Adiabatic quantum learning

Nannan Ma,[1] Wenhao Chu,[1] P. Z. Zhao,[2] and Jiangbin Gong[1,2,3,*]

[1]*Department of Physics, National University of Singapore, Singapore 117551, Singapore*
[2]*Centre for Quantum Technologies, National University of Singapore, Singapore 117543, Singapore*
[3]*Joint School of National University of Singapore and Tianjin University,*
International Campus of Tianjin University, Binhai New City, Fuzhou 350207, China

(Received 30 March 2023; accepted 3 October 2023; published 19 October 2023)

Quantum machine learning has attracted considerable interest due to its potential to improve certain learning tasks. In conventional quantum machine learning, the output is the expectation value of a preselected observable, and the projective measurement forces a quantum circuit to run many times to obtain the output with reasonable precision. In this work, we propose a protocol to utilize the adiabatic quantum evolution to execute quantum learning tasks, in which the output is obtained by the adiabatic weak measurement rather than the projective measurement. In comparison to previous protocols, we use only a single-shot measurement and therefore avoid the measurement repetition in the previous protocols. Moreover, our protocol allows us to extract the expectation values of multiple observables without disrupting the concerned quantum states.

DOI: 10.1103/PhysRevA.108.042420

I. INTRODUCTION

Quantum computation [1,2] is expected to lead the next computation revolution by exploiting quantum advantages [3]. Progress in quantum computation owes much to pioneering theoretical [4–7] and experimental [8–11] works. Meanwhile, inspired by the capability of machine learning [12], tremendous efforts have been made to transplant machine learning thinking and algorithms to quantum platforms [13–15], hence the emergence of the research topic of quantum machine learning. To date, quantum machine learning is mainly based on the so-called variational quantum circuits (VQCs) [16–22]. In VQCs, we apply a series of gate operations (unitary operators) to an initial state (e.g., the ground state $|0\rangle$ of the initial Hamiltonian). We use the parameterized gate operators to encode N-dimensional input $\mathbf{x} = (x_1, \ldots, x_N)$ as well as M training variables $\mathbf{w} = (w_1, \ldots, w_M)$. Then, we can achieve the expectation value $\langle 0|U^\dagger(\mathbf{x}, \mathbf{w})\hat{O}U(\mathbf{x}, \mathbf{w})|0\rangle$ of an observable $\hat{O}$, where $U^\dagger(\mathbf{x}, \mathbf{w})$ is the overall time-evolution operator. This expectation value is regarded as the prediction $f_{\mathbf{w}}(\mathbf{x})$ of the learning task. In general, the expectation value is made from the statistic result of many projective measurement repetitions. In a stimulating development, a quantum version of the universal approximation theorem [18,23] was obtained, confirming the expressivity of a quantum circuit to approximate any integrable functional relationships.

The outlined architecture of VQCs relies on the necessary repetition of many projective measurements [24] in extracting the expectation value of an observable of interest. The precision of measurement is inherently related to the variance of the eigenvalues of a concerned observable. In situations involving excitations in a high-dimensional Hilbert space, the convergence of the measured observable can be slow. Even for a low-dimensional Hilbert space, many quantum gates are still required to encode input $\mathbf{x}$ and training variables $\mathbf{w}$, and the needed repetition number is proportional to the inverse square of the statistical error. Repeating this large quantum circuit many times makes it very complex to obtain an output with high precision [25], hindering its application in practice [26–30].

In this work, we propose an adiabatic quantum learning protocol based on the weak measurement [31,32] rather than the projective measurement to effectively save the needed repetition number of measurements. In a nutshell, the weak measurement is based on the weak coupling between a quantum system and a pointer (the measurement device) over a sufficiently long integration time T. The weak coupling is described by the Hamiltonian $H_{\text{int}} = \frac{\hat{p}\hat{O}}{T}$, where $\hat{p}$ is the momentum operator of the pointer and $\hat{O}$ is the observable in the quantum system. From first-order perturbation theory, one can conclude that if the system is prepared in one of the energy eigenstates, then there will be a small energy shift $\frac{p\langle\hat{O}\rangle}{T}$. Assuming that quantum adiabatic conditions [33,34] are obeyed, the system will stay in its instantaneous eigenstates. Meanwhile, the pointer changes over a distance proportional to $\langle O\rangle$, which is caused by the energy shift of the system-pointer coupling. Throughout this process, the system and the pointer will not get entangled due to the perturbative nature of the system-pointer coupling. Indeed, the adiabatic condition ensures that the system remains at its slightly perturbed energy eigenstates. That being the case, upon finally checking the actual position shift of the pointer subject to the initial spreading of the pointer states as the only source of error without a lower bound [35], one can infer the expectation value $\langle O\rangle$ without collapsing the measured state to one of the eigenstates of $\hat{O}$. This implies that the readout of the expectation value

[*] phygj@nus.edu.sg

is achieved without incurring a wave-function collapse. Recent brilliant experiments have demonstrated the in-principle feasibility of such a weak-measurement protocol in trapped-ion [36] and photon [37] systems. It is important to note that the prerequisite for implementing this weak-measurement scheme is preparing the system under measurement in one of its energy eigenstates. Indeed, it is the Hamiltonian of the system itself that protects the system in the eigenstate when weak coupling between the system and a pointer is turned on.

In our protocol, the learning process is based entirely on adiabatic quantum evolution; that is, throughout the process the system is always in one of its energy eigenstates. In this way, weak quantum adiabatic measurement can be introduced at any step of a quantum learning algorithm without destroying the quantum state. The expectation value of an observable of interest can be extracted without the need to repeat the quantum gate sequences. The extraction of noncommuting observables, as pointed by Aharonov *et al.*'s paper [31], is also possible and hence potentially brings in even more resource saving. This paper is organized as follows: we will present (i) how to encode a learning algorithm into operations on instantaneous Hamiltonians, (ii) how to design operations to make adiabatic quantum evolution possible, and (iii) how to execute the training procedure for such architectures. We shall implement our protocol in two binary classification tasks.

II. LEARNING PROTOCOL

A. Representation and operations

Consider a D-dimensional quantum system described by a traceless and time-dependent Hamiltonian $H(t) = \vec{n}(t) \cdot \vec{e}$, where $\vec{n}(t)$ is a unit vector and e_i are the bases of Lie group SU(D). Under the adiabatic evolution, the quantum system, initially in the ground state of $H(\vec{n}(0))$, will eventually be in the ground state of $H(\vec{n}(T))$ at time T. Different tracks of $\vec{n}(t)$ determine different final states and, consequently, different measurement values. Therefore, we incorporate the learning process into the track of $\vec{n}(t)$. Constructing such an $\vec{n}(t)$ track in a high-dimensional space is difficult since the gap related to the ground state may be closed at some points of $\vec{n}(t)$, which leads to a breakdown of adiabatic evolution. We adopt a simple strategy to overcome this problem, that is, limiting the possible operations for constructing the track of $\vec{n}$ to a small subset of all rotation operations. Such rotation operations on a unit vector $\vec{n}$ should be equivalent to a unitary transformation U on the corresponding Hamiltonian, $\vec{n'} \cdot \vec{e} = U\vec{n} \cdot \vec{e}U^\dagger$. It can preserve the spectrum of the Hamiltonian and hence will not introduce any level crossing during the operation.

Specifically, let us map a Hamiltonian to a point on the $(D^2 - 1)$-dimensional unit sphere. We consider only the rotation operation with its axis characterized by a unit vector $\vec{m}$ and consider the rotation angle to be the parameter θ. Using A as the generator corresponding to the rotation axis $\vec{m}$, such that

$$A_{ij} = \sum_k C_{ijk} m_k, \tag{1}$$

the resulting rotation matrix can be expressed by $e^{\theta A}$. Here, the coefficient C_{ijk} is the structure constant of the Lie algebra such that $[e_i, e_j] = 2i \sum_{k=1}^N C_{ijk} e_k$, and A is skew symmetric with $A^T = -A$. When such a rotation operation acts on $\vec{n}$, the new vector may be rewritten as $\vec{n'} = e^{\theta A}\vec{n}$. In order to keep the energy-level gaps open, we need to select the operation acting on $\vec{n}$ in the form of $e^{\theta A}$. The effect of such $e^{\theta A}$ on $\vec{n}$ is equivalent to a unitary operation on the Hamiltonian, so that the gap in the system Hamiltonian will not be closed under adiabatic evolution once the initial gap is open. A detailed explanation is given in the Appendix.

B. Learning structure

The learning task is to learn the ground-truth relation between input $\vec{x}$ and output y. We know that, restricting the operations on the vector $\vec{n}$ to the set $e^{\theta A}$ under adiabatic evolution, the track of $\vec{n}(t)$ determines the output of a learning process. In our protocol, the ground state of the initial Hamiltonian H_0 is used as the initial state. The initial Hamiltonian H_0 is constructed in such a way that the gap between the ground state and the first excited state is sufficiently large. The classical input data $\vec{x}$ are then encoded into the rotation angle of $e^{\theta A}$, forming the encoding part, where the rotation axis is fixed. Meanwhile, some of the remaining rotation operations are reserved as the adjustable part, whose rotation angles are taken as the variational parameters. Under adiabatic evolution, the quantum system evolves along the constructed track of $\vec{n}$. After the whole evolution, we obtain the predicted y, from which an approximate relation can be constructed. However, this relation is usually different from the ground-truth one. We need to train this learning structure by adjusting variational parameters with some known samples to approximate the ground-truth relation. This is the training process, and the samples used are known as the training dataset.

In the practical learning structure, we choose a special operation configuration as the encoding block, with the rotation parameters x being the input data, and another configuration as the variational block, with the rotation parameters w being the variational parameters for the training step. These two blocks make up a learning unit. To enable our learning model to gain better learning capability, we repeat the learning unit several times, so that our protocol implements the data reuploading strategy [38]. This point is supported by the fact that our adiabatic learning process can be mapped to a quantum circuit. After we implement the data reuploading strategy, a quantum circuit obtains better expressivity since the restricted frequency spectrum in its Fourier expansion has been extended [23,39]. In different units, the encoding blocks are the same, but the parameters of the different variational blocks are not necessarily taken to be the same values.

Then, we outline the execution process of the learning. As mentioned above, the ground state of H_0 is set as the initial state. The rotation operations in encoding blocks and variational blocks, whose rotation angles are determined by input x and variational parameters w, act on the unit vector $\vec{n}$, forming a track for the adiabatic evolution. For our purposes, we introduce the parameter $g = \Delta_t/\Delta_\theta$, where Δ_t is the duration of angle rotation Δ_θ and g is hence the period needed for a unit change in the rotation angle. To make the evolution along this track adiabatic, the inverse of g, which determines the speed of the adiabatic time evolution, needs to

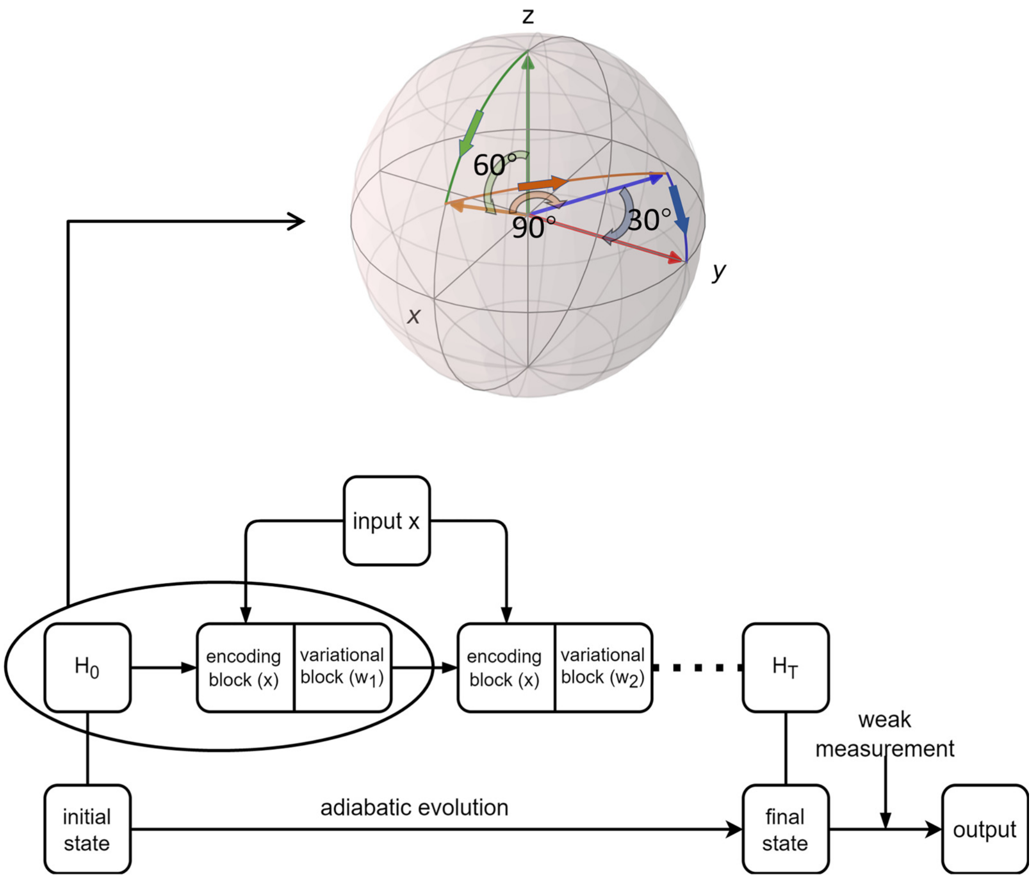

FIG. 1. The flow of the adiabatic learning protocol. The initial Hamiltonian H_0 and initial state are prefixed. The input x and variational parameters w control the rotation operations on the vector $\vec{n}(t)$ of the time-dependent Hamiltonian, and the state evolves adiabatically along the resulting evolution track of the Hamiltonian. During the entire learning process, the system remains in one of the energy eigenstates. This allows our learning protocol to be integrated with weak adiabatic measurement, such that the expectation value of an observable can be obtained without collapsing the state of the system. The repetition of a learning unit improves the expressivity, just like how data reuploading improves the learning capability of a conventional quantum circuit structure. For a two-dimensional Hilbert space with one-dimensional input $x = \pi/3$ and weight $w_1 = [\pi/2, \pi/6]$, there are three adiabatic rotations around the y axis (in an encoding clock) by x, around the z axis (in a variational block) by w_{11}, and around the x axis (in a variational block) by w_{12}. The Bloch sphere illustrates the track traced out by the Hamiltonian starting from $H_0 = \sigma_z$.

be sufficiently small in comparison to the spectral gap in the Hamiltonian. In this way, the final state is still the ground state of H_T, where T is the total evolution time. Consequently, we are able to extract the expectation values of some observables by using the weak adiabatic measurement advocated above. The layout of this learning protocol is shown in Fig. 1.

To understand the effect of the learning unit, we visualize a specific track of a time-evolving Hamiltonian determined by the first learning unit. As is illustrated in the Bloch sphere in Fig. 1, the encoding block has only one rotation operation around the y axis, and the variational block consists of two operations around the z and x axes, all preserving the spectrum of the Hamiltonian. The initial Hamiltonian is taken to be σ_z; thus, the initial state is $|1\rangle$. The one-dimensional input $x = \pi/3$ is encoded into the rotation angle around the y axis. Two-dimensional weights (variational parameters) $w_1 = [\pi/2, \pi/6]$ are the rotation angles of two rotation operations in the variational block. With these three rotation operations, the initial vector colored in green is rotated around y by 60° to the brown intermediate vector, then around z by 90° to the blue vector, and, finally, around x by 30° to the red vector. As a result, three arcs (green, brown, and blue) form the track depicting the adiabatic evolution path. The entire track of $\vec{n}$ can be obtained by repeating series of learning units.

C. Training

We have demonstrated the learning structure. Let us now move on to the training step for optimizing the variational parameters. We randomly select some samples with feature vectors $\vec{x}$ and known labels y. The training dataset consists of these selected sample $(\vec{x}, y)$, which are believed to be general. The constructed relation between $\vec{x}$ and y from the learning structure should approach the ground-truth relation. Therefore, we need an appropriate loss function that reflects the difference between predicted labels and the known labels of these training samples. One must attempt to minimize this loss function to get optimal variational parameters w in these variational blocks, with which an accurate y can be predicted for an input $\vec{x}$.

Assume that the adiabatic evolution can be achieved perfectly. Considering that a rotation operation on the

Hamiltonian is equivalent to a unitary operation acting on its ground state, there is a mapping between our adiabatic protocol and the conventional VQC. This implies that all optimization methods in the conventional VQC are applicable to our adiabatic protocol. There are two branches in optimization algorithms: one is based on gradient descent, and the other is gradient free. Numerical differentiation and parameter shift [19,40] are gradient-descent methods, and they can be executed solely on a quantum circuit without relying on a classical computer. Constrained optimization by linear approximation (COBYLA) is a gradient-free method. It is efficient for a problem with a relatively small number of variables and will be used in our verification cases. It is worth noting that these methods need the expectation values of some observables from experiments for optimization. In the conventional VQC, a large number of measurements will be executed. Differently, in our adiabatic learning protocol, the expectation value of concerned observables can be obtained with a single-shot measurement by invoking weak measurement.

In order to validate our learning protocol in principle, we adopt the strategy below. When a specific adiabatic learning structure has been constructed, we map it to an equivalent quantum circuit, and thus, various well-developed libraries for building and training a quantum circuit can be applied. In this way, we determine optimal variational parameters of this quantum circuit as well as those of the adiabatic learning structure with the training dataset $(\vec{x}, y)$. With an input and these optimal variational parameters, we then construct the track for the adiabatic evolution. Afterward, we simulate this adiabatic evolution numerically and calculate the final expectation value with the final evolving state, where a perfect weak measurement has been performed. There may be some errors due to the absence of perfect adiabaticity. However, our verification examples below show that this is a small issue.

III. IMPLEMENTATION

To demonstrate the feasibility of our adiabatic learning protocol, we implement our protocol in two binary classification tasks; one treats a one-dimensional data set, and the other treats a two-dimensional data set. We select COBYLA as our optimization method to get the optimal parameters.

A. Case I

We start with a working task for one-dimensional binary classification. We show the distribution of the label in Fig. 2(a). The input is a real number x. There are three isometric parts in the range $x \in (-1, 1)$. The outer two parts (blue) are arranged with the same label, $y = 1$, and the central part (green) is arranged with the label $y = 0$. For a sample i, the data are (x^i, y^i) with input $x^i \in (-1, 1)$ and label $y^i \in \{-1, 1\}$.

To accomplish this one-dimensional binary classification task, we execute an adiabatic learning protocol based on a two-dimensional Hilbert space, i.e., a one-qubit system. The initial state is taken to be the qubit ground state $|0\rangle$, and the initial Hamiltonian is taken to be $-\sigma_z$, whose vector representation is $\vec{n}_0 = (0, 0, -1)$ according to the notation used in the previous section. The unit vector is allowed to rotate on the Bloch sphere embedded in three-dimensional parameter space. In a learning unit, the encoding block contains only one rotation operation because the input is one dimension. The rotation axis of this operation can be chosen as the x axis, whose vector representation is (1,0,0). The rotation angle is the input x. The variational block consists of three rotation operations: rotation around the z axis (0,0,1) by w_1, rotation around the y axis (0,1,0) by w_2, and rotation around the z axis (0,0,1) by w_3. The variational parameter in this block is hence a three-dimensional vector (w_1, w_2, w_3). To achieve good classification accuracy, we repeat this learning unit three times. For a given input, the track of the instantaneous Hamiltonian $H(t)$ is determined by $4 \times 3 = 12$ rotations. This track of the qubit Hamiltonian determines the evolution of the quantum state, and the final state is given by $|\phi_T\rangle = \hat{T}\{\exp[-i\int_0^T dtH(t)]\}|0\rangle$, where $\hat{T}$ is the time-ordering operator. In the training step, the adjustable parameters are implemented by three three-dimensional vectors, each of which is from one learning unit. Under the adiabatic evolution, the final state is an eigenstate of the final instantaneous Hamiltonian $H(T)$. The outcome can be the expectation value of the observable σ_z, denoted by e, with e ranging from -1 to 1. The predicted label y is determined by e according to the classification criterion,

$$y = \begin{cases} 0 & e \leqslant 0, \\ 1 & e > 0. \end{cases} \tag{2}$$

Having explained the learning structure, in following numerical experiments, we transform these rotation operations

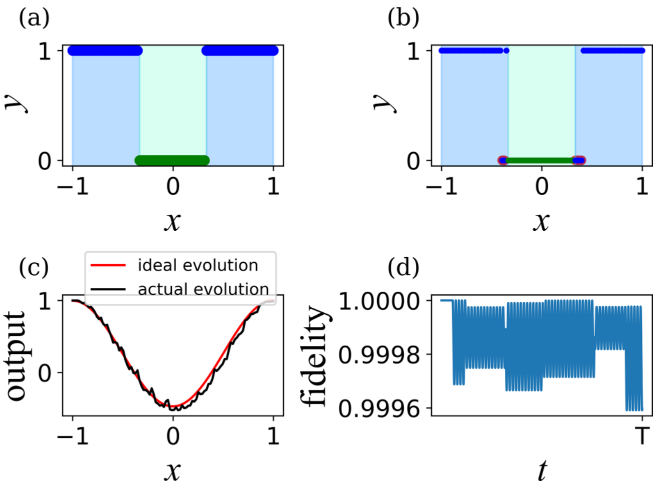

FIG. 2. One-dimensional binary classification. (a) The label y distribution in range $x \in [-1, 1]$. $y = 1$ when $|x| > \frac{1}{3}$, and $y = 0$ otherwise. (b) The classification result for 100 samples. The misclassifications marked with red circles occur in the neighborhood of the two boundaries. The accuracy of the classification is 94%. (c) The output difference between ideal evolution with perfect adiabaticity and the actual time evolution in our numerical simulation. The roughness of the actual simulation result is due to nonadiabatic effects. (d) The fidelity of the evolution state against the eigenstate of the instantaneous Hamiltonian for sample $x = 0$ for the whole adiabatic evolution. Note that although the fidelity remain at almost unity throughout the operation, the influence of the finite-time operation on adiabaticity is still visible in (c).

into their quantum circuit analogy. We then use the package QISKIT [41] to construct the quantum circuit and train it with COBYLA to get these optimal parameters.

In numerical experiments, we randomly select $N_t = 20$ samples from the distribution presented in Fig. 2(a) as the training dataset, which is $\{(x^1, y^1), \dots, (x^{N_t}, y^{N_t})\}$. After training, we get optimal parameters [−0.572, 0.643, 0.478, 1.57, 1.886, −1.225, −1.4, −1.568, 0.856]. The training score that represents the ratio of getting the correct prediction is 0.95. This means that there is only one sample classified wrongly. Then we uniformly select 100 samples from the distribution in Fig. 2(a) as the test dataset to verify our adiabatic learning structure. Again, for a given input, a quantum state prepared in $|0\rangle$ evolves along the track of the instantaneous Hamiltonians. We then get the output. In this process, we set the operation timescale parameter to be $g = 0.01/0.0005 = 20$ per angle change (the larger this number is, the closer to the adiabatic limit it is). Then, the evolution period for different inputs will be $\Delta_t = g \times \Delta_\theta$ (as mentioned in Sec. II B). The final classification result is presented in Fig. 2(b), and its accuracy can be up to 94%. This demonstrates that our adiabatic quantum learning works well. Here, it is also clear that the misclassifications (marked with red circles) all occur around two boundaries. This is understandable and typical because of the effect of the finite size of the training dataset.

The adiabatic evolution is guaranteed by the condition $T \gg \Gamma/\Delta^2$, where T is the entire period of time, Δ is the minimal energy gap, and $\Gamma = \max_{s\in[0,1]} \|\dot{H}(s)\|$ represents the evolution rate with $s = t/T \in [0, 1]$. In our protocol, the Hamiltonian $H(s)$ is determined by the learning structure, so the gap Δ and evolution rate Γ are fixed. Therefore, to guarantee the adiabatic evolution in an experimental implementation, we need to take a long evolution period T. It is adjusted by the scale parameter g. Considering that the adiabatic protocol is completed with a finite duration and thus the time-evolution state cannot exactly follow the track of the time-evolving Hamiltonian, we now turn to investigate nonadiabatic effects on the performance of our adiabatic learning. Figure 2(c) compares the output of actual time evolution with that of the ideal time evolution. The results match each other with high accuracy. To probe the nonadiabatic effects on the intermediate state, we calculate the fidelity $f = |\langle\phi_{\rm sim}(t)|\phi_g(t)\rangle|^2$ between the actual time-evolving state $|\phi_{\rm sim}(t)\rangle$ and the ideal state $|\phi_g(t)\rangle$. Figure 2(d) shows the fidelity f as a function of time t, where we select a sample with $x = 0$ from the test data. One can see that the fidelity stays close to unity for the finite-time evolution with $g = 20$. This implies that, to suppress possible nonadiabatic effects on the accuracy of our classification task, g should be on the order of 10, so the fidelity is expected to be around 99.9%.

B. Case II

Having benchmarked our protocol in a one-dimensional classification example, we stretch to a two-dimensional binary classification task. Figure. 3(a) depicts the distribution of this classification task. In a square with a width of 2, a circle divides it into two parts with the same area. The radius is $\sqrt{2/\pi}$. Like the situation in the previous case, we set the label of the part (blue) outside this circle as $y = 1$ and that of the other part (green) as $y = 0$. The input is a two-dimensional vector $[x_1, x_2]$, with $x_1, x_2 \in [-1, 1]$. The target output y for label 1 is 1, and that for label 0 is −1. In this task, a sample i has data $([x_1^i, x_2^i], y^i)$.

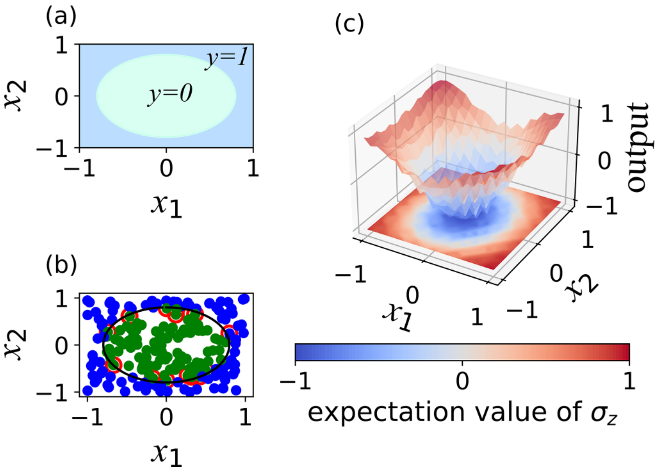

FIG. 3. Two-dimensional binary classification of a two-dimensional Hilbert space. (a) The label distribution inside a square of length 2, with the boundary of the two labels being a circle with a radius of $\sqrt{2/\pi}$. (b) The classification result of 200 randomly selected samples. The wrongly classified samples are marked with red circles and locate around the boundary. The accuracy is 92.5%. (c) The output distribution in this square. The surface is somewhat rough, which is again due to nonadiabatic effects in our actual time evolution. This is also reflected from its projection at the bottom, where the boundary (white part, output = 0) for classification is shown to be not a perfect circle.

Similarly, we execute an adiabatic learning structure based on a two-dimensional Hilbert space. The encoding block is composed of two rotations corresponding to the two elements of input. They are the rotation around the z axis by x_1 and the following rotation around the y axis by x_2. The variational block is the same as the previous one. We still repeat this learning unit three times. The initial state is taken to be $|0\rangle$, and the observable for the measurement is set to be σ_z. We also train our variational parameters on the corresponding quantum circuit from which 200 samples are selected randomly as the training dataset. The optimal parameter vector is obtained as [−0.268, 1.628, 0.0176, 2.367, 0.1684, 2.796, 1.044, 1.616, 0.866], and the training accuracy is up to 91.5%.

We still simulate the adiabatic evolution using the adiabatic timescale parameter $g = 0.01/0.0005 = 20$ per angle change. The performance of such an adiabatic learning protocol is verified with a test dataset consisting of 200 samples, the result of which is presented in Fig. 3(b). The points classified wrongly are marked with red circles, and the classification accuracy is found to be 92.5%. Like in the previous case, these wrong points are located around the boundary (the black circle). Evidently, the adopted learning structure has effectively learned the pattern of this two-dimensional binary distribution. The output of adiabatic learning for all the input data is shown in Fig. 3(c), where the color represents the value of the output. From the contour plane projected at the bottom of Fig. 3(c), we can confirm that although the classification boundary is

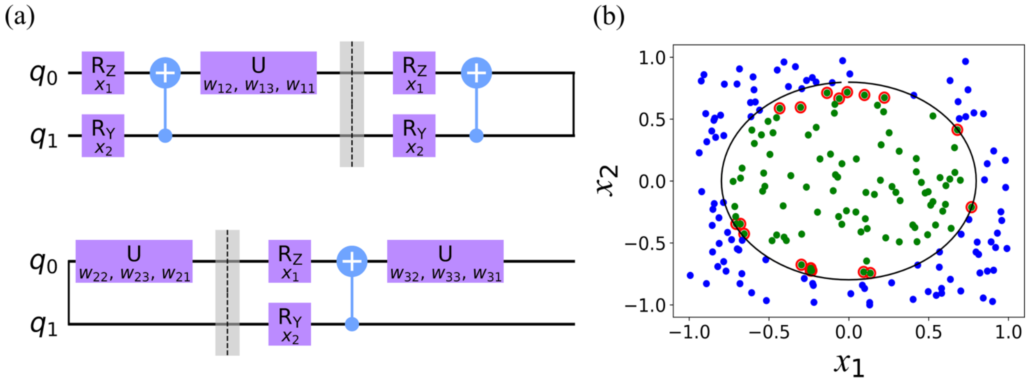

FIG. 4. Two-dimensional binary classification on a four-dimensional Hilbert space. (a) The equivalent quantum circuit on a two-qubit system. In a learning unit (separated by barriers), the encoding block consists of an R_Z gate (the first rotation in the encoding block) on qubit 0 and an R_Y gate (the second rotation) on qubit 1. The variational block is composed of a controlled-NOT gate, CNOT(1,0) (the first rotation in the variational block), and a U gate (the last three rotations) with three training parameters. (b) The classification result of 200 randomly selected samples. The wrongly classified samples are marked with red circles and locate around the boundary. The accuracy is 91%.

not a perfect circle, it really reflects the correct pattern with an acceptable error.

Besides the above learning structure, we also consider a learning structure based on a four-dimensional Hilbert space, i.e., a two-qubit system. A Hermitian operator in this Hilbert space can be expanded in term of 15 bases, $\{\frac{\sqrt{2}}{2}\sigma_{\lfloor (i+1)/4\rfloor} \otimes \sigma_{(i+1) \bmod 4}\}/\{\sigma_0 \otimes \sigma_0\}, i \in \{0, 1, \ldots, 14\}$. The initial state is taken to be $|00\rangle$. The vector representation of the initial Hamiltonian is $\vec{n}_0$, with the nonzero elements being $n_{0i} = -1/\sqrt{3},\ i \in \{2, 11, 14\}$. In the learning unit, the encoding block consists of two rotation operations corresponding to the two elements of input. They are selected as rotation around axis $\vec{r}_a$ (whose nonzero element in our vector representation is $r_{ai} = 1,\ i = 2$) by angle x_1 and a second rotation around axis $\vec{r}_b$ (whose nonzero element in our vector representation is $r_{bi} = 1,\ i = 7$) by angle x_2. In the variational block, we introduce four rotation operations: rotation around axis $\vec{r}_c$ with its nonzero elements $r_{c0} = r_{c11} = 1/\sqrt{3}$ and $r_{c12} = -1/\sqrt{3}$ by angle $2\pi/\sqrt{3}$, rotation around axis $\vec{r}_a$ (already defined above) by angle w_1, rotation around axis $\vec{r}_d$ with its nonzero element $r_{d1} = 1$ by angle w_2, and, finally, rotation around axis $\vec{r}_a$ by angle w_3. The three variational parameters in the variational block constructed this way form a three-dimensional vector. We repeat the learning unit three times, in the same fashion as for the above single-qubit learning structures, and then train this model after transforming it to the corresponding quantum circuit in Fig. 4(a). The optimal parameters are obtained as $[-1.553, -1.573, 0.6506, -1.967, -0.008876, -2.018, -2.996, -1.565, -0.9585]$, and the training accuracy is up to 95% using the quantum circuit. Our adiabatic learning model is then verified on our test data composed of 200 random samples with an adiabatic timescale parameter $g = 0.02/0.0003 = 66.6$ per angle change. The adiabatic learning classification result is presented in Fig. 4(b), and the obtained accuracy is found to be 91%.

Our results demonstrate that the adiabatic learning structure outlined above can also achieve good learning capability in comparison to others based on more general quantum circuits. The peculiar advantage of our protocol is that, even though some small nonadiabatic effects exist, the state of the system always stays in the ground state of the instantaneous Hamiltonian, so that our protocol, in principle, allows us to carry out adiabatic weak measurements at any point of the learning process. Hence, we can bypass the costly repetition of projective measurements to obtain the expectation value of an observable. Finally, we would like to mention that our simulations are computational and hence ideal. When our protocol is implemented on actual hardware, nonideal effects may arise. For that reason further engineering adiabaticity may be useful [42].

IV. CONCLUSION

In this work, we proposed an adiabatic learning structure which allows us to obtain the expectation value with the innovative weak-measurement scheme proposed by Aharonov *et al.* [31], instead of projective measurement in the conventional structure based on a quantum circuit. The output state in conventional quantum learning protocols is usually not the eigenstate of a prechosen observable, so repetition of the strong projective measurement is needed to obtain expectation values, which always destroys the final state. The key message from this paper is that the quantum learning structure can be integrated with weak measurement by letting the time-evolving states always be those of energy states. Therefore, at the end of quantum operations, the expectation value of an observable can be obtained by a single-shot measurement without collapsing the state. This will greatly facilitate the optimization process in variational learning since the complexity of output measurements in a quantum learning machine has been reduced. For our purposes, we encoded the learning process into the rotations on the Hamiltonian matrix around a set of axes to construct a learning structure. We showed that our learning protocol has the same learning capacity as the traditional protocols based on more general quantum circuits. However, our learning protocol allows us to probe the intermediate state without disturbing the computation or learning process. This is different from a conventional protocol in which the computation process is regarded as a black box. Hence, the flow of our protocol becomes transparent, and any unexpected error can be monitored along the way without destroying the time-evolving state. The actual performance of our protocol in real experimental platforms should be explored in the near future.

Recently, some other learning structures have been developed, such as deep multilayer perceptrons [43] and quantum-classical convolutional neural networks [44,45]. We note that more layers would require more measurements in the interior of their learning processes. A way to reduce complications in the measurement steps there is urgent, and our adiabatic learning protocol will likely provide a long-term solution when these learning structure can be implemented in our protocol.

APPENDIX: MAPPING BETWEEN ADIABATIC EVOLUTION AND QUANTUM CIRCUITS

Consider now a traceless Hamiltonian, which can be expressed with N Lie algebra group elements e_i, $i \in \{1, \ldots, N\}$. In other words, an N-dimensional unit vector $\vec{n}$ represents a Hamiltonian through the relation $H = \vec{n} \cdot \vec{e}$. We have also assumed that C_{abc} is the structure constant, with $[e_a, e_b] = 2i \sum_{c=1}^{N} C_{abc} e_c$. After a selected rotational operation R around an N-dimensional unit vector $\vec{m}$ by angle θ, the new unit vector $\vec{n'} = R(\vec{n})$ representing a new Hamiltonian H' can be regarded as the result of a unitary operator U acting on H, $H' = UHU^{\dagger}$.

Here, we give the explicit mapping relation to find the corresponding unitary operator U. As shown in the main text, the selected rotational operation around $\vec{m}$ by θ is in the form of $R = e^{\theta A}$, with $A_{ab} = \sum_c C_{abc} m_c$. As the rotation angle approaches zero, i.e., $\theta \to 0$, a straightforward relation emerges:

$$n'_a = \sum_b R_{ab} n_b = \sum_b \Bigg(\delta_{ab} + \theta \sum_c m_c C_{abc} + \frac{1}{2}\theta^2 \sum_{cde} C_{acd} C_{cbe} m_d m_e + \cdots \Bigg) n_b. \tag{A1}$$

The effect on the Hamiltonian then becomes

$$\begin{aligned}
\vec{n'} \cdot \vec{e} &= \vec{n} \cdot \vec{e} + \theta \sum_{abc} m_c n_b C_{abc} e_a + \frac{1}{2}\theta^2 \sum_{abcde} C_{acd} C_{cbe} m_d m_e n_b e_a + \cdots \\
&= \vec{n} \cdot \vec{e} + i(\theta/2) \sum_{ab} m_a n_b [e_a, e_b] + \frac{1}{2}(i\theta/2)^2 \sum_{abcd} 2i C_{cbd} [e_a, e_d] + \cdots \\
&= \vec{n} \cdot \vec{e} + i(\theta/2) \sum_{ab} m_a n_b [e_a, e_b] + \frac{1}{2}(i\theta/2)^2 \sum_{abc} m_a n_b m_c (e_a [e_c, e_b] + [e_b, e_a] e_c) + \cdots \\
&= \left\{ I + i(\theta/2)\vec{m} \cdot \vec{e} + \frac{1}{2} i[(\theta/2)\vec{m} \cdot \vec{e}]^2 + \cdots \right\} \cdot \vec{e} \left\{ I - i(\theta/2)\vec{m} \cdot \vec{e} + \frac{1}{2} - i[(\theta/2)\vec{m} \cdot \vec{e}]^2 + \cdots \right\} \\
&= e^{i(\theta/2)\vec{m} \cdot \vec{e}} \vec{n} \cdot \vec{e} e^{-i(\theta/2)\vec{m} \cdot \vec{e}} \\
&= U(\theta) \vec{n} \cdot \vec{e} U^{\dagger}(\theta).
\end{aligned} \tag{A2}$$

Thus, with an infinitesimal rotation, the relation between a unitary transformation $U(\theta) = e^{i(\theta/2)\vec{m} \cdot \vec{e}}$ experienced by a Hamiltonian and the corresponding rotational operation $R = e^{\theta A}$ on the corresponding unit vector is obvious. With multiplication of many such infinitesimal rotations, it becomes obvious that a corresponding unitary operator can be found for any rotation angle θ. This clarifies the mapping between a rotational operation in our protocol and unitary operators in a conventional quantum circuit. This also explains why we can first find optimal variational parameters from the corresponding quantum circuits in the training step and then finally do the finite-time rotational operations to implement our adiabatic learning protocol. Finally, if $N = D^2 - 1$ is now as large as the dimension of the Lie algebra associated with the whole Hilbert space of the system, then any unitary operator in a quantum circuit can be written as $U(\theta)$ via the exponential of some Hermitian operators (apart from a global phase). In this limiting case, any unitary operator considered in a quantum circuit can be realized by a corresponding rotational operator, whose experimental realization can, however, be challenging for a high-dimensional Hilbert space.

[1] S. Lloyd, Universal quantum simulators, Science **273**, 1073 (1996).

[2] D. P. DiVincenzo, Quantum computation, Science **270**, 255 (1995).

[3] M. A. Nielsen and I. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, Cambridge, 2002).

[4] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM Rev. **41**, 303 (1999).

[5] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual Symposium on the Theory of Computing* (ACM Press, New York, 1996), pp. 212–218.

[6] A. Montanaro, Quantum algorithms: An overview, npj Quantum Inf. **2**, 15023 (2016).

[7] Y. Liu, S. Arunachalam, and K. Temme, A rigorous and robust quantum speed-up in supervised machine learning, Nat. Phys. **17**, 1013 (2021).

[8] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. Obrien, A variational

eigenvalue solver on a photonic quantum processor, Nat. Commun. **5**, 4213 (2014).

[9] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

[10] L. M. Vandersypen, M. Steffen, G. Breyta, C. S. Yannoni, M. H. Sherwood, and I. L. Chuang, Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, Nature (London) **414**, 883 (2001).

[11] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. S. L. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler *et al.*, Quantum supremacy using a programmable superconducting processor, Nature (London) **574**, 505 (2019).

[12] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, Machine learning and the physical sciences, Rev. Mod. Phys. **91**, 045002 (2019).

[13] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature (London) **549**, 195 (2017).

[14] I. Cong, S. Choi, and M. D. Lukin, Quantum convolutional neural networks, Nat. Phys. **15**, 1273 (2019).

[15] M. Schuld and N. Killoran, Quantum machine learning in feature Hilbert spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[16] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New J. Phys. **18**, 023023 (2016).

[17] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Theory of variational quantum simulation, Quantum **3**, 191 (2019).

[18] T. Goto, Q. H. Tran, and K. Nakajima, Universal approximation property of quantum machine learning models in quantum-enhanced feature spaces, Phys. Rev. Lett. **127**, 090506 (2021).

[19] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[20] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature (London) **567**, 209 (2019).

[21] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, Quantum embeddings for machine learning, arXiv:2001.03622.

[22] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Phys. Rev. A **101**, 032308 (2020).

[23] A. Pérez-Salinas, D. López-Núñez, A. García-Sáez, P. Forn-Díaz, and J. I. Latorre, One qubit as a universal approximant, Phys. Rev. A **104**, 012405 (2021).

[24] J. Von Neumann, *Mathematical Foundations of Quantum Mechanics* (Princeton University Press, Princeton, NJ, 2018).

[25] This measurement procedure is irreversible because the initial state is destroyed after such a projective measurement. This flaw leads to a statistical problem when performing a measurement in practice, the accuracy. Assume that the expectation value is $\langle O\rangle$ and the variance is $\mathrm{var}(O) = [\Delta(O)]^2$; the measured expectation is the mean value of n measurements $\bar{O} = \frac{1}{n}\sum_{i=1}^{n} O_i$, where O_i is the result of the ith measurement. According to Chebyshev's inequality, the deviation of the final result $\bar{O}$ should be less than the error δ with a possibility larger than $1-\varepsilon$, that is, $P(|\bar{O} - \langle O\rangle| \leqslant \delta) \geqslant 1-\varepsilon$ if $n \geqslant \frac{\mathrm{var}(O)}{\varepsilon\delta^2}$ for any positive values ε and δ. Meanwhile, the Heisenberg uncertainty principle tells us that $\Delta(C)\Delta(D) \geqslant \frac{1}{2}|\langle[C, D]\rangle|$. This fact indicates that in some cases, we can obtain a good result for observable C with a few measurements if its variance is small enough. However, another observable, D, which does not commute with C, has a large variance, requiring multiple measurements to achieve good accuracy.

[26] D. Alonso and A. R. García, Single-energy-measurement integral fluctuation theorem and nonprojective measurements, Phys. Rev. E **108**, 024126 (2023).

[27] L. Banchi and G. E. Crooks, Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule, Quantum **5**, 386 (2021).

[28] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, Nat. Phys. **16**, 1050 (2020).

[29] A. Gresch and M. Kliesch, Guaranteed efficient energy estimation of quantum many-body Hamiltonians using shadow-grouping, arXiv:2301.03385.

[30] M. Kohda, R. Imai, K. Kanno, K. Mitarai, W. Mizukami, and Y. O. Nakagawa, Quantum expectation-value estimation by computational basis sampling, Phys. Rev. Res. **4**, 033173 (2022).

[31] Y. Aharonov, D. Z. Albert, and L. Vaidman, How the result of a measurement of a component of the spin of a spin-1/2 particle can turn out to be 100, Phys. Rev. Lett. **60**, 1351 (1988).

[32] Y. Aharonov and L. Vaidman, Measurement of the Schrödinger wave of a single particle, Phys. Lett. A **178**, 38 (1993).

[33] M. Born and V. Fock, Beweis des adiabatensatzes, Z. Phys. **51**, 165 (1928).

[34] T. Kato, On the adiabatic theorem of quantum mechanics, J. Phys. Soc. Jpn. **5**, 435 (1950).

[35] D.-J. Zhang and J. Gong, Dissipative adiabatic measurements: Beating the quantum Cramér-Rao bound, Phys. Rev. Res. **2**, 023418 (2020).

[36] Y. Pan, J. Zhang, E. Cohen, C.-W. Wu, P.-X. Chen, and N. Davidson, Weak-to-strong transition of quantum measurement in a trapped-ion system, Nat. Phys. **16**, 1206 (2020).

[37] F. Piacentini, A. Avella, E. Rebufello, R. Lussana, F. Villa, A. Tosi, M. Gramegna, G. Brida, E. Cohen, L. Vaidman, I. P. Degiovanni, and M. Genovese, Determining the quantum expectation value by measuring a single photon, Nat. Phys. **13**, 1191 (2017).

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

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

[40] J. Li, X. Yang, X. Peng, and C.-P. Sun, Hybrid quantum-classical approach to quantum optimal control, Phys. Rev. Lett. **118**, 150503 (2017).

[41] Qiskit Contributors, QISKIT: An open-source framework for quantum computing (2023), doi: 10.5281/zenodo.2573505.

[42] T. Chasseur, L. S. Theis, Y. R. Sanders, D. J. Egger, and F. K. Wilhelm, Engineering adiabaticity at an avoided crossing with optimal control, Phys. Rev. A **91**, 043421 (2015).

[43] M. Alam and S. Ghosh, DeepQMLP: A scalable quantum-classical hybrid deepneural network architecture for classification, *Proceedings of the 35th International Conference on VLSI Design and 2022 21st International Conference on Embedded Systems (VLSID)* (IEEE, New York, 2022), pp. 275–280.

[44] M. Henderson, S. Shakya, S. Pradhan, and T. Cook, Quanvolutional neural networks: Powering image recognition with quantum circuits, Quantum Mach. Intell. **2**, 2 (2020).

[45] J. Liu, K. H. Lim, K. L. Wood, W. Huang, C. Guo, and H.-L. Huang, Hybrid quantum-classical convolutional neural networks, Sci. China Phys. Mech. Astron. **64**, 290311 (2021).