

Quafu-RL: The cloud quantum computers based quantum reinforcement learning

Yu-Xin Jin(靳羽欣), Hong-Ze Xu(许宏泽), Zheng-An Wang(王正安), Wei-Feng Zhuang(庄伟峰), Kai-Xuan Huang(黄凯旋), Yun-Hao Shi(时运豪), Wei-Guo Ma(马卫国), Tian-Ming Li(李天铭), Chi-Tong Chen(陈驰通), Kai Xu(许凯), Yu-Long Feng(冯玉龙), Pei Liu(刘培), Mo Chen(陈墨), Shang-Shu Li(李尚书), Zhi-Peng Yang(杨智鹏), Chen Qian(钱辰), Yun-Heng Ma(马运恒), Xiao Xiao(肖骁), Peng Qian(钱鹏), Yanwu Gu(顾炎武), Xu-Dan Chai(柴绪丹), Ya-Nan Pu(普亚南), Yi-Peng Zhang(张翼鹏), Shi-Jie Wei(魏世杰), Jin-Feng Zeng(曾进峰), Hang Li(李行), Gui-Lu Long(龙桂鲁), Yirong Jin(金贻荣), Haifeng Yu(于海峰), Heng Fan(范桁), Dong E. Liu(刘东), and Meng-Jun Hu(胡孟军)

Citation: Chin. Phys. B, 2024, 33 (**5**): 050301. DOI: 10.1088/1674-1056/ad3061
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Quantum circuit-based proxy blind signatures: A novel approach and experimental evaluation on the IBM quantum cloud platform

Xiaoping Lou(娄小平), Huiru Zan(昝慧茹), and Xuejiao Xu(徐雪娇)
Chin. Phys. B, 2024, 33 (**5**): 050307. DOI: 10.1088/1674-1056/ad22d5

Quantum generative adversarial networks based on a readout error mitigation method with fault tolerant mechanism

Run-Sheng Zhao(赵润盛), Hong-Yang Ma(马鸿洋), Tao Cheng(程涛), Shuang Wang(王爽), and Xing-Kui Fan(范兴奎)
Chin. Phys. B, 2024, 33 (**4**): 040304. DOI: 10.1088/1674-1056/ad02e7

Analysis of learnability of a novel hybrid quantum—classical convolutional neural network in image classification

Tao Cheng(程涛), Run-Sheng Zhao(赵润盛), Shuang Wang(王爽), Rui Wang(王睿), and Hong-Yang Ma(马鸿洋)
Chin. Phys. B, 2024, 33 (**4**): 040303. DOI: 10.1088/1674-1056/ad1926

Remote entangling gate between a quantum dot spin and a transmon qubit mediated by microwave photons

Xing-Yu Zhu(朱行宇), Le-Tian Zhu(朱乐天), Tao Tu(涂涛), and Chuan-Feng Li(李传锋)
Chin. Phys. B, 2024, 33 (**2**): 020315. DOI: 10.1088/1674-1056/ad1747

Variational quantum semi-supervised classifier based on label propagation

Yan-Yan Hou(侯艳艳), Jian Li(李剑), Xiu-Bo Chen(陈秀波), and Chong-Qiang Ye(叶崇强)
Chin. Phys. B, 2023, 32 (**7**): 070309. DOI: 10.1088/1674-1056/acb9fb

---

SPECIAL TOPIC — Quantum computing and quantum sensing

Quafu-RL: The cloud quantum computers based quantum reinforcement learning

Yu-Xin Jin(靳羽欣)[1,7,†], Hong-Ze Xu(许宏泽)[1,†], Zheng-An Wang(王正安)[1], Wei-Feng Zhuang(庄伟峰)[1], Kai-Xuan Huang(黄凯旋)[1], Yun-Hao Shi(时运豪)[3,5,6], Wei-Guo Ma(马卫国)[3,5,6], Tian-Ming Li(李天铭)[3,5,6], Chi-Tong Chen(陈驰通)[3,5,6], Kai Xu(许凯)[3,1], Yu-Long Feng(冯玉龙)[1], Pei Liu(刘培)[1], Mo Chen(陈墨)[1], Shang-Shu Li(李尚书)[3,5,6], Zhi-Peng Yang(杨智鹏)[1], Chen Qian(钱辰)[1], Yun-Heng Ma(马运恒)[1], Xiao Xiao(肖骁)[1], Peng Qian(钱鹏)[1], Yanwu Gu(顾炎武)[1], Xu-Dan Chai(柴绪丹)[1], Ya-Nan Pu(普亚南)[1], Yi-Peng Zhang(张翼鹏)[1], Shi-Jie Wei(魏世杰)[1], Jin-Feng Zeng(曾进峰)[1], Hang Li(李行)[1], Gui-Lu Long(龙桂鲁)[2,1], Yirong Jin(金贻荣)[1], Haifeng Yu(于海峰)[1], Heng Fan(范桁)[3,1,5,6], Dong E. Liu(刘东)[2,1,4], and Meng-Jun Hu(胡孟军)[1,‡]

[1] *Beijing Academy of Quantum Information Sciences, Beijing* 100193, *China*
[2] *State Key Laboratory of Low Dimensional Quantum Physics, Department of Physics, Tsinghua University, Beijing* 100084, *China*
[3] *Institute of Physics, Chinese Academy of Sciences, Beijing* 100190, *China*
[4] *Frontier Science Center for Quantum Information, Beijing* 100184, *China*
[5] *School of Physical Sciences, University of Chinese Academy of Sciences, Beijing* 100190, *China*
[6] *CAS Center for Excellence in Topological Quantum Computation, University of Chinese Academy of Sciences, Beijing* 100190, *China*
[7] *School of Mathematical Sciences, Nankai University, Tianjin* 300071, *China*

(Received 23 December 2023; revised manuscript received 3 March 2024; accepted manuscript online 6 March 2024)

With the rapid advancement of quantum computing, hybrid quantum–classical machine learning has shown numerous potential applications at the current stage, with expectations of being achievable in the noisy intermediate-scale quantum (NISQ) era. Quantum reinforcement learning, as an indispensable study, has recently demonstrated its ability to solve standard benchmark environments with formally provable theoretical advantages over classical counterparts. However, despite the progress of quantum processors and the emergence of quantum computing clouds, implementing quantum reinforcement learning algorithms utilizing parameterized quantum circuits (PQCs) on NISQ devices remains infrequent. In this work, we take the first step towards executing benchmark quantum reinforcement problems on real devices equipped with at most 136 qubits on the BAQIS Quafu quantum computing cloud. The experimental results demonstrate that the policy agents can successfully accomplish objectives under modified conditions in both the training and inference phases. Moreover, we design hardware-efficient PQC architectures in the quantum model using a multi-objective evolutionary algorithm and develop a learning algorithm that is adaptable to quantum devices. We hope that the Quafu-RL can be a guiding example to show how to realize machine learning tasks by taking advantage of quantum computers on the quantum cloud platform.

Keywords: quantum cloud platform, quantum reinforcement learning, evolutionary quantum architecture search

PACS: 03.67.Lx, 03.67.Ac **DOI:** 10.1088/1674-1056/ad3061

1. Introduction

Recent years have witnessed a huge development of quantum computing, encompassing quantum hardware and abundant software platforms. Despite the proposition of quantum supremacy,[1–3] limitations persist in the current noisy intermediate-scale quantum (NISQ) regime, primarily stemming from a constrained number of qubits, quantum state decoherence, and imprecise quantum operations.[4,5] With the obstacle of constructing ideal quantum processing units (QPUs) in the short term, a current trend indicates that it is necessary to build hybrid quantum–classical infrastructures to explore useful applications while making efforts to scale up QPUs.[6] Quantum computing clouds such as IBMQ, Amazon Bracket, and Quafu[7] serve as bridges to connect developers in quantum software applications with advanced quantum computers in the NISQ era. Although quantum computing is promising to solve some intractable classical NP-hard problems, deficient quantum devices nowadays favorably excel at hybrid algorithms leveraging both quantum circuits and classical computing.

Variational quantum algorithms (VQAs) that run parameterized quantum circuits (PQCs) on quantum devices while integrating classical optimizers for parameter optimization satisfy the desire to demonstrate applications on NISQ devices[8] and have been wildly explored. Researchers in VQAs mainly focus on quantum chemistry,[9–11]

[†]These authors contributed equally to this work.
[‡]Corresponding author. E-mail: humj@baqis.ac.cn

© 2024 Chinese Physical Society and IOP Publishing Ltd

quantum optimization[12–14] and quantum machine learning (QML)[15,16] including interesting problems such as classification,[17–20] generative adversarial networks[21–24] and related theoretic analysis.[25–27]

Reinforcement learning (RL),[28] as an essential area in modern machine learning research, receives comparatively later attention in VQA-based approaches.[29–32] Nonetheless, these proposed PQC agents cannot reach satisfactory performance in benchmark environments from OpenAI Gym.[33] Until recently, policy-based[34] and value-based[35] quantum reinforcement learnings have made breakthroughs in both solving standard benchmark tasks and theoretical learning advantages over classical algorithms. In Refs. [34,35], apart from hyperparameters tuning, the importance of PQC architectures in successful RL training and better performance is proposed. This issue inspires some evolutionary methods[36–41] to search for suitable PQC architectures without human ingenuity at the aim of balancing expressivity[42,43] and trainability.

In this study, we leverage the theoretical guarantees and experimental validation of parameterized quantum policies for reinforcement learning, along with the availability of quantum computing clouds like Quafu, which offer scalable and stable services over the long term. With these resources, we have the ability to apply a benchmark RL task, CartPole, to real quantum devices and conduct numerous experiments. Firstly, we utilize a multi-objective evolutionary approach to search for the most suitable PQCs that can construct policies for higher performance as well as lower entanglement and design a learning algorithm for quantum devices. Next, we select the best policy and execute it on available quantum resources that include 10-qubit, 18-qubit, and 136-qubit quantum computers. Our experiments demonstrate that the environment can be successfully solved with some relaxation of the original goals. Overall, our proposed methods and pipelines offer a perspective for conducting PQC-based RL policies and other QML experiments on NISQ devices.

2. Reinforcement learning

Reinforcement learning illustrates a problem in which an agent learns to make the best decisions to get the maximum cumulative rewards through interactions with the environment.[28,44] Explicitly speaking, as shown in Fig. 1, an agent first observes a state $s \in \mathcal{S}$ from the environment. Then, according to the policy π, which can be some deterministic tables or neural networks with trainable parameters, the agent takes an action a from all possible actions $\mathcal{A}$. Additionally, a policy can be further described as a mapping from states to the probability of performing each possible action $\pi(a \mid s)$. After executing the action a, the environment returns a new state s' and a reward r. The aforementioned transitions usually formulated as $p(s' \mid s, a)$ describe the dynamics of the environment. In order to formalize the RL objective, discount factor γ is introduced which lies in $[0, 1]$. The corresponding discounted return is defined as the weighted sum of future rewards $G_t = \sum_{k=0}^{T-t-1} \gamma^k r_{t+k+1}$, which indicates the rewards obtained starting at time t until the final time T. In essence, gathering above all elements, reinforcement learning is mathematically modeled with Markov decision process (MDP) as a tuple $(\mathcal{S}, \mathcal{A}, p, G, \gamma)$.

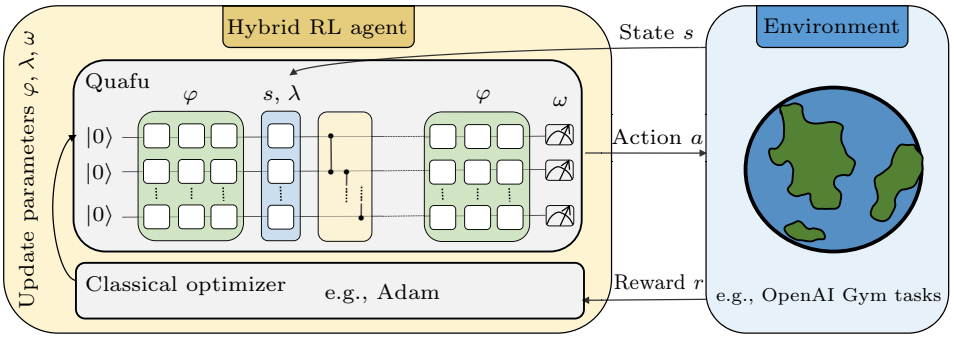


Fig. 1. Quantum reinforcement learning with Quafu. The entire pipeline can be described as follows: the RL agent, which is composed of a hardware-efficient PQC designed specifically for this task, receives a state s from the environment. The agent is then evaluated on Quafu and generates a policy $\pi_\theta(a \mid s)$ to sample an action a and receive a feedback reward r. The parameters within the PQC are updated using a classical optimizer and training algorithm introduced later.

2.1. Policy gradient methods

Reinforcement learning algorithms can be classified into two main categories: value-based and policy-based methods. Value-based methods aim to find the policy that maximizes the value function, while policy-based methods directly attempt to find the optimal policy by proposing a parameterized $\pi_{\boldsymbol{\theta}}(a \mid s)$ and optimizing its parameters $\boldsymbol{\theta}$. In this paper, we mainly focus on the policy gradient methods.

In most cases, policy gradient methods compute the gradient of the expected total return $\nabla_{\boldsymbol{\theta}} \mathbb{E}[G \mid \pi_{\boldsymbol{\theta}}]$ and apply gradient ascent to update the parameters. Specifically, the expectation value is taken over the trajectories τ which collect all states, actions and rewards $s_0, a_0, r_1, s_1, a_1, r_2, s_2, a_2, \ldots, s_T$ within one episode. Then, consistent with the policy gradient

theorem in Ref. [45], the expectation can be rewritten as

$$\nabla_{\boldsymbol{\theta}}\mathbb{E}\left[G \mid \pi_{\boldsymbol{\theta}}\right]=\mathbb{E}\left[G \sum_{t=0}^{T-1} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}\left(a_t \mid s_t\right) \mid \pi_{\boldsymbol{\theta}}\right]. \quad (1)$$

3. PQC-based quantum reinforcement learning

In the quantum realm, parameterized quantum circuits which can be represented by a unitary $U(s,\boldsymbol{\theta})$ with an input state s and trainable parameters $\boldsymbol{\theta}$ are utilized to build reinforcement learning policies, analogous to the function of classical neural networks. Recent studies[16,42,43,46] have revealed some hardware-efficient PQCs, typically composed of variational PQCs, data-encoding PQCs, entanglements, and final measurement. In the context, as displayed in Fig. 2, a variational PQC ($U_{\rm var}(\boldsymbol{\varphi})$) refers to a circuit made up of single-qubit rotations R_x, R_y, R_z acting on each qubit, assigning rotation angles $\boldsymbol{\varphi}$ as trainable parameters. Moreover, a data-encoding PQC ($U_{\rm enc}(s,\boldsymbol{\lambda})$) is formed by R_x applied to each qubit, with a state vector $s=(s_0,s_1,\ldots)$ scaled by trainable parameters $\boldsymbol{\lambda}$. Also, an entanglement ($U_{\rm ent}$) achieves circular entanglement by using multiple controlled-Z gates. At the end, measurement is performed just after a variational PQC.

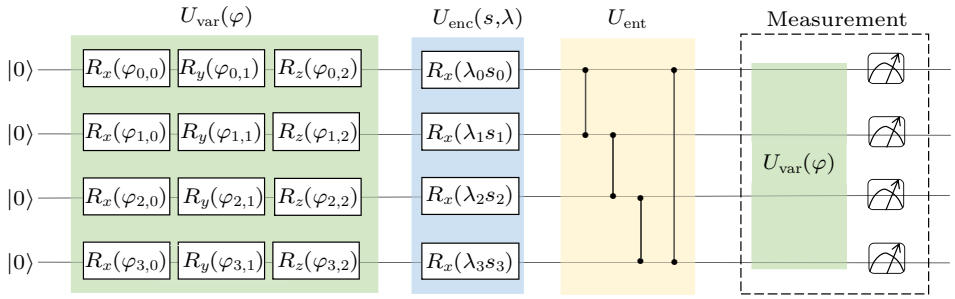


Fig. 2. PQC architecture components. The example 4-qubit circuit consists of four basic PQC architecture components including variational PQC denoted as $U_{\rm var}(\boldsymbol{\varphi})$ with trainable parameters $\boldsymbol{\varphi}$, data-encoding PQC termed as $U_{\rm enc}(s,\boldsymbol{\lambda})$ with input state s and scaling parameters $\boldsymbol{\lambda}$, entanglement identified as $U_{\rm ent}$ and the final measurement part.

3.1. Parameterized quantum policies

The four basic operations described above are the building blocks of the most common PQC architectures. However, to solve RL environments and obtain good performances, some data-encoding techniques and readout strategies are crucial for success. To enhance the expressive power, data-reuploading[43] which means using data-encoding PQC repeatedly in a circuit is a widely-adopted skill. In Refs. [34,35], a variational PQC, an entanglement and a data-encoding PQC are assembled as an alternating layer and are recurrently applied to create highly expressive parameterized quantum policies. Meanwhile, to handle the range of the output of measurements, weighted observables associated with action a are utilized with trainable parameters $\boldsymbol{\omega}$. The expectation value of the weighted observables then can be formulated as

$$\langle O_a\rangle_{s,\boldsymbol{\theta}}=\left\langle 0^{\otimes n}\left|U(s,\boldsymbol{\varphi},\boldsymbol{\lambda})^{\dagger} O_a U(s,\boldsymbol{\varphi},\boldsymbol{\lambda})\right| 0^{\otimes n}\right\rangle \cdot \boldsymbol{\omega}_a, \quad (2)$$

considering an n-qubit PQC with input state s, rotation angles $\boldsymbol{\varphi}$, scaling parameters $\boldsymbol{\lambda}$ and observable operators O_a, where $\boldsymbol{\theta}=(\boldsymbol{\varphi},\boldsymbol{\lambda},\boldsymbol{\omega})$. Furthermore, the non-linear activation function softmax is applied on the expectation values $\langle O_a\rangle_{s,\boldsymbol{\theta}}$, which defines a SOFTMAX-PQC policy as

$$\pi_{\boldsymbol{\theta}}(a \mid s)=\frac{e^{\beta\langle O_a\rangle_{s,\boldsymbol{\theta}}}}{\sum_{a'} e^{\beta\langle O_{a'}\rangle_{s,\boldsymbol{\theta}}}}, \quad (3)$$

where β is an inverse-temperature parameter.

3.2. Quantum architecture search

It is effective to adopt pre-selected alternating layers to construct the quantum policy architecture as shown in Ref. [34], but a recent work[36] proposes an evolutionary quantum architecture search method to build a more flexible PQC policy using the four fundamental architecture components in Fig. 2, achieving higher RL performance with lower depths.

To be more precise, the four elementary PQC structures, namely, $U_{\rm var}(\boldsymbol{\varphi})$, $U_{\rm enc}(s,\boldsymbol{\lambda})$, $U_{\rm ent}$, and measurement, are represented as genes and encoded into integer spaces with the values 1, 2, 3, and 0, respectively. Afterwards, non-dominated sorting genetic algorithm II (NSGA-II)[47] is implemented to optimize the quantum architecture search process which iteratively generates a population of candidates through genetic operations like crossover and mutation on the given parents, and selects parents for the next generation based on fitness evaluation with the RL average collected rewards as the objective.

4. Quantum reinforcement learning with Quafu

With the assistance of the SOFTMAX-PQC policy and evolutionary quantum architecture search method outlined in the preceding section, we can devise a PQC that is both efficient in hardware usage and effective in implementing quantum reinforcement learning tasks on the Quafu cloud platform.

4.1. Architecture selection

To select a hardware-efficient PQC for executing RL agents on Quafu, we utilize the method of evolutionary quantum architecture search (EQAS-PQC)[36] but take a further

step in refining our objective. We design a multi-objective function that not only takes into account RL training performance but also considers the number of entanglement components U_{ent}. The optimal architecture discovered during the search process for the CartPole environment can be represented using genes as follows: 3 – 1 – 1 – 2 – 1 – 1 – 3 – 1 – 3 – 2 – 1 – 2 – 0. To evaluate the ability and efficiency of our selected architecture, we compare it with the best architecture proposed in Ref. [36] and the alternating scheme presented in Ref. [34].

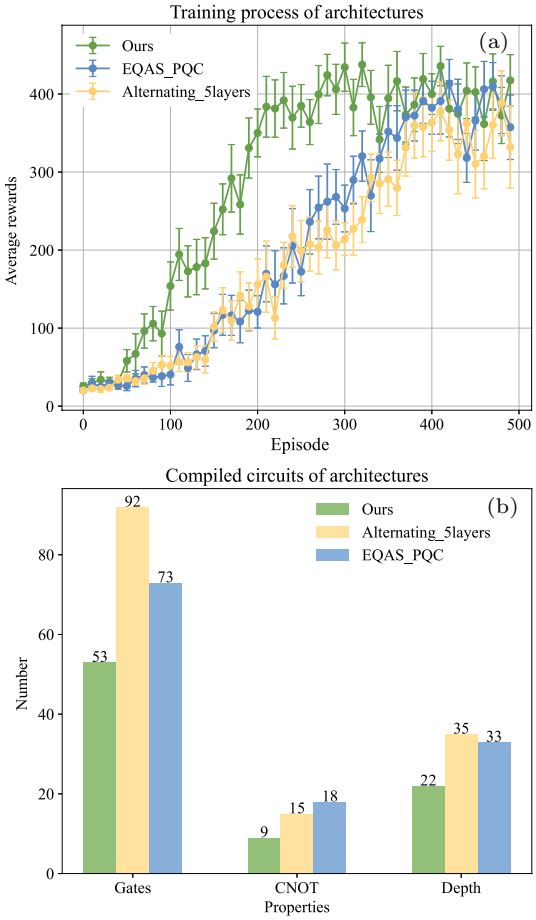


Fig. 3. Architecture comparison. Our proposed architecture, the 5-alternating-layer method, and EQAS-PQC are compared based on their training performance and compiled circuit properties. Panel (a) displays the average performance based on 10 trails, with error bars representing 0.3 times the standard deviation. Panel (b) shows the number of gates, CNOTs, and circuit depth of the compiled circuits.

As illustrated in Fig. 3(a), our PQC architecture achieves the best performance and is the first to reach the goal, whereas both the 5-alternating-layer method and EQAS-PQC slowly improve to solve the environment within 500 training episodes. Moreover, the Quafu cloud platform provides a compilation scheme for quantum circuits before their execution on the quantum computer. Therefore, certain hardware-sensitive characteristics (such as the number of total gates, the number of two-qubit gate CNOT, and the depth of the circuit) of the compiled circuits for the architectures are crucial for achieving optimal performance on Quafu. As shown in Fig. 3(b), our proposed architecture exhibits the least number of gates and CNOTs, as well as the shortest depth of the compiled circuit. In contrast, the other two methods have significantly more gates, CNOTs, and deeper circuit depths, which clearly indicates their lower hardware efficiency. Some implementation details to note in conducting the experiments shown in Fig. 3 are that we have removed the non-nearest neighbor CNOTs from the entanglement part. This is because including these CNOTs would drastically increase the number of gates and circuit depth, without adding significant benefits to the overall success of the task.

Briefly, our proposed multi-objective evolutionary quantum architecture search method enables us to identify a suitable PQC architecture that exhibits proficient RL performance and is also hardware-adaptive to the Quafu platform, as described in detail above.

4.2. Learning algorithm

After confirming the architecture used in SOFTMAX-PQC policy, it is important to specify an algorithm that connects classical optimizers and the Quafu platform. In Refs. [34,36], the classical REINFORCE algorithm[45] is implemented with quantum simulator. However, with the involvement of quantum devices, there will be slight modifications as shown in Algorithm 1.

Algorithm 1 REINFORCE with Quafu

Input: initialized SOFTMAX-PQC policy $\pi_{\boldsymbol{\theta}}(a \mid s)$, learning rate η, number of trajectories N, maximum time T

while True **do**
  for $i = 1$ to N **do**
    Initialize s_0
    for $t = 0$ to $T-1$ **do**
      Execute PQC_θ on Quafu and get policy $\pi_\theta(a_t \mid s_t)$
      Take action $a_t \sim \pi_\theta(a_t \mid s_t)$
      Move to next state s_{t+1} and store reward r_{t+1}
    end for
    Execute PQC_θ on simulator and get policy $\widehat{\pi}_\theta(a_t \mid s_t)$
    $G^{(i)} \leftarrow \sum_t \gamma^t r_{t+1}$
    Compute $\varepsilon = \pi_\theta(a_t \mid s_t) - \widehat{\pi}_\theta(a_t \mid s_t)$
    $z^{(i)} \leftarrow \sum_t \nabla_{\boldsymbol{\theta}} \log(\widehat{\pi}_\theta(a_t \mid s_t) + \varepsilon)$
  end for
  $\Delta\boldsymbol{\theta} \leftarrow (1/N)\sum_i G^{(i)} z^{(i)}$
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \eta\Delta\boldsymbol{\theta}$
end while

Table 1. Hyperparameters of the RL environment. Learning rates correspond to the parameters φ, ω, λ.

Environment	Qubits	Actions	Reward	Learning rates	γ	β	Observables
CartPole-v1	4	2	+1	0.01, 0.1, 0.1	1.0	1.0	$[Z_0Z_1Z_2Z_3]$

Combining the selected hardware-efficient PQC and specialized learning algorithm, we have built the whole pipeline of quantum reinforcement learning with Quafu, as pictured in Fig. 1.

5. Numerical results

In this section, we investigate a classical benchmark environment called CartPole from the OpenAI Gym[33] and apply our proposed PQC policy and algorithm to train the agent on three different quantum devices from Quafu:[7] ScQ-P10, ScQ-P18, and ScQ-P136, which have 10-qubit, 18-qubit, and 136-qubit capabilities, respectively.

5.1. Experimental setups

We determine the hyperparameters, such as learning rates and observables, based on the standard practices outlined in Ref. [34], which are also summarized in Table 1. Furthermore, instead of pursuing the initial goal of typically achieving 500 rewards within 500 episodes in quantum RL, we set a target of obtaining 100 rewards within 100 episodes, considering the high computational cost of training on real quantum computers. Besides, the classical optimization process is simulated on TensorFlow Quantum.[49]

5.2. Results

As presented in Fig. 4, we conduct a series of experiments on the Quafu cloud platform. In Fig. 4(a), an RL agent has been trained on the ScQ-P10 device and gets a maximum reward of 132 within 100 training episodes. Even in noiseless settings as demonstrated in Fig. 3, it is not easy to achieve such objectives with only 100 training episodes. Moreover, observing the rewards averaged by a time window of 5 episodes, the agent is able to reach a stable reward of over 50, which lasts for a certain period after 80 training episodes. Additionally, the mean reward over 100 training episodes is approximately 30, which is higher than 20, the reward achieved by random choice. Figures 4(b) and 4(c) depict the training process of RL agents on ScQ-P18 and ScQ-P136, which accomplish maximum rewards of 86 and 94, respectively. Due to the requirement of only a 4-qubit circuit for our task, having more qubits in a quantum computer does not necessarily lead to superior performance.

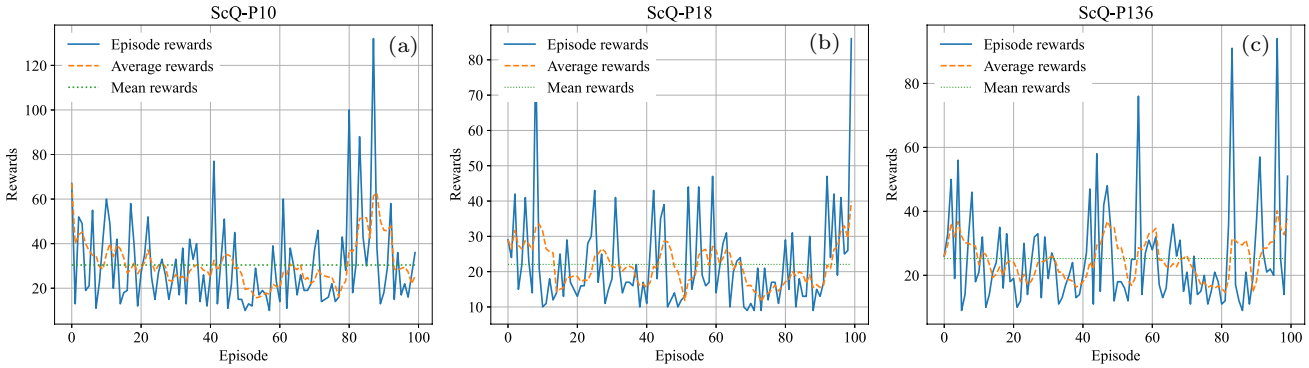


Fig. 4. RL training performance with Quafu within 100 episodes. RL agents have been trained on ScQ-P10, ScQ-P18 and ScQ-P136. The blue lines indicate episode rewards within 100 training episodes, the orange dashed lines take a moving average over episode rewards with a window of 5, and the dotted green lines show the mean rewards over all training processes.

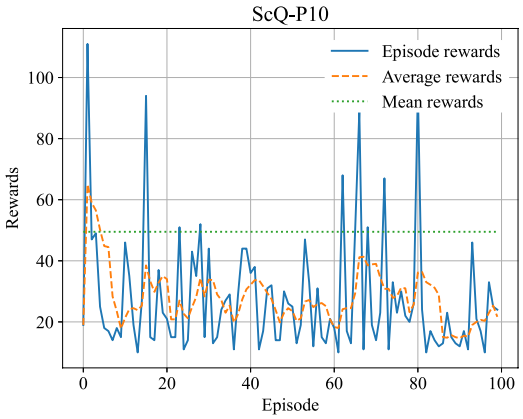


Fig. 5. Inference on ScQ-P10 within 100 episodes. The interpretation of the lines is similar to that in Fig. 4.

We also evaluate the pre-trained model on ScQ-P10 within 100 episodes as shown in Fig. 5. The results indicate that the RL agent trained on Quafu is capable of attaining a goal of over 100 rewards in the inference stage, with an overall mean reward of approximately 50. These numerical signs provide evidence of the agent's successful training on noisy devices. Although we limit the training and inference to 100 episodes in this study due to the expensive time cost, we believe that extending the training duration will finally lead to improved performance.

6. Conclusion

In this study, we implement quantum reinforcement learning on ScQ-P10, ScQ-P18, and ScQ-P136 from the Quafu cloud platform. We diligently choose our hardware-efficient PQCs using a multi-objective evolutionary algorithm and refine the REINFORCE learning algorithm to ensure the feasibility of training RL agents. Experimental results show that RL agents can be successfully trained and evaluated with

some relaxation of the original goals within 100 episodes. Based on the experimental results, the potential advantage of quantum reinforcement learning primarily stems from having fewer parameters while maintaining comparable performance.

Moving forward, there is potential to train agents on quantum devices with higher objectives, longer episode lengths, and a wider range of test environments. Additionally, the PQC search process could be conducted on real devices under finer restricted conditions to find a more suitable architecture for a specific device. Finally, other algorithms such as PPO could be introduced to stabilize the training process on quantum computers.

Acknowledgements

This work is supported by the Beijing Academy of Quantum Information Sciences. Haifeng Yu, Meng-Jun Hu and Wei-Feng Zhuang are supported by the National Natural Science Foundation of China (Grant No. 92365206). Hong-Ze Xu acknowledges the support of the China Postdoctoral Science Foundation (Certificate Number: 2023M740272). Zheng-An Wang is supported by the National Natural Science Foundation of China (Grant No. 12247168) and China Postdoctoral Science Foundation (Certificate Number: 2022TQ0036).

References

[1] Arute1 F, Arya1 K, Babbush R, *et al.* 2019 *Nature* **574** 505
[2] Wu Y, Bao W S, Cao S, *et al.* 2021 *Phys. Rev. Lett.* **127** 180501
[3] Zhu Q, Cao S, Chen F, *et al.* 2022 *Science Bullletin* **67** 240
[4] Preskill J 2018 *Quantum* **2** 79
[5] Leymann F and Barzen J 2020 *Quantum Science and Technology* **5** 044007
[6] Wendin G 2023 arXiv: 2302.04558 [quant-ph]
[7] Quafu quantum computing cloud platform http://quafu.baqis.ac.cn
[8] Cerezo M, Arrasmith A, Babbush R, Benjamin S C, Endo S, Fujii K, McClean J R, Mitarai K, Yuan X, Cincio L and Coles P J 2021 *Nat. Rev. Phys.* **3** 625
[9] Cao Y, Romero J, Olson J P, Degroote M, Johnson P D, Kieferov'a M, Kivlichan I D, Menke T, Peropadre B, Sawaya N P D, Sim S, Veis L and Aspuru-Guzik A 2019 *Chem. Rev.* **119** 10856
[10] Cao Y, Romero J and Aspuru-Guzik A 2018 *IBM J. Res. Dev.* **62** 6
[11] O'Malley P J J, Babbush R, Kivlichan I D, *et al.* 2016 *Phys. Rev. X* **6** 031007
[12] Zhou L, Wang S T, Choi S, Pichler H and Lukin M D 2020 *Phys. Rev. X* **10** 021067
[13] Farhi E, Goldstone J and Gutmann S 2014 arXiv: 1411.4028 [quant-ph]
[14] Zhuang W F, Pu Y N, Xu H Z, Chai X, Gu Y, Ma Y, Qamar S, Qian C, Qian P, Xiao X, Hu M J and Liu D E 2021 arXiv: 2112.11151 [quant-ph]
[15] Mitarai K, Negoro M, Kitagawa M and Fujii K 2018 *Phys. Rev. A* **98** 032309
[16] Zhou L, Wang S T, Choi S, Pichler H and Lukin M D 2019 *Quantum Science and Technology* **4** 043001
[17] Schuld M, Bocharov A, Svore K M and Wiebe N 2020 *Phys. Rev. A* **101** 032308
[18] Havlíček V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 *Nature* **567** 209
[19] Farhi E and Neven H 2018 arXiv: 1802.06002 [quant-ph]
[20] Abohashima Z, Elhosen M, Houssein E H and Mohamed W M 2020 arXiv: 2006.12270 [quant-ph]
[21] Dallaire-Demers P L and Killoran N 2018 *Phys. Rev. A* **98** 012324
[22] Lloyd S and Weedbrook C 2018 *Phys. Rev. Lett.* **121** 040502
[23] Tian J, Sun X, Du Y, Zhao S, Liu Q, Zhang K, Yi W, Huang W, Wang C, Wu X, Hsieh M H, Liu T, Yang W and Tao D 2022 arXiv: 2206.03066 [quant-ph]
[24] Yin X F, Du Y, Fei Y Y, Zhang R, Liu L Z, Mao Y, Liu T, Hsieh M H, Li L, Liu N L, Tao D, Chen Y A and Pan J W 2022 *Phys. Rev. Lett.* **128** 110501)
[25] Huang H Y, Broughton M, Mohseni M, Babbush R, Boixo S, Neven H and McClean J R 2021 *Nat. Commun.* **12** 2631
[26] Du Y, Hsieh M H, Liu T and Tao D 2020 *Phys. Rev. Res.* **2** 033125
[27] Liu Y, Arunachalam S and Temme K 2021 *Nat. Phys.* **17** 1013
[28] Sutton R S and Barto A G 1998 *IEEE Transactions on Neural Networks* **9** 1054
[29] Chen S Y C, Yang C H H, Qi J, Chen P Y, Ma X and Goan H S 2020 *IEEE Access* **8** 141007
[30] Lockwood O and Si M 2020 *Proceedings of the* 16*th AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, p. 245
[31] Wu S, Jin S, Wen D, Han D and Wang X 2023 arXiv: 2012.10711 [quant-ph]
[32] Jerbi S, Trenkwalder L M, Poulsen Nautrup H, Briegel H J and Dunjko V 2021 *PRX Quantum* **2** 010328
[33] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J and Zaremba W 2016 arXiv: 1606.01540 [cs.LG]
[34] Jerbi S, Gyurik C, Marshall S C, Briegel H J and Dunjko V 2021 arXiv: 2103.05577 [quant-ph]
[35] Skolik A, Jerbi S and Dunjko V 2022 *Quantum* **6** 720
[36] Ding L and Spector L 2022 *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, p. 2190
[37] Zhang A and Zhao S 2022 arXiv: 2212.00421 [quant-ph]
[38] Giovagnoli A, Ma Y and Tresp V 2023 arXiv: 2304.06981 [quant-ph]
[39] Rattew A G, Hu S, Pistoia M, Chen R and Wood S 2020 arXiv: 1910.09694 [quant-ph]
[40] Chivilikhin D, Samarin A, Ulyantsev V, Iorsh I, Oganov A R and Kyrienko O 2020 arXiv: 2007.04424 [quant-ph]
[41] Lu Z, Shen P X and Deng D L 2021 *Phys. Rev. Appl.* **16** 044039
[42] Schuld M, Sweke R and Meyer J J 2021 *Phys. Rev. A* **103** 032430
[43] Pérez-Salinas A, Cervera-Lierta A, Gil-Fuster E and Latorre J I 2020 *Quantum* **4** 226
[44] Duan Y, Chen X, Houthooft R, Schulman J and Abbeel P 2016 *Proceedings of the* 33*rd International Conference on Machine Learning*, p. 1329
[45] Sutton R S, McAllester D, Singh S and Mansour Y 1999 *Proceedings of the* 12*th International Conference on Neural Information Processing Systems*, p. 1057
[46] Kandala A, Mezzacapo A, Temme K, Takita M, Brink M, Chow J M and Gambetta J M 2017 *Nature* **549** 242
[47] Deb K, Pratap A, Agarwal S and Meyarivan T 2002 *IEEE Transactions on Evolutionary Computation* **6** 182
[48] Williams R J 1992 *Mach. Learn.* **8** 229
[49] Broughton M, Verdon G, Trevor M, *et al.* 2021 arXiv: 2003.02989 [quant-ph]