

## RESEARCH ARTICLE

## Reconstructing unknown quantum states using variational layerwise method

Junxiang Xiao<sup>1</sup>, Jingwei Wen<sup>1</sup>, Shijie Wei<sup>2,†</sup>, Guilu Long<sup>1,3,4,2,‡</sup><sup>1</sup>State Key Laboratory of Low-Dimensional Quantum Physics and Department of Physics, Tsinghua University, Beijing 100084, China<sup>2</sup>Beijing Academy of Quantum Information Sciences, Beijing 100193, China<sup>3</sup>Frontier Science Center for Quantum Information, Beijing 100084, China<sup>4</sup>Beijing National Research Center for Information Science and Technology, Beijing 100084, ChinaCorresponding authors. E-mail: <sup>†</sup>[weisj@baqis.ac.cn](mailto:weisj@baqis.ac.cn), <sup>‡</sup>[gllong@tsinghua.edu.cn](mailto:gllong@tsinghua.edu.cn)

Received December 19, 2021; accepted March 1, 2022

In order to gain comprehensive knowledge of an arbitrary unknown quantum state, one feasible way is to reconstruct it, which can be realized by finding a series of quantum operations that can refactor the unitary evolution producing the unknown state. We design an adaptive framework that can reconstruct unknown quantum states at high fidelities, which utilizes SWAP test, parameterized quantum circuits (PQCs) and layerwise learning strategy. We conduct benchmarking on the framework using numerical simulations and reproduce states of up to six qubits at more than 96% overlaps with original states on average using PQCs trained by our framework, revealing its high applicability to quantum systems of different scales theoretically. Moreover, we perform experiments on a five-qubit IBM Quantum hardware to reconstruct random unknown single qubit states, illustrating the practical performance of our framework. For a certain reconstructing fidelity, our method can effectively construct a PQC of suitable length, avoiding barren plateaus of shadow circuits and overuse of quantum resources by deep circuits, which is of much significance when the scale of the target state is large and there is no a priori information on it. This advantage indicates that it can learn credible information of unknown states with limited quantum resources, giving a boost to quantum algorithms based on parameterized circuits on near-term quantum processors.

**Keywords** variational quantum algorithm, layerwise learning, quantum state reconstructing

## 1 Introduction

Quantum information provides many of the most cutting-edge researches nowadays, including quantum simulations [1–3], quantum computing [4–8], quantum cryptology and quantum communication [9–18]. In the field of quantum information, information is encoded into quantum states. Many times we wish to obtain the full information of the states, in situations such as quantum device analysis [19], which promotes the emergence of technologies like quantum state tomography (QST). Although we can determine the classical expression of unknown quantum states through QST, how to reproduce the states and how to design the corresponding quantum circuit are still problems that need further discussion. Therefore, it is valuable to find a quantum circuit that can directly reconstruct the unknown quantum states since finding the unitary operations to prepare states can help us get rid of

the disadvantages that quantum states have a limited lifetime and are difficult to store, allowing us to manipulate them anytime and anywhere.

In 2018, an autonomous protocol based on quantum reinforcement learning (QRL) was put forward [20], which assumes enough copies of the input state, and requires no structural information of the inputs beforehand. The protocol introduces a quantum register to interact with the environment — the unknown state ensemble, and takes measurement on it, the result of which is used to adjust the “agent” state. Once the algorithm converges, the “agent” will have fidelities around 90% with respect to the input unknown qubit state. On the other hand, parameterized quantum circuits (PQCs) with certain objective functions are believed to be good solutions to many quantum problems, from solving quantum algebraic systems [21–24] and finding the ground state energy of a chemical system [25–31] to performing image classification [32, 33] and fitting unknown functions [34] in the noisy intermediate-scale quantum (NISQ) era [35]. In the field of quantum machine learning, PQCs are known as a type of quantum neural networks [36–39].

\* This article can also be found at <http://journal.hep.com.cn/fop/EN/10.1007/s11467-022-1157-2>.

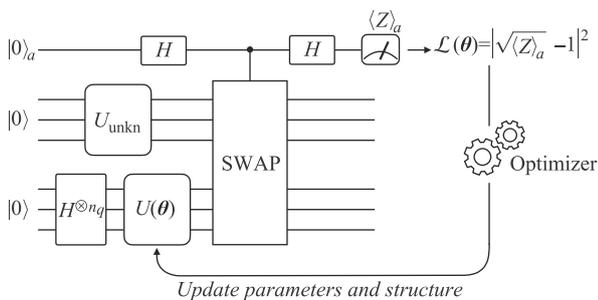


In this paper, we present a framework that can reconstruct unknown quantum states in an adaptive way. Different from the QRL method that keeps gate parameters adopted in previous iterations unchanged, we make the circuit parameterized, introduce classical gradient optimizers into our protocol to boost the final fidelities, and tune all parameters iteratively. We introduce the variational layerwise learning (LL) technique [33] as a strategy to dynamically adjust the PQC structure, yielding a circuit of suitable depth automatically and avoiding the requirement of experienced pre-designs of parameterized circuits in other ordinary non-adaptive algorithms, which is especially significant when the system scale gets large. This not-over-large depth of the obtained circuit can also serve as an exact upper bound for the minimal depth of the circuit that can reconstruct the input state at the required fidelity with the same layered structure. We test the performance of the framework for random states containing up to six qubits numerically, which shows that our framework can achieve the goal of quantum states reconstruction with more than 96% fidelities. We perform hardware experiments on IBM Quantum processor to demonstrate the practicality of our framework. Moreover, comparisons with the QRL-based method and non-layerwise-learning strategy are made.

## 2 Method

We assume that we have enough copies of the unknown state as in [20]. The schematic of the optimization process of this algorithm is shown in Fig. 1. For an unknown state  $|\psi_{\text{unkn}}\rangle$  that contains  $n_q$  qubits, another  $n_q$  work qubits are required for state reconstructing, which will be manipulated by the PQC  $U(\theta)$ . The PQC is composed of several “blocks”, the  $i$ -th block of which takes the form of

$$U^{(i)} = W \prod_{j=1}^{n_q} R_j(\theta_{j,1}^{(i)}, \theta_{j,2}^{(i)}, \theta_{j,3}^{(i)}), \quad (1)$$



**Fig. 1** Optimization process of our algorithm.  $U_{\text{unkn}}$  is the oracle operator that can produce the unknown state  $|\psi_{\text{unkn}}\rangle$  from  $|0\rangle$ , the all-zero state.  $U(\theta)$  is our parameterized quantum circuit, whose structure and parameter values will be changed during the training process.

as shown in Fig. 2. Here  $R_j(\cdot, \cdot, \cdot)$  is the general single-qubit rotation on the  $j$ -th qubit ( $X, Y, Z$  are the Pauli operators):

$$R_j(\theta_1, \theta_2, \theta_3) = R_z(\theta_3)R_y(\theta_2)R_z(\theta_1) \quad (2)$$

$$= e^{-iZ\theta_3/2}e^{-iY\theta_2/2}e^{-iZ\theta_1/2}, \quad (3)$$

while  $W$  is the connection operator between qubits. In the diagram of Fig. 2,  $W$  is a series of two-qubit interactions having all-to-all connectivity, and it can be customized for different quantum systems.

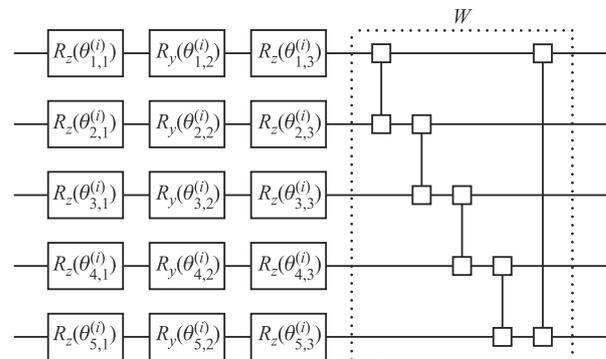
Similar to classical machine learning algorithms, a hybrid classical-quantum algorithm requires a proper loss function to train parameters. Here we take the overlap (i.e., fidelity) between work state and unknown state as the loss function:

$$\mathcal{L}(\theta) = \left| |\langle \psi(\theta) | \psi_{\text{unkn}} \rangle| - 1 \right|^2. \quad (4)$$

In order to evaluate its value from experimental results, we employ SWAP-Test to obtain the overlaps. Details of SWAP-Test are provided in Appendix A, and from Eq. (A8) we can derive that

$$\mathcal{L}(\theta) = \left| \sqrt{\langle Z \rangle_a} - 1 \right|^2. \quad (5)$$

Therefore, by introducing an auxiliary qubit and measuring its expectation value of Pauli  $Z$  operator, we can acquire the loss value experimentally. It is worth noticing that,  $\langle Z \rangle$  must be non-negative analytically, but if we perform our algorithm on a real-world quantum machine, a classical sampling simulator where we can only have frequencies of samples rather than exact probabilities, or a classical numerical simulator with limited precision, the approximated value of  $\langle Z \rangle$  may be negative when the exact value of  $\langle Z \rangle$  is near-zero. Therefore, enough number of



**Fig. 2** The  $i$ -th block of the PQC  $U(\theta)$ , which is composed of general rotations on each qubit followed by a connection operator  $W$ . In this diagram,  $W$  is a series of two-qubit interactions having all-to-all connectivity, and it can be customized for different quantum systems.

repetitions of sampling must be taken, and those negative  $\langle Z \rangle$  values close to zero should be clipped to zeros.

In the framework of Fig. 1, we choose layerwise learning strategy to adjust our PQC. In classical machine learning, layerwise learning strategy is proven to have comparable performance as complete depth learning (CDL) that trains all parameters in the whole network simultaneously. Skolik et al. introduced layerwise learning method into PQC-based quantum machine learning tasks [33]. They showed that when noiseless and analytic simulation is considered, LL and CDL display similar performance. But LL strategy works better on average when it comes to a real-world quantum device where noise is not negligible and measurements must be taken in order to evaluate the loss function and the gradients.

Another essential reason to use LL strategy is that it can gradually lengthen our PQC until convergence, which is adaptive as we wish. Compared with CDL strategy that requires us to determine the complete structure of PQC before running algorithms, LL strategy starts from a shallow PQC and can dynamically extend it on-demand, balancing the fidelity of our algorithm and quantum resources consumed by our PQC.

In general, our method adjusts not only the parameter values but also the structure of PQC by progressively increasing the number of blocks of the PQC and dividing all parameters into several groups which are trained iteratively. Before running our algorithm, several hyperparameters must be designated, whose meanings are listed in Table 1. As LL strategy in [33], our algorithm consists of two phases, which are detailed in Algorithm 1. For the clarity of presentation, we call lines 5 ~ 10 and line 18 a *training step* in phase I and phase II respectively.

In classical QST,  $O(2^{n_q})$  elements of the state density matrix need to be stored so that the state can be completely represented. Meanwhile, since these elements must be obtained via measurements individually, the execution time of classical QST also scales exponentially with the system size. In contrast, the qubit resources required by Algorithm 1 is  $2n_q = O(n_q)$ , and the state of the work qubits is directly the reconstructed state, without the need to convert classical density matrix data into quantum states. We use the number of queries to quantum gates through the entire process to estimate the upper

bound of time complexity of Algorithm 1. If the final number of blocks in the PQC is  $L$  and there are  $O(n_q)$  gates in each connection operator  $W$ , the gate complexity of the final PQC will be  $LO(n_q)$  and the total query complexity will be  $\sum_{l=L_s}^L O(n_q) l n_e g_1 + s \cdot (1/r) \cdot n_e g_2$ , where  $g_1, g_2$  are the query complexities of gradient evaluation in each training step of phase I and phase II respectively. For example, if parameter-shift rule [34, 40, 41] is employed in both phases, then  $g_1 = O(n_q)$ ,  $g_2 = O(rn_p) = O(Lrn_q)$ , and the total query complexity will be  $O(n_q^2)n_e L^2 + Ln_e s O(n_q) \leq O(n_q^2)n_e L(L+s) = O(n_q^2)$ , which also presents exponential improvement compared with classical QST. We owe this reduction of consumed resources to the hybrid quantum-classical framework, which combines the advantage of the intrinsic quantum representability of NISQ devices and well-developed optimization algorithms of classical computing.

## 3 Results

### 3.1 Numerical simulation

In order to run benchmarks on this quantum state reconstructing framework, we create random circuits and take them as unknown input states. The procedure is detailed in Appendix B.

We carry out 1000 numerical simulations for each  $n_q$  in  $\{1, 2, \dots, 6\}$ , and the configurations of our experiments are detailed in Appendix C. We record the losses at the end of each epoch and the evolution averaged over 1000 random input states during the complete training process is shown in Fig. 3(a). At the beginning of each training step, the loss value may have a sudden rise, which is the result of non-zero parameters in the newly appended block  $WU^{(i)}$ . Nevertheless, the overall performance can finally be improved by tuning these parameters and the model circuit will be brought away from local optima (and also barren plateaus) in the final parameter space. As a result, the entire curve exhibits a declining shape with fluctuations. For cases where  $n_q = 1$ , the loss values decrease so fast that they have been less than the threshold in the first several epochs before any early-stopping checks are

**Table 1** Hyperparameters of the state reconstructing algorithm.

|                              |   |
|------------------------------|---|
| $L_s$                        | Number of starting blocks.  |
| $L_m$                        | Maximal number of blocks in the full PQC.   |
| $n_e$                        | Maximal number of epochs in each training steps.  |
| $\mathcal{L}_{th}$           | Loss threshold, which is one of the early-stopping criteria. When the loss is less than this value, our algorithm is completed.   |
| $\delta\mathcal{L}, n_{pat}$ | Another criterion of early-stopping. During the running of gradient-based optimizer, if the reductions of loss do not exceed $\delta\mathcal{L}$ for $n_{pat}$ consecutive epochs, the optimizer will be early-stopped. |
| $r$                          | Parameter partition rate in phase II.   |
| $s$                          | Maximal number of sweeps in phase II.   |

**Algorithm 1** Reconstructing unknown quantum states using layerwise method.

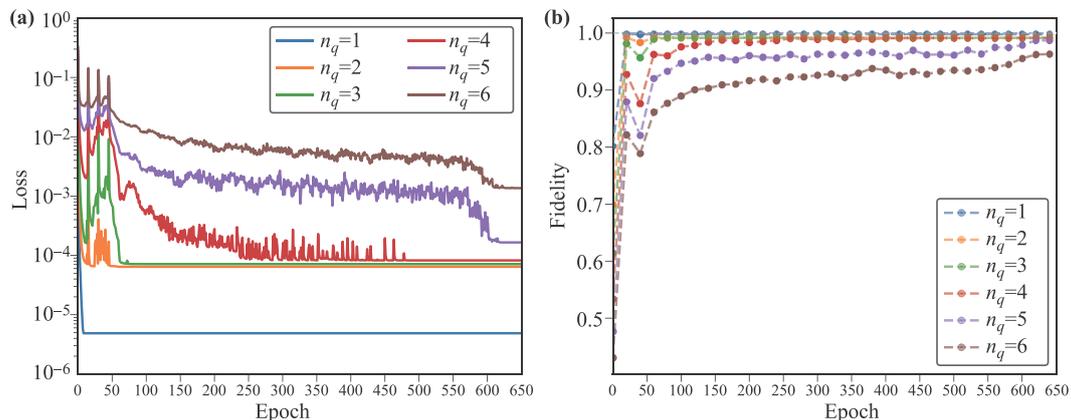
| Inputs   | unknown state $ \psi_{\text{unkn}}\rangle$ ; hyperparameters in Table 1   |
|--|---|
| Outputs  | trained PQC $U(\theta)$   |
| <pre> // Phase I 1 Prepare the working qubits on the evenly   superposed state <math>[( 0\rangle +  1\rangle)/\sqrt{2}]^{\otimes n_q}</math>; 2 <math>U(\theta) \leftarrow</math> circuit containing no blocks; 3 goOnTraining <math>\leftarrow</math> true; 4 while the number of blocks in <math>U(\theta) &lt; L_m</math> and   goOnTraining is true do 5   if there are no blocks in <math>U(\theta)</math> then 6       append <math>L_s</math> blocks into <math>U(\theta)</math>; 7   else 8       append 1 block into <math>U(\theta)</math>; 9   end 10  train the parameters in the newly appended     block(s) of <math>U(\theta)</math> (<math>n_e</math> epochs at most; if the     loss <math>&lt; \mathcal{L}_{\text{th}}</math>, stop optimizer and     goOnTraining <math>\leftarrow</math> false; if the reductions of     loss do not exceed <math>\delta\mathcal{L}</math> for <math>n_{\text{pat}}</math> consecutive     epochs, stop optimizer and turn to next     loop);     // with other parameter values fixed 11 end                 </pre> | <pre> // Phase II 12 <math>n_p \leftarrow</math> number of parameters in <math>\theta</math>; 13 paramGroups <math>\leftarrow</math>   <math>\{\theta[0, rn_p - 1], \theta[rn_p, 2rn_p - 1], \dots\}</math>;   // collect <math>rn_p</math> parameters into a group 14 <math>j \leftarrow 0</math>; 15 goOnTraining <math>\leftarrow</math> true; 16 while <math>j &lt; s</math> and goOnTraining is true do 17   foreach group in paramGroups do 18     train the parameters in group (<math>n_e</math> epochs       at most; if the loss <math>&lt; \mathcal{L}_{\text{th}}</math>, stop optimizer       and the foreach-loop, and       goOnTraining <math>\leftarrow</math> false; if the reductions of       loss do not exceed <math>\delta\mathcal{L}</math> for <math>n_{\text{pat}}</math>       consecutive epochs, stop optimizer and       turn to next group);       // with other parameter values       fixed 19   end 20   <math>j \leftarrow j + 1</math>; 21 end                 </pre> |

performed, while for other cases the early-stopping mechanism stops circuit training when the loss values gradually reach the threshold. This is the reason why the final value of the case  $n_q = 1$  is around 1/10 of those cases where  $n_q > 1$  in Fig. 3(a).

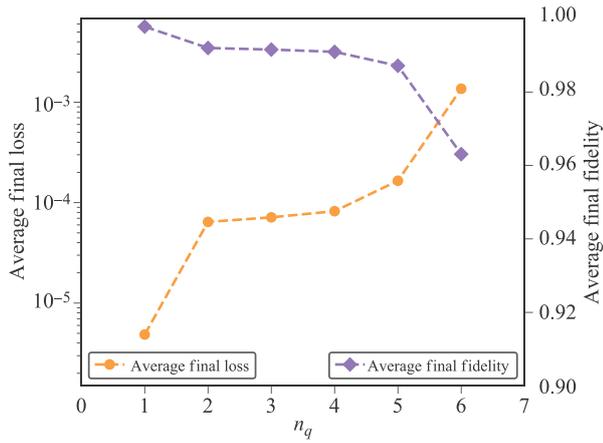
According to Eq. (4), we can evaluate the overlaps from loss values by  $|\langle \psi(\theta) | \psi_{\text{unkn}} \rangle| = 1 - \sqrt{\mathcal{L}(\theta)}$ . With this relationship, we calculate and plot the fidelities per 20 epochs in Fig. 3(b). For systems containing different number of qubits, the average final losses and final fidelities varies, as

revealed in Fig. 4. Figures 3(a), (b) and Fig. 4 show that for systems with less than 5 qubits, the final losses are smaller than  $10^{-4}$ , meaning that our PQC can learn unknown states with fidelities higher than 99% theoretically for systems with dimensions up to  $2^4 = 16$ . For larger systems like quantum states of 5 or 6 qubits, the fidelity of output state of our PQC is expected to be higher than 96%. Analyses on the convergence rate and training time are attached in Appendix D.

In order to study the effect of each circuit block on the



**Fig. 3** (a) Losses at the end of each epoch. (b) Fidelities evaluated per 20 epochs. All values are averaged over 1000 random input states. The number of qubits in the quantum systems varies from  $n_q = 1$  to  $n_q = 6$ .



**Fig. 4** Final losses and final fidelities (averaged over 1000 random input states) of quantum systems with  $n_q$  varying from 1 to 6. The quantum state reconstructing framework can effectively reproduce unknown quantum states with high fidelities.

reconstructed state, we conduct our algorithm to reconstruct the partially entangled state  $|\phi_0\rangle = |W_3\rangle \otimes |000\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle) \otimes |000\rangle$ . We start from a 0-block PQC  $|\psi_0\rangle = (H|0\rangle)^{\otimes 6}$  according to the procedure of Algorithm 1. The evolution of von Neumann entropy of the 6 subsystems

$$S_i = -\text{Tr}(\rho_i \log_2 \rho_i), \quad i = 0, 1, \dots, 5 \quad (6)$$

is monitored during the training process, where  $\rho_i$  is the partially traced density matrix of the  $i$ -th qubit. It is direct to derive that the analytical values of  $|\phi_0\rangle$  are

$$\rho_0 = \rho_1 = \rho_2 = \frac{1}{2}I, \quad \rho_3 = \rho_4 = \rho_5 = \text{diag}(1, 0); \quad (7)$$

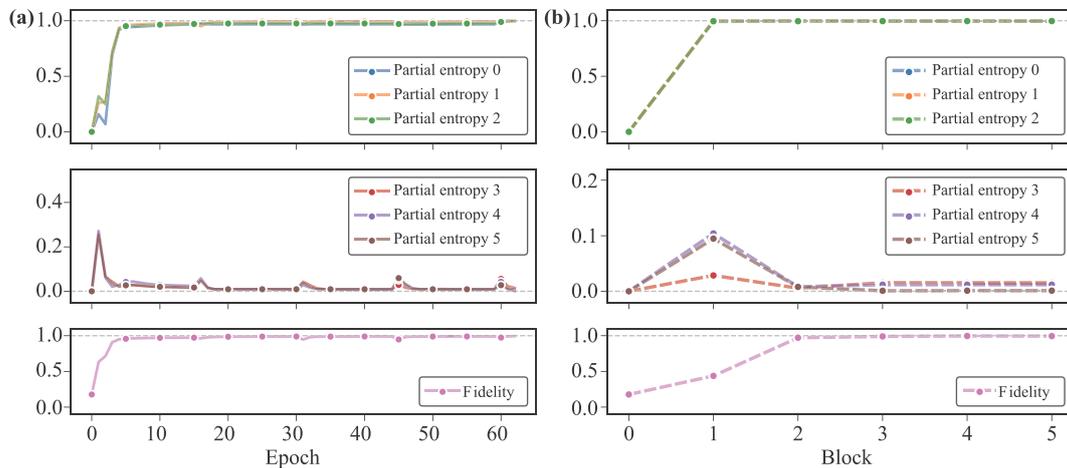
$$S_0 = S_1 = S_2 = 1, \quad S_3 = S_4 = S_5 = 0, \quad (8)$$

which match the simulation results in Fig. 5(a). As the algorithm proceeds, the parameterized circuit will lengthen gradually. Once converged, the algorithm gives out a circuit containing 5 blocks. We simulate this final circuit and evaluate the values of fidelity and entropy after the state passes each block, as in Fig. 5(b). After the first block, von Neumann entropy of entangled qubits (0, 1, 2) is mostly recovered, while the entanglement-free qubits (3, 4, 5) obtain extra entropy inappropriately. Therefore, a new block is added, with its parameters well-trained so that it can de-entangle (3, 4, 5) qubits, raising the entropy of (0, 1, 2) qubits at the same time. The last 3 blocks fine tune the state, making the degree of entanglement of each qubit more accurate. As the circuit deepens, the fidelity of state is gradually increased. Hence, different blocks in the PQC of our framework can adjust global or local properties of the work state according to the difference between input state and work state.

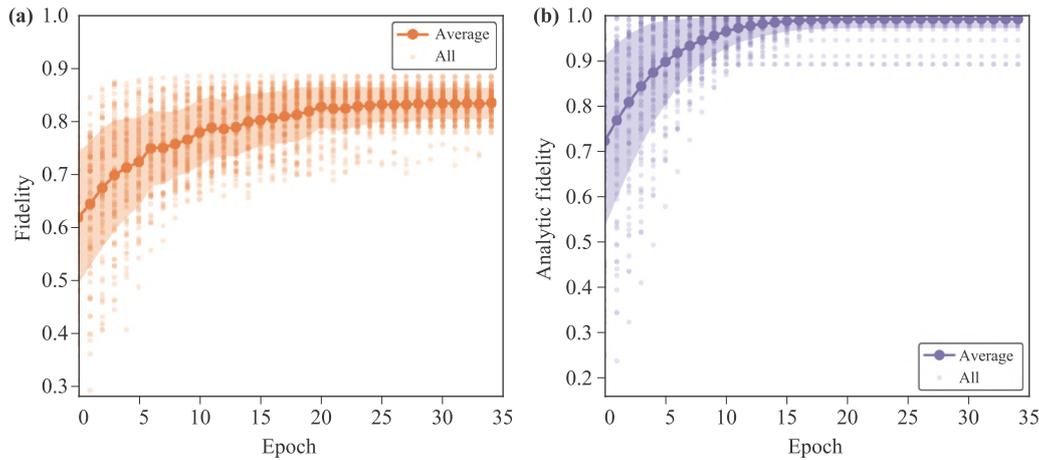
### 3.2 Experimental results

We deploy our framework on the 5-qubit IBM Quantum processor `ibmq_belem` [42] to reconstruct unknown single qubit states  $U(\alpha, \beta, \gamma)|0\rangle = R_z(\alpha)R_y(\beta)R_z(\gamma)|0\rangle$ , where  $\alpha, \beta, \gamma$  are randomized. The configurations are detailed in Appendix E. Due to hardware noise, the loss values evaluated from real measurement results will eventually be around 0.015 and not less than 0.010. As a result, the final fidelity is expected to be  $\sim 83.5\%$  according to Fig. 6(a), where the values represented by solid dots are averaged over 100 random cases.

Nevertheless, once the algorithm converges, the trained PQC can serve as a high precision noiseless approximation to  $U(\alpha, \beta, \gamma)$  when acted on  $|0\rangle$ , since the fidelity can reach  $\sim 99.1\%$  if we simulate the trained PQC analytically, as shown in Fig. 6(b). We owe this to the robustness of



**Fig. 5** The evolution of fidelity and von Neumann entropy per qubit as (a) training proceeds; (b) the state passes the blocks in the trained PQC.



**Fig. 6** We perform 100 random experiments for single qubit state reconstructing on `ibmq_belem`, and display the evolution of average fidelities from (a) real device output and (b) analytic noiseless simulation of PQC with parameter values same as (a). Here, the transparent dots represent all (epoch, fidelity) data points in the experiments, while the solid line stands for the average evolution of fidelity.

the parameter-shift rule adopted in gradient evaluation, since the gradient yielded still provides effective guidance to the optimization process even though the hardware is noisy [43].

### 3.3 Comparisons with other methods

We implement the QRL algorithm in [20], and do simulations on random input states generated with the same routine in Section 3.1 for systems containing 1 and 2 qubits. The QRL algorithm hyperparameter  $\epsilon$  ranges from 0.1 to 0.9, and we repeat the experiment 1000 times for each configuration. We compare the average final fidelities achieved by our method in Section 3.1 with the results of the QRL method in table 2, where the convergence rate stands for the proportion of converged cases in all 1000 experiments and the criterion for convergence differs in our algorithm and in the QRL algorithm, which is detailed in the first table annotation. For the simplest case — 1-qubit systems, the maximal value of average fidelity that can be achieved

by our method is raised to be higher than 99%. All 1000 simulations converged, and the learning process can be accomplished in 3 epochs on average. For larger systems, our advantage in fidelity is more obvious: the average final fidelity can be raised from around 90% (QRL) to 99% for 2-qubit systems, and is expected to reach 98.7% for 5-qubit systems if our PQCs converge.

We can also apply our PQC onto the input unknown state directly as proposed in [44], without need for SWAP-Test, the auxiliary qubit and the work qubits. In this construction, we need to optimize the PQC so that it can reset the state into  $|0\rangle$  (all-zero state). To achieve this goal we should measure all qubits of the state and take the loss function to be

$$\mathcal{L}(\theta) = |\langle \psi(\theta) | 0 \rangle \langle 0 | \psi(\theta) \rangle - 1|^2. \quad (9)$$

Once the loss function is reduced to near zero, we can invert the PQC and get  $\prod_{i=1}^{d_c} U^{(i)\dagger}(\theta^{(i)})$ . Applying this inverted series to  $|0\rangle$  will yield a state that is supposed to have high overlap with the input unknown state. This

**Table 2** The final fidelities achieved by the QRL method and the method of this paper averaged over 1000 random states.

|         | Method                 | Our method | QRL ( $\epsilon = 0.1$ ) | QRL ( $\epsilon = 0.3$ ) | QRL ( $\epsilon = 0.6$ ) | QRL ( $\epsilon = 0.9$ ) |
|---------|------------------------|------------|--------------------------|--------------------------|--------------------------|--------------------------|
| 1-qubit | Average final fidelity | 99.78%     | 90.43%                   | 91.62%                   | 93.48%                   | 96.35%                   |
|         | Convergence rate*      | 100%       | 93.10%                   | 91.20%                   | 89.30%                   | 66.10%                   |
| 2-qubit | Average final fidelity | 99.20%     | 83.29%                   | 84.43%                   | 86.33%                   | 76.99%**                 |
|         | Convergence rate       | 100%       | 99.50%                   | 99.70%                   | 98.70%                   | 55.30%                   |

\* The convergence rate is evaluated as  $n_{\text{conv}}/1000$ , where  $n_{\text{conv}}$  is the number of converged cases. For our method, those cases where final loss values are less than  $10^{-4}$  are considered as being converged. For the QRL method, the indicator of convergence is  $\Delta$ , the range of random rotation angles in each step [20]. We take those cases where  $\Delta < 10^{-5}$  as converged ones.

\*\* According to the original paper [20], as  $\epsilon$  grows, the algorithm achieves higher final fidelity but converges slower. Here more than 40% cases did not converge in the maximal number of iterations, which is set to be 1000 in our simulation. Therefore the corresponding average final fidelity is lower than cases with smaller  $\epsilon$ s.

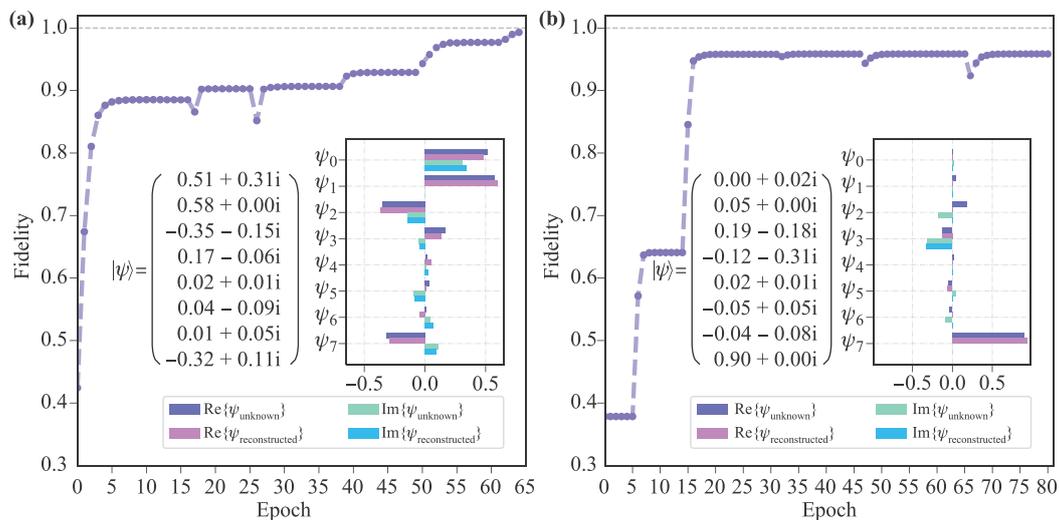
variant only consumes half of the quantum qubits of our framework, and avoids noisy executions of quantum circuits resulted from possibly erroneous controlled-SWAP gates, which involve interactions through the entire quantum system. However,  $O(n_q)$  times more measurements are required in order to estimate the fidelities with the same order of precision. Furthermore, on NISQ devices with noteworthy shot noise, if the probability of performing an erroneous measurement on a qubit is  $p_e$  and we need to evaluate the fidelity based on  $M$  circuit execution results, then  $p_e M$  results are expected to be correctly sampled from the probability distribution of the quantum state in our algorithm, while the value is  $p_e^n M$  for the cSWAP-free method since its fidelity relies on the measurement results of all qubits. As a consequence, the more qubits are to be measured, the less precise results we will obtain. Therefore, when we are working on a quantum device with significant multi-qubit interaction error, the cSWAP-free method in [44] exhibits higher applicability, while our method is more robust and more suitable if the measurement noise is not negligible.

We conduct numerical experiments to compare the performances of our algorithm using layerwise learning strategy and non-layerwise learning strategy, i.e., complete depth learning. Figure 7(a) illustrates how the fidelities are raised in a specific example using LL-strategy, where readout noise with bit-flip probability being 20% is assumed. We run CDL to reconstruct the same state, with the circuit structure same as the final PQC of LL, and initial values of all parameters identical to those of LL. Optimization of CDL converges in only 10 epochs and early-stops, with the final fidelity being 86.36%, while LL gives out a fidelity up to 99.32%, from which we can see

that our framework can avoid converging too early and being stranded in barren plateaus. According to [33], LL-strategy takes more epochs than CDL-strategy, but less measurements are to be taken in each epoch due to fewer parameters to be updated, which leads to the same amount of calls to quantum devices.

However, LL may not surpass CDL regarding to the final fidelity under all circumstances, suggested by another example shown in Fig. 7(b). In this case LL method converges at fidelity 95.84%, while CDL method reaches 96.05%. We opine that given the block structure of the multi-layer (i.e., block) PQC, the significance of layerwise learning strategy is to help our algorithm find a relatively short circuit that can reconstruct the unknown state at high fidelity in an automatic and heuristic manner, avoiding fixing the structure of the complete PQC beforehand like in CDL, which may overuse quantum resources if the PQC contains too many blocks, especially when there is significant circuit noise, or may not able to achieve the fidelity goal if the PQC is too short. Our algorithm starts training with a low-depth circuit. If the PQC is too shallow to converge to the input state, which means the algorithm sticks in the barren plateau far away from the input state, LL-strategy will dynamically extend the circuit by appending a new block, trying to enlarge the reachable region in the state space of our PQC to cover the unknown state. Once phase I accomplished, LL-strategy will stop extending and bring our PQC into phase II, fine-tuning each parameter group. To sum up, LL can find the suitable circuit which is neither too short nor over-expressive, saving quantum computing resources with the final fidelities guaranteed.

Suppose that there exists a minimal circuit that can



**Fig. 7** Instances of fidelity evolution curves in the training process using LL-strategy. Here we consider 3-qubit systems with readout noise, where the bit-flip probability is 20%. The fidelities are evaluated as  $|\langle \psi(\theta) | \psi_{unkn} \rangle|$ , where  $|\psi(\theta)\rangle$  is simulated using the PQC without noise. The sub-figures display the real and imaginary parts of elements in the input unknown state  $|\psi_{unkn}\rangle$  (denoted as  $|\psi_{unknown}\rangle$ ) and the output state  $|\psi(\theta)\rangle$  of the noiseless trained PQC (denoted as  $|\psi_{reconstructed}\rangle$ ).

achieve the target fidelity threshold with the same layered structure, whose depth is denoted as  $L_{\min}$ . The depth of the trained circuit from layerwise learning can naturally be regarded as a limited upper bound of  $L_{\min}$ , which is found adaptively, even though we lack preliminary knowledge of the input state. Therefore, the parameterized circuit of LL provides a valuable reference if we wish to further find the exact value of  $L_{\min}$ .

Similar to the classical analogue, neural network algorithms, the performance of hybrid quantum-classical algorithms is greatly dependent on the form of ansatz, the optimization algorithm, and even the initialization method of trainable parameters. Hence we can improve CDL by handpicking more adequate instances of these training facilities, in which case they can also be applied to the training process of LL. For example, the structure of each block in a multi-layer CDL parameterized circuit and the corresponding classical optimizer can be adopted into LL training as well. The benefits of these facilities are also effective against the LL-strategy since each LL training step can be regarded as a CDL-training of the sub-PQC. Consequently, the adoption of better training techniques in CDL is expected to bring an equal boost to LL.

## 4 Discussion

As illustrated above, the algorithm of this paper can find a suitable PQC to reconstruct an unknown state with high fidelity in an adaptive way. We design the framework based on parameterized quantum circuits, and introduce classical gradient optimizers into the quantum machine learning protocol to raise the final fidelities. The adopted layerwise learning strategy lengthens PQCs and optimizes newly added parameters gradually, avoiding barren plateaus of short-depth PQCs while keeping the entire structure minimal and optimal. As a result, our method provides high practicality for state reconstructing in the NISQ-era, reducing the consumed computational resources and time complexity compared with classical state tomography, and can reach efficiently high fidelities for systems of different scales. The output state of PQC can have more than 96% fidelities for 1–6-qubit systems in theory. Real device experiments indicate the practicality of our framework, and comparisons with other methods reveal its advantage. Our method is helpful for gaining enough information about the structures of complex non-analytic quantum systems, or mitigating and eliminating noises on real quantum devices such as superconducting systems and ion-trap systems. Furthermore, it can serve as a general framework for hybrid variational quantum-classical algorithms that are based on fidelities and provide a boost for them.

It is worth noticing that given an ansatz of PQC in our quantum state reconstructing framework, its representation ability is fixed. The block structure employed

in our experiment, as shown in Fig. 2, determines that our ansatz is sufficient for us to learn and reproduce quantum states of 1–4 qubits, while for larger systems the loss values may not be optimized to be less than the convergence threshold. There are many ways to increase the fidelity and the probability of convergence, one of which is adopting ansatz more suitable to the system to be considered, or heuristic ansatz like [29]. Another important idea is to apply better initialization strategies and optimization algorithms. We will conduct further studies in our future works.

**Acknowledgements** We gratefully acknowledge support from the National Natural Science Foundation of China under Grant Nos. 11974205, and 11774197; the National Key Research and Development Program of China (2017YFA0303700); the Key Research and Development Program of Guangdong province (2018B030325002); Beijing Advanced Innovation Center for Future Chip (ICFC). S. W. also acknowledges the National Natural Science Foundation of China under Grant No. 12005015.

We create a Python package `paulicirq` [45] for building quantum circuits and layers of quantum machine learning models which utilizes `Cirq` [46] and `TensorFlow Quantum` [47], and is used in the numerical analysis in this work. We take advantage of `Qiskit` [48] to conduct hardware experiments.

We acknowledge the use of IBM Quantum services for this work. The views expressed are those of the authors, and do not reflect the official policy or position of IBM or the IBM Quantum team.

## Appendices

### A SWAP test

Suppose we have two quantum states  $|\psi_1\rangle$  and  $|\psi_2\rangle$  of equal dimension, between which there is no entanglement. SWAP test can help us measure the overlap between them, i.e.,  $|\langle\psi_1|\psi_2\rangle|$  [49]. To do this job, an auxiliary qubit initialized in  $|0\rangle$  state needs to be introduced. Without loss of generality, we assume that the two quantum states are composed of  $n$  qubits, and they take the following form under the computational basis:

$$|\psi_1\rangle = \sum_{i=0}^{N-1} c_{1;i} |i\rangle_1, \quad (\text{A1})$$

$$|\psi_2\rangle = \sum_{i=0}^{N-1} c_{2;i} |i\rangle_2, \quad (\text{A2})$$

where  $N = 2^n$  is the dimension of the whole Hilbert space of one state. Firstly, an Hadamard gate  $H$  is applied to the auxiliary qubit  $|0\rangle_a$ , which is transformed into  $(|0\rangle_a + |1\rangle_a)/\sqrt{2}$ . Then we act the ancilla-controlled SWAP gate onto the two states, i.e., swap each pair of qubits from the two states if the auxiliary qubit is in  $|1\rangle_a$  state, which yields

$$|\Psi\rangle = \frac{1}{\sqrt{2}} \left[ |0\rangle_a \left( \sum_{i=0}^{N-1} c_{1;i} |i\rangle_1 \right) \left( \sum_{j=0}^{N-1} c_{2;j} |j\rangle_2 \right) + |1\rangle_a \left( \sum_{i=0}^{N-1} c_{2;i} |i\rangle_1 \right) \left( \sum_{j=0}^{N-1} c_{1;j} |j\rangle_2 \right) \right] \quad (\text{A3})$$

$$= \frac{1}{\sqrt{2}} [ |0\rangle_a |\psi_1\rangle |\psi_2\rangle + |1\rangle_a |\psi_2\rangle |\psi_1\rangle ]. \quad (\text{A4})$$

After the controlled-SWAP gate, another  $H$  gate is acted onto the auxiliary qubit, mixing the unswapped substate and the swapped substate:

$$|\Psi\rangle = \frac{1}{2} [ |0\rangle_a ( |\psi_1\rangle |\psi_2\rangle + |\psi_2\rangle |\psi_1\rangle ) + |1\rangle_a ( |\psi_1\rangle |\psi_2\rangle - |\psi_2\rangle |\psi_1\rangle ) ]. \quad (\text{A5})$$

In the last step, we measure the auxiliary qubit in  $\{|0\rangle_a, |1\rangle_a\}$  basis, and the probabilities of obtaining 0 and 1 are

$$P(i) = \frac{1}{4} \left\| |\psi_1\rangle |\psi_2\rangle + (-1)^i |\psi_2\rangle |\psi_1\rangle \right\|^2 \quad (\text{A6})$$

$$= \frac{1}{2} [ 1 + (-1)^i |\langle \psi_2 | \psi_1 \rangle|^2 ], \quad i = 0, 1. \quad (\text{A7})$$

From these relationships we have

$$|\langle \psi_2 | \psi_1 \rangle|^2 = P(0) - P(1) = \langle Z \rangle, \quad (\text{A8})$$

which is just the expectation value of the Pauli operator  $\hat{Z}$  of the auxiliary qubit. The quantum circuit of SWAP test is shown in Fig. 8.

### B Procedure for creating random circuits

We adopt the concept of “moments” in [46]: A moment is a time slice in a quantum circuit, which contains operations to be performed at the same time. In order to create random states, we build random symbolic circuits firstly, with the number of symbols in the random circuit  $d_s$  designated beforehand: For the first  $d_s$  moments, we sample operations which can fill in it from the set

$$\{ I, X, Y, Z, XX, YY, ZZ, H, CZ, \text{CNOT, SWAP, iSWAP, fsm} \} \quad (\text{B1})$$

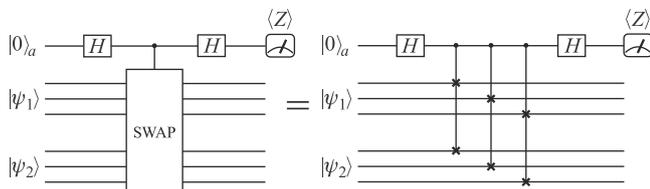


Fig. 8 Quantum circuit of SWAP test.

where fsm is the two-qubit Fermionic simulation gate [50]:

$$\text{fsm}(\theta, \phi) = e^{-i\theta(XX+YY)/2} e^{-i\phi(I-Z)\otimes(I-Z)/4} \quad (\text{B2})$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -i \sin \theta & 0 \\ 0 & -i \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & e^{-i\phi} \end{pmatrix}. \quad (\text{B3})$$

Each operation will be raised to a symbolic power which is sampled from the symbol list, and then be added into the moment with 90% probability. For remaining unused symbols,  $H$  gates parameterized with them will be appended to the circuit. Finally, all symbols will be replaced with random values, generating a fully random quantum circuit which can be used to produce our input state.

In numerical simulations in Section 3.1, parameter  $d_s$  is set to be 10.

### C Configurations of numerical simulations

We change the number of qubits  $n_q$  from 1 to 6, and generate random input state following the procedure in Appendix B with  $d_s = 10$ . Due to the random generating mechanism, the actual number of moments of input random circuits varies from 14 to 26.

We initialize the work qubits as the evenly superposed state  $\left[ \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \right]^{\otimes n_q}$  via Hadamard gates  $U_0 = H^{\otimes n_q}$ . The  $W$  parts in our PQC blocks are composed of a series of parameterized two-qubit gates  $e^{-iX_i \otimes X_j J_{ij}}$  with  $J_{ij}$  being the two-body interaction parameter, having all-to-all connectivity in our experiment, as shown in Fig. 2. Parameters in each block are initially set to be random values close to zero.

In each numerical experiment, our PQC starts with  $L_s = 2$  blocks. We set the upper limit of the depths of PQC blocks to be  $L_m = 5$ . In phase II, we set the partition rate  $r = 0.3$  and the maximal number of sweeps  $s = 10$ . In each training step of both phases, we optimize our PQC with  $n_e = 15$  epochs. For each epoch, we take 10 copies of the input random circuit to make up a dataset, which means that there are 10 mini trainings in an epoch. If the value of loss function is not reduced by a difference greater than  $\delta \mathcal{L} = 2 \times 10^{-5}$  for  $n_{\text{pat}} = 5$  epochs, the training step will be stopped early. And if the value of loss function is reduced to be less than  $\mathcal{L}_{\text{th}} = 10^{-4}$  after a certain training step, the entire training phase will early-stop. Because of this early-stopping mechanism, the final depth of our PQC may be less than  $L_m$  and is different from case to case.

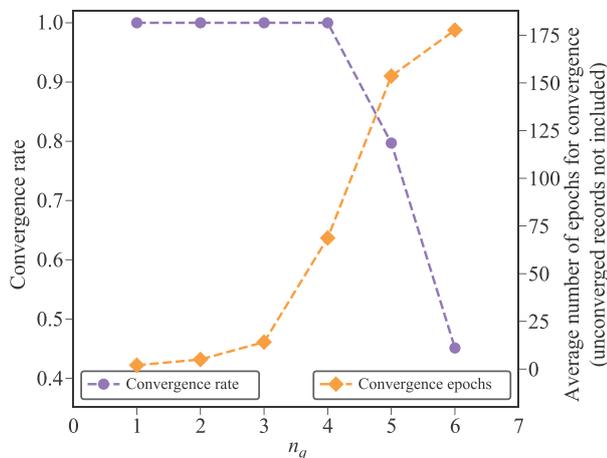
The gradients of loss function are evaluated via adjoint method [47, 51, 52]. In phase I, we use Nadam algorithm [53] with initial learning rate being 0.09 to optimize the parameters, and Adam optimizer [54] with initial learning rate being 0.09 is used in phase II.

## D Convergence rate and training time

We note that as the scale of quantum system increases, the probability for our PQC to converge to the threshold may decrease, and the training process may become longer, which are the natural results of the limited representation ability of our PQC structure. On the one hand, not all cases can be optimized to our goal (i.e., loss value being reduced to less than  $10^{-4}$ ) in the experiment: All 1000 cases of  $n_q = 1, 2, 3, 4$  converged, while the rate of converged cases is 80% for  $n_q = 5$  and decreases to 45% for larger systems with  $n_q = 6$ , as illustrated in Fig. 9. Here the convergence rate is evaluated as  $n_{\text{conv}}/1000$ , with  $n_{\text{conv}}$  representing the number of converged cases. On the other hand, averagely the training process takes more epochs when  $n_q$  is enlarged, which displays a sub-exponential relationship between  $N_e$  (the average value of total number of epochs for convergence) and  $n_q$ , according to Fig. 9, where only converged cases are taken into account.

## E Configurations of IBM quantum experiments

We initialize the work qubit as  $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ . For the purpose of demonstration, our PQC starts from  $L_s = 1$  block, and each block takes the form of Eq. (3). The upper limit of the depth of PQC blocks is set to be  $L_m = 3$ . Parameters in each block are initially set to be random values close to zero. In phase II, the partition rate  $r = 0.3$ , and at most  $s = 3$  sweeps through all parameters are performed.



**Fig. 9** Relations between convergence rate/time and  $n_q$ . For larger systems it takes more time for the PQC to converge and the rate of converged cases decreases, which is because of the limited representation ability of the PQC structure used in our experiment.

In each training step, the optimizer runs for 10 iterations at most, which means that parameters in the newly appended block/parameters collected in a group will be updated 10 times at most in phase I /in each sweep of phase II. If the loss value evaluated from hardware measurement is not optimized by a reduction greater than  $\delta\mathcal{L} = 0.005$  for  $n_{\text{pat}} = 5$  iterations, the training step will be early-stopped and turn to next step. At the end of each training step, if the loss value is reduced to be lower than  $\mathcal{L}_{\text{th}} = 0.05$ , the training phase will be early-stopped.

The gradients of loss function are evaluated via parameter-shift rule [34, 40, 41]. For each evaluation of loss function and gradient components, the quantum processor will be called 8192 times to obtain high-precision values. Adam optimizer with initial learning rate being 0.09 is used in both phases.

## References and notes

1. S. Lloyd, Universal quantum simulators, *Science* 273(5278), 1073 (1996)
2. X. Qiang, T. Loke, A. Montanaro, K. Aungkunsiri, X. Zhou, J. L. O'Brien, J. B. Wang, and J. C. F. Matthews, Efficient quantum walk on a quantum processor, *Nat. Commun.* 7(1), 11511 (2016)
3. N. N. Zhang, M. J. Tao, W. T. He, X. Y. Chen, X. Y. Kong, F. G. Deng, N. Lambert, and Q. Ai, Efficient quantum simulation of open quantum dynamics at various Hamiltonians and spectral densities, *Front. Phys.* 16(5), 51501 (2021)
4. P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM J. Comput.* 26(5), 1484 (1997)
5. L. K. Grover, Quantum computers can search arbitrarily large databases by a single query, *Phys. Rev. Lett.* 79(23), 4709 (1997)
6. G. L. Long, General quantum interference principle and duality computer, *Commun. Theor. Phys.* 45(5), 825 (2006)
7. G. L. Long, Duality quantum computing and duality quantum information processing, *Int. J. Theor. Phys.* 50(4), 1305 (2011)
8. A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, *Phys. Rev. Lett.* 103(15), 150502 (2009)
9. C. H. Bennett and G. Brassard, in: Proceedings of IEEE International Conference on Computers, Systems and Signal Processing, 1984, pp 175-179
10. A. K. Ekert, Quantum cryptography based on Bell's theorem, *Phys. Rev. Lett.* 67(6), 661 (1991)
11. C. H. Bennett, Quantum cryptography using any two nonorthogonal states, *Phys. Rev. Lett.* 68(21), 3121 (1992)
12. G. L. Long and X. S. Liu, Theoretically efficient high-capacity quantum-key-distribution scheme, *Phys. Rev. A* 65(3), 032302 (2002)

13. Z. D. Ye, D. Pan, Z. Sun, C. G. Du, L. G. Yin, and G. L. Long, Generic security analysis framework for quantum secure direct communication, *Front. Phys.* 16(2), 21503 (2021)
14. Z. X. Cui, W. Zhong, L. Zhou, and Y. B. Sheng, Measurement-device-independent quantum key distribution with hyper-encoding, *Sci. China Phys. Mech. Astron.* 62(11), 110311 (2019)
15. Z. Qi, Y. Li, Y. Huang, J. Feng, Y. Zheng, and X. Chen, A 15-user quantum secure direct communication network, *Light Sci. Appl.* 10(1), 183 (2021)
16. G. L. Long and H. Zhang, Drastic increase of channel capacity in quantum secure direct communication using masking, *Sci. Bull. (Beijing)* 66(13), 1267 (2021)
17. X. Liu, Z. Li, D. Luo, C. Huang, D. Ma, M. Geng, J. Wang, Z. Zhang, and K. Wei, Practical decoy-state quantum secure direct communication, *Sci. China Phys. Mech. Astron.* 64(12), 120311 (2021)
18. Y. B. Sheng, L. Zhou, and G. L. Long, One-step quantum secure direct communication, *Sci. Bull. (Beijing)* 67(4), 367 (2022)
19. G. M. D'Ariano, M. D. Laurentis, M. G. A. Paris, A. Porzio, and S. Solimeno, Quantum tomography as a tool for the characterization of optical devices, *J. Opt. B* 4(3), S127 (2002)
20. F. Albarrán-Arriagada, J. C. Retamal, E. Solano, and L. Lamata, Measurement-based adaptation protocol with quantum reinforcement learning, *Phys. Rev. A* 98(4), 042315 (2018)
21. X. Xu, J. Sun, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, Variational algorithms for linear algebra, arXiv: 1909.03898 [quant-ph] (2019)
22. C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, Variational quantum linear solver, arXiv: 1909.05820 [quant-ph] (2019)
23. X. Wang, Z. Song, and Y. Wang, Variational quantum singular value decomposition, arXiv: 2006.02336 [quant-ph] (2020)
24. M. Cerezo, K. Sharma, A. Arrasmith, and P. J. Coles, Variational quantum state eigensolver, arXiv: 2004.01372 [quant-ph] (2020)
25. S. McArdle, S. Endo, A. Aspuru-Guzik, S. Benjamin, and X. Yuan, Quantum computational chemistry, arXiv: 1808.10402 [quant-ph] (2018)
26. A. Peruzzo, J. McClean, P. Shadbolt, M. H. Yung, X. Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, *Nat. Commun.* 5(1), 4213 (2014)
27. J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, *New J. Phys.* 18(2), 023023 (2016)
28. 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* 549(7671), 242 (2017)
29. I. G. Ryabinkin, T. C. Yen, S. N. Genin, and A. F. Izmaylov, Qubit coupled cluster method: A systematic approach to quantum chemistry on a quantum computer, *J. Chem. Theory Comput.* 14(12), 6317 (2018)
30. I. G. Ryabinkin, R. A. Lang, S. N. Genin, and A. F. Izmaylov, Iterative qubit coupled cluster approach with efficient screening of generators, arXiv: 1906.11192 [quant-ph] (2019)
31. D. B. Zhang, Z. H. Yuan, and T. Yin, Variational quantum eigensolvers by variance minimization, arXiv: 2006.15781 [quant-ph] (2020)
32. E. Farhi and H. Neven, Classification with quantum neural networks on near term processors, arXiv: 1802.06002 [quant-ph] (2018)
33. A. Skolik, J. R. McClean, M. Mohseni, P. van der Smagt, and M. Leib, Layerwise learning for quantum neural networks, arXiv: 2006.14904 [quant-ph] (2020)
34. K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, *Phys. Rev. A* 98(3), 032309 (2018)
35. J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* 2, 79 (2018)
36. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, *Nature* 549(7671), 195 (2017)
37. M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, *Quantum Sci. Technol.* 4(4), 043001 (2019)
38. S. Wei, Y. Chen, Z. Zhou, and G. Long, A quantum convolutional neural network on NISQ devices, arXiv: 2104.06918 [quant-ph] (2021)
39. F. Hu, B. N. Wang, N. Wang, and C. Wang, Quantum machine learning with  $D$ -wave quantum computer, *Quantum Engineering* 1(2), e12 (2019)
40. J. Li, X. Yang, X. Peng, and C. P. Sun, Hybrid quantum-classical approach to quantum optimal control, *Phys. Rev. Lett.* 118(15), 150503 (2017)
41. M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, *Phys. Rev. A* 99(3), 032331 (2019)
42. IBM quantum, 2021
43. A. Mari, T. R. Bromley, and N. Killoran, Estimating the gradient and higher-order derivatives on quantum hardware, *Phys. Rev. A* 103(1), 012405 (2021)
44. T. Xin, X. Nie, X. Kong, J. Wen, D. Lu, and J. Li, Quantum pure state tomography via variational hybrid quantum-classical method, *Phys. Rev. Appl.* 13(2), 024013 (2020)
45. J. Xiao, Paulicirq, 2020
46. Quantum AI Team and Collaborators, Cirq, 2020
47. M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, M. Y. Niu, R. Halavati, E. Peters, M. Leib, A. Skolik, M. Streif, D. V. Dollen, J. R. McClean, S. Boixo, D. Bacon, A. K. Ho, H. Neven, and M. Mohseni, Tensorflow quantum: A software framework for quantum machine learning, arXiv: 2003.02989 [quant-ph] (2020)

48. Qiskit: An open-source framework for quantum computing, 2019
49. J. C. Garcia-Escartin and P. Chamorro-Posada, SWAP test and Hong–Ou–Mandel effect are equivalent, *Phys. Rev. A* 87(5), 052330 (2013)
50. F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, et al., Quantum supremacy using a programmable superconducting processor, *Nature* 574(7779), 505 (2019)
51. X. Z. Luo, J. G. Liu, P. Zhang, and L. Wang, Yao.jl: Extensible, efficient framework for quantum algorithm design, *Quantum* 4, 341 (2020)
52. R. E. Plessix, A review of the adjoint-state method for computing the gradient of a functional with geophysical applications, *Geophys. J. Int.* 167(2), 495 (2006)
53. T. Dozat, in: *ICLR* (2016)
54. D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv: 1412.6980 [cs.LG] (2014)