www.nature.com/scientificreports

scientific reports

OPEN

Quantum architecture search via truly proximal policy optimization

Xianchao Zhu[1]✉ & Xiaokai Hou[2]

Quantum Architecture Search (QAS) is a process of voluntarily designing quantum circuit architectures using intelligent algorithms. Recently, Kuo et al. (Quantum architecture search via deepreinforcement learning. arXiv preprint arXiv:2104.07715, 2021) proposed a deep reinforcement learning-based QAS (QAS-PPO) method, which used the Proximal Policy Optimization (PPO) algorithm to automatically generate the quantum circuit without any expert knowledge in physics. However, QAS-PPO can neither strictly limit the probability ratio between old and new policies nor enforce well-defined trust domain constraints, resulting in poor performance. In this paper, we present a new deep reinforcement learning-based QAS method, called Trust Region-based PPO with Rollback for QAS (QAS-TR-PPO-RB), to automatically build the quantum gates sequence from the density matrix only. Specifically, inspired by the research work of Wang, we employ an improved clipping function to implement the rollback behavior to limit the probability ratio between the new strategy and the old strategy. In addition, we use the triggering condition of the clipping based on the trust domain to optimize the policy by restricting the policy within the trust domain, which leads to guaranteed monotone improvement. Experiments on several multi-qubit circuits demonstrate that our presented method achieves better policy performance and lower algorithm running time than the original deep reinforcement learning-based QAS method.

Reinforcement learning (RL)[1] has achieved great success and demonstrated human or superhuman abilities in various tasks, such as mastering video games[2–5] and the game of Go[6,7]. With such success, it is natural to apply such technologies to scientific fields that require complex control capabilities. In fact, RL has been used to study quantum control[8–14], quantum error correction[15–18] and the optimization of variational quantum algorithms[19–22].

RL has also been used to optimize the structure and parameters of neural networks, which is called Neural Architecture Search (NAS)[23]. Specifically, NAS trains an RL agent to sequentially add different neural networks components (such as convolution operation, residual connection, and pooling) and then automatically generates a high-performance neural network by evaluating the model's performance to adjust these components structure. NAS is already comparable to human experts in specific tasks, effectively reducing neural networks' use and implementation costs[24–31].

Quantum algorithms are proven to have exponential or quadratic operational efficiency improvements in solving specific problems compared to classical algorithms[32,33], such as integer factorization[34] and unstructured database searches[35]. Recent studies in variational quantum algorithms (VQA) have applied quantum computing to many scientific domains, including molecular dynamical studies[36], quantum optimization[37,38] and various quantum machine learning (QML) applications such as regression[39–41], classification[40,42–56], generative modeling[57–62], deep reinforcement learning[63–69], sequence modeling[39,70,71], speech identification[72], distance metric learning[73,74], transfer learning[46] and federated learning[75]. However, designing a quantum circuit to solve a specific task is not easy because it requires domain knowledge and sometimes extraordinary insight.

Recently, a deep reinforcement learning-based Quantum Architecture Search (QAS-PPO) approach is proposed to automatically generate the quantum circuit via the Proximal Policy Optimization (PPO) algorithm without any expert knowledge in physics[76]. Specifically, QAS-PPO uses the PPO[77] method to optimize the interaction of the RL agent with a quantum simulator to learn the target quantum state. During the interaction, the agent sequentially generates an output action as a candidate for a quantum gate or quantum operation placed on the circuit. Then the fidelity of generated quantum circuit is evaluated to determine the agent whether the agent has reached the goal. This process is performed iteratively to train the RL agent. Despite its success, QAS-PPO can

[1]School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou 450001, China. [2]Institute of Fundamental and Frontier Sciences, University of Electronic Science and Technology of China, Chengdu, China. ✉email: xczhuiffs@163.com

neither strictly limit the probability ratio between old and new policies nor enforce well-defined trust domain constraints, resulting in poor performance.

This paper proposes a new deep reinforcement learning-based QAS approach, named Trust Region-based PPO with Rollback for QAS (QAS-TR-PPO-RB), to automatically build the quantum gates sequence from the density matrix only. Specifically, inspired by the research work of Wang et al.[78], we adopt an improved clipping function to implement the rollback behavior to limit the probability ratio between the new strategy and the old strategy to prevent the strategy from being pushed away during training. Moreover, we optimize the strategy within the trust region by replacing the clipped trigger conditions with those based on the trust region to guarantee monotonic improvement. Experimental results on several benchmark tasks demonstrate that the proposed method observably improves policy performance and algorithm running time compared to the original deep reinforcement learning-based QAS methods.

The rest of this paper is arranged as follows. "Preliminaries" presents preliminaries on reinforcement learning, advantage actor-critic (A2C), proximal policy optimization (PPO) and quantum architecture search. The QAS-PPO method is reviewed in "Quantum architecture search with deep reinforcement learning". "Methods" proposes a new deep reinforcement learning-based QAS algorithm, called Trust Region-based PPO with Rollback for QAS (QAS-TR-PPO-RB). Specifically, we adopt an improved clipping function to implement the rollback behavior to limit the probability ratio between the new strategy and the old strategy to prevent the strategy from being pushed away during training. In "Experiments", we present several experimental comparative results for the automatic generation of quantum circuits for multi-qubit target states to show the superiority of our presented method. Finally, we conclude this paper in "Conclusion".

Preliminaries

Reinforcement learning.

The reinforcement learning (RL) algorithm that maximizes the value function is called value-based reinforcement learning. Unlike value-based RL, which learns a value function and uses it as a reference to generate decisions at each step, another RL method is called policy gradient. In this method, the strategy function $\pi(a|s;\theta)$ is parameterized with the parameters θ. Then θ will be affected by the optimization procedure, which rises gradient ascent on the expected total return $\mathbb{E}[R_t]$. One of the classic examples of strategy gradient algorithm is the REINFORCE algorithm[79]. In the standard REINFORCE algorithm, the parameters θ is updated along the direction $\nabla_\theta \log \pi(a_t|s_t;\theta)R_t$, which is the unbiased estimate of $\nabla_\theta \mathbb{E}[R_t]$. However, the strategy gradient method is affected by the variance of the $\nabla_\theta \mathbb{E}[R_t]$, making the training very difficult. To reduce the estimate variance and keep it unbiased, the learning function of the state $b_t(s_t)$, which is the baseline, can be substracted from the return value. So the result is $\nabla_\theta \log \pi(a_t|s_t;\theta)(R_t - b_t(s_t))$.

Advantage actor-critic (A2C).

The estimation of the value function is a common choice for the baseline $b_t(s_t) \approx V^\pi(s_t)$. This choice usually results in a much lower variance estimation of the strategy gradient. When using the approximation value function as the basic line, the quantity $R_t - b_t = Q(s_t, a_t) - V(s_t)$ can be regarded as the advantage function $A(s_t, a_t)$ of the action in the state s_t. Intuitively, one can see this advantage as how nice or nasty the action is compared to the average value in this state $V(s_t)$. For example, if the $Q(s_t, a_t)$ equals to 10 at a given time-step t, it is not clear whether a_t is a good action or not. However, if we also know that the $V(s_t)$ equals to, say 2 here, we will imply that a_t may not be bad. Conversely, if the $V(s_t)$ equals to 15, then the advantage is -5, meaning that the Q value for this action at is well below the average $V(s_t)$ and therefore that action is not good. This approach is called advantage actor-critic (A2C) approach where the strategy π is the actor and the value function V is the critic [1].

Proximal policy optimization (PPO).

In the strategy gradients method, the policy is optimized by gradient descent according to the policy loss function $L_{policy}(\theta) = \mathbb{E}_t[-\log \pi(a_t|s_t;\theta)]$. However, the training itself may suffer from instabilities. If the step size of policy update is too small, the training process will be too slow. On the other hand, if the step size is too larger, the training will have a high variance. Proximal policy optimization (PPO) solves this problem by restricting the strategy update step size at each training step[77]. Specifically, The PPO introduces a loss function called the clipped proxy loss function that will restrict the strategy change a small range with the help of a clip. Consider the ratio between the probability of action under present strategy and the probability under anterior strategy $q_t(\theta) = \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta_{old})}$. If $q_t(\theta) > 1$, it means the action is with higher probability in the present strategy than in the old one. And if $0 < q_t(\theta) < 1$, it means that the action is less probable in the present strategy than in the old one. The new loss function can then be defined as $L_{policy}(\theta) = \mathbb{E}_t[q_t(\theta)A_t] = \mathbb{E}_t[\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta_{old})}A_t]$, where $A_t = R_t - V(s_t;\theta)$ is the advantage function. However, if the action under current policy is much more probable than in the previous policy, the ratio q_t may be large, leading to a large policy update step. To circumvent this problem, the original PPO algorithm adds a constraint on the ratio, which can only be in the range 0.8 to 1.2. The modified loss function is defined as follow:

$$L_{policy}(\theta) = \mathbb{E}_t[-\min(q_t(\theta)A_t, \mathcal{F}^{CLIP}(q_t(\theta), \epsilon)A_t)]. \tag{1}$$

The clipping function $\mathcal{F}^{CLIP}$ is denoted as

$$\mathcal{F}^{CLIP}(q_t(\theta),\epsilon) = \begin{cases} 1-\epsilon, & \text{if } \ q_t(\theta) \le 1-\epsilon \\ 1+\epsilon, & \text{if } \ q_t(\theta) \ge 1+\epsilon \ , \\ q_t(\theta) & \text{else} \ , \end{cases} \tag{2}$$

where the $(1-\epsilon, 1+\epsilon)$ represents the clipping range, $\epsilon \in [0,1]$ is the clip hyperparameter (common choice is 0.2).

Finally, the value loss and entropy bonus are added into the total loss function as usual: $L(\theta) = L_{policy}(\theta) + c_1 L_{value}(\theta) - c_2 H(\theta)$ where $H(\theta) = \mathbb{E}_t[H_t](\theta) = \mathbb{E}_t[-\sum_j \pi(a_j|s_t;\theta)\log(\pi(a_j|s_t;\theta))]$ is the entropy bonus which is used to encourage exploration and $L_{value}(\theta) = \mathbb{E}_t[\|R_t - V(s_t;\theta)\|^2]$ is the value loss.

Quantum architecture search. Quantum architecture search (QAS) is a class of approaches using algorithms such as quantum simulated annealing (QSA)[80, 81], quantum evolutionary algorithm (QEA)[82, 83], quantum machine learning (QML)[84–87], and quantum reinforcement learning (QRL)[76, 88–90] intelligent algorithms to voluntarily search for the best quantum circuit for a given target quantum state. Existing research work shows that quantum circuits generated by QAS methods based on variational quantum algorithms have reached or even surpassed quantum circuits designed based on human expertise. However, when such quantum architecture search algorithms automatically generate quantum circuits in discrete environments, they often need to evaluate the performance of many quantum circuits with different structures, resulting in colossal resource consumption. Recently, Zhang et al. presented a Differentiable Quantum Architecture Search (DQAS) method, which expanded the space to be searched from discrete domain to continuous domain and used gradient descent to optimize the entire quantum circuit generation process to achieve relatively high performance[91].

Quantum architecture search with deep reinforcement learning

Given the original quantum state $|0...0\rangle$ and the target quantum state, the goal is to produce a quantum circuit that converts the original state into the target state within a specific fidelity threshold. En-Jui Kuo et al. use the Pauli measurement as an observation, which is a often-used setting for quantum mechanics. Then they adopt two RL algorithms (PPO and A2C) respectively to achieve the above goal[76]. Specifically, environment E represents a quantum computer or quantum simulator. The RL agent is hosted on a classic computer and interacts with environment E. At each iteration step, the RL agent selects an action a from the set of possible actions A, consisting of different quantum operations. After the RL agent updates the quantum circuit based on the selected action, environment E tests the newly generated circuit and computes the fidelity between the given target quantum state and the currently developed state quantum state. If the calculated fidelity has reached or exceeded a predefined threshold, the round ends, and the RL agent will receive a positive feedback reward. Elsewise, the RL agent will receive a negative feedback reward. This process continues until the maximum number of steps required for the iteration will terminate. The optimization of the algorithm in this interaction can be realized by using reinforcement learning algorithm A2C or PPO.

Given the number of qubits $n \in N$, the initial quantum state $|0\rangle^{\otimes^n}$, the target state, the tolerance error, and a set of quantum gates $\mathbb{G}$, the goal of the algorithm is to discover a quantum circuit $\mathcal{C}$ by construsting an objective function $\mathcal{F}$:

$$\mathcal{F} : (|0\rangle^{\otimes^n}, |\psi\rangle, \epsilon, \mathbb{G}) \rightarrow \mathcal{C} \tag{3}$$

such that $1 \ge D(|\psi\rangle, \mathcal{C}(|0\rangle^{\otimes^n})) \ge 1-\epsilon$, where $\mathcal{C}$ is composed of gates $g \in \mathbb{G}$ and D is a distance metric between two quantum states (larger is better). In this paper, we use the fidelity[92] as our distance D. Given two density operators ρ and σ, the fidelity between two operators is usually expressed as $F(\rho,\sigma) = [tr\sqrt{\sqrt{\rho}\sigma\sqrt{\rho}}]^2$. In particular, in the case where ρ and σ represent pure quantum states, i.e., $\rho = |\phi_\rho\rangle\langle\phi_\rho|$ and $\sigma = |\phi_\sigma\rangle\langle\phi_\sigma|$, respectively, the original expression can be reduced to the inner product of the two quantum states: $F(\rho,\sigma) = |\langle\phi_\rho|\phi_\sigma\rangle|^2$.

Furthermore, En-Jui Kuo et al. verified the performance of their proposed deep reinforcement learning-based QAS algorithm using Bell states and Greenberg–Horn–Zehlinger (GHZ) states as target quantum states, respectively.

A Bell state achieves maximal two-qubit entanglement,

$$|Bell\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{4}$$

To generate a Bell state, En-Jui Kuo et al. picked the observation to be the expectation values of Pauli matrices on each qubits $\{\langle\sigma_j^i\rangle | i \in 0,1, j \in x,y,z\}$. The action set $\mathbb{G}$ is

$$\mathbb{G} = \bigcup_{0}^{n-1} U_i(\frac{\pi}{4}), X_i, Y_i, Z_i, H_i, CNOT_{i,(i+1)(mod2)}, \tag{5}$$

where $n = 2$ (for two qubits), $U_i(\theta) = \begin{pmatrix} 1 & 0 \\ 0 & exp(i\theta) \end{pmatrix}$ is the single-qubit rotation applied to the i-th qubit around the Z-axis, $X_i \equiv \sigma_x^i$ denotes the Pauli-X gate and likewise for Y_i and Z_i, H_i represents the Hadamard gate, and $CNOT_{i,j}$ is the CNOT gate where the i-th qubit is the control bit, and the j-th qubit is the target bit, so there are 12 actions in total.

A GHZ state is a multi-qubit generalization of the Bell state, in which an equal superposition between the lowest and the highest energy states is created.

$$|GHZ\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle). \tag{6}$$

To generate the 3-qubit GHZ state, En-Jui Kuo et al. adopted the expectation values of individual qubit's Pauli matrices, resuting in 9 observables in the aggregate. For the actions, En-Jui Kuo et al. selected the same single-qubit gates as in Eq. (6), and sixe *CNOT* gate as two-qubit gates.

Despite its success, QAS-PPO can neither strictly limit the probability ratio between old and new policies nor enforce well-defined trust domain constraints, resulting in poor performance. The former problem is mainly due to the inability of PPO to eliminate the incentives of pushing away the strategy, while the latter situation is primarily due to the essential difference of the two types of constraints used by PPO and Trust Region Policy Optimization (TRPO), respectively.

Methods

In this section, to address above issue, we propose a new deep reinforcement learning-based QAS approach, called Trust Region-based PPO with Rollback for QAS (QAS-TR-PPO-RB). More realistically adhering to the "proximal" property-bound strategy within the trust region, our method can significantly improve over original deep reinforcement learning-based QAS approaches in terms of policy performance and sample efficiency.

Analysis of the "proximal" property of PPO. PPO limits the strategy by reducing the probability ratio between old and new policies. However, in practice, the known probability ratio is not limited to the clipping range. A significant factor in this problem is that the limiting mechanism cannot eliminate the excitation from the overall target function $L(\theta)$, pushing this out-of-the-range $q_t(\theta)$ further beyond the limit[93]. Moreover, PPO does not explicitly impose a trust domain constraint on the probability ratio between old and new policies, i.e., the *KL*-divergence between the two strategies. Even if the probability ratio $q_t(\theta)$ is bounded, the corresponding *KL*-divergence $D_{KL}^{s_t}(\theta_{old}, \theta)$ is not necessarily bounded[78].

PPO with rollback for quantum architecture search. As mentioned in "Analysis of the "proximal" property of PPO", the PPO method used in the QAS-PPO method cannot strictly constrain the range of probability ratio: the limiting mechanism cannot eliminate the motivation to drive $q_t(\theta)$ beyond the limiting range, in fact, $q_t(\theta)$ often deviates from the constraints of this mechanism ultimately lead to poor performance. We solve this problem by replacing the clip function $\mathcal{F}^{CLIP}$ with a rollback function whose mathematical expression follows.

$$\mathcal{F}^{RB}(q_t(\theta), \epsilon, \alpha) = \begin{cases} -\alpha q_t(\theta) + (1+\alpha)(1-\epsilon), & \text{if } q_t(\theta) \leq 1-\epsilon \\ -\alpha q_t(\theta) + (1+\alpha)(1+\epsilon), & \text{if } q_t(\theta) \geq 1+\epsilon \\ q_t(\theta) & \text{otherwise}, \end{cases} \tag{7}$$

where α represents the hyper-parameter that controls the intensity of the rollback. The new overall target function is $L^{RB}(\theta)$. When $q_t(\theta)$ exceeds the limit range, the rollback function $\mathcal{F}^{RB}(q_t(\theta), \epsilon, \alpha)$ will produce passive stimulation. Therefore, it can offset the excitation from the overall target function $L^{RB}(\theta)$ to a certain extent. The rollback operation prevents the probability ratio $q_t(\theta)$ from being squeezed out more strongly than the original clip function[78].

The pseudocode for the PPO with Rollback for Quantum Architecture Search approach is shown below:

Algorithm 1 PPO with Rollback for Quantum Architecture Search (QAS-PPO-RB)

Require: Initialize the parameters of the number of total episode M, the max steps in a single episode S, the update timestep U, epoch number K, the epsilon clip C, the slope rollback α, the trajectory buffer T, the timestep number t, and the model parameters θ and θ_{old}.

```
for $episode = 1, 2, ..., M$ do
    $Initialize\ state\ s_1$
    for $step = 1, 2, ..., S$ do
        $Update\ the\ timestep\ t = t + 1$
        $Choose\ the\ action\ a_t\ from\ the\ policy\ \pi(a_t|s_t;\theta_{old})$
        $Execute\ the\ action\ a_t\ in\ emulator\ and\ then\ obtain\ reward\ r_t\ and\ next\ state\ s_{t+1}$
        $Record\ the\ transition\ matrixs\ (s_t, a_t, \log_\alpha \pi(a_t|s_t;\theta_{old}), r_t)\ in\ T$
        if $t = U$ then
            $Compute\ the\ discounted\ reward\ R_t\ for\ each\ state\ s_t\ in\ the\ trajectory\ buffer\ T$
            for $k = 1, 2, ..., K$ do
                $Calculate\ the\ log\ probability\ \log_\alpha \pi(a_t|s_t;\theta),\ state\ values\ V(s_t,\theta)\ and$
                $entropy\ H_t.$
                $Calculate\ the\ ratio\ q_t = exp(\log_\alpha \pi(a_t|s_t;\theta) - \log_\alpha \pi(a_t|s_t;\theta_{old}))$
                $Compute\ the\ advantage\ function\ A_t = R_t - V(s_t,\theta)$
                if $q_t \leq (1 - C)$ then
                    $surr = (-\alpha q_t + (1 + \alpha)(1 - C))A_t$
                else if $ratio \geq (1 + C)$ then
                    $surr = (-\alpha q_t + (1 + \alpha)(1 + C))A_t$
                else
                    $surr = q_t \times A_t$
                end if
                $Compute\ the\ total\ loss\ L = \mathbb{E}_t[surr + 0.5||V(s_t,\theta) - R_t||^2 - 0.01H_t]$
                $Update\ the\ agent\ policy\ parameters\ \theta\ with\ gradient\ descent\ on\ the\ loss\ L$
            end for
            $Update\ the\ \theta_{old}\ to\ \theta$
            $Reset\ the\ trjectory\ buffer\ T$
            $Reset\ the\ timestep\ number\ t = 0$
        end if
    end for
end for
```

Trust region-based PPO for quantum architecture search. As mentioned in "Analysis of the "proximal" property of PPO", the clipping function in the original deep reinforcement learning-based QAS method uses the probability ratio as the element of the clipping trigger condition, which makes the difference between the ratio-based constraints and the trust domain-based constraints used: the constraint probability ratio is not enough to constrain the KL-divergence, which ultimately leads to poor performance. Therefore, we replace the ratio-based clipping function with a trust domain-based clipping function. Formally, when the strategy π_θ exceeds the trust domain, the probability ratio is tailored,

$$\mathcal{F}^{TR}(q_t(\theta), \delta) = \begin{cases} q_t(\theta_{old}), & \text{if } D_{KL}^{s_t}(\theta_{old}, \theta) \geq \delta \\ q_t(\theta) & \text{else ,} \end{cases} \tag{8}$$

where δ is the hyperparameter, $q_t(\theta_{old}) = 1$ is a constant. When the strategy π_θ exceeds the trust region, that is, $D_{KL}^{s_t}(\theta_{old}, \theta) \geq \delta$, the incentive for updating the strategy is removed. Although the clipped value $q_t(\theta_{old})$ may make the proxy target function uncontinuous, this discontinuity will not influence the optimization of the parameter θ because the gradient will not be affected by the constant value.

In general, our proposed QAS-TR-PPO method combines the advantages of TRPO and PPO: it is theoretically reasonable (subject to the trust domain), is easy to implement, and only need to do one-rank optimization. On the one hand, our approach does not require KL-divergence $D_{KL}^{s_t}(\theta_{old}, \theta)$ to optimize θ. $D_{KL}^{s_t}(\theta_{old}, \theta)$ calculates to determine whether to $q_t(\theta)$ or not. Compared with the PPO used in the original method, our method uses a different strategy metric to limit the strategy. Specifically, unlike PPO, the ratio-based metric $\frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta_{old})}$ is used to impose element-by-element constraints on the sampling action points. Our method uses a trusted domain the KL-divergence $\sum_a \pi(a_t|s_t;\theta_{old}) \log \frac{\pi(a_t|s_t;\theta)}{\pi(a_t|s_t;\theta_{old})}$ based on the trust region to impose a summation constraint on the action space. Crucially, the central willingness is that ratio-based regulations can impose relatively strict restrictions on actions that the old strategy does not like, that is, $\pi_{\theta_{old}}$ is small, which may result in finite sample efficiency when the strategy is initialized from a wrong solution. On the contrary, the trust domain-based approach we adopted has no such prejudice and tends to show higher sample efficiency in reality[78].

Finally, we should pay attention to the importance of the $\min(\cdot, \cdot)$ operation for all variants of PPO. The operation function $\min(\cdot, \cdot)$ is denoted as:

$$L_{policy}^{TR}(\theta) = \mathbb{E}_t[-\min(q_t(\theta)A_t, \mathcal{F}^{TR}(q_t(\theta), \delta)A_t)]. \tag{9}$$

Schulman et al. proposed that this additional $\min(\cdot,\cdot)$ operation made $L_{policy}^{TR}\theta$ become a lower bound on the unclipped target function $q_t(\theta)A_t$[94]. As Eq. (11) expresses, there is no target of $\min(\cdot,\cdot)$ operation, namely $\mathcal{F}^{TR}(q_t(\theta), \delta)A_t$. Once the policy violates the trust region, it will stop updating, even if the target value is less than the original value, that is, $q_t(\theta)A_t \le q_t(\theta_{old})A_t$. The $\min(\cdot,\cdot)$ operation virtually provides a remedy for this problem. Therefore, Eq. (12) can be rewritten as

$$L_{policy}^{TR}(\theta) = \begin{cases} q_t(\theta_{old})A_t, & \text{if } D_{KL}^{s_t}(\theta_{old}, \theta) \ge \delta \textit{ and } q_t(\theta)A_t \ge q_t(\theta_{old})A_t \\ q_t(\theta)A_t & \text{else .} \end{cases} \tag{10}$$

In this way, the ratio will be clipped only when the target value increases (and the policy violates the constraints). The Trust Region-based PPO for Quantum Architecture Search method is as follows (as shown in Algorithm 2):

Algorithm 2 Trust Region-based PPO for Quantum Architecture Search (QAS-TR-PPO)

Require: Initialize the parameters of the number of total episode M, the max steps in a single episode S, the update timestep U, the epoch number K, the epsilon clip C, the klrange δ, the trajectory buffer T, the timestep number t, and the model parameters θ and θ_{old}

```
for $episode = 1, 2, ..., M$ do
    $Initialize\ state\ s_1$
    for $step = 1, 2, ..., S$ do
        $Update\ the\ timestep\ t = t + 1$
        $Choose\ the\ action\ a_t\ from\ the\ policy\ \pi(a_t|s_t; \theta_{old})$
        $Execute\ the\ action\ a_t\ in\ emulator\ and\ then\ obtain\ reward\ r_t\ and\ next\ state\ s_{t+1}$
        $Record\ the\ transition\ matrixs\ (s_t, a_t, \log_\alpha \pi(a_t|s_t; \theta_{old}), r_t)\ in\ T$
        if $t = U$ then
            $Compute\ the\ discounted\ reward\ R_t\ for\ each\ state\ s_t\ in\ the\ trajectory\ buffer\ T$
            for $k = 1, 2, ..., K$ do
                $Compute\ the\ log\ probability\ \log_\alpha \pi(a_t|s_t; \theta),\ state\ values\ V(s_t, \theta)\ and$
                $entropy\ H_t.$
                $Compute\ the\ ratio\ q_t = exp(\log_\alpha \pi(a_t|s_t; \theta) - \log_\alpha \pi(a_t|s_t; \theta_{old}))$
                $Compute\ the\ advantage\ function\ A_t = R_t - V(s_t, \theta)$
                $Compute\ the\ KL\ divergence\ D_{KL}^{s_t}(\theta_{old}, \theta)\ between\ \theta_{old}\ and\ \theta\ at\ state\ s_t$
                if $D_{KL}^{s_t}(\theta_{old}, \theta) \ge \delta$ then
                    $surr = q_t(\theta_{old})A_t$
                else
                    $surr = q_tA_t$
                end if
                $Compute\ the\ total\ loss\ L = \mathbb{E}_t[surr + 0.5||V(s_t, \theta) - R_t||^2 - 0.01H_t]$
                $Update\ the\ agent\ policy\ parameters\ \theta\ with\ gradient\ descent\ on\ the\ loss\ L$
            end for
            $Update\ the\ \theta_{old}\ to\ \theta$
            $Reset\ the\ trjectory\ buffer\ T$
            $Reset\ the\ timestep\ number\ t = 0$
        end if
    end for
end for
```

Trust region-based PPO with rollback for quantum architecture search. However, the tailoring based on the trust domain may still have the problem of an unbounded probability ratio. When the strategy exceeds the trust region, the method proposed above does not provide any negative incentives, leading to poor performance. Therefore, we solve this problem by combining the tailoring based on the trust domain and the rollback mechanism.

$$\mathcal{F}^{TR-RB}(q_t(\theta), \delta, \alpha) = \begin{cases} -\alpha q_t(\theta_{old}), & \text{if } D_{KL}^{s_t}(\theta_{old}, \theta) \ge \delta \\ q_t(\theta) & \text{else .} \end{cases} \tag{11}$$

As shown in Eq. (13), when π_θ exceeds the trust domain, our proposed $\mathcal{F}^{TR-RB}(q_t(\theta), \delta, \alpha)$ method will produce negative excitation. Trust Region-based PPO with Rollback for Quantum Architecture Search method is as follows (as shown in Algorithm 3):

Algorithm 3 Trust Region-based PPO with Rollback for Quantum Architecture Search

Require: Initialize the episode number M, the max steps in a single episode S, the timestep U, the epoch number K, the epsilon clip C, the slope rollback α, the klrange δ, the trajectory buffer T, the timestep number t, and the model parameters θ and θ_{old}.

```
for $episode = 1, 2, ..., M$ do
    $Initialize\ state\ s_1$
    for $step = 1, 2, ..., S$ do
        $Update\ the\ timestep\ t = t + 1$
        $Choose\ the\ action\ a_t\ from\ the\ policy\ \pi(a_t|s_t; \theta_{old})$
        $Excute\ the\ action\ a_t\ in\ emulator\ and\ then\ obtain\ reward\ r_t\ and\ next\ state\ s_{t+1}$
        $Record\ the\ transition\ matrix\ (s_t, a_t, \log_\alpha \pi(a_t|s_t; \theta_{old}), r_t)\ in\ T$
        if $t = U$ then
            $Compute\ the\ discounted\ reward\ R_t\ for\ each\ state\ s_t\ in\ the\ trajectory\ buffer\ T$
            for $k = 1, 2, ..., K$ do
                $Compute\ the\ log\ probability\ \log_\alpha \pi(a_t|s_t; \theta),\ state\ values\ V(s_t, \theta)\ and$
                 $entropy\ H_t.$
                $Compute\ the\ ratio\ q_t = exp(\log_\alpha \pi(a_t|s_t; \theta) - \log_\alpha \pi(a_t|s_t; \theta_{old}))$
                $Compute\ the\ advantage\ function\ A_t = R_t - V(s_t, \theta)$
                $Compute\ the\ KL\ divergence\ D^{s_t}_{KL}(\theta_{old}, \theta)\ between\ \theta_{old}\ and\ \theta\ at\ state\ s_t$
                if $D^{s_t}_{KL}(\theta_{old}, \theta) \geq \delta$ then
                    $surr = -\alpha q_t(\theta_{old})A_t$
                else
                    $surr = q_t A_t$
                end if
                $Compute\ the\ total\ loss\ L = \mathbb{E}_t[surr + 0.5||V(s_t, \theta) - R_t||^2 - 0.01H_t]$
                $Update\ the\ agent\ policy\ parameters\ \theta\ with\ gradient\ descent\ on\ the\ loss\ L$
            end for
            $Update\ the\ \theta_{old}\ to\ \theta$
            $Reset\ the\ trjectory\ buffer\ T\ and\ the\ timestep\ number\ t = 0$
        end if
    end for
end for
```

Experiments

Experimental settings.

Optimizer. In this paper, we employ the Adam optimizer for training the RL agent in the A2C, PPO, PPO-RB, TR-PPO and TR-PPO-RB cases[95–99]. Adam is one of the gradient-descent methods which calculates the self-adaptive learning rates of each parameter. Furthermore, Adam stores both the exponentially damping mean of gradient g_t and its square g_t^2,

$$\mu_t =\zeta_1\mu_{t-1} + (1-\zeta_1)g_t, \tag{12}$$

$$\nu_t =\zeta_2\nu_{t-1} + (1-\zeta_2)g_t^2, \tag{13}$$

where ζ_1 and ζ_2 are hyperparameters. We use $\zeta_1 = 0.9$ and $\zeta_2 = 0.999$ in this papaer. The μ_t and ν_t are adjustable according to the following equation to offset the biases towards 0,

$$\hat{\mu}_t =\frac{m_t}{1-\zeta_1^t}, \tag{14}$$

$$\hat{\nu}_t =\frac{\nu_t}{1-\zeta_2^t}. \tag{15}$$

The parameters θ_t in the our method in the time step t are then updated according to the following equation,

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{\nu}_t} + \epsilon}\hat{\mu}_t. \tag{16}$$

Quantum noise in quantum simulator. In this paper, we consider two forms of errors: gate errors and measurement errors[100]. The gate error refers to the defect in any quantum operation during the algorithm's execution, and the measurement error refers to the error generated in the quantum measurement process. Specifically, for gate error, we consider the depolarizing noise, which replaces the state of any qubit with a stochastic state of probability p_{gate}. For the measurement error, we think about a stochastic flip between 0 and 1 with probability p_{meas} immediately before the actual measurement. We use the following noise configuration in the simulation software to test the manifestation of the agent of our proposed approach:

- error rate (both p_{gate} and p_{meas})= 0.001
- error rate (both p_{gate} and p_{meas})= 0.005

Density matrix of quantum states. The generic form of a density matrix ρ of a quantum state under the basis $|\psi_j\rangle$ is,

$$\rho = \sum_j p_j |\psi_j\rangle\langle\psi_j|, \tag{17}$$

where p_j denotes the probability that the quantum system is in the pure state $|\psi_j\rangle$ such that $\sum_j p_j = 1$. For example, the density matrix of the Bell state adopted in this paper is $|Bell\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. Its corresponding density matrix ρ is then given by

$$|Bell\rangle\langle Bell| = \frac{1}{2}(|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|). \tag{18}$$

Quantum state tomography. Quantum state tomography (QST), which reconstructs quantum states of quantum systems through quantum measurements, plays an important role in verifying and benchmarking quantum devices in various quantum information processing tasks. Expand the density matrix in the Pauli basis of N qubits,

$$\rho = \frac{1}{2^N} \sum_{i_1,\ldots,i_N=0}^{3} C_{i_1,\ldots,i_N} \sigma_{i_1} \otimes \cdots \otimes \sigma_{i_N}, \tag{19}$$

where $4^N - 1$ measurement operations are required to determine ρ (minus one due to the conservation of probability, $Tr(\rho) = 1$). More generally, the measurement using $4^N - 1$ linearly independent projection operators can uniquely determine the density matrix, where Eq. (14) is a special case with the projectors being the Pauli operators. Therefore, the number of measurements increase exponentially in the qubit number N, which poses a huge challenge for verifying multi-qubit quantum states in any experiments, and under a limited number of shots, the expectation values of $\rho_{i_1,\ldots,i_N}$ can only be measured within certain accuracy. In this paper, we adopt IBM's Qiskit software package to perform the quantum state tomography simulations[100].

Customized OpenAI gym environment. We use a customized OpenAI gym environment[101] to verify the performance of our proposed algorithm. In this experimental environment, the objective quantum state, the fidelity threshold, and the quantum computation backend (real machine or simulator software) are set by the user in the form of parameters. In addition, users can also customize the noise mode. Specifically, we use the following parameter settings to build the test environment:

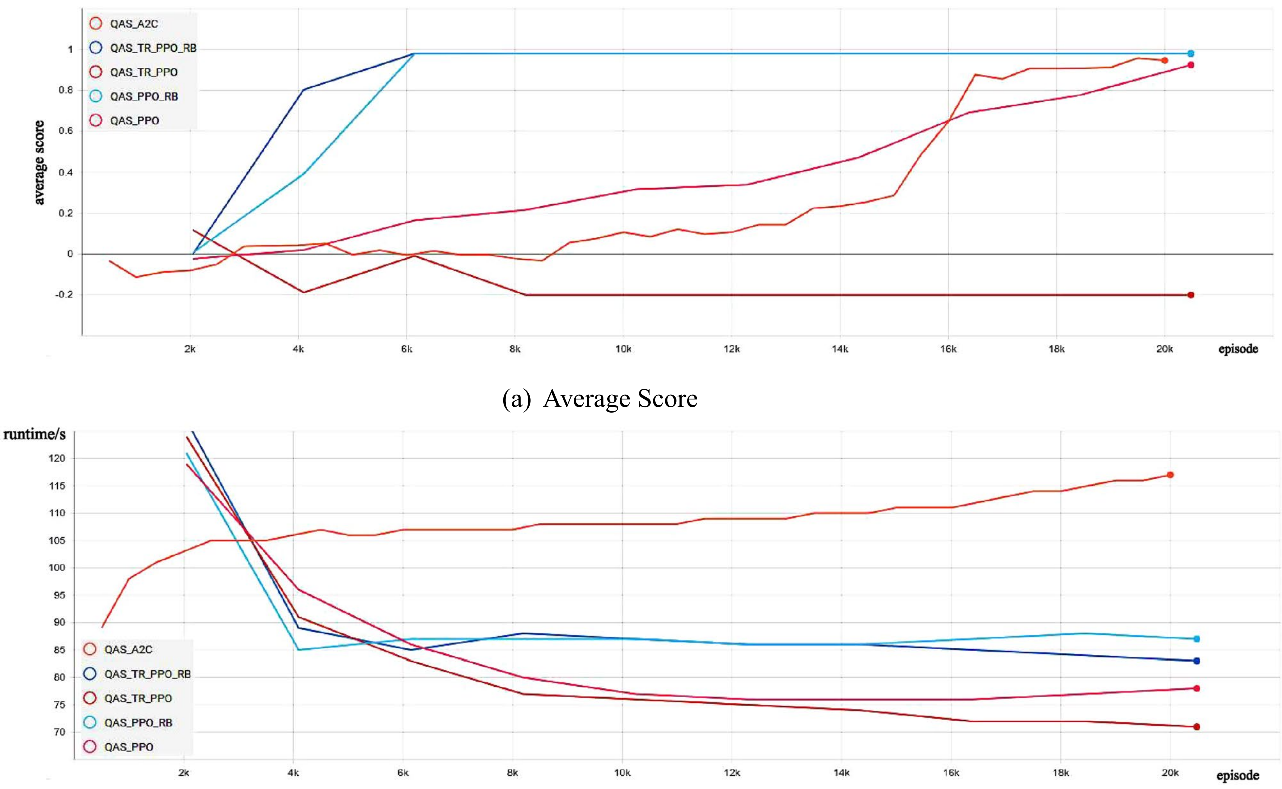
- Observation: The agent receives Pauli-X, Y, Z expected values on each qubit.
- Action: The RL agent will choose a quantum gate that runs on a specific qubit.
- Reward: Before successfully reaching the goal, the agent will receive a -0.01 reward for every step to encourage exploring the shortest path. When the agent reaches the goal, it will obtain a reward of F.

Parameter settings. In this paper, we think about five RL algorithms in this paper, their parameter setting are shown as follow:

- A2C: learning rate $\eta = 10^{-4}$, discount factor $\gamma = 0.99$.
- PPO: $\eta = 0.002$, $\gamma = 0.99$, clip range parameter $C = 0.2$, update epoch number $K = 4$.
- PPO-RB: $\eta = 0.002$, $\gamma = 0.99$, clip range parameter $C = 0.2$, update epoch number $K = 4$, slope rollback $\alpha = -0.3$.
- TR-PPO: $\eta = 0.002$, $\gamma = 0.99$, clip range parameter $C = 0.2$, update epoch number $K = 4$, klrange $\delta = 0.03$.
- TR-PPO-RB: $\eta = 0.002$, $\gamma = 0.99$, clip range parameter $C = 0.2$, update epoch number $K = 4$, klrange $\delta = 0.03$, slope rollback $\alpha = -0.3$.

Noise-free environments performance. *2-Qubit Bell state.* Firstly, we show the experimental comparison results of different deep reinforcement learning-based QAS methods generating the 2-qubit Bell state from scratch in a noise-free environment (as shown in Fig. 1). We can find that these deep reinforcement learning-based QAS methods can successfully train RL agents to synthesize Bell states, however, under the same neural network, our proposed algorithm obtains better policy performance and less running time than other methods. Figure 2 shows the Bell state quantum circuit generated by our proposed method on a noise-free two-qubit system.

3-Qubit GHZ state. Secondly, we show the experimental comparison results of different deep reinforcement learning-based QAS methods generating the 3-qubit GHZ state from scratch in a noise-free environment (as



(a) Average Score

(b) Runtime

Figure 1. Comparison of the average score and the runtime with different deep reinforcement learning-based QAS methods for Quantum Architecture Search on noise-free Two-Qubit system.

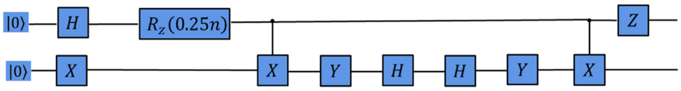


Figure 2. Quantum circuit for the Bell state generated by the RL agent on noise-free Two-Qubit system.

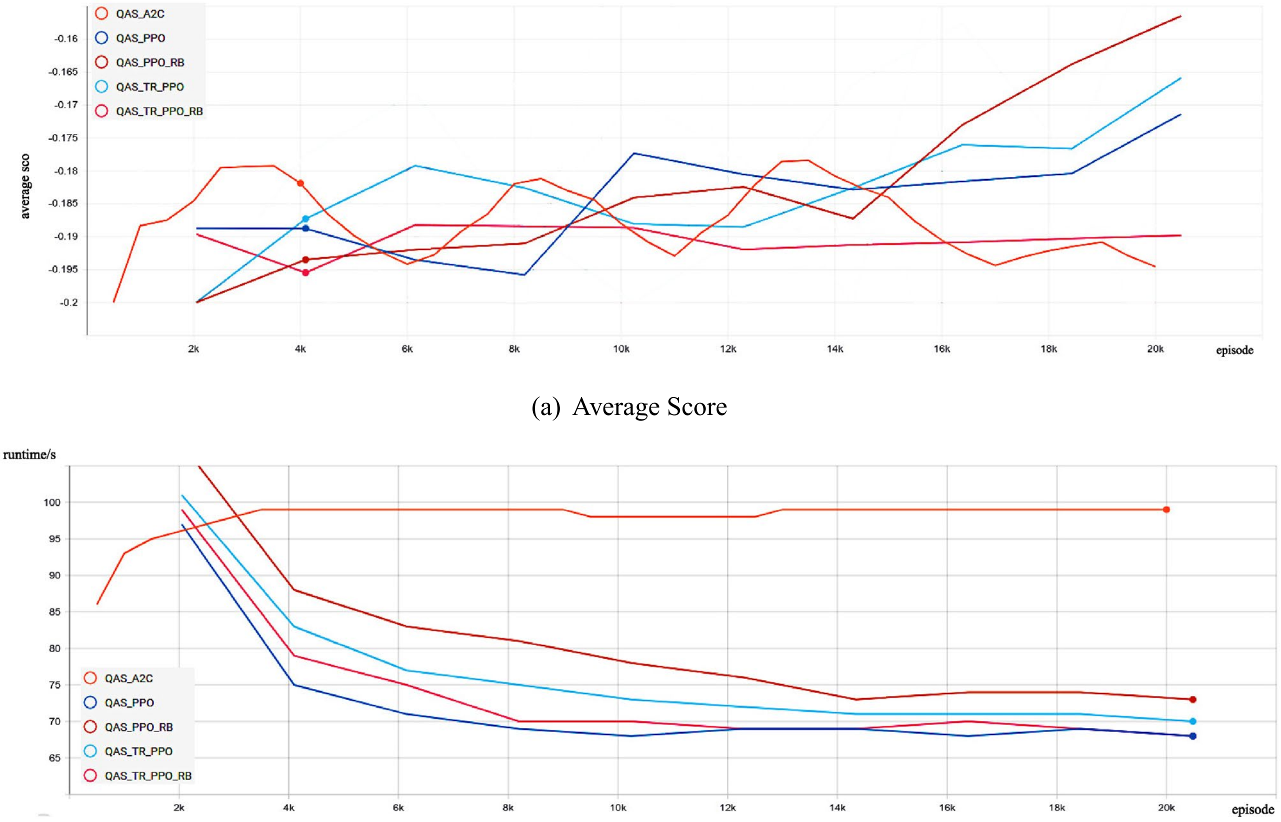
shown in Fig. 3). We can find that these deep reinforcement learning-based QAS methods can successfully train RL agents to synthesize GHZ states, however, under the same neural network, our proposed method reaches optimal policy performance faster and the algorithm running time is less compared to other methods. In Fig. 4, we provide the quantum circuit for GHZ state generated by our proposed method on a noise-free three-qubit system.

4-Qubit SK Ising spin glass state. Thirdly, we focus on a classical problem in combination optimization, namely, the SK Ising spin glass with the energy function

$$C = \frac{1}{\sqrt{n}} \sum_{i,j=1}^{n} J_{ij}\sigma_i^z\sigma_j^z + \sum_{i=1}^{n} h_i\sigma_i^z, \tag{20}$$

where J_{ij} and h_i represent independent Gaussian stochastic variables with zero-mean and zero-variance $J^2 = h^2 = 1$, and each σ^z spin can take the values ±1. We use the Metropolis algorithm to calculate the ground state of the Hamiltonian system of the SK Ising Spin Glass model as the target quantum state.

We show the experimental comparison results of different deep reinforcement learning-based QAS methods generating the 4-qubit SK Ising spin galss state from scratch in a noise-free environment (as shown in Fig. 5). We can observe that these deep reinforcement learning-based QAS methods can successfully train RL agents to synthesize SK Ising spin galss states, however, under the same neural network, our proposed algorithm obtains better policy performance and less running time than other methods. Figure 6 shows the SK Ising spin galss state quantum circuit generated by our proposed method on a noise-free four-qubit system.



(a) Average Score

(b) Runtime

Figure 3. Comparison of the average score and the runtime with different deep reinforcement learning-based QAS methods on noise-free Three-Qubit system.

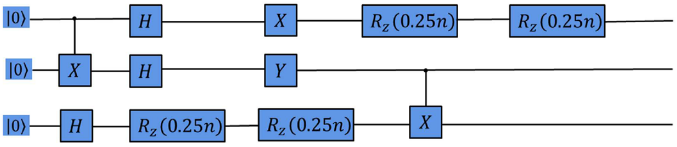


Figure 4. Quantum circuit for the GHZ state generated by the RL agent on noise-free Three-Qubit system.

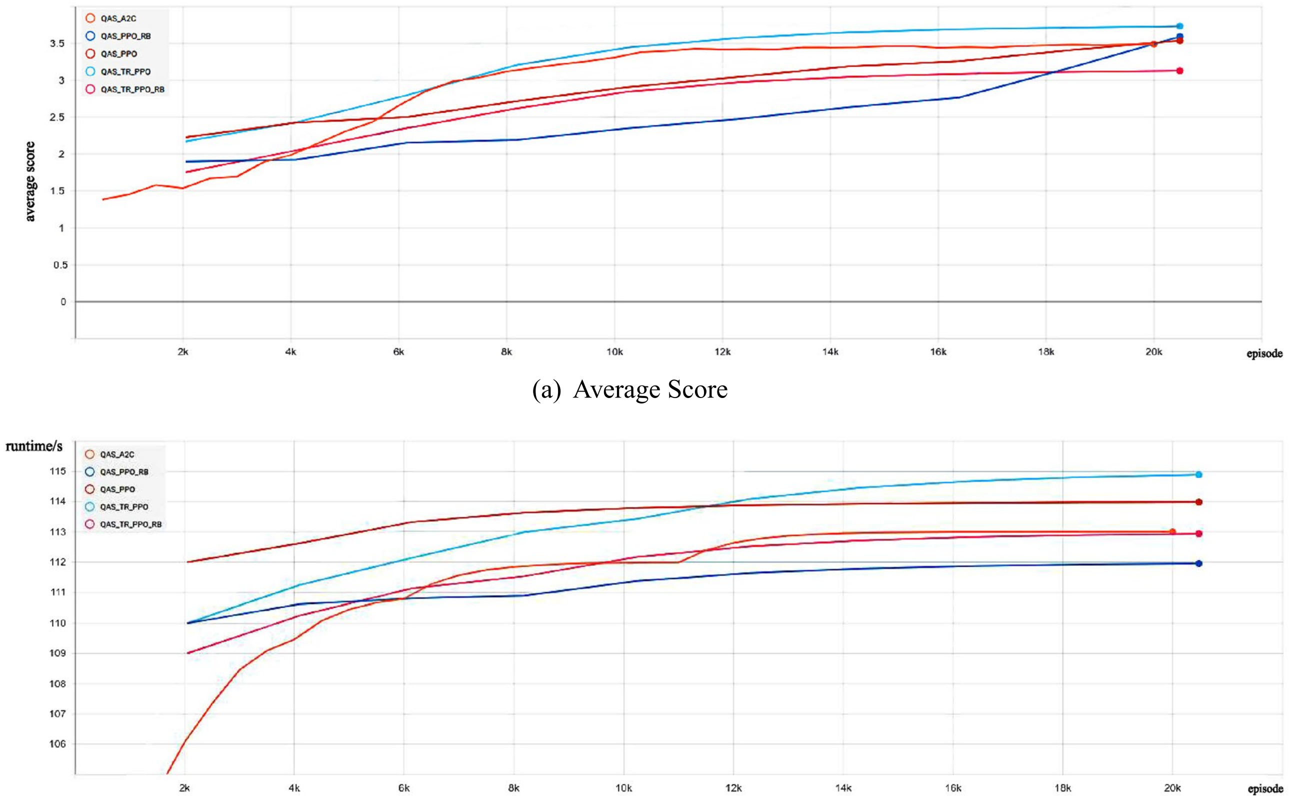
Noisy environments performance.

2-Qubit Bell state. Fourthly, we show the experimental comparison results of different deep reinforcement learning-based QAS methods generating the 2-qubit Bell state from scratch in a noisy simulation environment (as shown in Fig. 7). We can observe that these deep reinforcement learning-based QAS methods can successfully train RL agents to synthesize Bell states, however, under the same neural network, our proposed algorithm obtains better policy performance and less running time than other methods. Figure 8 shows the Bell state quantum circuit generated by our proposed method on a noisy simulation two-qubit system.

As shown in Figs. 1 and 7, for the trust region-based clipping methods (QAS-TR-PPO and QAS-TR-PPO-RB), the KL divergences are also smaller than those of QAS-PPO. Especially, QAS-TR-PPO shows the enhanced restriction ability on the KL divergence even it does not incorporate the rollback mechanism. Furthermore, the proportions of out-of-range probability ratios of QAS-TR-PPO-RB are much less than those of the original QAS-PPO during the training process. The probability ratios and the KL divergences of QTR-TR-PPO-RB are also much smaller than those of QAS-PPO. The "rollback" operation on the KL divergence can be regarded as a penalty (regularization) term:

$$L_t^{penalty}(\theta) = q_t(\theta)A_t - \alpha D_{KL}^s(\theta_{old}, \theta). \tag{21}$$

The penalty-based methods are usually notorious by the difficulty of adjusting the trade-off coefficient. And PPO-penalty addresses this issue by adaptively adjusting the rollback coefficient α to achieve a target value of the KL divergence. However, the penalty-based PPO does not perform well as the clipping-based one, as it is difficult



(a) Average Score

(b) Runtime

Figure 5. Comparison of the average score and the runtime with different deep reinforcement learning-based QAS methods on noise-free Four-Qubit system.

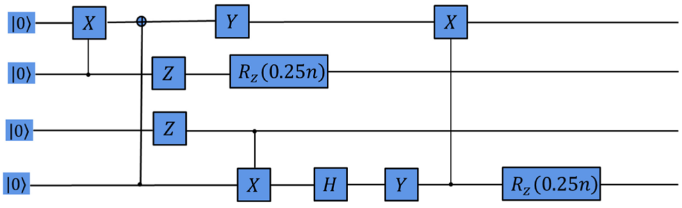


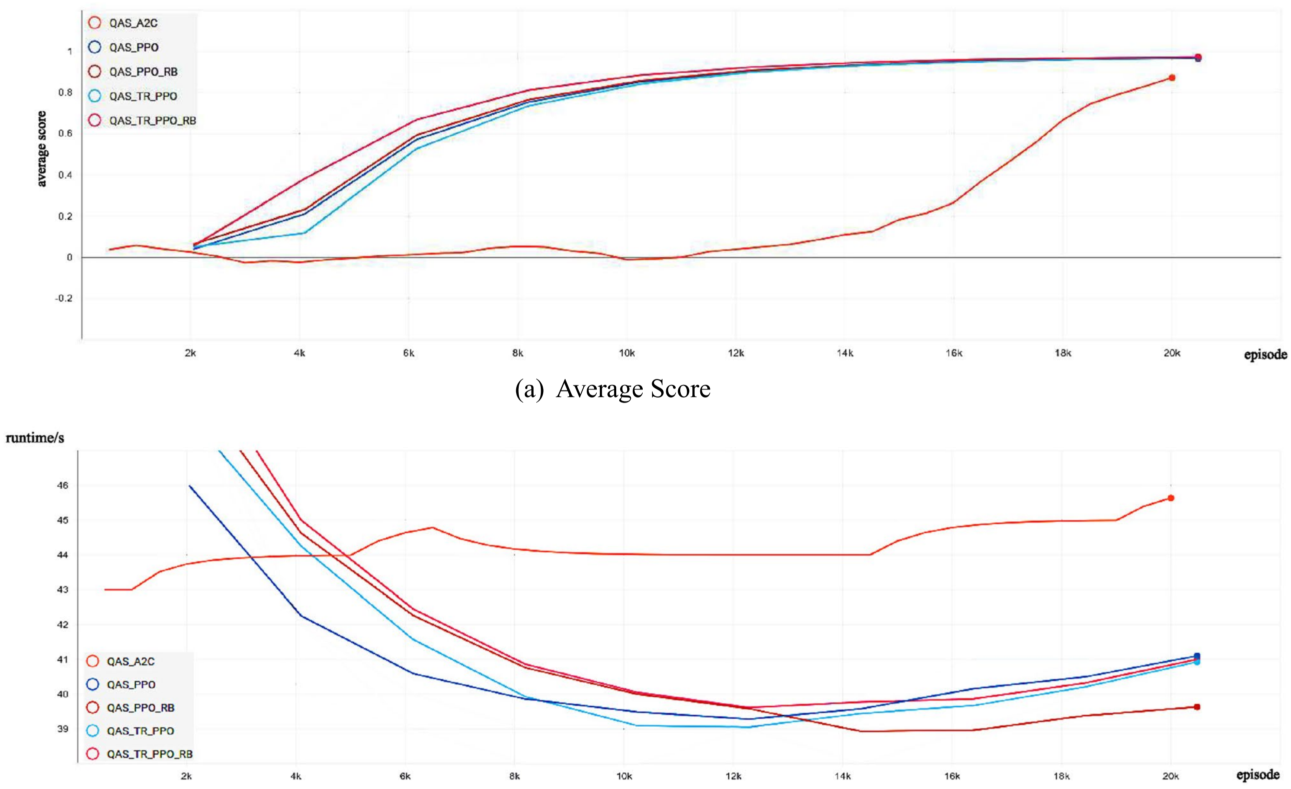
Figure 6. Quantum circuit for the 4-qubit SK Ising spin galss state generated by the RL agent on noise-free Four-Qubit system.

to find an effective coefficient-adjustig strategy across different tasks. Our method introduces the "clipping" strategy to assisst in restricting policy, i.e., the penalty is enforced only when the policy is out of the trust region. As for when the policy is inside the trust region, the objective function is not affected by the penalty term. Such a mechanism could relieve the difficulty on adjusting the trade-off coefficient, and it will not alter the theoretical property of monotonic improvement (as we will show below). In practice, we found QAS-TR-PPO-RB to be more robust to the coefficient and achieve better performance across different tasks. The clipping technique may be served as an effective method to enforce the restriction, which enjoys low optimization complexity and seems to be more robust.

To analyse the monotonic improvement property, we use the maximum KL divergence instead, i.e.,

$$L_{policy}^{TR\_RB}(\theta) = q_t(\theta)A_t - \begin{cases} \alpha \max_{s_{t+1}\in S} D_{KL}^{s_{t+1}}(\theta_{old},\theta), & \text{if } \max_{s_{t+1}\in S} D_{KL}^{s_{t+1}}(\theta_{old},\theta) \geq \delta \\ & and\, q_t(\theta)A_t \geq q_t(\theta_{old})A_t \\ \delta & \text{else} . \end{cases} \tag{22}$$

in which the maximum KL divergence is also used in TRPO for theoretical analysis. Such objective function also possesses the theoeretical property of the guaranteed monotonic improvement. Let $\theta_{new}^{TR\_RB} = \arg\max_\theta L_{policy}^{TR\_RB}(\theta)$ and $\theta_{new}^{TRPO} = \arg\max_\theta M(\theta)$ denote the optimal solution of QAS-TR-PPO-RB and TRPO respectively. We have the follow theorem.



(a) Average Score

(b) Runtime

Figure 7. Comparison of the average score and the runtime with different deep reinforcement learning-based QAS methods on noisy Two-Qubit system.

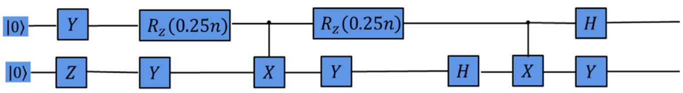


Figure 8. Quantum circuit for the Bell state generated by the RL agent on noisy Two-Qubit system.

Theorem 1 *If* $\alpha = C \triangleq max_t|A_t|4\gamma/(1-\gamma)^2$ *and* $\delta \le \max_{s_t \in S} D_{KL}^{s_t}(\theta_{old}, \theta_{new}^{TRPO})$, *then* $\zeta(\theta_{new}^{TR\_RB}) \ge \zeta(\theta_{old})$, *where* $\zeta(\theta) = E_{s^t,a^t}[r(s^t, a^t)]$.

Proof Firstly, we prove two properties of θ_{new}^{TRPO}.

Note that $M(\theta) = E_t[q_t(\theta)A_t] - \alpha \max_{s_{t+1} \in S} D_{KL}^{s_{t+1}}(\theta_{old}, \theta)$. As θ_{new}^{TRPO} is the optimal solution of $M(\theta)$, we have

$$E_a[q_{s^t,a}(\theta_{new}^{TRPO})A_{s^t,a}] \ge E_a[q_{s^t,a}(\theta_{old})A_{s^t,a}], \forall s_t \tag{23}$$

Assume θ_{new} is an optimal solution of $M(\theta)$ and there exists some s^{t+1} such that $E_a[q_{s^{t+1},a}(\theta')A_{s^{t+1},a}] \ge E_a[q_{s^{t+1},a}(\theta_{old})A_{s^{t+1},a}]$, then we can construst a new policy

$$\pi_{\theta''}(\cdot|s_t) = \begin{cases} \pi_{\theta_{old}}(\cdot|s_t), & \text{if } E_a[q_{s^{t+1},a}(\theta')A_{s^{t+1},a}] \ge E_a[q_{s^{t+1},a}(\theta_{old})A_{s^{t+1},a}] \\ \pi_{\theta'}(\cdot|s_t) & \text{else} \end{cases} \tag{24}$$

We have $M(\pi_{\theta'}) \le M(\pi_{\theta''})$, which contradicts that $\pi_{\theta'}(\cdot|s_t)$ is an optimal policy.

Besides, by Eq. (23), we can also obtain that for any s_t there exists at least one a' such that $q_{s^t,a'}(\theta)A_{s^t,a'} \ge q_{s^t,a'}(\theta_{old})A_{s^t,a'}$. Therefore, by Eq. (22), we have

$$L_{policy}^{TR\_RB}(\theta_{new}^{TRPO}) + \zeta(\theta_{old}) = L_{policy}(\theta_{new}^{TRPO}) - \alpha \max_{s_t \in S} D_{KL}^{s_t}(\theta_{old}, \theta_{new}^{TRPO}). \tag{25}$$

Then, we prove that θ_{new}^{TRPO} is the optimal solution of $L_{policy}^{TR\_RB}$. There are three cases.

- For θ' which satisfies $\max_{s_t \in S} D_{KL}^{s_t}(\theta_{old}, \theta') \ge \delta$ and there exist some a' such that $q_{s^t,a'}(\theta')A_{s^t,a'} \ge q_{s^t,a'}(\theta_{old})A_{s^t,a'}$ for any s^t, we have

$$\begin{aligned} L_{policy}^{TR\_RB}(\theta') + \zeta(\theta_{old}) &= L_{policy}(\theta') - \alpha \max_{s_t \in S} D_{KL}^{s_t}(\theta_{old}, \theta') \\ &\leq L_{policy}(\theta_{new}^{TRPO}) - \alpha \max_{s_t \in S} D_{KL}^{s_t}(\theta_{old}, \theta_{new}^{TRPO}) \\ &= L_{policy}^{TR\_RB}(\theta_{new}^{TRPO}) + \zeta(\theta_{old}). \end{aligned} \tag{26}$$

- For θ' which satisfies $\max_{s_t \in S} D_{KL}^{s_t}(\theta_{old}, \theta') \geq \delta$, we have

$$\begin{aligned} L_{policy}^{TR\_RB}(\theta') + \zeta(\theta_{old}) &= L_{policy}(\theta') - \alpha\delta \\ &\leq L_{policy}(\theta') - \alpha \max_{s_t \in S} D_{KL}^{s_t}(\theta_{old}, \theta') \\ &\leq L_{policy}(\theta_{new}^{TRPO}) - \alpha \max_{s_t \in S} D_{KL}^{s_t}(\theta_{old}, \theta_{new}^{TRPO}) \\ &= L_{policy}^{TR\_RB}(\theta_{new}^{TRPO}) + \zeta(\theta_{old}). \end{aligned} \tag{27}$$

- We now prove the case of θ' which satisfies $\max_{s_t \in S} D_{KL}^{s_t}(\theta_{old}, \theta') \geq \delta$ and there exist some s^t such that $q_{s^t,a'}(\theta')A_{s^t,a'} < q_{s^{t+1},a}(\theta_{old})A_{s^{t+1},a}$ for any a, we have

$$\begin{aligned} E_a[L_{policy,s^t,a}^{TR\_RB}(\theta')] &= E_a[q_{s^{t+1},a}(\theta')] - \alpha\delta \\ &< E_a[q_{s^{t+1},a}(\theta_{old})] - \alpha\delta \\ &\leq E_a[q_{s^{t+1},a}(\theta_{new}^{TRPO}) - \alpha \max_{s_{t+1} \in S} D_{KL}^{s_{t+1}}(\theta_{old}, \theta_{new}^{TRPO}) \\ &= E_a[L_{policy,s^{t+1},a}^{TR\_RB}(\theta_{new}^{TRPO})]. \end{aligned} \tag{28}$$

We can construst a new policy

$$\pi_{\theta''}(\cdot|s_t) = \begin{cases} \pi_{\theta_{new}^{TRPO}}(\cdot|s_t), & \text{if } s \in \{s^{t+1}\} \\ \pi_{\theta'}(\cdot|s_t) & \text{else} \end{cases} \tag{29}$$

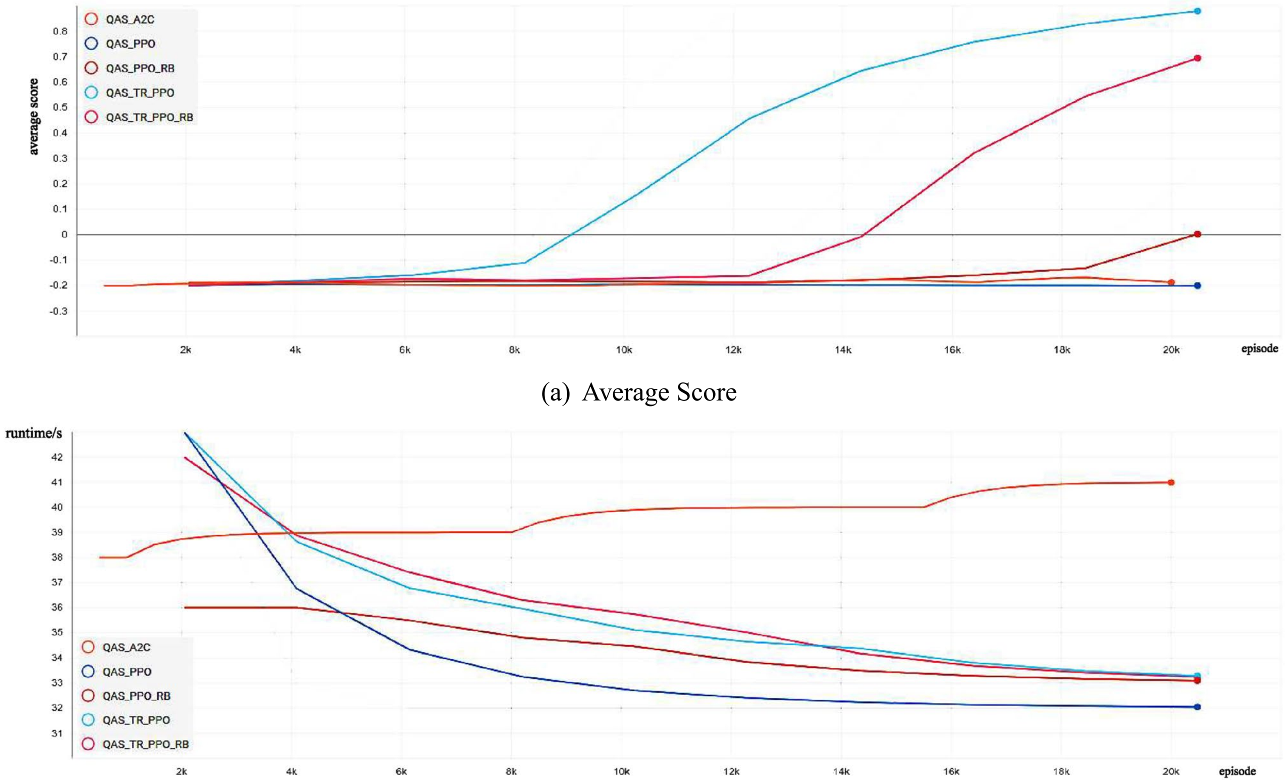
for which we have

$$\begin{aligned} L_{policy}^{TR\_RB}(\theta') + \zeta(\theta_{old}) &= E_{s,a}[L_{policy,s^t,a}^{TR\_RB}(\theta')] + \zeta(\theta_{old}) \\ &<_{s,a} [L_{policy,s^t,a}^{TR\_RB}(\theta'')] + \zeta(\theta_{old}) \\ = L_{policy}(\theta'') - \alpha \max_{s_t \in S} D_{KL}^{s_t}(\theta_{old}, \theta'') &= E_{s,a}[L_{policy,s^t,a}^{TR\_RB}(\theta'')] + \zeta(\theta_{old}) \\ &\leq M(\theta_{new}^{TRPO}) \\ &= L_{policy}^{TR\_RB}(\theta_{new}^{TRPO}) + \zeta(\theta_{old}). \end{aligned} \tag{30}$$

In summary, we have $\zeta(\theta_{new}^{TR\_RB}) = \zeta(\theta_{new}^{TRPO}) \geq M(\theta_{new}^{TRPO})M(\theta_{old}) = \zeta(\theta_{old})$ □

3-Qubit GHZ state. Then, we show the experimental comparison results of different deep reinforcement learning-based QAS methods generating the 3-qubit GHZ state from scratch in a noisy simulation environment. As shown in Fig. 9, we observe that, give the same neural network architeure, our method performs signficantly better than original deep reinforcement learning-based QAS methods in terms of the runtime and the policy performance. Figure 10 shows the GHZ state quantum circuit generated by our proposed method on a noisy simulation three-qubit system.

4-Qubit SK Ising spin glass state. Finally, we show the experimental comparison results of different deep reinforcement learning-based QAS methods generating the 4-qubit SK Ising spin galss state from scratch in a noisy environment (as shown in Fig. 11). We can observe that although these deep reinforcement learning-based QAS algorithms can successfully train RL agents to synthesize SK Ising spin galss states, under the same neural network, our proposed approach obtains better policy performance and less running time than other methods. Fig. 12 shows the SK Ising spin galss state quantum circuit generated by our proposed method on a noisy four-qubit system.



(a) Average Score

(b) Runtime

Figure 9. Comparison of the average score and the runtime with different deep reinforcement learning-based QAS methods on noisy Three-Qubit system.

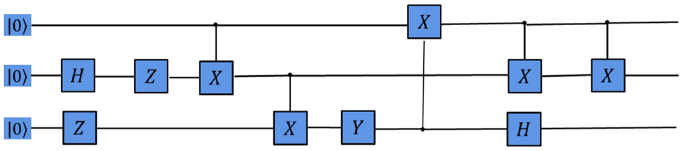
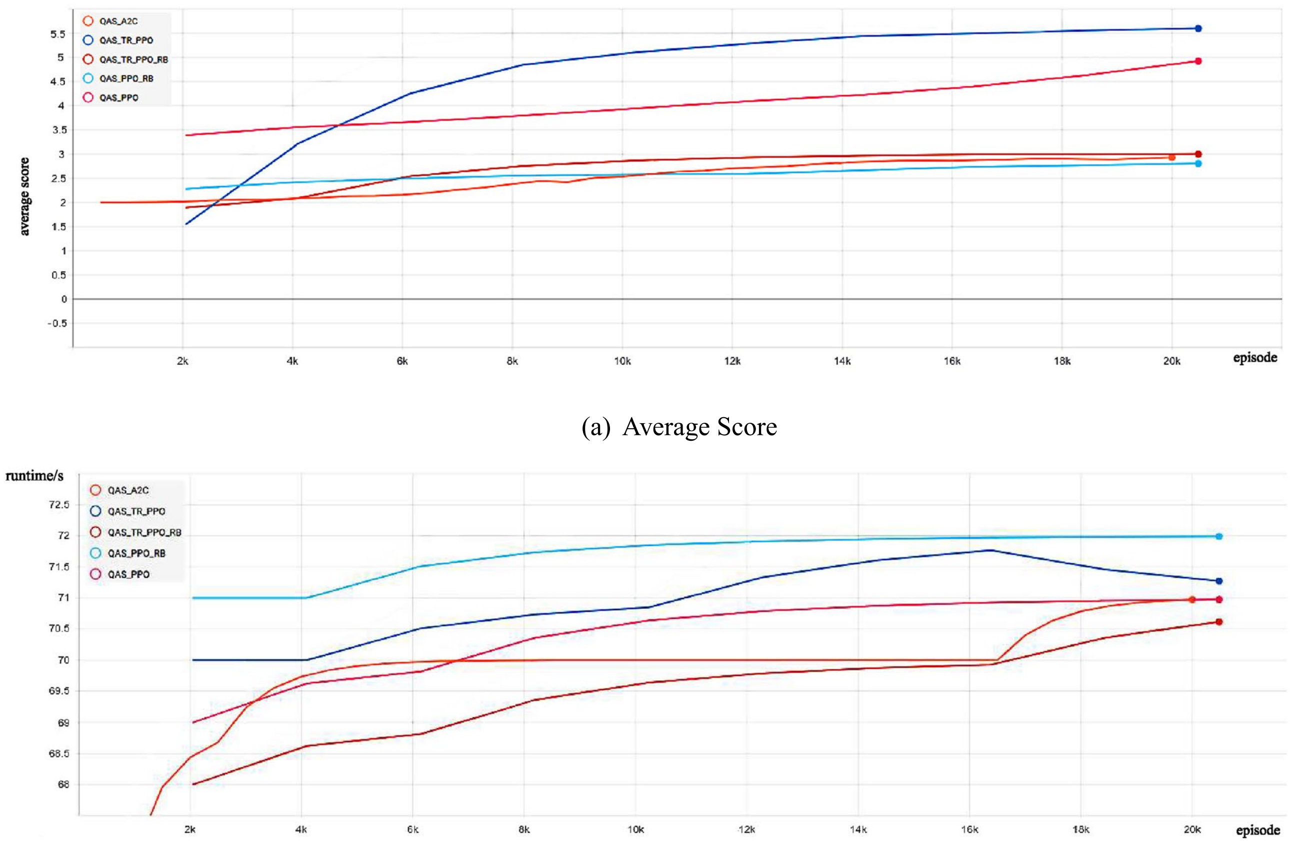


Figure 10. Quantum circuit for the GHZ state generated by the RL agent on noisy Three-Qubit system.

Conclusion

In this paper, we present a new deep reinforcement learning-based QAS approach, named Trust Region-based PPO with Rollback for QAS (QAS-TR-PPO-RB), to automatically build the quantum gates sequence from the density matrix only. Specifically, inspired by the research work of Wang, we adopt an improved clipping function to implement the rollback behavior to limit the probability ratio between the new strategy and the old strategy. Moreover, we optimize the strategy within the trust region by replacing the clipped trigger conditions with those based on the trust region to guarantee monotonic improvement. In this way, our method can improve the original deep reinforcement learning-based QAS methods on policy performance and algorithm running time.



(a) Average Score

(b) Runtime

Figure 11. Comparison of the average score and the runtime with different deep reinforcement learning-based QAS methods on noisy Four-Qubit system.

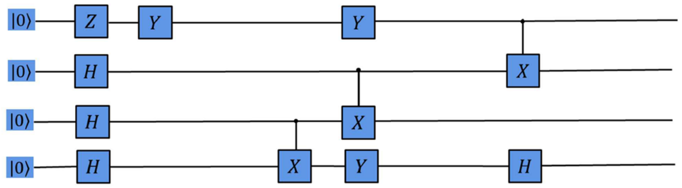


Figure 12. Quantum circuit for the 4-qubit SK Ising spin galss state generated by the RL agent on noisy Four-Qubit system.

Data availability

The datasets used during the current study are available in the Qiskit and Stable-Baselines3 repositories, https://github.com/Qiskit/qiskit and https://stable-baselines3.readthedocs.io/en/master/, respectively.

Received: 23 November 2022; Accepted: 26 March 2023
Published online: 29 March 2023

References

1. Sutton, R. S., Barto, A. G. *Reinforcement Learning: An Introduction*. (MIT Press, 2018).
2. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**(7540), 529–533 (2015).
3. Schrittwieser, J. *et al.* Mastering atari, go, chess and shogi by planning with a learned model. *Nature* **588**(7839), 604–609 (2020).
4. Puigdomènech Badia, A., Piot, B., Kapturowski, S., Sprechmann, P., Vitvitskyi, A., Daniel Guo, Z., Blundell, C. Agent57: Outperforming the atari human benchmark. In *Proceedings of the 37th International Conference on Machine Learning, 13–18 July, Virtual Event*, vol. 119, pp. 507–517 (PMLR, 2020).
5. Kapturowski, S., Ostrovski, G., Quan, J., Munos, R., Dabney, W. Recurrent experience replay in distributed reinforcement learning. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6–9*. (OpenReview.net, 2019).
6. Silver, D. *et al.* Mastering the game of go with deep neural networks and tree search. *Nature* **529**(7587), 484–489 (2016).
7. Silver, D. *et al.* Mastering the game of go without human knowledge. *Nature* **550**(7676), 354–359 (2017).
8. Bukov, M. *et al.* Reinforcement learning in different phases of quantum control. *Phys. Rev. X* **8**(3), 031086 (2018).

9. Fösel, T., Tighineanu, P., Weiss, T. & Marquardt, F. Reinforcement learning with neural networks for quantum feedback. *Phys. Rev. X* **8**(3), 031084 (2018).
10. Niu, M. Y., Boixo, S., Smelyanskiy, V. N. & Neven, H. Universal quantum control through deep reinforcement learning. *NPJ Quantum Inf.* **5**(1), 1–8 (2019).
11. An, Z. & Zhou, D. L. Deep reinforcement learning for quantum gate control. *EPL (Europhys. Lett.)* **126**(6), 60002 (2019).
12. Zhang, X.-M., Wei, Z., Asad, R., Yang, X.-C. & Wang, X. When does reinforcement learning stand out in quantum control? a comparative study on state preparation. *NPJ Quantum Inf.* **5**(1), 1–7 (2019).
13. Palittapongarnpim, P., Wittek, P., Zahedinejad, E., Vedaie, S. & Sanders, B. C. High-dimensional global optimization for noisy quantum dynamics. Learning in quantum control. *Neurocomputing* **268**, 116–126 (2017).
14. Xu, H. *et al.* Generalizable control for quantum parameter estimation through reinforcement learning. *NPJ Quantum Inf.* **5**(1), 1–8 (2019).
15. Andreasson, P., Johansson, J., Liljestrand, S. & Granath, M. Quantum error correction for the toric code using deep reinforcement learning. *Quantum* **3**, 183 (2019).
16. Fitzek, D., Eliasson, M., Kockum, A. F. & Granath, M. Deep q-learning decoder for depolarizing noise on the toric code. *Phys. Rev. Res.* **2**(2), 023230 (2020).
17. Nautrup, H. P., Delfosse, N., Dunjko, V., Briegel, H. J. & Friis, N. Optimizing quantum error correction codes with reinforcement learning. *Quantum* **3**, 215 (2019).
18. Colomer, L. D., Skotiniotis, M. & Muñoz-Tapia, R. Reinforcement learning for optimal error correction of toric codes. *Phys. Lett. A* **384**(17), 126353 (2020).
19. Wilson, M. *et al.* Optimizing quantum heuristics with meta-learning. *Quantum Mach. Intell.* **3**(1), 1–14 (2021).
20. Verdon, G., Broughton, M., McClean, J. R., Sung, K. J., Babbush, R., Jiang, Z., Neven, H., Mohseni, M. Learning to learn with quantum neural networks via classical neural networks. arXiv preprint arXiv:1907.05415 (2019).
21. Wauters, M. M., Panizon, E., Mbeng, G. B. & Santoro, G. E. Reinforcement-learning-assisted quantum optimization. *Phys. Rev. Res.* **2**(3), 033446 (2020).
22. Yao, J., Bukov, M., Lin, L. Policy gradient based quantum approximate optimization algorithm. In *Mathematical and Scientific Machine Learning*, 605–634. (PMLR, 2020).
23. Zoph, B., Le, Q. V. Neural architecture search with reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, Conference Track Proceedings*. (OpenReview.net, 2017).
24. Baker, B., Gupta, O., Naik, N., Raskar, R. Designing neural network architectures using reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24–26, Conference Track Proceedings*. (OpenReview.net, 2017).
25. Cai, H., Chen, T., Zhang, W., Yu, Y., Wang, J. Efficient architecture search by network transformation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2–7*, 2787–2794. (AAAI Press, 2018).
26. Zoph, B., Vasudevan, V., Shlens, J., Le, Q. V. Learning transferable architectures for scalable image recognition. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 18–22*, 8697–8710. (IEEE Computer Society, 2018).
27. Zhong, Z., Yan, J., Wu, W., Shao, J., Liu, C.-L. Practical block-wise neural network architecture generation. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, June 18–22*, 2423–2432. (IEEE Computer Society, 2018).
28. Schrimpf, M., Merity, S., Bradbury, J., Socher, R. A flexible approach to automated RNN architecture generation. In *6th International Conference on Learning Representations, Vancouver, BC, Canada, April 30–May 3, Workshop Track Proceedings*. (OpenReview.net, 2018).
29. Pham, H. , Guan, M. Y., Zoph, B., Le, Q. V., Dean, J. Efficient neural architecture search via parameter sharing. In *Proceedings of the 35th International Conference on Machine Learning, Stockholmsmässan, Stockholm, Sweden, July 10–15*, vol. 80, 4092–4101. (PMLR, 2018).
30. Cai, H., Yang, J., Zhang, W., Han, S., Yu, Y. Path-level network transformation for efficient architecture search. In *Proceedings of the 35th International Conference on Machine Learning, Stockholmsmässan, Stockholm, Sweden, July 10–15*, vol. 80, 677–686. (PMLR, 2018).
31. Elsken, T., Metzen, J. H. & Hutter, F. Neural architecture search: A survey. *J. Mach. Learn. Res.* **20**(1), 1997–2017 (2019).
32. Harrow, A. W. & Montanaro, A. Quantum computational supremacy. *Nature* **549**(7671), 203–209 (2017).
33. Arute, F. *et al.* Quantum supremacy using a programmable superconducting processor. *Nature* **574**(7779), 505–510 (2019).
34. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **41**(2), 303–332 (1999).
35. Grover, L. K. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.* **79**(2), 325 (1997).
36. Peruzzo, A. *et al.* A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**(1), 1–7 (2014).
37. Farhi, E., Goldstone, J., Gutmann, S. A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014).
38. Zhou, L., Wang, S.-T., Choi, S., Pichler, H. & Lukin, M. D. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Phys. Rev. X* **10**(2), 021067 (2020).
39. Chen, S. Y.-C., Yoo, S., Fang, Y.-L.L. *Quantum Long Short-Term Memory* (Bulletin of the American Physical Society, 2021).
40. Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev. A* **98**(3), 032309 (2018).
41. Kyriienko, O., Paine, A. E. & Elfving, V. E. Solving nonlinear differential equations with differentiable quantum circuits. *Phys. Rev. A* **103**(5), 052416 (2021).
42. Schuld, M., Bocharov, A., Svore, K. M. & Wiebe, N. Circuit-centric quantum classifiers. *Phys. Rev. A* **101**(3), 032308 (2020).
43. Havlíček, V. *et al.* Supervised learning with quantum-enhanced feature spaces. *Nature* **567**(7747), 209–212 (2019).
44. Farhi, E., Neven, H. Classification with quantum neural networks on near term processors. arXiv preprint arXiv:1802.06002 (2018).
45. Benedetti, M., Lloyd, E., Sack, S. & Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **4**(4), 043001 (2019).
46. Mari, A., Bromley, T. R., Izaac, J., Schuld, M. & Killoran, N. Transfer learning in hybrid classical-quantum neural networks. *Quantum* **4**, 340 (2020).
47. Abohashima, Z., Elhosen, M., Houssein, E.H., Mohamed, W.M. Classification with quantum machine learning: A survey. arXiv preprint arXiv:2006.12270 (2020).
48. Easom-McCaldin, P., Bouridane, A., Belatreche, A., Jiang, R. Towards building a facial identification system using quantum machine learning techniques. arXiv preprint arXiv:2008.12616 (2020).
49. Sarma, A., Chatterjee, R., Gili, K., Yu, T. Quantum unsupervised and supervised learning on superconducting processors. arXiv preprint arXiv:1909.04226 (2019).
50. Stein, S. A., Baheri, B., Tischio, R. M., Chen, Y., Mao, Y., Guan, Q., Li, A., Fang, B. A hybrid system for learning classical data in quantum states. arXiv preprint arXiv:2012.00256 (2020).
51. Yen-Chi Chen, S., Huang, C.-M., Hsing, C.-W., Kao, Y.-J. Hybrid quantum-classical classifier based on tensor network and variational quantum circuit. arXiv preprint arXiv:2011.14651 (2020).

52. Yen-Chi Chen, S., Wei, T.-C., Zhang, C., Yu, H., Yoo, S. Quantum convolutional neural networks for high energy physics data analysis. arXiv preprint arXiv:2012.12177 (2020).
53. Wu, S.L., Chan, J., Guan, W., Sun, S., Wang, A., Zhou, C., Livny, M., Carminati, F., Di Meglio, A., Li, A.C.Y., *et al.* Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the lhc on ibm quantum computer simulator and hardware with 10 qubits. *J. Phys. G Nucl. Part. Phys.* (2021).
54. Yen-Chi Chen, S., Wei, T.-C., Zhang, C., Yu, H., Yoo, S. Hybrid quantum-classical graph convolutional network. arXiv preprint arXiv:2101.06189 (2021).
55. Stein, S. A., Baheri, B., Chen, D., Mao, Y., Guan, Q., Li, A., Xu, S., Ding, C. Quclassi: A hybrid deep neural network architecture based on quantum state fidelity. arXiv preprint arXiv:2103.11307 (2021).
56. Jaderberg, B., Anderson, L. W., Xie, W., Albanie, S., Kiffner, M., Jaksch, D. Quantum self-supervised learning. arXiv preprint arXiv:2103.14653 (2021).
57. Dallaire-Demers, P.-L. & Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A* **98**(1), 012324 (2018).
58. Lloyd, S. & Weedbrook, C. Quantum generative adversarial learning. *Phys. Rev. Lett.* **121**(4), 040502 (2018).
59. Stein, S. A., Baheri, B., Tischio, R. M., Mao, Y., Guan, Q., Li, A., Fang, B., Xu, S. Qugan: A generative adversarial network through quantum states. arXiv preprint arXiv:2010.09036 (2020).
60. Zoufal, C., Lucchi, A. & Woerner, S. Quantum generative adversarial networks for learning and loading random distributions. *NPJ Quantum Inf.* **5**(1), 1–9 (2019).
61. Situ, H., He, Z., Wang, Y., Li, L. & Zheng, S. Quantum generative adversarial network for generating discrete distribution. *Inf. Sci.* **538**, 193–208 (2020).
62. Nakaji, K. & Yamamoto, N. Quantum semi-supervised generative adversarial network for enhanced data classification. *Sci. Rep.* **11**(1), 1–10 (2021).
63. Yen-Chi Chen, S., Huck Yang, C.-H., Qi, J., Chen, P.-Y., Ma, X., Goan, H.-S. Variational quantum circuits for deep reinforcement learning. *IEEE Access***8**, 141007–141024 (2020).
64. Lockwood, O. & Si, M. Reinforcement learning with quantum variational circuit. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, vol. **16**, 245–251 (2020).
65. Jerbi, S., Trenkwalder, L. M., Nautrup, H. P., Briegel, H. J. & Dunjko, V. Quantum enhancements for deep reinforcement learning in large spaces. *PRX Quantum* **2**(1), 010328 (2021).
66. Chen, C.C., Shiba, K., Sogabe, M., Sakamoto, K., Sogabe, T. Hybrid quantum-classical ulam-von neumann linear solver-based quantum dynamic programing algorithm. In *Proc. Annu. Conf. JSAI, page 2K6ES203* (2020).
67. Wu, S., Jin, S., Wen, D., Wang, X. Quantum reinforcement learning in continuous action space. arXiv preprint arXiv:2012.10711 (2020).
68. Skolik, A., Jerbi, S., Dunjko, V. Quantum agents in the gym: A variational quantum algorithm for deep q-learning. arXiv preprint arXiv:2103.15084 (2021).
69. Jerbi, S., Gyurik, C., Marshall, S., Briegel, H. J., Dunjko, V. Variational quantum policies for reinforcement learning. arXiv preprint arXiv:2103.05577 (2021).
70. Bausch, J. Recurrent quantum neural networks. In *Advances in Neural Information Processing Systems, December 6–12, Virtual*, vol. 33, pp. 1368–1379 (2020).
71. Takaki, Y., Mitarai, K., Negoro, M., Fujii, K. & Kitagawa, M. Learning temporal data with a variational quantum recurrent neural network. *Phys. Rev. A* **103**(5), 052414 (2021).
72. Yang, C.-H. H., Qi, J., Chen, S. Y.-C., Chen, P.-Y., Siniscalchi, S.M., Ma, X., Lee, C.-H. Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing, Toronto, ON, Canada, June 6–11*, 6523–6527. (IEEE, 2021).
73. Lloyd, S., Schuld, M., Ijaz, A., Izaac, J., Killoran, N.. Quantum embeddings for machine learning. arXiv preprint arXiv:2001.03622 (2020).
74. Nghiem, N. A., Chen, S. Y.-C., Wei, T.-C. A unified classification framework with quantum metric learning. arXiv preprint arXiv:2010.13186 (2020).
75. Samuel Yen-Chi Chen and Shinjae Yoo. Federated quantum machine learning. *Entropy* **23**(4), 460 (2021).
76. Kuo, E.-J., Fang, Y.-L. L., Chen, S.Y.-C. Quantum architecture search via deep reinforcement learning. arXiv preprint arXiv:2104.07715 (2021).
77. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017).
78. Wang, Y., He, H., Tan, X. Truly proximal policy optimization. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, Tel Aviv, Israel, July 22–25*, vol. 115, 113–122. (AUAI Press, 2019).
79. Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.* **8**(3), 229–256 (1992).
80. Cincio, L., Rudinger, K., Sarovar, M. & Coles, P. J. Machine learning of noise-resilient quantum circuits. *PRX Quantum* **2**(1), 010324 (2021).
81. Sharma, K., Khatri, S., Cerezo, M. & Coles, P. J. Noise resilience of variational quantum compiling. *N. J. Phys.* **22**(4), 043006 (2020).
82. Rattew, A. G., Hu, S., Pistoia, M., Chen, R., Wood, S. A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver. arXiv preprint arXiv:1910.09694 (2019).
83. Chivilikhin, D., Samarin, A., Ulyantsev, V., Iorsh, I., Oganov, A.R., Kyriienko, O. Mog-vqe: Multiobjective genetic variational quantum eigensolver. arXiv preprint arXiv:2007.04424 (2020).
84. Zhang, S.-X., Hsieh, C.-Y., Zhang, S., Yao, H.. Neural predictor based quantum architecture search. arXiv preprint arXiv:2103.06524 (2021).
85. Wu, X.-C., Davis, M.G., Chong, F.T., Iancu, C. Optimizing noisy-intermediate scale quantum circuits: A block-based synthesis. arXiv e-prints, pages arXiv–2012 (2020).
86. Du, Y., Huang, T., You, S., Hsieh, M.-H., Tao, D. Quantum circuit architecture search: Error mitigation and trainability enhancement for variational quantum solvers. arXiv preprint arXiv:2010.10217 (2020).
87. Pirhooshyaran, M. & Terlaky, T. Quantum circuit design search. *Quantum Mach. Intell.* **3**(2), 1–14 (2021).
88. Zhang, Y.-H., Zheng, P.-L., Zhang, Y. & Deng, D.-L. Topological quantum compiling with reinforcement learning. *Phys. Rev. Lett.* **125**(17), 170501 (2020).
89. He, Z., Li, L., Zheng, S., Li, Y. & Situ, H. Variational quantum compiling with double q-learning. *N. J. Phys.* **23**(3), 033002 (2021).
90. Ostaszewski, M., Trenkwalder, L. M., Masarczyk, W., Scerri, E., Dunjko, V. Reinforcement learning for optimization of variational quantum circuit architectures. arXiv preprint arXiv:2103.16089 (2021).
91. Zhang, S., Hsieh, C.-Y., Zhang, S., Yao, H. Differentiable quantum architecture search. *Bull. Am. Phys. Soc.* (2021).
92. Nielsen, M. A., Chuang, I. *Quantum Computation and Quantum Information*. (American Association of Physics Teachers, 2002).
93. Ilyas, A., Engstrom, L., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., Madry, A. Are deep policy gradient algorithms truly policy gradient algorithms? arXiv preprint arXiv:1811.02553 (2018).

94. Baker, B., Gupta, O., Naik, N., Raskar, R. Designing neural network architectures using reinforcement learning. In *5th International Conference on Learning Representations, Toulon, France, April 24–26, Conference Track Proceedings*. (OpenReview.net, 2017).
95. Ruder, S. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747 (2016).
96. Hinton, G., Srivastava, N., Swersky, K. Rmsprop: Divide the gradient by a running average of its recent magnitude. *Neural Netw. Mach. Learn. Coursera Lect.***6e**, 13 (2012).
97. Kingma, D. P., Adam, J. B. A method for stochastic optimization. In *3rd International Conference on Learning Representations* (eds Bengio, Y. & LeCun, Y.) 7–9 (CA, USA, May, San Diego, 2015).
98. Shen, X., Zhu, X., Jiang, X., Gao, L., He, T., Hu, X. Visualization of non-metric relationships by adaptive learning multiple maps t-sne regularization. In *2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11–14*, 3882–3887. (IEEE Computer Society, 2017).
99. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J. *et al.* Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32, December 8–14, Vancouver, BC, Canada*, 8024–8035 (2019).
100. Cross, A. The IBM q experience and qiskit open-source quantum computing software. *APS March Meet. Abst.* **2018**, L58-003 (2018).
101. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W. Openai gym. arXiv preprint arXiv:1606.01540 (2016).

Acknowledgements

This work is supported by the Research Foundation for Advanced Talents 31401529 of Henan University of Technology. The authors would like to thank Xiaoting Wang of the Institute of Fundamental and Frontier Sciences of University of Electronic Science and Technology of China for helpful discussions on topics related to this work.

Author contributions

X.Z. wrote the main manuscript text and X.H. helped plan the experimental analysis. All authors reviewed the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to X.Z.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.