General-purpose quantum architecture search based on deep reinforcement learning

Xiao-Yu Bi, Yi-Ming Yu, Ye-Hong Chen, and Zhi-Rong Zhong*
Fujian Key Laboratory of Quantum Information and Quantum Optics, College of Physics and Information Engineering, Fuzhou University, Fuzhou, Fujian 350108, China

(Received 10 June 2025; accepted 26 September 2025; published 4 November 2025)

Reinforcement learning (RL) shows promise for automated quantum circuit design but often stalls due to a "fidelity trap": by optimizing only state fidelity, agents overlook entanglement structure and become stranded in suboptimal, overly complex circuits, resulting in significantly reduced search efficiency. In this paper, we propose a scheme that overcomes this barrier by implementing an entanglement-aware learning framework and enhancing the agent's reward function with a direct, quantitative measure of entanglement. This approach offers a more comprehensive physical description of the state space. We demonstrate the efficacy of this principle on three- and four-qubit state-synthesis tasks within an expanded gate set. For this problem, where the fidelity-driven agent systematically fails to discover the minimal-depth circuit, our entanglement-aware agent consistently succeeds. This transformative result is highly robust against variations in initial random seeds and extends to multiqubit systems even in the presence of noise. Our findings establish a generalizable principle that incorporating entanglement as an auxiliary reward can significantly enhance RL-based solutions for a broad class of fidelity-centric tasks in quantum physics and pave the way for scalable, automated discovery on near-term quantum devices.

DOI: 10.1103/7rc4-p446

I. INTRODUCTION

The precise engineering of quantum states and operations is a cornerstone of modern quantum information science, underpinning the potential of quantum computation [1], quantum simulation [2], and even fields like drug discovery [3] and materials science [4]. A central task in harnessing the power of near-term quantum devices is the synthesis of optimal quantum circuits—those that implement a desired algorithm with minimal depth and maximal robustness against inherent noise [5]. However, the manual design of such circuits is a notoriously challenging endeavor. The combinatorial explosion of the available gate set and the vastness of the Hilbert space render brute-force enumeration intractable and intuitive design feasible only for the simplest of cases. This bottleneck has motivated a shift toward automated strategies, a field broadly known as quantum architecture search (QAS) [6–16]; however, many powerful approaches, like machine learning, face a critical shared challenge: the search effectiveness often decreases exponentially as the problem complexity grows, leading to considerable computational overhead and a tendency to get lost in the immense search landscape.

To navigate this complex optimization landscape, reinforcement learning (RL) [17–26] has emerged as a promising automated strategy, with deep RL (DRL) [27] frameworks showing aptitude for mapping raw state descriptions to actions without manual feature engineering [17,27]. RL has been successfully applied to a variety of quantum tasks, including quantum error correction [28–32], quantum state generation [33–37], quantum control [37–43], quantum gate design [44,45], and quantum state diagonalization [6]. However, a critical factor limits these algorithms, especially when tasked with synthesizing states in large quantum gate sets. We identify this as a physical information blind spot: guided by state fidelity alone, the RL agent is oblivious to the internal multipartite entanglement structure of the quantum states it explores. This "fidelity trap" leads the agent to exhibit low search efficiency, becoming stranded in highly suboptimal circuit solutions that may ultimately achieve high fidelity but at the cost of an unacceptably large gate count.

In this paper, we overcome this fundamental barrier by introducing an entanglement-aware DRL-QAS framework. Our approach enriches the agent's reward signal by integrating fidelity with a direct, quantitative measure of multipartite entanglement. This addition provides the agent with a more complete physical description of the state space, enabling it to distinguish between states that are superficially similar but structurally distinct. To rigorously validate our framework, we benchmark its performance on challenging synthesis tasks within an expanded and more realistic gate set, deliberately increasing the search space complexity. The consequence is a qualitative leap in performance: in this difficult landscape, our method consistently discovers minimal-depth circuits for tasks where fidelity-driven agents systematically fail. Furthermore, our approach exhibits profound robustness against noise and reduced sensitivity to initialization, establishing a more principled and efficient pathway for automated quantum discovery. The remainder of this paper details our

*Contact author: zhirz@fzu.edu.cn

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

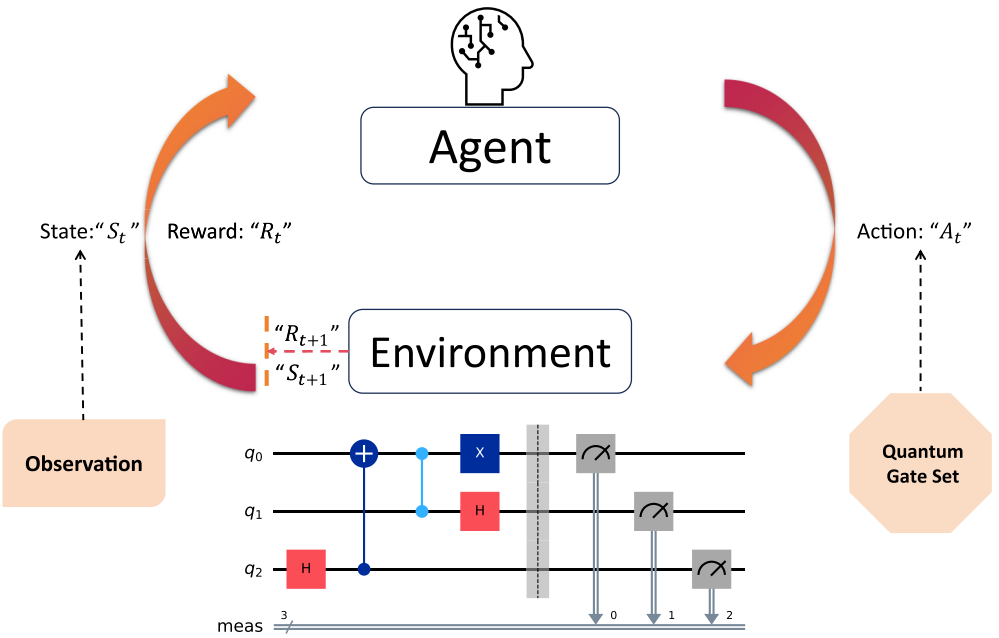


FIG. 1. The agent interacts with the environment. At each discrete time step t, the agent observes the environment's state, represented as S_t. Based on S_t, the agent selects an action A_t from a predefined action set A. The action space A can be categorized as discrete, continuous, or hybrid, depending on the nature of the available actions. The execution of A_t influences the environment, causing it to transition to a new state S_{t+1}. Concurrently, the agent receives a scalar reward signal from the environment. This reward serves as evaluative feedback on the preceding action or state transition. The reward signal R is a crucial component as it implicitly defines the objective of the RL task. During the learning process, the agent strives to learn a policy (a mapping from states to actions) that maximizes the expected cumulative reward over time.

methodology, presents comprehensive numerical results across various tasks, and discusses the broader implications of our findings.

II. REINFORCEMENT LEARNING

Prior to detailing our work, we provide a brief overview of the RL mechanism. RL addresses the problem of how an agent can maximize its cumulative rewards within complex and uncertain environments. As illustrated in Fig. 1, RL comprises two key components: the agent and the environment. The agent learns optimal decision-making policies through iterative interactions with the environment.

At each time step t, the agent observes the environment and acquires a state S_t through observation O_t. Based on this state, the agent selects an action A_t according to policy $\pi(a|s;\theta)$, which is parametrized by θ. This action is then executed in the environment, resulting in a transition to the next state S_{t+1}, observed as O_{t+1}, and the receipt of a reward r_t associated with the action. The agent's objective is to maximize its cumulative rewards from the environment.

The total reward is defined as $R = \sum_{t'=t}^{T} r_{t'}$. The total discounted reward at t time step usually has been defined as $G_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'} = r_t + \gamma G_{t+1}$, where T is total time steps and γ is the discount factor that lies in $(0, 1]$. The expected total return $\mathbb{E}[G_t]$ is used to update the policy through parameter θ. A classic algorithm example of a policy gradient algorithm is the REINFORCE algorithm [46]. Considering a general case about the policy gradient, in one episode, all combinations of state-action pairs $\{s_1, a_1, r_1, \ldots, s_{t+1}, a_{t+1}, r_{t+1}, \ldots, s_T, a_T, r_T\}$ are labeled as trajectory τ. For a certain θ, we can calculate the probability of a τ by $p_\theta(\tau) = p(s_1)\prod_{t=1}^{T} p_\theta(a_t|s_t)p_\theta(s_{t+1}|a_t, s_t)$. The expected reward can be written $\bar{R}_\theta = \sum_\tau R(\tau)p_\theta(\tau)$. In order to achieve the maximum reward, the method of gradient ascent is used. $\nabla\bar{R}_\theta = \sum_\tau R(\tau)\nabla p_\theta(\tau) = \sum_\tau R(\tau)p_\theta\nabla\log p_\theta(\tau)$. Then, the policy will be updated to $\theta = \theta + \eta\nabla\bar{R}_\theta$, where the η is the learning rate. We can also interpret this process through value function. The value function is defined as

$$v_\pi(s) = \mathbb{E}_\pi[G_t|S_t = s], \tag{1}$$

and the action value function is

$$q_\pi(s, a) = \mathbb{E}_\pi[G_t|S_t = s, A_t = a]. \tag{2}$$

Based on $v_\pi(s)$ and $q_\pi(s, a)$, the agent updates policy by

$$\begin{aligned}\pi(s) &= \arg\max_a \; q_\pi(s, a) \\ &= \arg\max_a \; \mathbb{E}_\pi(r + \gamma v_\pi(s')|s, a).\end{aligned} \tag{3}$$

Our schema is not limited in a certain RL algorithm, but in previous works, the proximal policy optimization (PPO) [18] showed better performance [16]. So in this paper, our schema is shown via the PPO algorithm, which is a policy gradient algorithm.

III. QUANTUM ARCHITECTURE SEARCH WITH DEEP REINFORCEMENT LEARNING

To implement QAS via DRL, we must formulate the QAS process as a Markov decision process (MDP). An MDP consists of four key elements: the state space S, the action space

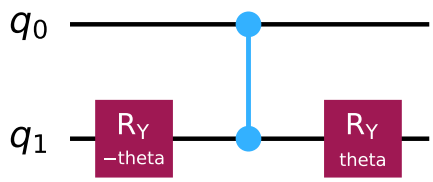


FIG. 2. Composition of F Gate, which is composed of ry gates and cz gate. $\theta = \arccos(\sqrt{1/(n-k+1)})$, and n is the total number of qubits of the quantum circuit.

A, the reward function R, and the terminal condition. The state space S is accessible to the agent through observations.

The first aspect we discuss is the action space. In general, a smaller quantum gate set implies a smaller action space; as the action space increases, the computational cost rises rapidly. Previous works have critically limited the variety of gates to maintain a small action space. In this work, our scheme can handle a significantly larger action space, so we use discrete gate sets

$$\begin{aligned}\mathbb{A} = \bigcup_{i=1}^{n} \{&X_i, Y_i, Z_i, H_i, \mathrm{CNOT}_{(i,i+1)(\mathrm{mod}2)},\\ &\mathrm{CZ}_{(i,i+1)(\mathrm{mod}2)}, Rz_i(\pi/4), Ry_i(\pi/2),\\ &F(k=1)_{(i,i+1)(\mathrm{mod}2)}, F(k=2)_{(i,i+1)(\mathrm{mod}2)}\}. \end{aligned} \quad (4)$$

The action space can be changed according to the assignment.

To prepare a specific zero-entanglement state, as demonstrated by qiskit, we augment the gate set with an F gate. The F gate is a composite gate composed of two ry gates and one cz gate, which is shown in Fig. 2.

The observation represents the portion of the current state that the agent can observe. There are two possible scenarios regarding the observation: the agent obtains a fully observed state when $O_t = S_t$, and the agent obtains a partially observed state when $O_t \neq S_t$.

In the QAS case, we use the expectation values of Pauli-X, -Y, -Z; the number of quantum gates (NQG); and the current state entanglement ($\mathrm{Ent}_{\mathrm{current}}$) as the observation. Thus, the observation space can be defined as

$$O = \bar{X}, \bar{Y}, \bar{Z}, \mathrm{NQG}, \mathrm{Ent}_{\mathrm{current}}. \quad (5)$$

Many algorithms cannot handle episodes with infinite steps, so we limit the maximum number of steps. Here, we include the NQG as part of the observation space, which makes the agent time aware. As demonstrated in Ref. [47], this is a useful method to address truncation when we limit the maximum number of steps.

When the agent reaches the terminal condition, it signifies the completion of one episode, and the agent will then initiate the next episode. We define the terminal condition using a fidelity threshold and a maximum gate limit. This implies that an episode concludes either when the maximum number of quantum gates in the circuit is reached or when the fidelity between the current state and the target state attains the fidelity threshold.

The next aspect we will discuss is the reward function. The reward function is a crucial component of RL, as it defines the agent's objective. The reward function directly influences the agent's performance; in other words, we can improve the agent's performance by providing a more effective reward function.

In the way described in Sec. II, the reward influences policy updating. Related works utilize fidelity as a sparse reward: the agent receives a reward of fidelity -0.01 upon reaching the fidelity threshold; otherwise, the agent receives a penalty of -0.01. Sparse rewards are easy to design and introduce minimal bias, making them a reasonable choice for QAS. However, with larger state and action spaces, the "sparse reward problem" arises. The agent only receives a reward different from -0.01 upon reaching the goal. Consequently, most rewards within an episode are -0.01, preventing the agent from clearly distinguishing the value of different state-action pairs, so in this opinion, such reward function can be considered as a sparse reward.

In the early stages of RL exploration, the agent explores almost blindly. As task complexity and the size of the state-action space increase, the probability of discovering the target state through random exploration diminishes significantly. This scarce feedback signal fails to guide the agent's exploration, hindering the formation of local knowledge. Blind exploration leads to few, if any, effective transitions, resulting in extremely low sample efficiency and difficulty in converging the RL algorithm.

Even after achieving the goal, when the agent receives a reward of fidelity -0.01, a problem known as the "credit assignment problem" [48] persists: are all (S, a) pairs equally valuable? Consider a trajectory τ represented as $\{s_1, a_1, r_1, \ldots, s_{t+1}, a_{t+1}, r_{t+1}, \ldots, s_T, a_T, r_T\}$. If the agent reaches the goal, r_T is fidelity -0.01, while all rewards from r_1 to r_{T-1} are -0.01. This makes it challenging for the agent to differentiate the value of (S, a) pairs before step T.

While the credit assignment problem is often associated with sparse rewards, the fundamental issue is information sparsity, not merely reward sparsity. Consider a reward function

$$R'(S, a) = R(S, a) + C. \quad (6)$$

Clearly, $R'(S, a)$ is no longer sparse and guarantees nonzero feedback at each time step. However, such a modification neither alters the optimal policy π nor alleviates the challenges of credit assignment that impede efficient learning π. As argued in Ref. [48], this type of reward function remains informationally sparse.

A common approach to address the sparse reward problem is to redesign the reward function. However, the design space for reward functions is vast, and crafting a sufficiently effective reward function is often challenging. Reward design [49–51] is a important field in RL, but in many RL-QAS works, the reward design has not been designed rigorously. The reward function serves as both the objective and the guide for the agent, enabling the agent to distinguish between favorable and unfavorable states. Generally, introducing additional reward signals can introduce bias, leading to reward hacking, wireheading, cowardice, and reckless behavior. Therefore, any redesign of the reward function must carefully avoid these issues.

Using fidelity as a dense reward simply is a reasonable choice, but it still has potential problems. Considering a

simple case of a three-qubit cluster state, the initial state is $|000\rangle$ where the fidelity with the cluster state is 12.5%. After adding an H gate at the first qubit, the state evolves to $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)|00\rangle$ where the fidelity is 25%, and after adding an H gate to the second qubit, the fidelity is changed to 12.5%. After adding the cz gate between the first and the second qubit, the fidelity is changed to 25%, and after adding the final cz gate between the second and third qubit, the fidelity becomes 1. During such process, the fidelity does not increase monotonically.

Another noticeable thing is that, like training the agent to get high fidelity, getting the target entanglement state also needs be trained. For a certain state, we can calculate the fidelity with the current state, and the entanglement is the same thing. The fidelity can show the distance between the current state and target state—when the fidelity is equal to 1, two states can be considered as the same state; in other words, the state is unique. But when the fidelity is not 1, there is more than one state at the same fidelity. Here, another question is posed: are the states which have equal fidelity the same? The answer is clearly no. If we want to distinguish those states, additional physical information is necessary.

To provide more frequent and distinguishable reward signals, we incorporate an entanglement measure into the reward function, in addition to fidelity. In our work, since we focus on three-qubit quantum circuits, we use the residual entanglement [52–55] to measure entanglement. However, our approach does not limit the choice of entanglement measure. The residual entanglement can be expressed in terms of the coefficients $\{\phi_{000}, \phi_{001}, \ldots, \phi_{111}\}$:

$$\mathrm{Ent} = 4|d_1 - 2d_2 + 4d_3|, \tag{7}$$

where the d_1, d_2, d_3 are

$$d_1 = \phi_{000}^2\phi_{111}^2 + \phi_{001}^2\phi_{110}^2 + \phi_{010}^2\phi_{101}^2 + \phi_{100}^2\phi_{011}^2, \tag{8}$$

$$\begin{aligned} d_2 = {} & \phi_{000}\phi_{111}\phi_{011}\phi_{100} + \phi_{000}\phi_{111}\phi_{101}\phi_{010} \\ & + \phi_{000}\phi_{111}\phi_{110}\phi_{001} + \phi_{011}\phi_{100}\phi_{101}\phi_{010} \\ & + \phi_{011}\phi_{100}\phi_{110}\phi_{001} + \phi_{101}\phi_{010}\phi_{110}\phi_{001}, \end{aligned} \tag{9}$$

$$d_3 = \phi_{000}\phi_{110}\phi_{101}\phi_{011} + \phi_{111}\phi_{001}\phi_{010}\phi_{100}. \tag{10}$$

To incorporate the entanglement measure into the reward function, we need to balance the proportion between fidelity and entanglement. Our reward function can be expressed as

$$\mathrm{reward} = \begin{cases} C_{g1} \times \mathrm{GP} + C_{f1} \times F + (1 - \mathrm{IW}) \times (1 - \mathrm{Ent_{diff}}) & (F \geqslant \mathrm{Threshold}) \\ [C_{g2} \times \mathrm{GP} - 1 + C_{f2} \times (1 - \mathrm{IW}) \times F + C_e \times (1 - \mathrm{IW}) \times (1 - \mathrm{Ent_{diff}})] \times 0.5 & (\mathrm{Threshold} > F \geqslant 0.6) \\ [C_{g3} \times \mathrm{GP} - 1 + \mathrm{IW} \times (1 - \mathrm{Ent_{diff}} + F)] \times 0.5 & (0.6 > F \geqslant 0.4) \\ [C_{g4} \times \mathrm{GP} - 1 - \mathrm{IW} \times (1 - F)] \times 0.5 & (0.4 > F) \end{cases}. \tag{11}$$

The terms $C_{g1}, C_{g2}, C_{g3}, C_{g4}, C_{f1}, C_{f2}, C_e$, and C_{iw} are hyperparameters. The GP represents the gate penalty, which is defined by the current quantum gate count (CQGC) and the maximum quantum gate count (MQGC):

$$\mathrm{GP} = \frac{\mathrm{CQGC}}{\mathrm{MQGC}}, \tag{12}$$

which represents the ratio of the current number of quantum gates to the maximum number of quantum gates allowed in the quantum circuit. We use the GP term as a penalty; when the agent adds a new quantum gate to the quantum circuit, the GP increases. A larger quantum gate count implies a larger penalty, so the agent tends to find shorter quantum circuits. In the reward function, the coefficient C_{g1} to C_{g4} of GP gradually increases but is still smaller than 0. Thus, when the fidelity is low, the agent can use more quantum gates to increase the fidelity without incurring a significant penalty. However, as the fidelity increases, the proportion of GP in the reward function grows increasingly larger. This setting encourages the agent to choose circuits with fewer quantum gates to maximize the reward. The C_{f1} is the coefficient of fidelity when the agent achieves the goal. The C_{f2} and the C_e are weighting coefficients used to control the proportion of the fidelity term and the entanglement term in the reward function. For convenience in calculations, these two coefficients simply sum up to 1:

$$C_{f2} + C_e = 1. \tag{13}$$

The $\mathrm{Ent_{diff}}$ is the difference of entanglement between the target state and the current state:

$$\mathrm{Ent_{diff}} = |\mathrm{Ent_{target}} - \mathrm{Ent_{current}}|. \tag{14}$$

We calculate the entanglement using the residual entanglement. By using the absolute value of the difference between the target state's entanglement and the current state's entanglement, $\mathrm{Ent_{diff}}$ increases as the current state deviates from the target state. One of the agent's objectives is to minimize $\mathrm{Ent_{diff}}$. However, $\mathrm{Ent_{diff}} = 0$ does not necessarily imply that the current state is identical to the target state. Introducing the $\mathrm{Ent_{diff}}$ term too early may mislead the agent. Therefore, in our reward function, we do not incorporate $\mathrm{Ent_{diff}}$ when the fidelity is below 0.4. The agent only needs to consider entanglement when the quantum circuit has achieved a certain level of fidelity.

The IW is the infidelity weight:

$$\mathrm{IW} = C_{iw} \times (1 - F). \tag{15}$$

The IW is equal to a coefficient multiplied by the infidelity. In our reward function, we use IW to control the weight changes of fidelity (F) and $\mathrm{Ent_{diff}}$ across different fidelity ranges. As the fidelity increases, the IW decreases. When the fidelity is

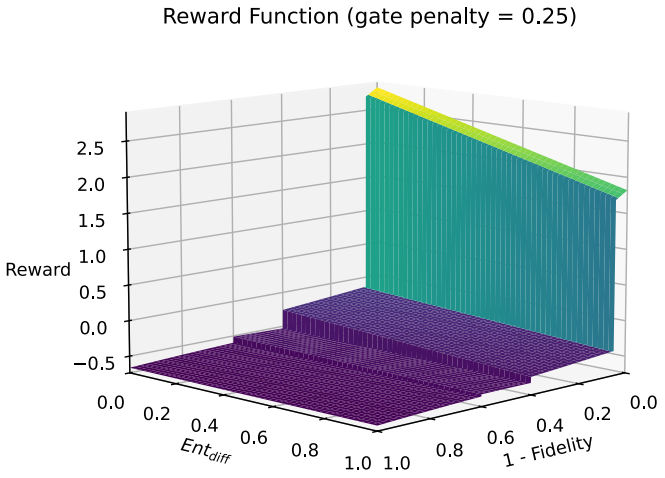


FIG. 3. Reward function. Based on Ent_{diff} and fidelity, with a fixed gate penalty of 0.25, $C_{g1} = -1.5$, $C_{g2} = -1$, $C_{g3} = -0.75$, $C_{g4} = -0.5$, $C_{f1} = 2.2$, and here the C_{iw} is equal to 0.2.

below 0.6, we use the −IW term as the weight, and when the fidelity is greater than 0.6, we use (1 − IW) as the weight. Here, the coefficient of IW is C_{iw}, and the coefficient is adjustable. As the coefficient increases, the variation of IW with fidelity becomes more pronounced.

For simplicity, we set GP equal to 0.25. As shown in Fig. 3, the overall trend of the reward function is to increase with increasing fidelity. Until the fidelity reaches the fidelity threshold, all reward function values are less than 0, which is achieved by the term −1 [the value is determined by Eq. (13)]. This setting helps prevent reward hacking. Without the term −1, the reward function would be positive in the fidelity range [0.4, fidelity threshold], and the agent might choose an unintended way to obtain an extremely high reward, such as selecting multiple self-inverting quantum gate operations like the $\{X, X, X, X\}$ gate sequence on one qubit. We do not desire such circuits, so a way to avoid this problem is to make the reward negative before the agent achieves the goal.

Another potential problem we address is cowardice. The cowardice problem is a phenomenon where the agent receives negative rewards at the beginning, and no matter what action the agent chooses, the reward remains negative, potentially leading the agent to take no action at all. To mitigate this issue, we multiply each negative reward term by 0.5, which does not alter the shape of the reward function.

IV. QUANTUM ARCHITECTURE SEARCH EXPERIMENTS

In this section, we use our method to generate three different quantum states as test cases, comparing it with the previous QAS-PPO method [16] (we use the same name as in Ref. [14] to describe the previous baseline method). We set all hyperparameters to the same values; the specific hyperparameter settings are as follows: fidelity threshold = 0.95, learning rate: $\eta = 0.004, \gamma = 0.97, n$ epochs = 4, clip range = 0.2, max steps = 20, optimizer = Adam, $C_{g1} = -1.5$, $C_{g2} = -1$, $C_{g3} = -0.75$, $C_{g4} = -0.5$, $C_{f1} = 2.2$, $C_{iw} = 0.2$, $C_{f2} = C_e = 0.5$.

To mitigate the influence of random seeds, we randomly select five different random seeds for each task, ensuring that both methods use the same seeds. Because we set a terminal condition, the episode only concludes when either the current fidelity exceeds the fidelity threshold or the total number of quantum gates in the circuit reaches the gate limit. This means that only when the fidelity reaches the fidelity threshold can the number of quantum gates be less than the gate limit. Therefore, we compare the two methods based on the number of gates used.

First, we consider generating the three-qubit cluster state from the Fock state $|000\rangle$ in a noise-free environment, the cluster state can be written as

$$|\text{CLUSTER}\rangle = \frac{1}{\sqrt{8}}(|000\rangle + |001\rangle + |010\rangle - |011\rangle + |100\rangle + |101\rangle - |110\rangle + |111\rangle). \quad (16)$$

In the second experiment, the agent is tasked with generating the cluster state commencing from the Greenberger-Horne-Zeilinger (GHZ) state. The initiation of quantum state transformations from states other than the Fock state $|000\rangle$ has not been extensively considered in prior literature, yet such transformations are valuable in specific scenarios. When the objective is to transform an existing, nontrivial initial state into a desired target state, constraining the agent to construct the quantum circuit solely from the $|000\rangle$ state is inappropriate. While Ref. [56] documents research on deriving the GHZ state from the cluster state, the primary objective of the present experiment is the reverse transformation: to generate the cluster state from the GHZ state via a quantum circuit.

In the third experiment, to investigate the impact of target state entanglement on performance, we use a zero-entanglement state as the target, which has the same residual entanglement as the initial state:

$$|0\text{ENT}\rangle = \frac{1}{\sqrt{3}}(|100\rangle + |110\rangle + |111\rangle). \quad (17)$$

As shown in Fig. 4, our method outperforms the QAS-PPO method in all tested scenarios, exhibiting superior convergence speed and requiring fewer quantum gates. Furthermore, our method demonstrates a smaller standard deviation compared to QAS-PPO, indicating its reduced sensitivity to the initial random seed. This enhanced robustness and efficiency can be attributed to a more effective reward signal that addresses the information sparsity problem, and to the expanded quantum gate set relative to previous work, which necessitates greater exploration capabilities within the action and state spaces for task completion.

Another noteworthy observation pertains to the success rates. In Figs. 4(a) and 4(c), the success rate of the QAS-PPO method is demonstrably lower than that of our approach. In particular, in the task of preparing the cluster state with the GHZ state as the initial state, the success rate of QAS-PPO is 0%. In addition, our method effectively improves the search efficiency by about 100%. Across all evaluated tasks, our method consistently achieves a 100% success rate in identifying the shallowest-depth quantum circuit. This disparity suggests that the performance of the baseline fidelity-driven

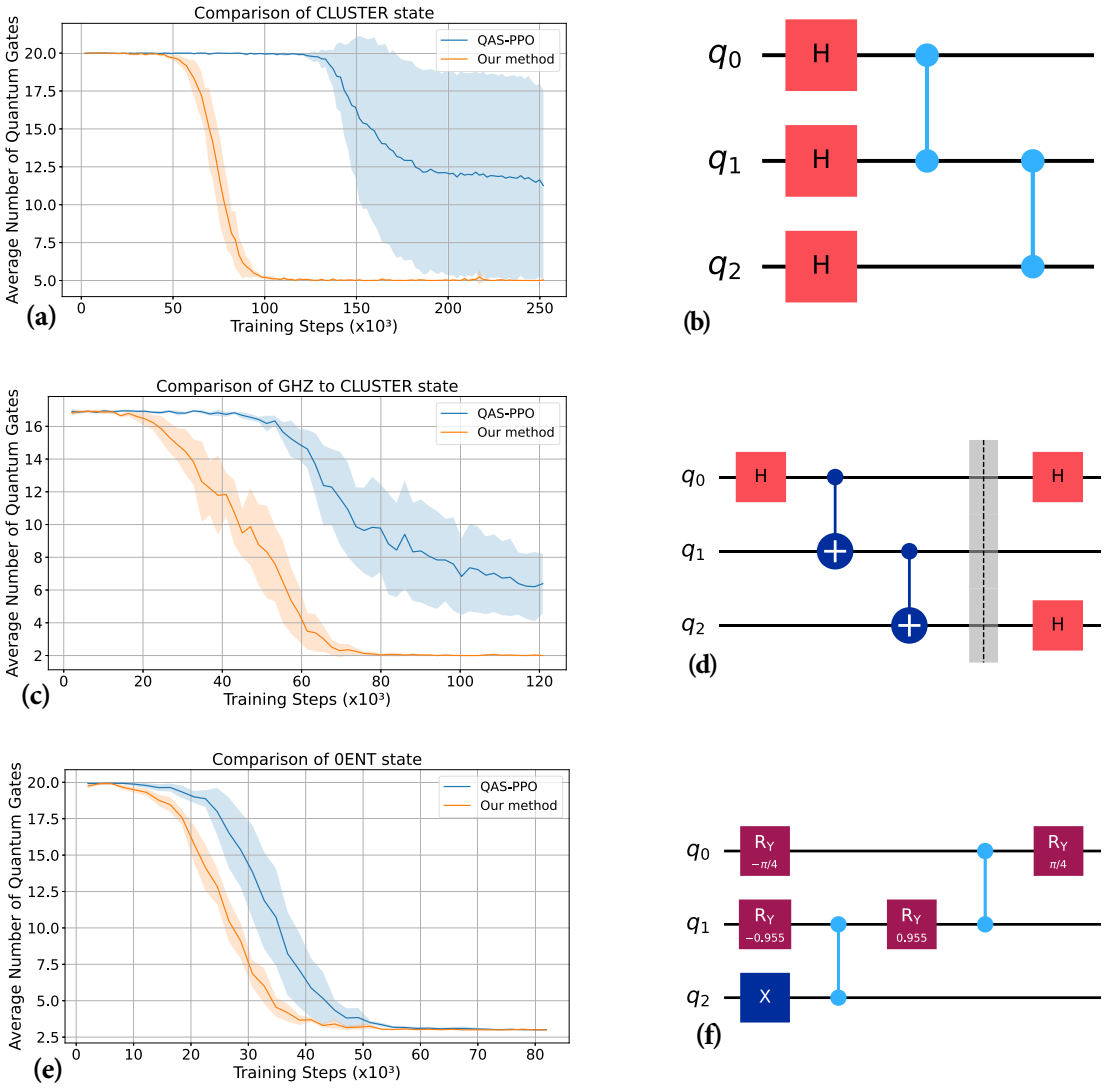


FIG. 4. Comparison of two methods in the same assignment. The x-axis label is the total training steps, and the y-axis label is the average number of quantum gates. The results are averaged across five random seeds for each experiment, and shaded areas represent plus or minus one standard deviation. (a) The target state is a cluster state, and the minimum number of quantum gates is 5. (c) The target is generating the cluster state from a GHZ state, and the minimum number of quantum gates is 2. Because generating the initial GHZ state requires three gates, without changing the limit number of quantum gates, the agent's addable gate count is 17. (e) The target is generating the zero-entanglement state and the minimum number of quantum gates is three. The column on the right is the quantum circuit generated by our scheme. (b) The cluster state quantum circuit, where our method designs the same circuit as a human-level design by applying a Hadamard gate to each qubit and then applying a cz gate to each pair of qubits. (d) Generating the cluster state from the GHZ state, where our method finds a quantum circuit with two gates. (f) A zero-entanglement state, consisting of an X gate and two F gates.

QAS-PPO method is more susceptible to the initial probability distribution of quantum gate selection.

The results for the four tasks are shown in Figs. 4(b), 4(d), and 4(f). We observe that the quantum circuits generated for cluster states achieve human-level design. Previous studies have indicated that the standard quantum circuit for the cluster state requires five quantum gates. Generating the cluster state from the GHZ state requires applying just two Hadamard gates. Furthermore, generating the specific zero-entanglement state necessitates one X gate and two F gates.

In Fig. 5, the two methods illustrate different relationships between fidelity and entanglement. Initially, the fidelity of the QAS-PPO method does not exhibit the same growth trend as entanglement, fluctuating between 0 and 0.6, while entanglement varies from 0 to 1. In contrast, our method demonstrates a consistent growth trend between fidelity and entanglement from the beginning. This observation suggests that the entanglement term in our reward function effectively guides the agent's behavior.

V. MULTIQUBITS AND NOISY ENVIRONMENT CONDITION

The measurement and characterization of multiqubit entanglement have consistently represented a pivotal challenge in the field of quantum physics. In our preceding discussion

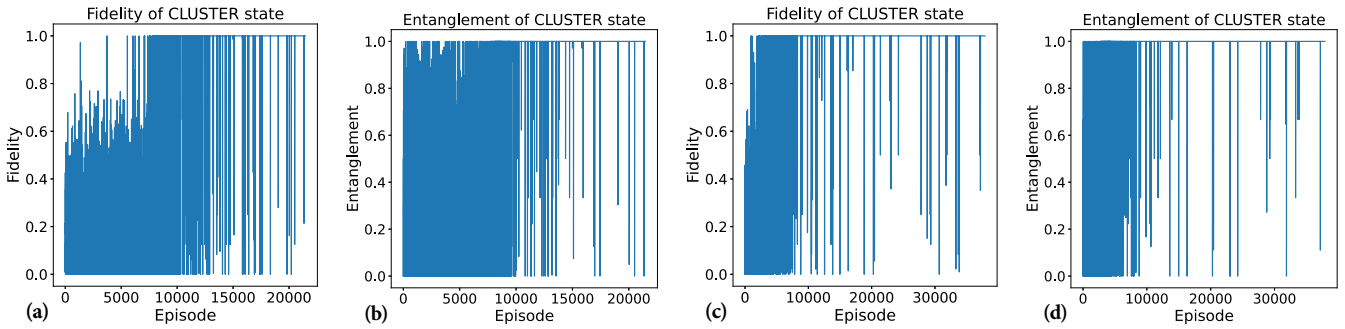


FIG. 5. Raw data on fidelity and entanglement per episode for cluster state generation using the same initial seed. (a) and (b) are the fidelity and entanglement of the QAS-PPO method. (c) and (d) are the fidelity and entanglement of our method. For tasks involving the generation of cluster states, our method demonstrates a consistent relationship between the variations of entanglement and fidelity during the early stages of RL exploration.

[specifically, as formulated in Eq. (11)], the entanglement term was introduced within the context of a three-qubit system. This section extends the discourse to scenarios involving a greater number of qubits. To quantify entanglement in such many-body systems, a common strategy involves partitioning the total system into two complementary subsystems. If the information pertaining to one subsystem is disregarded (i.e., traced out), the remaining subsystem typically evolves into a classical statistical mixture of pure quantum states. As demonstrated in Ref. [57], Rényi entropy serves as an efficacious tool for entanglement detection. The αth order Rényi entropy is formally defined as

$$S_\alpha(A) = \frac{1}{1-\alpha}\log \mathrm{Tr}\left(\rho_A^\alpha\right). \tag{18}$$

It is well established that when the parameter $\alpha = 2$, Rényi entropy reduces to the von Neumann entanglement entropy. For the case where $\alpha = 2$, Rényi entropy is directly related to the purity of the quantum state. Given that Rényi entropy, particularly for $\alpha \neq 1$, can furnish more comprehensive information about entanglement than the von Neumann entropy, this study adopts the second-order Rényi entropy ($\alpha = 2$) as the primary entanglement quantifier.

In an idealized, noise-free quantum circuit, the generated quantum state is typically a pure state. For any pure state, the global density matrix ρ_{total} satisfies $\mathrm{Tr}(\rho_{\text{total}}^2 = 1)$, irrespective of whether it is an entangled state such as the GHZ state. In such instances, calculating the Rényi entropy of the total system is uninformative (the second-order Rényi entropy of a pure state is zero). Consequently, we focus on the Rényi entropy of one of its subsystems as a measure of entanglement. As established in Ref. [57], the specific manner of partitioning the system significantly influences the resultant Rényi entropy of the subsystems. A bisection, or a division as close to the midpoint as possible, generally tends to yield a relatively small entropy difference between the two resulting subsystems. Adopting this approach, the Rényi entropy of a single subsystem can then serve as a characteristic measure of the entanglement present in the current quantum state. This selection strategy effectively preserves crucial entropy information while concurrently reducing the computational overhead. Given that the maximum value of Rényi entropy scales with the number of qubits, normalization is necessary to apply this entanglement measure consistently across systems of arbitrary qubit counts. We perform this normalization using a base-2 logarithm:

$$\mathrm{Ent} = \frac{S_2(A)}{N}, \tag{19}$$

where N is the number of qubits.

To demonstrate the feasibility of our proposed approach, this study involved a search for a quantum circuit capable of preparing a four-qubit GHZ state. Consistent with the methodology of previous experiments, five identical initial random seeds were utilized for each of the two methods under comparison to ensure a fair comparison. The quantum gate set employed in this experiment is defined as follows:

$$\begin{aligned}\mathbb{A} = \bigcup_{i=1}^{n} \{&X_i, Y_i, Z_i, H_i, \mathrm{CNOT}_{(i,i+1)(\mathrm{mod}2)},\\ &\mathrm{CZ}_{(i,i+1)(\mathrm{mod}2)}, Rz_i(\pi/4), Ry_i(\pi/2), S\}.\end{aligned} \tag{20}$$

We kept all hyperparameters as consistent as possible; however, since the exploration space of the four-qubit system is larger than that of the three-qubit system, we slightly increased the learning rate to accelerate training, slightly decreased the C_{iw} term to reduce reward fluctuations and thereby improve training stability, and simultaneously increased C_{f1} to further encourage the agent to achieve high fidelity. The specific settings are as follows: fidelity threshold = 0.95, learning rate $\eta = 0.005$, $\gamma = 0.97$, n epochs = 4, clip range = 0.2, max steps = 20, optimizer = Adam, $C_{g1} = -1.5$, $C_{g2} = -1$, $C_{g3} = -0.75$, $C_{g4} = -0.5$, $C_{f1} = 10.2$, $C_{iw} = 0.1$, $C_{f2} = C_e = 0.5$.

The experimental results are presented in Fig. 6. It is evident from the figure that our proposed method requires significantly fewer interaction steps to achieve the target state and exhibits a smaller standard deviation when compared to the QAS-PPO method. This indicates a higher search efficiency and superior convergence stability for our approach.

The inherent error rates of quantum gates represent a critical factor for consideration in quantum circuit design, particularly when targeting real-world quantum devices. The execution of quantum gates under nonideal conditions not only leads to a degradation in operational fidelity but also adversely affects the learning efficiency of RL agents. To

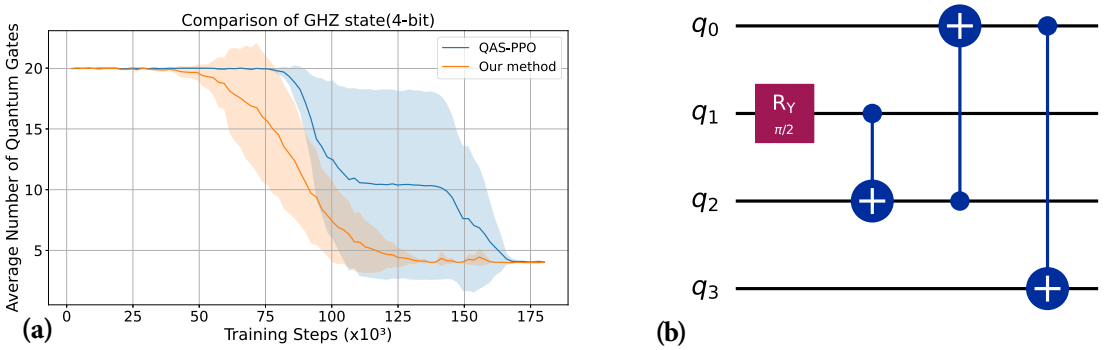


FIG. 6. Comparison of two methods in generating a four-qubit GHZ state. (a) the x-axis label is the total training steps, and the y-axis label is the average number of quantum gates. The results are averaged across five random seeds for each experiment and shaded areas represent plus or minus one standard deviation. (b) The four-qubit GHZ state quantum circuit, which is constructed by one $Ry(\pi/2)$ gate and three CNOT gates.

evaluate the performance of our approach in such noisy environments, we conducted experiments similarly focused on the generation of a four-qubit GHZ state. In these noisy simulations, the error rate for single-qubit gates was set to 0.002. For multiqubit gates, an equivalent per-qubit error rate of 0.002 was applied to each qubit involved in the operation. In these noisy simulations, the relevant reward function weighting coefficients were set to $C_{f2} = 0.7$ and $C_e = 0.3$.

As depicted in Fig. 7, the introduction of noise led to an expected decrease in the fidelity of the generated quantum state compared to the ideal, noise-free scenario. Notably, however, the structure of the shortest quantum circuit required to generate the target GHZ state (comprising four quantum gates) remained unaltered. Under these noisy conditions, while all methods required more training iterations to achieve the predefined objective, our proposed approach demonstrated a faster convergence rate. In contrast, the QAS-PPO method not only converged more slowly but also failed to reduce the average number of quantum gates to the theoretically minimal value by the culmination of the training process. Furthermore, when compared to the noise-free environment, the standard deviation in performance across different initial random seeds for the QAS-PPO method increased, indicating heightened sensitivity to noise. Our method, conversely, exhibited greater robustness, being less significantly affected by the noise. Consequently, these experimental findings suggest that our proposed approach can search for quantum circuits more efficiently even in the presence of noise and possesses significant advantages in terms of noise resilience. It is also worth noting that in the noisy condition, our method shows a larger performance improvement than the method driven by fidelity only, compared to the noise-free case.

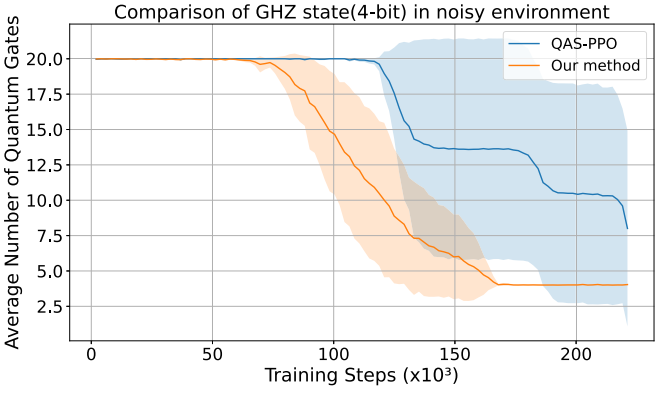


FIG. 7. Experimental results for the generation of a four-qubit GHZ state in a noisy environment. Comparison with the noise-free environment reveals that while the optimal quantum circuit structure (four quantum gates) remains unchanged, training efficiency is notably impacted by noise. Specifically, the QAS-PPO method is significantly more affected than our proposed approach in terms of both the standard deviation of training steps and sensitivity to initial random seeds. Furthermore, unlike in the noise-free scenario, the average number of quantum gates employed by the QAS-PPO method did not converge to the theoretical minimum by the end of the training process.

VI. DISCUSSION AND CONCLUSIONS

In this work, we propose an "entanglement-aware" framework to address a key inefficiency in reinforcement learning for quantum circuit synthesis. This inefficiency stems from a fundamental information deficit: the inability of agents to account for the internal entanglement structure of quantum states when guided solely by fidelity. By enriching the agent's reward signal with a quantitative measure of entanglement, our framework significantly improves search efficiency and helps the agent avoid "fidelity traps"—local optima where states of high fidelity possess a complex circuit structure. This allows the agent to effectively discover minimal-depth circuits for complex tasks that are challenging for traditional, fidelity-driven approaches.

Experimental results in three- and four-qubit systems, including the synthesis of GHZ, cluster, and zero-entanglement states, demonstrate the practical viability and robustness of our approach. We observed a marked reduction in both training standard deviation and sensitivity to initialization. For instance, in the challenging task of synthesizing a cluster state from diverse initial conditions, our method achieves consistently high success rates where conventional methods often fail, effectively doubling the search efficiency in our tests. To assess its performance on realistic platforms, we evaluated

our method under noisy conditions. Notably, the performance advantage of our method becomes even more pronounced under noise, suggesting that entanglement guidance is particularly valuable for complex and noisy conditions. This enhanced stability and noise resilience translate directly into a more reliable and resource-efficient design process for near-term quantum devices. Overall, the method's advantages are reflected in lower gate counts, improved success rates and robustness, and faster convergence.

Regarding scalability, our framework's design hinges on selecting computationally tractable entanglement measures, such as transitioning to subsystem Rényi entropy for larger systems. The principal limitation, common to similar simulation-based approaches, is the classical computational overhead, which can become a limitation as system size increases. A further consideration is that the reward's efficacy is contingent on the chosen measure's ability to capture the target state's specific correlations. Nevertheless, the framework's modularity allows for the future integration of more advanced measures, reinforcing the core principle of leveraging physical information beyond fidelity to enhance the search process.

The core innovation, the balanced reward that trades off fidelity against entanglement structure, provides a versatile tool adaptable to a wide range of reinforcement learning applications in quantum physics. Beyond circuit synthesis, any task where fidelity guides the agent—such as error-robust quantum logic gates design [44], state generation [58], and quantum control [40]—is expected to benefit from entanglement aware framework, providing a promising direction for future RL-based quantum physics applications. We thus envision this framework as a general pathway for resource-efficient, physically grounded RL across diverse quantum information and dynamics problems.

In addition to the entanglement-aware RL framework presented here, alternative paradigms such as evolutionary algorithms (EAs) [59,60] offer complementary strengths, and their integration with RL may open new pathways for efficient quantum circuit synthesis. These methods offer powerful capabilities for parallel, global exploration of the vast circuit space. A promising avenue for future research lies in creating hybrid approaches that synergize EAs and RL. This direction is further motivated by recent successes in related fields, such as the use of quantum-inspired genetic algorithms (QGAs) combined with machine learning surrogate models to efficiently design complex photonic structures [61].

Inspired by this approach, one could envision a framework where an EA discovers promising circuit topologies, which then serve as a curriculum or an initialization for an RL agent to fine-tune and optimize. Such a hybrid model could potentially overcome the respective bottlenecks of each approach, leading to highly efficient and trainable agents that learn a robust, predictive model of quantum circuit synthesis.

ACKNOWLEDGEMENTS

We express our gratitude to Ye-Xiong Zeng and Yan-Ming Che for their valuable suggestions for the revision of this article. This work was supported by the National Natural Science Foundation of China under Grant No. 12074070.

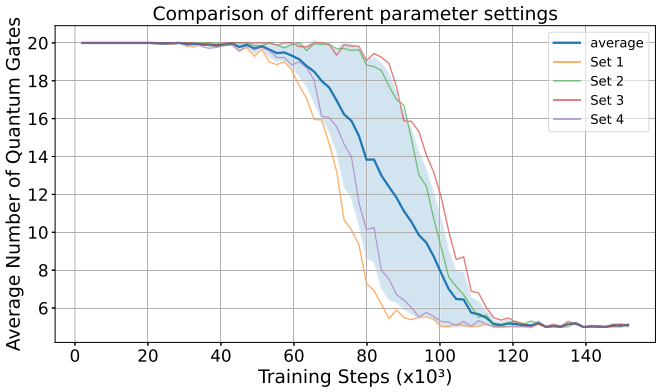


FIG. 8. Robustness analysis of our method under different hyperparameter settings under generation cluster state. Except for the blue line, each colored solid line represents the performance of one of the four hyperparameter sets. The solid blue line depicts the grand mean, averaged across all four settings, with the standard deviation shown as a shaded area. The consistent high performance across all configurations, with only minor variations in convergence speed, demonstrates the robustness of our proposed method.

DATA AVAILABILITY

The data that support the findings of this article are openly available [62].

APPENDIX A: PERTINENT INFORMATION OF HYPERPARAMETERS

In this section, we provide a more detailed rationale for the hyperparameter selection in our reward function and present a sensitivity analysis to demonstrate the robustness of our proposed method. The reward function's design incorporates several hyperparameters. They are selected based on general principles intended to guide the agent's learning process effectively. The key principles are as follows:

C_{g1} to C_{g4}: These coefficients control the gate count penalty (GP). Their magnitudes are designed to increase with fidelity. This strategy encourages exploration with deeper circuits in the early training stages and promotes the exploitation of shorter, more efficient circuits as the agent nears the solution.

C_{f2} and C_e: These coefficients balance the relative importance of fidelity (F) and entanglement difference ($\mathrm{Ent}_{\mathrm{diff}}$). Since fidelity is the ultimate measure of success, and a minimal entanglement difference is a necessary but not sufficient condition for achieving the target state, we adopt the setting $C_{f2} \geqslant C_e$ to prioritize the fidelity term.

C_{f1}: This coefficient provides a large positive reward upon reaching the target state. Its primary role is to serve as a strong terminal incentive. A sufficiently large value ensures the agent is motivated to achieve the final goal, even if it must accumulate penalties (e.g., from gate usage) during intermediate steps. This is particularly crucial for complex tasks requiring longer gate sequences.

C_{iw}: This parameter controls the infidelity weight (IW), which dynamically adjusts the balance between the fidelity and entanglement terms. The value of C_{iw} determines the magnitude of fluctuations in the reward signal.

TABLE I. Hyperparameters for the PPO algorithm. The configuration is based on the stable-baselines3 (v2.4.0) implementation, and we adopt its default values for most parameters. In the table, VF denotes the value function, GAE represents the generalized advantage estimator, SED refers to state-dependent exploration, and KL stands for the Kullback-Leibler divergence.

n steps	2048	Batch size	64	n epochs	4
Gamma	0.97	GAE lambda	0.95	Clip range	0.2
Clip range VF	None	Normalize advantage	True	Entropy coefficient	0.0
VF coefficient	0.5	Max grad norm	0.5	Use SDE	False
SDE sample freqency	-1	Rollout buffer class	None	Rollout buffer kwargs	None
Target KL	None	Stats window size	100	Tensorboard log	None
Policy kwargs	None	Verbose	0	Policy	MultiInputPolicy
Device	"Auto"	Initial setup model	True	Learning rate	0.004

To validate the robustness of our approach, we tested the cluster state generation task using multiple hyperparameter sets, with Set 1 being the main text configuration and all unlisted parameters kept the same across sets. Set 2: $C_{f1} = 3.2, C_{f2} = 0.7, C_e = 0.3, C_{iw} = 0.25$; Set 3: $C_{g1} = -1.7, C_{f1} = 3.2, C_{iw} = 0.15, C_{g3} = -0.7$; Set 4: $C_{g2} = -1.1, C_{iw} = 0.22, C_{g4} = -0.6$.

The results are shown in Fig. 8. The training curves demonstrate that our method consistently outperforms the QAS-PPO baseline across all tested configurations. Although there are minor variations in convergence speed, the final performance remains stable and does not deviate significantly from the original experiment. This confirms that our method is robust to reasonable variations in the choice of hyperparameters.

APPENDIX B: TRAINING INFORMATION

This section provides additional details on the training procedure. Our training environment is built upon cirq (v1.4.1), stable-baselines3 (v2.4.0), and gymnasium (v1.0.0). We used the default hyperparameters for the PPO algorithm, with the exceptions of those specified in the main text. The complete set of PPO parameters is listed in Table I.

Furthermore, to enhance reproducibility and provide a transparent view of performance, we present the detailed experimental data for both our proposed method and the baseline models discussed in the main text. The results are visualized in the subsequent figures. For each method, we plot the raw performance curves from all independent runs, along with their corresponding mean curve and standard deviation (represented by the shaded area). The noisy environment is shown in Fig. 9, and noise-free experiments are shown in Fig. 10. This allows for a direct inspection of the performance distribution and stability of each approach.

APPENDIX C: ANOTHER REWARD DESIGN

In this section, we provide another dense reward design which is only guided by fidelity (F) and can be considered as reward shaping. The reward function is

$$r = F_t - F_{t-1}. \tag{C1}$$

It means that the reward is determined by the difference in fidelity between the current state and the previous state. Although we describe the inappropriateness of such a design (in Sec. III), here we test this reward function in a three-bit system like in the main text.

To further validate the design of our proposed reward function, we conducted a comparative analysis using a simplified reward function based solely on dense fidelity. In this setup, the agent receives a reward at each step proportional to the increase in fidelity toward the target state. For fairness, all hyperparameters were kept identical to those used in the main

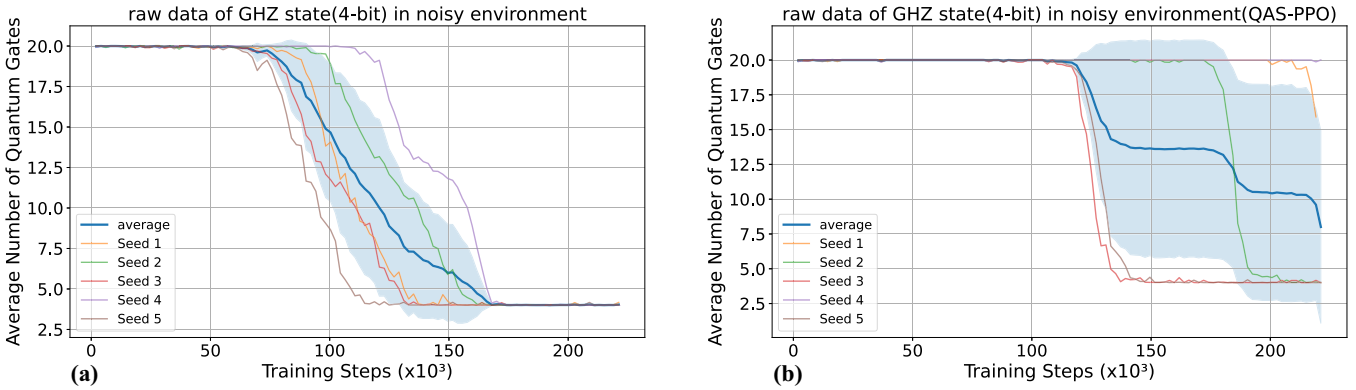


FIG. 9. Detailed performance comparison under noisy conditions for the four-qubit GHZ state generation task. This figure provides the raw experimental data corresponding to the noisy simulation results presented in Fig. 7. Both subplots show raw performance curves from five independent runs, their collective mean (blue line), and standard deviation (shaded area). (a) Performance of our method. (b) Performance of the QAS-PPO baseline.

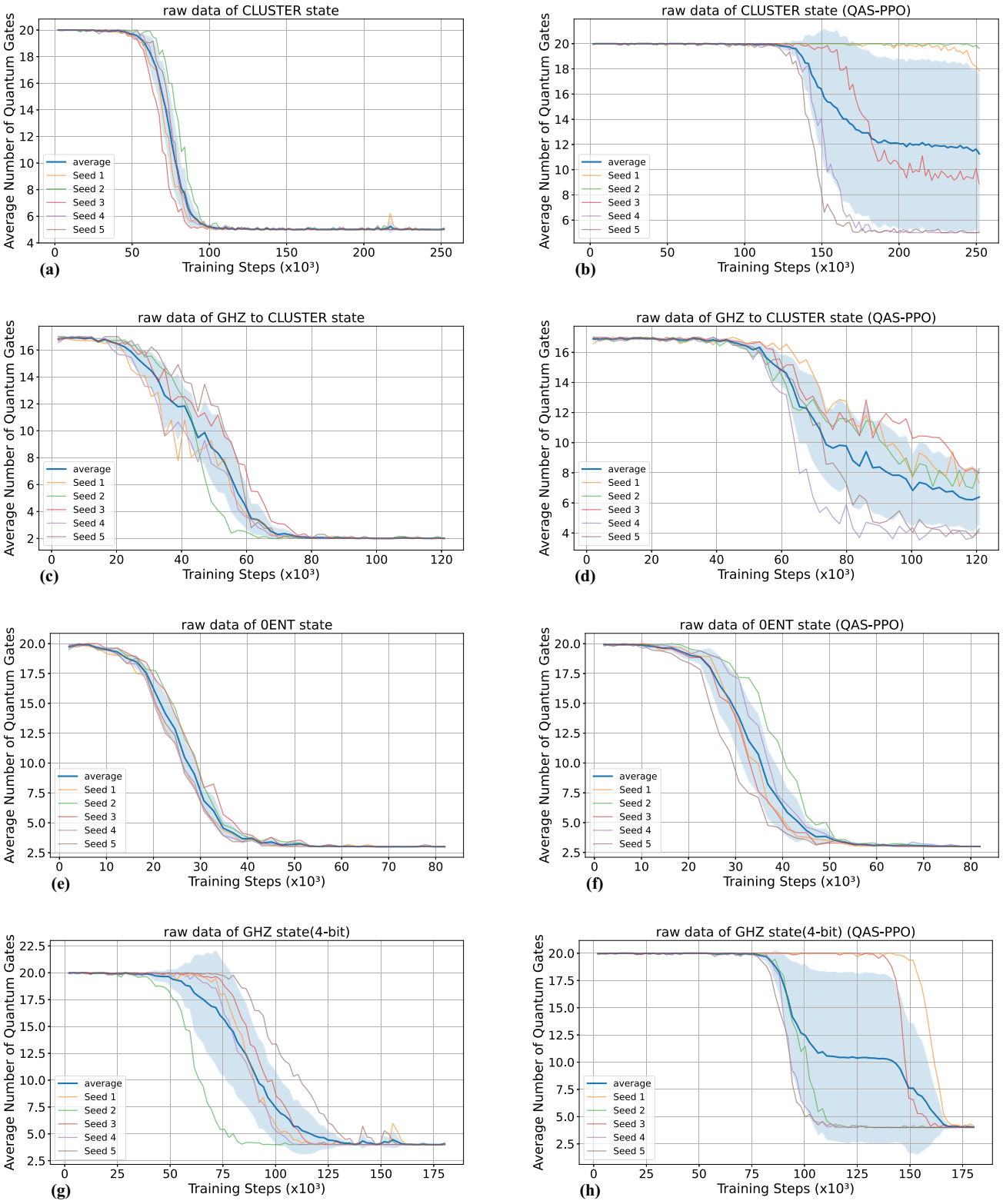


FIG. 10. Detailed performance comparison for noiseless simulations across four different quantum state generation tasks. This figure provides the raw experimental data corresponding to the results presented in Figs. 4 and 6. For each subplot, the left column shows the performance of our method, and the right column shows the performance of the QAS-PPO. All plots visualize raw data from five independent runs, their average value (blue line), and standard deviation (shaded area). The tasks are (a) and (b) cluster state, (c) and (d) GHZ to cluster state, (e) and (f) zero-entanglement state, and (g) and (h) four-bit GHZ state.

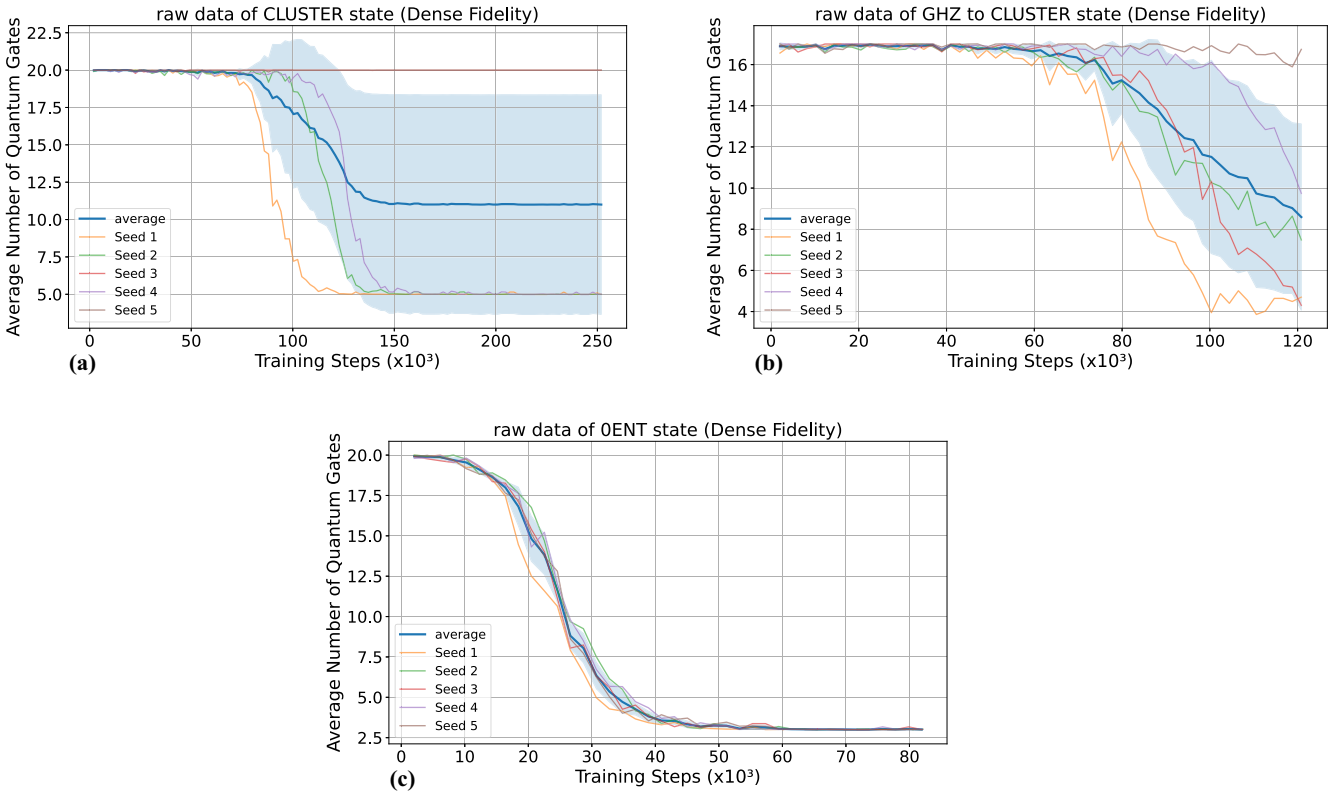


FIG. 11. Performance analysis of an agent trained with an alternative, dense, fidelity-based reward function. The figure shows the results across three distinct tasks, with each subplot displaying raw performance curves from five independent runs, their average (blue line), and standard deviation (shaded area). The results highlight the challenges associated with this reward, including slower convergence and high variance, particularly on more complex tasks. The tasks shown are (a) cluster state, (b) GHZ to cluster state, and (c) 0ENT state.

text experiments. The results for this approach on Tasks (a)–(c) are presented in Fig. 11.

The agent trained with the dense fidelity reward exhibited significant performance challenges. As shown in Figs. 11(a) and 11(b), it demonstrates substantially slower convergence and high performance variance across different random seeds, indicating a strong sensitivity to initialization. This is particularly evident in Task (a), where two of the five runs failed to learn to reduce the quantum gate count. Similarly, in Task (b), this approach, much like the QAS-PPO baseline, was unable to find the optimal solution with the minimum number of gates. While the agent achieved acceptable results on the relatively simple Task (c) Fig. 11(c), its limitations on more complex tasks are apparent.

This phenomenon demonstrates the existence of what can be termed a "fidelity trap," a scenario where an agent, guided solely by fidelity, converges to suboptimal solutions. These solutions may be resource inefficient or structurally incorrect despite having high fidelity scores. This observation underscores the necessity of incorporating entanglement as supplementary physical information in our reward function. The entanglement term can guide the agent to recognize the correct structural properties of the target state, thereby helping it to avoid these fidelity traps and find globally superior solutions.

[1] H.-K. Lau and M. B. Plenio, Universal quantum computing with arbitrary continuous-variable encoding, Phys. Rev. Lett. **117**, 100501 (2016).

[2] I. M. Georgescu, S. Ashhab, and F. Nori, Quantum simulation, Rev. Mod. Phys. **86**, 153 (2014).

[3] M. D'Arcangelo, L.-P. Henry, L. Henriet, D. Loco, N. Gouraud, S. Angebault, J. Sueiro, J. Forêt, P. Monmarché, and J.-P. Piquemal, Leveraging analog quantum computing with neutral atoms for solvent configuration prediction in drug discovery, Phys. Rev. Res. **6**, 043020 (2024).

[4] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, Low-depth quantum simulation of materials, Phys. Rev. X **8**, 011044 (2018).

[5] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[6] A. Kundu, P. Bedełek, M. Ostaszewski, O. Danaci, Y. J. Patel, V. Dunjko, and J. A. Miszczak, Enhancing variational quantum state diagonalization using reinforcement learning techniques, New J. Phys. **26**, 013034 (2024).

[7] E. Ye and S. Y.-C. Chen, Quantum architecture search via continual reinforcement learning, arXiv:2112.05779.

[8] Y. Du, T. Huang, S. You, M.-H. Hsieh, and D. Tao, Quantum circuit architecture search for variational quantum algorithms, npj Quantum Inf. **8**, 62 (2022).

[9] W. Zhu, J. Pi, and Q. Peng, A brief survey of quantum architecture search, in *Proceedings of the 6th International Conference on Algorithms, Computing and Systems, ICACS'22* (Association for Computing Machinery, New York, 2023).

[10] D. Martyniuk, J. Jung, and A. Paschke, Quantum architecture search: A survey, in *2024 IEEE International Conference on Quantum Computing and Engineering (QCE)*, Vol. 1 (IEEE, New York, 2024), pp. 1695–1706.

[11] Y. J. Patel, A. Kundu, M. Ostaszewski, X. Bonet-Monroig, V. Dunjko, and O. Danaci, Curriculum reinforcement learning for quantum architecture search under hardware errors, arXiv:2402.03500.

[12] S. Y.-C. Chen, Quantum reinforcement learning for quantum architecture search, in *Proceedings of the 2023 International Workshop on Quantum Classical Cooperative, QCCC'23* (Association for Computing Machinery, New York, 2023), pp. 17–20.

[13] S.-X. Zhang, C.-Y. Hsieh, S. Zhang, and H. Yao, Differentiable quantum architecture search, Quantum Sci. Technol. **7**, 045023 (2022).

[14] X. Zhu and X. Hou, Quantum architecture search via truly proximal policy optimization, Sci. Rep. **13**, 5157 (2023).

[15] Y. Sun, Z. Wu, Y. Ma, and V. Tresp, Quantum architecture search with unsupervised representation learning, arXiv:2401.11576.

[16] E.-J. Kuo, Y.-L. L. Fang, and S. Y.-C. Chen, Quantum architecture search via deep reinforcement learning, arXiv:2104.07715.

[17] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT, Cambridge, 2018).

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, Proximal policy optimization algorithms, arXiv:1707.06347.

[19] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Harley, T. P. Lillicrap, D. Silver, and K. Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, edited by M. F. Balcan and K. Q. Weinberger, Proceedings of Machine Learning Research Vol. 48 (PMLR, New York, NY, 2016), pp. 1928–1937.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, Playing Atari with deep reinforcement learning, arXiv:1312.5602.

[21] J. Fan, A review for deep reinforcement learning in Atari: Benchmarks, challenges, and solutions, arXiv:2112.04145.

[22] L. Kaiser, M. Babaeizadeh, P. Miłos, B. Osiński, R. H. Campbell, K. Czechowski, D. Erhan, C. Finn, P. Kozakowski, S. Levine, A. Mohiuddin, R. Sepassi, G. Tucker, and H. Michalewski, Model-based reinforcement learning for Atari, in *8th International Conference on Learning Representations, Addis Ababa, Ethiopia* (OpenReview.net, 2020).

[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, Human-level control through deep reinforcement learning, Nature (London) **518**, 529 (2015).

[24] H. Van Hasselt, A. Guez, and D. Silver, Deep reinforcement learning with double Q-learning, in *Proceedings of the AAAI Conference on Artificial Intelligence* (AAAI Press, Palo Alto, CA, 2016).

[25] W. Ye, S. Liu, T. Kurutach, P. Abbeel, and Y. Gao, Mastering Atari games with limited data, *35th Conference on Neural Information Processing Systems (NeurIPS)* (Curran Associates, New York, 2021), Vol. 34, pp. 25476–25488.

[26] B. M. Babayan, N. Uchida, and S. J. Gershman, Belief state representation in the dopamine system, Nat. Commun. **9**, 1891 (2018).

[27] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, Deep reinforcement learning: A brief survey, IEEE Signal Proces. Mag. **34**, 26 (2017).

[28] P. Andreasson, J. Johansson, S. Liljestrand, and M. Granath, Quantum error correction for the toric code using deep reinforcement learning, Quantum **3**, 183 (2019).

[29] D. Fitzek, M. Eliasson, A. F. Kockum, and M. Granath, Deep Q-learning decoder for depolarizing noise on the toric code, Phys. Rev. Res. **2**, 023230 (2020).

[30] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, Optimizing quantum error correction codes with reinforcement learning, Quantum **3**, 215 (2019).

[31] Y. Zeng, Z.-Y. Zhou, E. Rinaldi, C. Gneiting, and F. Nori, Approximate autonomous quantum error correction with reinforcement learning, Phys. Rev. Lett. **131**, 050601 (2023).

[32] L. Domingo Colomer, M. Skotiniotis, and R. Muñoz-Tapia, Reinforcement learning for optimal error correction of toric codes, Phys. Lett. A **384**, 126353 (2020).

[33] S. Z. Baba, N. Yoshioka, Y. Ashida, and T. Sagawa, Deep reinforcement learning for preparation of thermal and prethermal quantum states, Phys. Rev. Appl. **19**, 014068 (2023).

[34] X. Zhao, Y. Zhao, M. Li, T. Li, Q. Liu, S. Guo, and X. Yi, A strategy for preparing quantum squeezed states using reinforcement learning, Ann. Phys. **536**, 2400056 (2024).

[35] R. Porotti, A. Essig, B. Huard, and F. Marquardt, Deep reinforcement learning for quantum state preparation with weak nonlinear measurements, Quantum **6**, 747 (2022).

[36] R. H. He, R. Wang, S. S. Nie, and Z. M. Wang, Deep reinforcement learning for universal quantum state preparation via dynamic pulse control, EPJ Quantum Technol. **8**, 29 (2021).

[37] X.-M. Zhang, Z. Wei, R. Asad, X.-C. Yang, and X. Wang, When does reinforcement learning stand out in quantum control? A comparative study on state preparation, npj Quantum Inf. **5**, 85 (2019).

[38] S. Borah, B. Sarma, M. Kewming, G. J. Milburn, and J. Twamley, Measurement-based feedback quantum control with deep reinforcement learning for a double-well nonlinear potential, Phys. Rev. Lett. **127**, 190403 (2021).

[39] V. V. Sivak, A. Eickbusch, H. Liu, B. Royer, I. Tsioutsios, and M. H. Devoret, Model-free quantum control with reinforcement learning, Phys. Rev. X **12**, 011059 (2022).

[40] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Reinforcement learning in different phases of quantum control, Phys. Rev. X **8**, 031086 (2018).

[41] Y. Zeng, J. Shen, S. Hou, T. Gebremariam, and C. Li, Quantum control based on machine learning in an open quantum system, Phys. Lett. A **384**, 126886 (2020).

[42] T. Neema, S. Jha, and T. Sahai, Non-Markovian quantum control via model maximum likelihood estimation and reinforcement learning, arXiv:2402.05084.

[43] Z. An and D. L. Zhou, Deep reinforcement learning for quantum gate control, Europhys. Lett. **126**, 60002 (2019).

[44] Y. Baum, M. Amico, S. Howell, M. Hush, M. Liuzzi, P. Mundada, T. Merkh, A. R. R. Carvalho, and M. J. Biercuk, Experimental deep reinforcement learning for error-robust gate-set design on a superconducting quantum computer, PRX Quantum **2**, 040324 (2021).

[45] H. Nam Nguyen, F. Motzoi, M. Metcalf, K. Birgitta Whaley, M. Bukov, and M. Schmitt, Reinforcement learning pulses for transmon qubit entangling gates, Mach. Learn.: Sci. Technol. **5**, 025066 (2024).

[46] R. J. Williams, Simple statistical gradient-following algorithms for connectionist reinforcement learning, Mach. Learn. **8**, 229 (1992).

[47] F. Pardo, A. Tavakoli, V. Levdik, and P. Kormushev, Time limits in reinforcement learning, in *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80, edited by J. Dy and A. Krause (PMLR, Cambridge, MA, 2018), pp. 4045–4054.

[48] D. Arumugam, P. Henderson, and P.-L. Bacon, An information-theoretic perspective on credit assignment in reinforcement learning, arXiv:2103.06224.

[49] S. Singh, R. L. Lewis, and A. G. Barto, Where do rewards come from, Proc. Ann. Meet. Cogn. Sci. Soc. **49**, 2601 (2009).

[50] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, Concrete problems in AI safety, arXiv:1606.06565.

[51] D. Dewey, Reinforcement learning and the reward engineering principle, in *AAAI Spring Symposia* (AAAI Press, Palo Alto, CA, 2014).

[52] J. G. G. de Oliveira, Jr., J. G. Peixoto de Faria, and M. C. Nemes, Residual entanglement and sudden death: A direct connection, Phys. Lett. A **375**, 4255 (2011).

[53] K. Cao, Z.-W. Zhou, G.-C. Guo, and L. He, Efficient numerical method to calculate the three-tangle of mixed states, Phys. Rev. A **81**, 034302 (2010).

[54] V. Coffman, J. Kundu, and W. K. Wootters, Distributed entanglement, Phys. Rev. A **61**, 052306 (2000).

[55] M. Ma, Y. Li, and J. Shang, Multipartite entanglement measures: A review, Fundam. Res. **55**, 145303 (2024).

[56] J. de Jong, F. Hahn, N. Tcholtchev, M. Hauswirth, and A. Pappa, Extracting GHZ states from linear cluster states, Phys. Rev. Res. **6**, 013330 (2024).

[57] R. Islam, R. Ma, P. M. Preiss, M. Eric Tai, A. Lukin, M. Rispoli, and M. Greiner, Measuring entanglement entropy in a quantum many-body system, Nature (London) **528**, 77 (2015).

[58] Y.-H. Liu, Q.-S. Tan, L.-M. Kuang, and J.-Q. Liao, Deterministic generation of nonclassical mechanical states in cavity optomechanics via reinforcement learning, Phys. Rev. A **111**, 053517 (2025).

[59] H. Wang, Y. Ding, J. Gu, Y. Lin, D. Z. Pan, F. T. Chong, and S. Han, Quantumnas: Noise-adaptive search for robust quantum circuits, in *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)* (IEEE, New York, 2022), pp. 692–708.

[60] Y. Huang, Q. Li, X. Hou, R. Wu, M.-H. Yung, A. Bayat, and X. Wang, Robust resource-efficient quantum variational ansatz through an evolutionary algorithm, Phys. Rev. A **105**, 052414 (2022).

[61] Z. Xu, W. Shang, S. Kim, A. Bobbitt, E. Lee, and T. Luo, Quantum-inspired genetic algorithm for designing planar multilayer photonic structure, NPJ Comput. Mater. **10**, 257 (2024).

[62] X. Bi, Code for "General-purpose quantum architecture search based on deep reinforcement learning (version 1.0.0) (2025)", https://github.com/vojhs/QAS-entanglement.