hybrid quantum-classical algorithms, which demonstrates the effectiveness and practicability of our results in meaningful tasks.

Our results may also shed light on the advantages of quantum machine learning over classical machine learning, with respect to the training data. The main goal of using quantum machine learning to simulate quantum dynamics is to provide speedups over classical methods, as the computational cost of quantum simulation using classical computers is believed to grow exponentially with system size. Thus any exponential scaling in the quantum machine learning algorithm places a barrier on this speedup. Our results reduce the size of the dataset from exponentially large to linear, and further to a constant, which removes such a barrier.

As the minimum training dataset problem for unitary learning has been answered, one future step is to study the more general case. Note that the most general quantum process is a quantum channel, it is of great interest to study the sufficient training dataset for learning an unknown quantum channel. Possible applications of quantum channel learning include quantum error mitigation [49–54] and quantum channel simulation [55–58]. The training dataset problem for learning a quantum channel may also have implications for the learnability of QML models. Another interesting direction is to study the relationship between the size of the quantum dataset and the generalization error of unitary learning, without the assumption of perfect training. Our work shows applications of unitary learning via a quantum dataset, it will also be interesting to discover more such applications in quantum information processing and quantum machine learning [2–4].

## ACKNOWLEDGMENTS

We would like to thank Runyao Duan and Yin Mo for their helpful discussions. We also thank Christiane Koch for pointing out the connections between differentiating unitaries and learning unitaries. This work was done when Z.Y., X.Z., and B.Z. were research interns at Baidu Research.

## APPENDIX: PROOFS OF THE MAIN RESULTS

**Lemma 1.** Consider learning an  $n$ -qubit unitary  $U$  using a QML model  $V$  with training set  $\mathcal{D} = \{(|x_j\rangle, U|x_j\rangle)\}_{j=1}^N$ , where  $\{|x_j\rangle\}_{j=1}^N$  is an orthonormal basis. Then the worst case of learning is  $R_U(V) = 1 - 1/2^n + 1$ .

*Proof:* Since the training is perfect, i.e.,  $V|x_j\rangle = e^{i\theta_j} U|x_j\rangle$  for  $j = 1, 2, \dots, N$ , we have  $\langle x_j | U^\dagger V | x_j \rangle = e^{i\theta_j}$ . Then we can write

$$U^\dagger V = \sum_{j=1}^N e^{i\theta_j} |x_j\rangle\langle x_j|. \quad (\text{A1})$$

Note that each  $\theta_j$  are not necessarily the same. Consider a worst case scenario,  $\theta_j = \theta$  for  $j = 1, \dots, N/2$  and  $\theta_j = \theta + \pi$  for  $j = N/2 + 1, \dots, N$ , where  $\theta$  is an arbitrary angle, then we have  $\text{Tr}[U^\dagger V] = 0$  and  $R_U(V) = 1 - 1/2^n + 1$ . ■

**Proposition 4.** (Upper bound of quantum risk). Consider learning an  $n$ -qubit  $U$  using the QML model  $V$  with a training set  $\mathcal{D} = \{(|x_j\rangle, U|x_j\rangle)\}_{j=1}^t$ , where  $\{|x_j\rangle\}_{j=1}^t$  is a set of nonorthogonal but linearly independent vectors. Assume that  $N/2 \leq t \leq N$ , then we have  $R_U(V) \leq 1 - N + (2t - N)^2 / N(N + 1)$ .

*Proof*: Training the QML model  $V$  with the training set  $\mathcal{D}$  makes

$$U^\dagger V = \left[ \begin{array}{ccc|c} e^{i\theta_1} & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \\ 0 & \cdots & e^{i\theta_t} & \\ \hline & & 0 & Y \end{array} \right], \quad (\text{A2})$$

where  $Y$  is a unitary matrix on a  $(N-t)$ -dimensional Hilbert space. Since  $\{|x_j\rangle\}_{j=1}^t$  is a set of nonorthogonal but linearly independent vectors, we have  $\theta_j = \theta_k, \forall j, k \in \{1, \dots, t\}$  [28]. Then we can write

$$U^\dagger V = \left[ \begin{array}{c|c} e^{i\theta} \mathcal{I} & 0 \\ \hline 0 & Y \end{array} \right]. \quad (\text{A3})$$

Thus we have

$$|\text{Tr}[U^\dagger V]| = |\text{Tr}[e^{i\theta} \mathcal{I}] + \text{Tr}[Y]| \quad (\text{A4})$$

$$= |\text{Tr}[e^{i\theta} \mathcal{I}] - \text{Tr}[-Y]| \quad (\text{A5})$$

$$\geq ||\text{Tr}[e^{i\theta} \mathcal{I}]| - |\text{Tr}[-Y]|| \quad (\text{A6})$$

$$= |t - |\text{Tr}[Y]|| \quad (\text{A7})$$

$$\geq t - (N-t) \quad (\text{A8})$$

$$= 2t - N. \quad (\text{A9})$$

Inequality (A6) is from the reverse triangle inequality. Inequality (A8) follows from the fact that the maximum of  $\text{Tr}[Y]$  is  $(N-t)$ . The upper bound of the quantum risk is

$$R_U(V) = 1 - \frac{N + |\text{Tr}[U^\dagger V]|^2}{N(N+1)} \leq 1 - \frac{N + (2t-N)^2}{N(N+1)}. \quad (\text{A10})$$

**Corollary 2.** Consider learning an  $n$ -qubit  $U$  using the QML model  $V$  with a training set  $\mathcal{D} = \{(|x_j\rangle, U|x_j\rangle)\}_{j=1}^N$ , where  $\{|x_j\rangle\}_{j=1}^N$  is a set of nonorthogonal but linear independent vectors. Then we have  $R_U(V) = 0$ .

*Proof*: Directly follows from Proposition 1. ■

**Lemma 3.** A unitary operator  $W$  satisfies

$$W\rho_j W^\dagger = \rho_j \quad (\text{A11})$$

for all  $1 \leq j \leq n$  only if  $W$  is diagonal in the computational basis.

*Proof*: A state  $\rho_j$  defined in Eq. (9) can be written out in the computational basis as

$$\rho_j = \frac{1}{2^{n-1}} \sum_{k=0}^{2^j-1} \sum_{l=0}^{2^{n-j}-1} |l+2^{n-j}k\rangle\langle l+2^{n-j}k|. \quad (\text{A12})$$

The unitary operator  $W$  can be written as  $W = \sum_{k=0}^{2^n-1} e^{-i\theta_k} |\phi_k\rangle\langle k|$  for an orthonormal set of states  $\{|\phi_k\rangle\}_{k=0}^{2^n-1}$ . Hence we have

$$W\rho_j W^\dagger = \left( \sum_{r=0}^{2^n-1} e^{-i\theta_r} |\phi_r\rangle\langle r| \right) \rho_j \left( \sum_{s=0}^{2^n-1} e^{i\theta_s} |s\rangle\langle \phi_s| \right) \quad (\text{A13})$$

$$= \frac{1}{2^{n-1}} \sum_{k=0}^{2^j-1} \sum_{l=0}^{2^{n-j}-1} |\phi_{l+2^{n-j}k}\rangle\langle \phi_{l+2^{n-j}k}| \quad (\text{A14})$$

$$= \frac{1}{2^{n-1}} \sum_{k=0}^{2^j-1} \sum_{l=0}^{2^{n-j}-1} |l+2^{n-j}k\rangle\langle l+2^{n-j}k|, \quad (\text{A15})$$

where Eq. (A15) follows from Eq. (A11). Then, the orthonormal sets of vectors  $\Phi_j \equiv \{|\phi_{l+2^{n-j}k}\rangle\}$  and  $\Lambda_j \equiv \{|l+2^{n-j}k\rangle\}$  must span the same subspace, and so do  $\bar{\Phi}_j \equiv \{|\phi_m\rangle | m=0, \dots, 2^n-1\} - \Phi_j$  and  $\bar{\Lambda}_j \equiv \{|m\rangle | m=0, \dots, 2^n-1\} - \Lambda_j$ . Hence, we can say that the space spanned by  $\Phi_j$  is decoupled from the space spanned by  $\bar{\Phi}_j$  in the sense that  $W = U_{\Phi_j} + U_{\bar{\Phi}_j}$ , where  $U_{\Phi_j}$  acts nontrivially only on the space spanned by  $\Phi_j$ .

Writing  $l+2^{n-j}k$  into its binary representation, we note that the set  $\Lambda_j$  corresponds to all the bit strings whose  $(j+1)$ th bits (from left to right) are 0. Since for every pair of different computational basis states, there is at least one-bit difference in their corresponding bit strings, there exists a number  $j$  such that one of this pair is in  $\Lambda_j$  and the other is in  $\bar{\Lambda}_j$ . Thus, Eq. (A11) holding for all  $j$  implies that the space spanned by each computational basis state is decoupled from each other: that is,

$$W = \sum_{k=0}^{2^n-1} U_k = \sum_{k=0}^{2^n-1} e^{-i\theta_k} |k\rangle\langle k|, \quad (\text{A16})$$

where  $U_k = e^{-i\theta_k} |k\rangle\langle k|$ . Therefore,  $W$  is a diagonal matrix. ■

**Lemma 4.** If a diagonal unitary  $\Lambda$  satisfies

$$\Lambda\rho_0\Lambda^\dagger = \rho_0, \quad (\text{A17})$$

then  $\Lambda = e^{-i\theta} \mathcal{I}$ , where  $\mathcal{I}$  is the identity matrix.



*Proof:* The pure state  $|+\rangle = 1/\sqrt{2}(1, 1)^T$ , then we can write

$$\begin{aligned} & \left( \sum_{j=0}^N e^{-i\theta_j} |j\rangle\langle j| \right) \cdot (|+\rangle\langle +|)^{\otimes n} \cdot \left( \sum_{k=0}^N e^{i\theta_k} |k\rangle\langle k| \right) \\ &= \left( \sum_{j=0}^N e^{-i\theta_j} |j\rangle\langle j| \cdot |+\rangle^{\otimes n} \right) \left( \sum_{k=0}^N e^{i\theta_k} \langle +|^{\otimes n} |k\rangle\langle k| \right) \end{aligned} \quad (\text{A18})$$

$$= (|+\rangle\langle +|)^{\otimes n}. \quad (\text{A19})$$

Since  $\langle j | (|+\rangle^{\otimes n}) = (\langle + |^{\otimes n}) |k\rangle = 1/\sqrt{N}$ , we have

$$\frac{1}{N} \sum_{j=0}^N \sum_{k=0}^N e^{-i\theta_j} e^{i\theta_k} |j\rangle\langle k| = (|+\rangle\langle +|)^{\otimes n}. \quad (\text{A20})$$

The state  $\rho$  is the matrix with all elements are the same, i.e.,  $\rho = (|+\rangle\langle +|)^{\otimes n} = 1/N\mathbb{I}$ , the elements in  $\mathbb{I}$  are all 1. Thus we have

$$\sum_j \sum_k e^{-i\theta_j} e^{i\theta_k} |j\rangle\langle k| = \mathbb{I}, \quad (\text{A21})$$

which implies each element on the left-hand side equals to 1, so the phase term must be 1, i.e.,  $\theta_j = \theta_k = \theta$ . Then the diagonal matrix can be written into

$$\sum_j e^{-i\theta} |j\rangle\langle j| = e^{-i\theta} \sum_j |j\rangle\langle j| = e^{-i\theta} \mathbb{I}. \quad (\text{A22})$$

■

**Proposition 5.** For an  $n$ -qubit unitary  $U$  and a QML model  $V$ , if it satisfies that

$$U\rho_i U^\dagger = V\rho_i V^\dagger \quad (\text{10})$$

for  $0 \leq i \leq n$ , then we have  $U = V$ .

*Proof:* Writing Eq. (10) as  $U^\dagger V\rho_i V^\dagger U = \rho_i$ . By Lemmas 3 and 4, we have  $U^\dagger V = e^{-i\theta} \mathbb{I}$ . ■

**Lemma 5.** Consider learning an  $n$ -qubit  $U$  using the QML model  $V$  with only one training data  $\mathcal{D} = \{(\rho, U\rho U^\dagger)\}$ , where  $\rho$  is a mixed state. The worst case of learning is  $R_U(V) = 1 - 1/2^n + 1$ .

*Proof:* We assume the training is perfect, i.e.,  $U\rho U^\dagger = V\rho V^\dagger$ . Then, we rewrite it into

$$U^\dagger V\rho V^\dagger U = \rho. \quad (\text{A23})$$

Since  $U^\dagger V$  commutes with  $\rho$ ,  $U^\dagger V$  and  $\rho$  share the same eigenvectors [37, 5F]. Suppose  $\rho$  has eigenvectors  $\{|\psi_j\rangle\}_j$ ,

we can write

$$U^\dagger V = \sum_{j=1}^N e^{i\theta_j} |\psi_j\rangle\langle\psi_j|. \quad (\text{A24})$$

Then we have

$$|\text{Tr}[U^\dagger V]| = \left| \sum_{i=1}^N e^{i\theta_j} \right|. \quad (\text{A25})$$

Note that each  $\theta_j$  are not necessarily the same. Consider a worst case scenario,  $\theta_j = \theta$  for  $j = 1, \dots, N/2$  and  $\theta_j = \theta + \pi$  for  $j = N/2 + 1, \dots, N$ , where  $\theta$  is an arbitrary angle, then we have  $\text{Tr}[U^\dagger V] = 0$  and  $R_U(V) = 1 - 1/2^n + 1$ . ■

**Proposition 6.** Suppose a unitary  $W$  satisfies  $W\rho W^\dagger = \rho$  and  $W\sigma W^\dagger = \sigma$ , where  $\rho$  and  $\sigma$  are full-ranked mixed states that have nondegenerate eigenvalues, and eigenvectors of  $\rho$  and  $\sigma$  are nonorthogonal. Then  $W = \mathbb{I}$ .

*Proof:* Write mixed states  $\rho$  and  $\sigma$  in the form of spectral decomposition:

$$\rho = \sum_j b_j |x_j\rangle\langle x_j|, \quad (\text{A26})$$

$$\sigma = \sum_j c_j |y_j\rangle\langle y_j|, \quad (\text{A27})$$

where  $\{|x_j\rangle\}_j$  and  $\{|y_j\rangle\}_j$  are two different eigenbases. By Ref. [37, 5F],  $W$  shares a common eigenbasis with  $\rho$  and  $\sigma$ , respectively. Suppose that  $\{\theta_j\}_{j=1}^N$  is the eigenphases of  $W$ , we can write  $W$  as

$$W = \sum_{j=1}^N e^{i\theta_j} |x_j\rangle\langle x_j|, \quad (\text{A28})$$

$$W = \sum_{j=1}^N e^{i\theta_j} |y_j\rangle\langle y_j|, \quad (\text{A29})$$

where  $\{|x_j\rangle\}_{j=1}^N$  and  $\{|y_j\rangle\}_{j=1}^N$  are two different eigenbases of that satisfy  $\langle x_j | y_k \rangle \neq 0$  for each  $1 \leq j, k \leq N$ . Since eigenvectors with distinct eigenvalues are orthogonal,  $|x_j\rangle$  and  $|y_k\rangle$  being nonorthogonal implies that  $\alpha_j = \alpha_k$  for each  $1 \leq j, k \leq N$ . Thus we have  $W = e^{i\theta} \mathbb{I}$ . ■

**Corollary 6.** Consider learning an  $n$ -qubit  $U$  using the QML model  $V$  with a training set  $\mathcal{D}_2 = \{(\rho_r, U\rho_r U^\dagger), (\sigma_r, U\sigma_r U^\dagger)\}$ , where  $\rho_r$  and  $\sigma_r$  are two random generated

full-ranked mixed states according to the Hilbert-Schmidt measure. Then we have  $\mathbb{E}_U[\mathbb{E}_{\mathcal{D}}[R_U(V)]] = 0$ .

*Proof*: The Hilbert-Schmidt measure is equivalent to the measures induced by partial tracing of bipartite composite systems on the Hilbert space  $\mathcal{H}_N \otimes \mathcal{H}_K$  if  $N = K$  [39]. One could generate an  $N \times N$  random density matrix  $\rho_r$  according to the Hilbert-Schmidt measure by taking normalized random Wishart matrices  $AA^\dagger$ , with  $A$  belonging to the Ginibre ensemble of Hermitian matrices of appropriate dimension. The probability distribution in the simplex of eigenvalues of  $\rho_r$  is

$$P(\lambda_1, \lambda_2, \dots, \lambda_N) = C_N \delta \left( 1 - \sum_{j=1}^N \lambda_j \right) \prod_{j < k}^N (\lambda_j - \lambda_k)^2, \quad (\text{A30})$$

where  $C_N$  is the normalization constant. From the probability distribution in Eq. (A30), we observe that the probability of  $\rho_r$  having two identical eigenvalues is 0. It implies that a random density matrix has nondegenerate eigenvalues. In addition, eigenvectors of two randomly generated mixed states are nonorthogonal, as the probability that two independent random vectors that follow an absolutely continuous distribution will be exactly orthogonal is 0. Thus  $\rho_r$  and  $\sigma_r$  satisfy the condition in Proposition 3, which implies that  $\mathbb{E}_U[\mathbb{E}_{\mathcal{D}}[R_U(V)]] = 0$ . ■

- 
- [1] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, *Nature* **521**, 436 (2015).
- [2] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, and Quantum machine learning, *Nature* **549**, 195 (2017).
- [3] M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers, Quantum Science and Technology* (Springer International Publishing, Cham, 2018).
- [4] S. Arunachalam and R. de Wolf, A Survey of Quantum Learning Theory, *ACM SIGACT News* **48**, 41 (2017).
- [5] E. Farhi and H. Neven, Classification with Quantum Neural Networks on Near Term Processors, 1 (2018), [ArXiv:1802.06002](https://arxiv.org/abs/1802.06002).
- [6] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, *Nat. Phys.* **15**, 1273 (2019).
- [7] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, *Quantum Sci. Technol.* **4**, 043001 (2019).
- [8] Z. Yu, H. Yao, M. Li, and X. Wang, in *36th Conf. Neur. Inf. Process. Syst. (NeurIPS 2022)* (2022) [ArXiv:2205.07848](https://arxiv.org/abs/2205.07848).
- [9] J. Romero, J. P. Olson, and A. Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, *Quantum Sci. Technol.* **2**, 045001 (2017).
- [10] K. H. Wan, O. Dahlsten, H. Kristjánsson, R. Gardner, and M. S. Kim, Quantum generalisation of feedforward neural networks, *npj Quantum Inf.* **3**, 36 (2017).
- [11] C. Cao and X. Wang, Noise-Assisted Quantum Autoencoder, *Phys. Rev. Appl.* **15**, 054012 (2021).
- [12] M. Schuld and N. Killoran, Quantum Machine Learning in Feature Hilbert Spaces, *Phys. Rev. Lett.* **122**, 040504 (2019).
- [13] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, Power of data in quantum machine learning, *Nat. Commun.* **12**, 2631 (2021).
- [14] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, *Nat. Phys.* **17**, 1013 (2021).
- [15] M. Arjovsky, A. Shah, and Y. Bengio, in *Proceedings of The 33rd International Conference on Machine Learning, Proceedings of Machine Learning Research, Vol. 48*, edited by M. F. Balcan and K. Q. Weinberger (PMLR, New York, New York, USA, 2016), p. 1120.
- [16] S. L. Hyland and G. Rätsch, in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI'17* (AAAI Press, 2017), p. 2050.
- [17] A. Bisio, G. Chiribella, G. M. D'Ariano, S. Facchini, and P. Perinotti, Optimal quantum learning of a unitary transformation, *Phys. Rev. A* **81**, 032324 (2010).
- [18] I. Marvian and S. Lloyd, Universal quantum emulator (2016), [ArXiv:1606.02734](https://arxiv.org/abs/1606.02734).
- [19] K. Heya, Y. Suzuki, Y. Nakamura, and K. Fujii, Variational quantum gate optimization (2018), [ArXiv:1810.12745](https://arxiv.org/abs/1810.12745).
- [20] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, Quantum-assisted quantum compiling, *Quantum* **3**, 140 (2019).
- [21] K. Sharma, S. Khatri, M. Cerezo, and P. J. Coles, Noise resilience of variational quantum compiling, *New J. Phys.* **22**, 043006 (2020).
- [22] T. Jones and S. C. Benjamin, Robust quantum compilation and circuit optimisation via energy minimisation, *Quantum* **6**, 628 (2022).
- [23] L. Cincio, Y. Subasi, A. T. Sornborger, and P. J. Coles, Learning the quantum algorithm for state overlap, *New J. Phys.* **20**, 113022 (2018).
- [24] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, Training deep quantum neural networks, *Nat. Commun.* **11**, 808 (2020).
- [25] L. Cincio, K. Rudinger, M. Sarovar, and P. J. Coles, Machine Learning of Noise-Resilient Quantum Circuits, *PRX Quantum* **2**, 1 (2021).
- [26] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, Generalization in quantum machine learning from few training data, *Nat. Commun.* **13**, 4919 (2022).
- [27] K. Poland, K. Beer, and T. J. Osborne, No free lunch for quantum machine learning, (2020), [ArXiv:2003.14103](https://arxiv.org/abs/2003.14103).
- [28] K. Sharma, M. Cerezo, Z. Holmes, L. Cincio, A. Sornborger, and P. J. Coles, Reformulation of the No-Free-Lunch Theorem for Entangled Datasets, *Phys. Rev. Lett.* **128**, 070501 (2022).
- [29] S. Chakrabarti, Y. Huang, T. Li, S. Feizi, and X. Wu, Quantum Wasserstein generative adversarial networks, (2019), [ArXiv:1911.00111](https://arxiv.org/abs/1911.00111).
- [30] M.-D. Choi, Completely positive linear maps on complex matrices, *Linear Algebra Appl.* **10**, 285 (1975).

- [31] A. Jamiolkowski, Linear transformations which preserve trace and positive semidefiniteness of operators, *Rep. Math. Phys.* **3**, 275 (1972).
- [32] M. Suzuki, General theory of fractal path integrals with applications to many-body theories and statistical physics, *J. Math. Phys.* **32**, 400 (1991).
- [33] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996* (1996), p. 212.
- [34] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, New York, USA, 2010).
- [35] R. Chen, Z. Song, X. Zhao, and X. Wang, Variational quantum algorithms for trace distance and fidelity estimation, *Quantum Sci. Technol.* **7**, 015019 (2021).
- [36] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Quantum Fingerprinting, *Phys. Rev. Lett.* **87**, 167902 (2001).
- [37] G. Strang, *Linear Algebra and its Applications* (Thomson, Brooks/Cole, Belmont, CA, 2006).
- [38] D. M. Reich, G. Gualdi, and C. P. Koch, Minimum number of input states required for quantum gate characterization, *Phys. Rev. A* **88**, 042309 (2013).
- [39] K. Życzkowski and H.-J. Sommers, Induced measures in the space of mixed quantum states, *J. Phys. A: Math. General* **34**, 7111 (2001).
- [40] K. Nakaji and N. Yamamoto, Expressibility of the alternating layered ansatz for quantum computation, *Quantum* **5**, 434 (2021).
- [41] M. Bilkis, M. Cerezo, G. Verdon, P. J. Coles, and L. Cincio, A semi-agnostic ansatz with variable structure for quantum machine learning, (2021), [ArXiv:2103.06712](https://arxiv.org/abs/2103.06712).
- [42] Paddle Quantum: A quantum machine learning toolkit (2020).
- [43] Y. Ma, D. Yu, T. Wu, and H. Wang, PaddlePaddle: An Open-Source Deep Learning Platform from Industrial Practice, *Front. Data Computing* **1**, 105 (2019).
- [44] R. Feynman, Simulating physics with computers, *Int. J. Theor. Phys.* **21**, 467 (1982).
- [45] A. M. Childs, D. Maslov, Y. Nam, N. J. Ross, and Y. Su, Toward the first quantum simulation with quantum speedup, *Proc. Nat. Acad. Sci.* **115**, 9456 (2018).
- [46] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* **26**, 1484 (1997).
- [47] M. Szegedy, in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, FOCS '04* (IEEE Computer Society, USA, 2004), p. 32.
- [48] A. Gilyén, Y. Su, G. H. Low, and N. Wiebe, in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019* (Association for Computing Machinery, New York, NY, USA, 2019), p. 193.
- [49] S. Endo, Z. Cai, S. C. Benjamin, and X. Yuan, Hybrid quantum-classical algorithms and quantum error mitigation, *J. Phys. Soc. Jpn.* **90**, 032001 (2021).
- [50] K. Temme, S. Bravyi, and J. M. Gambetta, Error Mitigation for Short-Depth Quantum Circuits, *Phys. Rev. Lett.* **119**, 180509 (2017).
- [51] A. Strikis, D. Qin, Y. Chen, S. C. Benjamin, and Y. Li, Learning-Based Quantum Error Mitigation, *PRX Quantum* **2**, 040330 (2021).
- [52] J. Jiang, K. Wang, and X. Wang, Physical implementability of linear maps and its application in error mitigation, *Quantum* **5**, 600 (2021).
- [53] C. Piveteau, D. Sutter, and S. Woerner, Quasiprobability decompositions with reduced sampling overhead, *npj Quantum Inf.* **8**, 12 (2022).
- [54] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, Quantum error mitigation, (2022), arXiv preprint [ArXiv:2210.00921](https://arxiv.org/abs/2210.00921).
- [55] K. Fang, X. Wang, M. Tomamichel, and M. Berta, Quantum channel simulation and the channel's smooth max-information, *IEEE Trans. Inf. Theory* **66**, 2129 (2020).
- [56] F. Caruso, V. Giovannetti, C. Lupo, and S. Mancini, Quantum channels and memory effects, *Rev. Mod. Phys.* **86**, 1203 (2014).
- [57] M. M. Wilde, Entanglement cost and quantum channel simulation, *Phys. Rev. A* **98**, 042338 (2018).
- [58] M. G. Díaz, K. Fang, X. Wang, M. Rosati, M. Skotiniotis, J. Calsamiglia, and A. Winter, Using and reusing coherence to realize quantum processes, *Quantum* **2**, 100 (2018).