Curriculum-Based Deep Reinforcement Learning for Quantum Control

Hailan Ma, Daoyi Dong, Senior Member, IEEE, Steven X. Ding, and Chunlin Chen, Senior Member, IEEE

Abstract—Deep reinforcement learning (DRL) has been recognized as an efficient technique to design optimal strategies for different complex systems without prior knowledge of the control landscape. To achieve a fast and precise control for quantum systems, we propose a novel DRL approach by constructing a curriculum consisting of a set of intermediate tasks defined by fidelity thresholds, where the tasks among a curriculum can be statically determined before the learning process or dynamically generated during the learning process. By transferring knowledge between two successive tasks and sequencing tasks according to their difficulties, the proposed curriculum-based DRL (CDRL) method enables the agent to focus on easy tasks in the early stage, then move onto difficult tasks, and eventually approaches the final task. Numerical comparison with the traditional methods [gradient method (GD), genetic algorithm (GA), and several other DRL methods] demonstrates that CDRL exhibits improved control performance for quantum systems and also provides an efficient way to identify optimal strategies with few control pulses.

Index Terms—Curriculum learning, deep reinforcement learning (DRL), quantum control.

NOMENCLATURE

| S      | State space. |
| A      | Action space. |
| P      | State transition probability. |
| π      | Policy function. |
| s_t    | State at time step t. |
| a_t    | Selected action according to policy π(s_t) at time t. |
| r_t    | Scalar reward at time step t. |
| F_t    | Fidelity at time step t. |
| e_t    | The tth experience. |
| T      | Terminal time of one episode. |
| dt     | Time duration for each control piece. |
| ε      | ε-greedy factor. |
| γ      | Discount factor. |
| Q(s,a) | State-action values. |
| H_0    | Free Hamiltonian. |
| H_m    | Control Hamiltonian operators. |
| N_{max} | Maximum steps. |
| | Unit complex vector. |
| ρ      | Density operator. |
| Y_l    | Orthogonal generators. |
| y      | Coherent vector. |
| χ      | Unified form for quantum states. |
| u      | Control fields. |
| M      | Number of control fields. |
| G_i    | Bounds for control fields. |
| J(u)   | Cost function. |
| k_1    | Slope of reward function. |
| k_2    | Bias of reward function. |
| v      | Task. |
| D(v)   | Difficulty for task v. |
| δ_t    | Gaps between two successive tasks. |
| ω      | Times of hitting a task. |
| SC     | Threshold for success count. |
| ρ       | Probability threshold. |

I. INTRODUCTION

Manipulating quantum systems with high efficiency [1]–[3] is a major challenge in developing quantum technology and provides recipes for many applications such as quantum computation [1], quantum communication, and quantum sensing [4]. Due to unique quantum characteristics such as quantum entanglements and quantum measurements that are different from classical systems, it is challenging to design control strategies for quantum systems. To achieve quantum operations with high efficiency, control methods, such as optimal control theory [3], closed-loop learning control algorithms [5], and quantum feedback control approaches [6], have been developed. In closed-loop learning control approach, for each trial of control, it is an open-loop control, while the control performance will be sent back to the learning algorithm to suggest a better control policy.
It has achieved great success in quantum control applications using different learning algorithms. Among them, gradient algorithms have been used for numerically finding an optimal field [7]. Evolutionary computing methods such as genetic algorithm (GA) and differential evolution (DE) have been utilized in optimizing molecular systems [8]–[11]. However, in many practical applications, the gradient information may not be easy to obtain and evolutionary algorithms usually involve a process of evolving a population and tend to be time-consuming when solving complex problems.

Machine learning has attracted increasing attention owing to its powerful computing capability and has been gradually applied in various quantum tasks in recent years [12]. Particularly, reinforcement learning (RL) [13] offers a considerable advantage in controlling systems without prior knowledge about the environment and it can be viewed as an alternative method to achieve a similar performance to dynamic programming, with less computation and without assuming perfect system models. For example, a fidelity-based probabilistic Q-learning approach has been proposed to achieve a better balance between exploitation and exploration when dealing with quantum systems [14]. It has also been found that RL-aided approaches succeed in identifying variational protocols with nearly optimal fidelity [15]. However, for systems with high-dimensional inputs, it is impractical to store a $Q(s, a)$ table. As such, deep reinforcement learning (DRL) has been recognized as a universal data-driven tool for complex systems [16]–[19] by combining RL and deep learning [20], [21] and has achieved efficient control of different quantum systems [22]–[32]. For example, a network-based “agent” is designed to protect a collection of qubits against noise [26]. With the help of DRL, nearly extreme spin squeezing with a one-axis twisting interaction is achieved using a handful of rotation pulses [29]. In addition, DRL realizes efficient and precise gate control [24] and exhibits strong robustness when dealing with parameter fluctuations [25]. Besides, DRL methods are also employed for violating various Bell inequalities [28], and designing generalizable control for quantum parameter estimation [27], [32].

When optimizing the control fields using RL, the learning process is usually quite slow due to complex dynamics, or sparse reward signals. The early transitions are likely to terminate on states that are easy to reach, and those states that are difficult to reach are usually found later in the training process [33]. However, these easy-to-reach states may not provide a reward signal, which might hinder the training process of the DRL agent. In addition, it usually takes millions of episodes to learn a good policy for a difficult problem, with many suboptimal actions taken during the learning process. When optimizing complex quantum systems, such as multi-level quantum systems, the complexity of the system dynamics scales sharply with the size of quantum systems [25]. Also, the dissipation part in a quantum system may irreversibly lead it away from an equilibrium state [34], greatly increasing the difficulty of manipulating its dynamics. It is highly desirable to design an efficient DRL approach to achieve a fast and reliable control of complex quantum systems.

Owing to the observation that students usually learn easy courses before they start to learn complex courses, curriculum learning [33], [35] has emerged as a general and powerful tool for solving difficult problems and has also been applied in optimizing RL agents [36], [37]. For example, the introduction of curriculum learning allows the RL agents to make the best use of transitions and achieves high scores when playing with complex games [36]. Actually, curriculums can be defined in different levels, including ordering of tasks or ordering of individual samples [36]. However, creating a curriculum at the sample level can be computationally difficult for large sets of tasks since the entire set of samples from a task (or multiple tasks) is typically not available ahead of time. In addition, the samples experienced in a task depend on the agent’s behavior policy, which can be influenced by previous tasks.

In this article, we introduce a task-level curriculum to the DRL control for quantum systems and propose a novel curriculum-based DRL (CDRL) with the purpose of achieving reliable and fast manipulation of quantum systems. In quantum control community, fidelity is widely used to evaluate the similarity (closeness) between two quantum states [1]. Here, a task is defined by a threshold of fidelity (also called as a target fidelity), which represents the difficulty of one task. In particular, two methods of generating tasks for a curriculum are introduced, including presetting static fixed tasks using empirical knowledge and dynamically generating tasks based on the performance of the agent, which do not significantly add computational burden compared with the traditional DRL methods. By sequencing a set of tasks with increasing difficulties and reusing knowledge between different tasks, the RL agent is able to focus on easy tasks at the early stage, gradually move onto difficult ones after grasping basic skills, and therefore achieve the final task.

In CDRL, a target fidelity is closely related to each task, which allows for an early termination of each episode once a satisfactory effect is achieved, enabling the agent to search for shorter control pulses compared with the traditional DRL methods. Such a mechanism makes it possible to manipulate quantum systems within short time durations thus protecting systems against decoherence. This is particularly significant for open quantum systems, where dissipation typically drives the system to a mixed state and an optimal control might keep it from the maximum entropy state for some specific dissipation channels [38]. From this respective, the proposed method is an attempt to utilize machine learning techniques to realize efficient manipulation of quantum systems, which not only saves operation time but also keeps systems away from unwanted decoherence [39].

In this article, we focus on a basic and crucial issue of quantum state preparation, which aims at steering a quantum system from an initial state toward a target state. To test the effectiveness of the proposed method, numerical simulations on closed quantum systems and open quantum systems are implemented and compared with traditional optimization methods. The main contributions of this article are summarized as follows.
1) A CDRL method is proposed for quantum systems where tasks among a curriculum are defined by fidelities and are linearly sequenced with increasing fidelities. 

2) Numerical comparison with GA, gradient method (GD), and several other DRL methods suggests that CDRL has potential to design effective control strategies with high fidelities for both closed and open quantum systems. 

3) CDRL has the advantage of searching for shorter control pulses, thus providing insights for utilizing machine learning techniques to realize efficient manipulation of quantum systems, which not only saves operation time but also keeps systems’ robustness against unwanted decoherence. 

The rest of this article is organized as follows. Section II introduces several basic concepts about RL, curriculum learning, and quantum systems. In Section III, CDRL is presented in detail with the curriculum design and the algorithm implementation for quantum systems. Numerical results for both closed and open quantum systems are presented in Section IV. Concluding remarks are drawn in Section V.

II. PRELIMINARIES

A. Reinforcement Learning

1) Markov Decision Process: RL is commonly studied based on the framework of the Markov decision process (MDP). An MDP can be described by a tuple of \((S, A, P, R, \gamma)\) [13], where \(S\) is the state space, \(A\) is the action space, \(P: S \times A \times S \rightarrow [0, 1]\) is the state transition probability, \(R: S \times A \rightarrow \mathbb{R}\) is the reward function, and \(\gamma \in [0, 1]\) is the discount factor. A policy is a mapping from the state space \(S\) to the action space \(A\). At each time step \(t \in [0, T]\), where \(T\) is the terminal time, the agent forms the state \(s_t\), takes an action \(a_t \in A\) according to a policy \(\pi: s_t = \pi(s_t)\), transits to the next state \(s_{t+1}\) and gets a scalar reward signal \(r_t\) from the environment. RL aims at determining an optimal action \(a^*_t\) at each state \(s_t\) so as to maximize the cumulative discounted future rewards of return \(R_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}\).

2) Deep Q-Network: Similar to Q-learning [40], deep Q-network (DQN) is a value-based DRL method, and it employs a function with parameters \(\xi\) to approximate \(Q\) value for each state-action pair \(Q(s, a)\), i.e., \(Q(s, a; \xi) \approx Q(s, a)\). Such a network can be trained by minimizing the loss function as: \(\text{Loss}(\theta) = \{Q_{\text{target}} - Q(s, a; \xi^-)\}^2\) with \(Q_{\text{target}} = r + \gamma \max_{a'} Q(s', a'; \xi^-)\) [41], where \(s'\) is the next state after taking action \(a\) at state \(s\). \(\xi^-\) denotes the parameters of the target network which is fixed during the computation of \(Q_{\text{target}}\) and is usually updated after some training iterations. Differentiate the loss function with respect to \(\xi\), the gradient is formulated as

\[
\nabla_\xi \text{Loss} = \left[ r + \gamma \max_{a'} Q(s', a'; \xi^-) - Q(s, a; \xi^-) \right] \nabla_\xi Q(s, a; \xi^-). 
\]

(1)

After training the network, an optimal control strategy is generated by selecting the action with the largest Q-function, i.e., \(a^* = \max_a Q(s, a)\).

B. Curriculum Learning

Curriculum learning is a methodology to optimize the order in which experiences are accumulated by the agent [33]. During this process, knowledge is transferred from easy tasks to difficult ones, which helps to achieve an enhanced performance on a hard problem or reduce the time it takes to converge to an optimal policy. A curriculum is commonly defined as an ordering of tasks; while an ordering of individual experience samples can also be regarded as a curriculum at a more fundamental level. One task building upon knowledge gained from multiple source tasks is also acceptable and useful, which is similar to the case in human education, where courses can build off multiple prerequisites.

There have been growing interests in exploring how to leverage curriculum learning to speed up the RL agents. To achieve this, there are three key elements to be considered.

1) Task Generation: A set of intermediate tasks can be prespecified ahead of time or dynamically generated based on the previous learning performance.

2) Sequencing: Similar to task generation, it can be predefined or automatically realized.

3) Knowledge Transfer: Before moving to the next task, reusable knowledge acquired from one task is required to extract and pass to the next one [42], [43].

From this respective, designing a good curriculum depends on generating appropriate and useful tasks to transfer or reuse knowledge between different tasks.

C. Quantum Dynamics

The state of a finite-dimensional closed quantum system can be represented by a unit complex vector \(|\psi\rangle\) and its dynamics can be described by the Schrödinger equation

\[
\frac{d}{dt}|\psi(t)\rangle = -\frac{i}{\hbar}(H_0 + \sum_{m=1}^{M} u_m(t)H_m)|\psi(t)\rangle
\]

(2)

where \(\hbar\) is the reduced Planck constant (hereafter, we set \(\hbar = 1\)), \(H_0\) denotes the time-independent free Hamiltonian of the system, and the control Hamiltonian operators \(H_m\) represent the interaction of the system with the control fields. The fidelity of two quantum states is defined as

\[
F(|\psi_1\rangle, |\psi_2\rangle) = |\langle \psi_1 | \psi_2 \rangle|^2. 
\]

(3)

Denoting quantum states in vector forms as \(|\psi_1\rangle = (a_1, ..., a_n)^T\), \(\sum_i |a_i|^2 = 1\) and \(|\psi_2\rangle = (b_1, ..., b_n)^T\), \(\sum_i |b_i|^2 = 1\), their fidelity can be formulated as \(F(|\psi_1\rangle, |\psi_2\rangle) = \sum_i |a_i^*b_i|^2 \in [0, 1]\). A large (small) value of fidelity means a high (low) similarity of two states. We adopt the fidelity between the actual state \(|\psi(T)\rangle\) and the target state \(|\psi_f\rangle\) to evaluate the control performance [2], i.e., \(J(u) = |\langle \psi(T) | \psi_f \rangle|^2\).

In practical applications, a quantum system usually suffers from the interaction with its environments and is then regarded as an open quantum one [34]. In such a case, the state of the quantum system is described by a Hermitian, positive semidefinite matrix \(\rho\) satisfying \(\text{Tr}(\rho) = 1\) and \(\text{Tr}(\rho^2) \leq 1\).
Its dynamics under Markovian approximation can be described by the Lindblad master equation [3]

\[
i\dot{\rho}(t) = \left[ H_0 + \sum_{m=1}^{M} u_m(t) H_m, \rho(t) \right] + \sum_{k} \eta_k D[L_k](\rho(t))
\]  

with \( D[L_k](\rho) = L_k \rho L_k^\dagger - (1/2)L_k^\dagger L_k \rho - (1/2)\rho L_k^\dagger L_k \), where \( L_k \) represents the Lindblad operators and the coefficients \( \eta_k \geq 0 \) characterize the relaxation rates. For an \( n \)-level quantum system, denote a set of orthogonal generators as \( \{Y_l^N \}_{l=1}^N \) of \( SU(n) \) with each generator satisfies: 1) \( \text{tr}(Y_l) = 0 \); 2) \( Y_l^N = Y_l \); and 3) \( \text{tr}(Y_l Y_m) = 2\delta_{lm} \).

Then, a coherent Bloch vector can be defined as \( y(t) := (y_1(t), \ldots, y_N(t))^T \), and then a density operator can be rewritten as \( \rho(t) = I/n + (1/2) \sum_{l=1}^{N} y_l(t) Y_l \).

The evolution of \( y(t) \) is deduced as

\[
\dot{y}(t) = (\mathcal{L}_{H_0} + \mathcal{L}_D)y(t) + \sum_{m=1}^{M} u_m(t) \mathcal{L}_{H_m} y(t) + s_0, \quad y(0) = y_0
\]

(5)

where \( \mathcal{L}_{H_0} \) and \( \mathcal{L}_{H_m} \) are \( N \times N \) superoperators and the inhomogeneous source term \( s_0 \) is a \( N \times 1 \) column vector [44]. The goal is to steer the system from an initial state \( y_0 \) to a final state \( y(T) \) as close to a target state \( y_f \) as possible, and we can take the following cost function [44]:

\[
J(u) = 1 - \frac{n}{8(n-1)} \| y_f - y(T) \|^2
\]

(6)

where \( y(T) \) is dependent on \( u \). The constant \( n/8(n-1) \) is introduced to guarantee \( J(u) \in [0, 1] \) and \( J(u) \) in (6) plays the same role as quantum fidelity. To simplify the following notation, we introduce a unified form \( \chi(t) \) to represent the states for closed systems |\( \psi(t) \rangle \) and the states for open systems |\( y(t) \rangle \). A new function \( F(\chi_1, \chi_2) \) is introduced to measure the similarity between two states \( \chi_1 \) and \( \chi_2 \). Hereafter, the similarity metric is called as fidelity in this work. The control performance for quantum systems can be unified as \( J(u) = F(\chi(T), \chi_f) \).

III. CDRL FOR QUANTUM CONTROL

In this section, the framework of CDRL for quantum control is presented, and the key elements for designing a curriculum are elaborated in detail, and then the ingredients of applying DRL to quantum systems are provided. Finally, the implementation of CDRL for quantum control is summarized.

A. Framework of CDRL

Curriculum learning aims at generating appropriate and useful tasks and reusing knowledge between different tasks. For quantum systems, the difficulty of one task can be defined using a fidelity threshold that reflects the efficiency of state preparation. Hence, for a quantum system, one task can be to drive the system to an (intermediate) state exceeding a certain fidelity value (see Definition 1). Furthermore, a curriculum can be defined as a sequential list of tasks with increasing fidelity thresholds (see Definition 2). The framework of CDRL is presented in Fig. 1, which can be summarized as two aspects: 1) curriculum construction and management and 2) train the DRL agent for one task.

In Fig. 1(a), a set of tasks is constructed and then compose a curriculum. During the learning process, once a task is generated, it is sent to the RL agent for learning. Meanwhile, the training performance collected from the RL agent is sent back to the curriculum agent for determining whether a stop criterion is met. Before moving to the next task, experiences acquired from the previous task are collected and passed to the next one. In addition, some measures are taken to excite the RL agent to maintain a certain exploration for a new task.

In Fig. 1(b), the RL agent trains its network by trial-and-error for each task. After receiving a task, the RL agent observes its current state \( s_t \) and determines an action \( a_t \) by inferring from the deep neural networks. The control fields used \( u(t) = \{u_m(t), m = 1, 2, \ldots, M \} \) based on \( a_t \) are performed on the quantum system, resulting in a next state \( s_{t+1} \) and fidelity \( F'_t \) and reward signal \( r_t \). The transition \( (s_t, a_t, r_t, s_{t+1}, F'_t) \) is collected and stored into a memory pool. Meanwhile, a batch of samples is selected from the memory and fed into the neural networks to update their parameters. It is worthy to note that the experiences from the past task are collected as reusable knowledge for the next task.

B. Curriculum Design for Quantum Systems

1) Task and Curriculum: Most quantum control problems can be generalized as a goal task to drive a quantum system from an initial state to a target state as close as possible with a given time period. Recall the closeness between two quantum states is usually measured by fidelity. There are multiple states \(|\psi'\rangle = \exp(i\theta)|\psi(T)\rangle, \theta \in R \) that can achieve the same fidelity, i.e., \( F(|\psi'(T)\rangle, |\psi_f\rangle) = F(|\psi'\rangle, |\psi_f\rangle) \), which means that \(|\psi'\rangle \) is equivalent to \(|\psi(T)\rangle \) neglecting the global phase. According to (6), there exist multiple states \( y(T) \) that achieve the same similarity with \( y_f \) for open systems. For ideal fidelity of 1, the final states are not unique, and the states in the last but one step can be multiple when tracing one step backward. Hence, there exist multiple branches or trajectories that result in final states with the same fidelity. When designing a curriculum for quantum systems, we do not take the actual physical states as the intermediate tasks. Instead, a target fidelity is utilized to distinguish between different tasks and we have the following definition.

Definition 1 (Task): For a quantum system with a target state \( \chi_f \), one task \( v \) is defined as the process of driving the system from an initial state \( \chi_0 \) to an actual state \( \chi(T) \) with \( F(\chi(T), \chi_f) \geq D(v) \). \( D(v) \) represents the difficulty of the task \( v \).

Recall that a curriculum is composed of a set of tasks. The ordering of tasks among a curriculum is similar to the way that vertices and edges form a graph. Hence, we adopt the concept of graph to define a curriculum. In particular, a task is defined as a vertex of a graph, and the relationship between two successive tasks is used to define an edge of a graph.

To achieve a final goal, it is natural to sequence the tasks according to their difficulties. Denote the \( i \)th task as a vertex

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 1. Framework of CDRL approach for quantum control. (a) Curriculum construction and management. The curriculum agent generates a task for training the RL agent and utilizes the performance of the RL agent to determine whether a stop criterion is satisfied. Once the previous task has been achieved, knowledge is transferred between different tasks by reusing past samples. Meanwhile, measures are taken to excite the RL agent before the subsequent task is scheduled. (b) Train the RL agent for one task. At each time step $t$, the RL agent observes its current state $s_t$ (step 1) and suggests a control action $a_t$ (step 2), which is mapped to the control fields $\{\alpha_t(t)\}$ (step 3). Then, the quantum system takes the proposed control strategy and obtains the next state $s_{t+1}$, with fidelity $F_t$ and reward signal $r_t$ (step 4). The transition $e_t = (s_t, a_t, r_t, s_{t+1}, F_t)$ is put into a large memory buffer (step 5). Finally, a batch of transitions is selected from the buffer and then fed into the networks to update its weights or parameters (step 6).

Then a directed edge $<v_i, v_j>$ can be utilized to denote the relationship between two tasks, which means that samples associated with $v_i$ should be trained before samples associated with $v_j$. By sequencing a set of different tasks in a similar fashion, a task-level curriculum can be in the following way.

Definition 2 (Curriculum): Let the task set be $V$. A curriculum can be defined as a directed acyclic graph $C = (V, E)$, where $V$ is the set of vertices, $E \subseteq \{(v_i, v_j)\mid (v_i, v_j) \in V \times V \land D(v_i) < D(v_j)\}$ is the set of directed edges.

Based on the definition of tasks, a curriculum for quantum problems can be simplified by sequencing tasks in a linear way. Hence, the graph is reduced to a chain and the whole learning can be described as Fig. 2, where the learning process is in line with a flow of tasks with increasing difficulties. For each task $v_i$, the agent tries to achieve a better performance than the target fidelity $D(v_i)$. After learning for some episodes, the agent moves on to the next task $v_{i+1}$. Although there are oscillations during each task, the average performance reveals that the agent has climbed to a higher point.

2) Task Generation for a Curriculum: During the construction of a curriculum, generating useful and appropriate tasks is a crucial procedure [45], [46] since the quantity of the created tasks has a strong impact on the search space and efficiency of curriculum sequencing algorithms. A common choice is to manually design a set of intermediate tasks such that transferring knowledge through them is beneficial. This is usually realized based on the empirical knowledge about the problem. Recently, there are efforts to automatically create tasks online using a set of rules or a generative process [46], even though these approaches may still rely on tuning hyperparameters, such as the number of tasks generated.

Recall that a task for quantum systems is determined by a fidelity and it can achieve any value that lies between $0$ and $1$ when arbitrary control laws are available. Hence, any value lying in $[0, 1]$ can be utilized as an indicator for one task. Moreover, tasks are usually ordered with increasing difficulties. Generally, the intermediate tasks for the curriculum should be ordered as a sequential chain, with each point related with a threshold fidelity. In particular, the preset tasks should be arranged in such a way that the front vertices are assigned with low target fidelities; while those vertices at the rear are attached with difficulties approaching $1$. Moreover, the difficulty of controlling quantum systems increases tremendously. When determining a set of tasks with increasing difficulties,
the gap between two successive tasks $\delta_i = D(v_{i+1}) - D(v_i)$ should be reduced with increasing $i$. It is natural to take advantage of some typical values of fidelities (such as 0.9, 0.99, 0.999) to generate a set of tasks before the learning process. 0.9 corresponds to an easy task and 0.99 means a higher requirement, and the ability to achieve 0.999 usually means an ideal case. Since a direct increase from 0.9 to 0.99 may not be achieved in the learning process, finely grained transitions are required to generate more tasks, such as 0.991, 0.992, ..., 0.999. We call it a static way to generate tasks for a curriculum.

Another way is to dynamically generate tasks during the learning process. To start a curriculum, an initial task should be assigned with an appropriate difficulty. For the remaining tasks, it is important to perform data statistics on the performance for the previous task to suggest a candidate fidelity for the next task. For example, it is useful to measure the “mean/median/max” of the fidelities during the past episodes. Likewise, the determination of the next task should follow the principle that the new task is assigned with higher difficulty compared with the previous task.

The static way is to assign tasks with fixed fidelities before the learning process, while the dynamic way features in assigning $D(v_{i+1})$ based on the performance (i.e., the actual fidelity at the end of each episode) during task $v_i$, which does not rely on knowing quantum dynamics. Hence, the determination of the difficulties of tasks does not rely on the dynamics of quantum systems while relates to fidelity for both closed and open quantum systems.

C. CDRL for Quantum Control

1) DRL for Quantum Control: When applying RL to quantum systems, a maximum number of control pieces is usually defined as $N_{\text{max}}$, which limits the maximum steps in one episode. However, the performance does not always increase with steps and may decrease to a low value after performing additional control pulses, especially when the previous step has achieved excellent performance. A natural solution is to terminate the current episode when certain requirements are satisfied [22], [23]. Since we already define a task using a fidelity threshold (see Definition 1), it is practical to terminate the current episode when the fidelity exceeds the difficulty of the current task. Then, we introduce the definition of smart episode, where the number of control pieces can be less than $N_{\text{max}}$. This measure enables the RL agent to search for shorter control pulses.

Definition 3 (Smart Episode): For an RL agent aiming at achieving task $v$, a smart episode is defined as a state-action chain $s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \ldots \xrightarrow{s_j} s_{j+1} \ldots$ with termination conditions as: 1) the number of steps taken reaches the maximum steps, i.e., $j = N_{\text{max}}$; or 2) the current performance surpasses the target fidelity, i.e., $F'(\chi_{j+1}, \chi_f) \geq D(v)$.

Another significant procedure for RL methods is to determine an appropriate reward signal [13]. Recalling that the performance in one episode is not necessarily increasing, giving reward signal at the end of the episode helps to avoid a performance decline [23], [24]. In addition, the log of infidelity is also a common metric in quantum control community [2], since the difficulty of controlling quantum systems increases tremendously with fidelity. In this work, the reward is calculated based on log of infidelity to attach a higher reward signal to a higher fidelity. Finally, the reward signal can be formulated as

$$r_j = \begin{cases} k_1 + k_2 \log_{10} (1 - F'_j) & \text{if } F'_j \geq D(v) \text{ or } j = N_{\text{max}} \\ 0 & \text{else} \end{cases}$$

(7)

where $k_2$ represents the slope of reward function, and $k_1$ acts as a bias. Actually, $k_1$ and $k_2$ can be adjusted according to the different fidelity intervals [22]. In principle, a large value of $k_2$ should be set for a large fidelity such as $[0.99, 0.999]$, while the value of $k_1$ guarantees that reward increases with fidelity between different intervals.

When training DRL agents, experience replay is utilized to store the agents’ past experiences into a big memory pool for replaying [47]. Generally, the memory pool is filled with past experiences that were met previously and the weights of the DRL agent have been learned well using these past experiences. When attempting a new task, the DRL agent takes actions based on the previous model, and replay transitions from the memory, which achieves the reuse of previous states and data. Considering that transitions may be more or less surprising, redundant, or task-relevant, we employ a smart store mechanism to make better use of significant samples. For each step, the fidelity between $s_{j+1}$ and the target state is informative, and we define a transition as $e_j = (s_j, a_j, r_j, s_{j+1}, F'_j)$. Here, $s_j$ is obtained by splitting its real and imaginary parts from the complex vector $|\psi\rangle$ using (10) for closed systems or from the Bloch vector $\mathbf{y}$ for open systems. $a_j$ can be obtained from (8). $r_j$ can be measured based on the fidelity information according to (7). If the older one has higher fidelity than the new one, the new sample should be moved onto the next pointer for replacement and storing. This helps retain some significant samples while storing the latest experiences into the memory. Such a practice is based on the hypothesis that good transitions are usually not ordered nearby and the next sample after a good sample is rarely a good one owing to the smart episode. Besides, this measure only takes effect with a certain probability.

2) Knowledge Transfer Between Different Tasks: After training the DRL agent for each task, it is important to transfer and reuse knowledge among different tasks. Given two tasks, the process of transferring knowledge can be summarized as four procedures in Fig. 3.

1) Exploration: When first jumping to $v_A$, it is usually difficult for the agent to achieve a fidelity larger than $D(v_A)$ and it requires a process of exploration by learning from trial-and-error to satisfy the requirement of $v_A$.

2) Storing Transitions: Meanwhile, those useful transitions are collected and stored in the memory pool.

3) Transferring Knowledge: Those transitions from task $v_A$ provide useful resources and can be replayed to train the agent when striving for the subsequent tasks. Before jumping to the next task, the memory pool is actually
filled with past experiences and the weights of the agent have been learned well.

4) **Exploitation:** When attempting to achieve task \( v_B \), the agent already has the capacity to achieve a fidelity \( F'(\chi', \mathcal{X}) \geq D(v_A) \) within \( n_a \) steps by repeating the past transitions.

In the early stage, the agent makes use of its past experiences to achieve a fidelity larger than \( D(v_A) \) without learning from scratch. After that, the agent begins to explore in the later stage. Continuing the trajectory that leads to \( F'(\chi, \mathcal{X}) \geq D(v_A) \), the agent attempts to explore additional steps with final state \( \chi' \), hopefully to achieve a better fidelity \( F'(\chi', \mathcal{X}) \geq D(v_B) \). This guarantees that the agent reviews the knowledge accumulated from task \( v_A \) and explores new transitions under task \( v_B \), which is a combination process of exploitation and exploration.

3) **Stop Criterion and Excitation:** When designing a curriculum for RL agents, it is important to set a suitable stopping criterion for each task. Typically, training is terminated when performance on a task or a set of samples has converged. However, training to convergence on an intermediate task is not always necessary [33]. In standard RL algorithms, the stopping criterion can be based on parameter updates or a fixed number of episodes, and we adopt the second one.

In practical applications, the performance of different episodes tends to vary frequently since the randomness or uncertainties in actions and the stochastic nature of batch sampling for updating parameters might result in fluctuations of the DRL agent. When dealing with quantum systems, the subsequent state varies greatly when taking one step following different actions, thus leading to great difference in the final performance. A fixed number of episodes may not be enough to guarantee good performance on one task and the process of winning a task may take more episodes for a hard task. Considering that the fidelity metric is adopted in the definition of tasks, we make the best use of the fidelity information to determine the stop criterion. For task \( v_i \), a nonnegative integer \( w \) is introduced to measure the time of hitting the task, that is to update \( w \leftarrow w + 1 \) once the current episode achieves \( F'_j \geq D(v_i) \) with \( j \leq N_{max} \). Besides, an integer \( SC \) is defined to terminate the current task when \( w \geq SC \).

During each task, the randomness factor for the DRL agent is usually decreased using a decay factor \( \zeta \in (0, 1) \) to guarantee a certain convergence. For example, the greedy factor \( \epsilon \) for discrete control cases tends to reach 0, or the disturbance amplitude for continuous control cases approaches nearly 0. To achieve a smooth transition between two tasks, an excitation operation is required to reset the randomness degree of the DRL agent to guarantee a certain exploration for the new task.

### D. Integrated Algorithm

The proposed CDRL features in using a set of tasks to help the DRL focus on easy tasks at an early stage and difficult ones at a later stage. The key to applying CDRL to quantum systems lies in determining a set of tasks with increasing difficulties. For the implementations of CDRL for quantum control, we take the approach of dynamically generating tasks as an example and adopt DQN to train the agent toward intermediate tasks owing to its wide applications [16].

When searching for the optimal control fields, the set of \( 2^M \) possible choices for the action vector are formulated as [24]

\[
\mathcal{A}(u) = \{u_1, \ldots, u_M\}, u_m \in \{-G_m, G_m\}, m = 1, \ldots, M
\]

(8)

where \( \{G_m > 0\} \) are preset bounds for each control. During the training process, the \( Q \) values for all possible actions, i.e., \( Q(s, a_1), Q(s, a_2), \ldots \) are predicted and the actual action is obtained in an \( \epsilon \)-greedy way. That is to select a random action \( a \), with probability \( \epsilon \) or to select \( a_i = \max_{a_j} Q(s_i, a) \) with probability \( (1-\epsilon) \). The randomness factor \( \epsilon \) reflects the greedy degree, which means that a high value favors exploration while a low value encourages exploitation.

Finally, an integrated CDRL algorithm using dynamically generated tasks and DQN is summarized as in Algorithm 1. The curriculum agent launches a task \( v_i \), with a difficulty \( D(v_i) \), to activate the DRL agent to learn a good policy through trial-and-error. During each step, the agent encounters one transition \( (s_j, a_j, r_j, s_{j+1}, F'_j) \). By evaluating \( F'_j \), the transition \( e_j \) is stored in the memory pool in the right order. Meanwhile, the DQN agent updates its weights by sampling a batch of those transitions. Meanwhile, the performance of each episode is collected and transmitted to the agent for the determination of the stop criterion. Once it is met, the curriculum agent schedules a new task based on the past performance. This procedure is carried out iteratively until the final task is achieved.

**Remark 1:** In practical implementations, the gap of difficulty between successive tasks, i.e., \( \delta_i = D(v_{i+1}) - D(v_i) \), should be evaluated to determine whether or not to terminate the learning process. For example, if \( \delta_i \) is below a small threshold (such as 0.00005), there is no need to introduce additional tasks, since the actual performance of one episode can be in principle larger than \( D(v_i) \). As for the case of manually generated tasks, it is required to judge whether or not the current task is too hard for the agent. For example, the
Algorithm 1 Algorithm Description for CDRL

**Input:** Initial state \( \chi_0 \), target state \( \chi_f \), maximum steps \( N_{\text{max}} \), decay period \( E \), \( \epsilon \)-decay factor \( \zeta \), batch size \( B \), memory size \( K \), count threshold \( SC \), probability threshold \( \mu_1 \)

**Output:** Optimal set of control pulses \( a_j = \max_a Q(s_j, a) \)

1. Build a DQN agent \( Q(s, a|z) \) and initialize \( i = 0 \), \( k = 0 \) and a memory pool \( M \);
2. Initialize task \( v_0 \) with an initial target fidelity \( D(v_0) \);
3. **while** curriculum not finished **do**
   4. Initialize the number of hitting a task as \( w = 0 \) and a null list \( L \) to store the past performance;
   5. **while** stop criterion for task \( v_i \) not met **do**
      6. Initialize step \( j = 0 \) and transform initial quantum state \( \chi_0 \) into \( s_0 \);
      7. **while** \( s_j \) is not termination state **do**
         8. Generate actions \( a_j \) based on \( Q(s_j, a_j|z) \) following the \( \epsilon \)-greedy action selection policy;
         9. Perform the control pulses obtained from \( a_j \), obtain next state \( s_{j+1} \) and fidelity \( F'_j = F'(\chi_j, \chi_f) \);
         10. Determine \( r_j \) based on \( F_j \) and \( D(v_i) \);
          11. if \( F'_j \geq D(v_i) \) or \( j \geq N_{\text{max}} \) then
              12. \( s_j \) is termination state;
          13. end
          14. \( j \leftarrow j + 1 \);
          15. if Memory pool is full then
              16. Sample a batch of samples and update parameters \( z \) according to Eq. (1);
              17. Observe the fidelity of \( k \)th sample in \( M \), denoted as \( h(k) \);
              18. if \( h(k) > F'_j \) and \( \text{rand}(1) < \mu_1 \) then
                  19. \( k \leftarrow k + 1 \);
              20. end
          21. end
          22. Store transition \( e_j = (s_j, a_j, r_j, s_{j+1}, F'_j) \) as the \( k \)th sample into \( M \) and set \( k \leftarrow k + 1 \);
      23. end
      24. Decrease \( \epsilon \) every \( E \) episodes as \( (1 - \epsilon) \leftarrow (1 - \zeta \epsilon) \);
      25. Label the performance of current episode as \( F^* = F'_j \) and append \( F^* \) into \( L \);
      26. if \( F^* \geq D(v_i) \) then
          27. Update the number of hitting current task \( w \leftarrow w + 1 \);
      28. end
      29. if \( w \geq SC \) then
          30. Stop criterion for the current task is met;
      31. end
   32. end
   33. Determine the candidate difficulty based on \( L \) and assign it to the next task task \( v_{i+1} \);
   34. Reactivate DQN agent and reset \( \epsilon \) to guarantee exploration;
   35. \( i \leftarrow i + 1 \)
36. end

agent is thought to reach its final task if it fails to hit the task in the early episodes.

**Remark 2:** The introduction of curriculum learning to the proposed CDRL helps the agent to focus on proper tasks. In particular, it improves the learning efficiency by making better use of experiences by learning easy samples at the early stage and difficult ones at the later stage without affecting the convergence of the baseline DRL methods [36].

**Remark 3:** Although CDRL is implemented on a classical computer, it can be applied to experimental implementations by collecting the states, the actions, and the reward signals from quantum systems. For example, when searching for optimal laser pulse sequences to excite specified molecular states in experiments [48], the policy can be the control fields of the laser and the reward signal can be given based on the fitness value (some observable function of the molecular products).

**Remark 4:** The definition of (intermediate) tasks in a curriculum is similar to subgoals in hierarchical reinforcement learning (HRL) [49], but they do not have the meaning of hierarchy. Tasks in a curriculum generally belong to the same level with increasing difficulties, which is different from hierarchy where higher-level parent-tasks invoke lower-level child tasks. Despite this, the application of HRL to quantum phenomena is an interesting work that deserves further investigation.
IV. NUMERICAL SIMULATIONS

To test the performance of the proposed CDRL algorithm for manipulating quantum systems, several groups of numerical simulations are carried out for closed and open quantum systems, and the results are demonstrated and analyzed with detailed parameter settings.

A. Parameter Settings

There are two ways to generate tasks for a curriculum. For the static way, the first task is assigned with difficulty 0.9. In addition, 0.99 means a higher requirement and 0.999 usually means an ideal case. Since a direct increase from 0.9 to 0.99 may not be achieved in the learning process, finely grained transitions between two tasks are required to generate more tasks. In the simulations, an incremental value of 0.001 between 0.99 and 0.999 is adopted.

As for dynamic task generation, we initialize the first task with a fixed target fidelity of 0.9. For the remaining tasks, we take the median value of the fidelities collected during the previous task since a median value reflects the average performance and is robust to extreme values among the data.

For the reward signal, we design four piecewise functions to indicate different rewarding schemes. The parameters for $k_1$ and $k_2$ in (7) are given as follows:

\[
(k_1, k_2) = \begin{cases} 
(0, -10) & F_j \in (0, 0.9) \\
(60, -10) & F_j \in [0.9, 0.99) \\
(-10, -100) & F_j \in [0.99, 0.999) \\
(-800, -400) & F_j \in [0.999, 1.000). 
\end{cases}
\]  

The discount factor is set as $\gamma = 0.95$, and the $\epsilon$-greedy factor is initialized as $\epsilon = 0.2$. The trained policy of DRL is approximated by neural networks with two 256-unit hidden layers. Parameters are optimized using Adam with the learning rate of $\alpha = 0.0001$. The decay period is set as $E = 10$ and the batch size is set as $B = 128$. The threshold for the stop criterion of each task is set as $SC = 2000$. The above parameters are used in all simulations. The other parameters in different simulations are summarized as in Table I.

A prioritized experience replay method [47] is applied and the probability threshold for the smart storing mechanism is set as $\mu_1 = 0.8$.
To verify the effectiveness of CDRL, we introduce two traditional DRL methods for comparison: 1) DRL-1 searches the exact $N_{\text{max}}$ pulses of control for each episode [15], [24] and 2) DRL-2 takes an empirical target fidelity to terminate an episode [22], [23]. DRL-1 can be regarded as a single task with difficulty $D(v_1) = 1.000$ and DRL-2 with a fidelity threshold (e.g., 0.999) can also be regarded as a case of one single task with a certain difficulty. The comparison between CDRL and the two traditional DRL methods that do not involve intermediate tasks may demonstrate the effectiveness of the curriculum learning mechanism for quantum control.

In addition, we also include two traditional optimization methods including GD [10] and GA [10] for comparison.

In this article, the simulations for each task are implemented multiple times and a seed is utilized to control the randomness for each running. The simulations can be implemented in two scenarios according to the initial states: 1) a benchmark case means ten runs with an identical initial state and different seeds and 2) a random case means ten runs with identical seed and different initial states.

B. Numerical Results for Closed Quantum Systems

To apply DRL methods to quantum systems, a map is built to transform a quantum state to a real vector, which can be formulated as

$$s_j = [\text{Re}(\langle 0|\psi_j \rangle), \text{Im}(\langle 0|\psi_j \rangle), \ldots, \langle n-1|\psi_j \rangle, \text{Im}(n-1|\psi_j \rangle)]$$

(10)

where $\{|k\rangle\}_{k=0}^{n-1}$ is a set of basis states for an $n$-level quantum system. $\text{Re}(\cdot)$ and $\text{Im}(\cdot)$ are the functions of taking the real and imaginary parts of a complex number, respectively.

1) Two-Qubit Quantum Systems: Consider a two-qubit system with Hamiltonian $H(t) = H_0 + \sum_{m=1}^{4} u_m(t) H_m$. Denote $I_n$ as the identity operator for $n$-level systems. Denote the Pauli operators as

$$\sigma_z = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

(11)

We assume the free Hamiltonian is $H_0 = \sigma_z \otimes \sigma_z$ and the control Hamiltonian operators are $H_1 = \sigma_x \otimes I_2$, $H_2 = I_2 \otimes \sigma_x$, $H_3 = \sigma_y \otimes I_2$, $H_4 = I_2 \otimes \sigma_y$, respectively.
the value of fidelity is clipped within 

\[ I \] defined as CDRL ranks first regarding the average reward in an episode stage in Fig. 5(a). From Fig. 5(b), the average steps learned DRL-2, although they display a little difference in the early Fig. 5, CDRL converges to a similar fidelity to DRL-1 and performance regarding fidelity among ten runnings. From higher than the other two lines. Similar to the benchmark case, CDRL tends to achieve a better fidelity, as the violet line is the maximum control step is defined as \( N_{\text{max}} = 40 \) and the control magnitudes are constrained as \( G_1 = G_2 = G_3 = G_4 = 4 \).

Here, we take a static way to generate tasks for curriculum construction. The numerical comparison of CDRL, GD, and GA under \( dt = 0.0275 \) is revealed in Fig. 4. From Fig. 4(a) and (b), GD achieves the best fidelities, closely followed by CDRL in both benchmark and random scenarios. In addition, CDRL achieves better fidelities than GA in searching for discrete control fields. The trajectories of the optimal control pulses learned by three methods are depicted in Fig. 4(c), where CDRL finds an optimal control strategy with 13 control pulses to achieve a final fidelity of 0.9999. While the optimal control pulses searched by GA and GD take the maximum steps. In particular, the fidelity of CDRL increases with step, while the other two methods do not have this benefit. In this sense, CDRL method has an advantage to search for short control pulses without sacrificing its fidelity.

The comparison between CDRL, DRL-1, and DRL-2 for the benchmark case is summarized as in Fig. 5. It is worthy to note that the lines represent the average performance (i.e., mean value) and the shallow areas denote the standard deviation (std). In particular, the value of upper border represents mean(•) + std(•) and the value of lower border is mean(•) - std(•). Considering the physical meaning of fidelity, the value of fidelity is clipped within [0, 1]. This technique is applied to all the other figures that involve the learning performance regarding fidelity among ten runnings. From Fig. 5, CDRL converges to a similar fidelity to DRL-1 and DRL-2, although they display a little difference in the early stage in Fig. 5(a). From Fig. 5(b), the average steps learned by CDRL are lower than the other two methods. In Fig. 5(c), CDRL ranks first regarding the average reward in an episode [defined as \((1/N) \sum_{j=0}^{N} r_j\)]. For the random case in Fig. 6, CDRL tends to achieve a better fidelity, as the violet line is higher than the other two lines. Similar to the benchmark case, CDRL also converges to a small value of steps compared with the other two DRL methods. Here, the curriculum learning mechanism enables the agent to exploit its past knowledge and thus to search for its optimal control pulses with different initial states. Hence, the average performance of CDRL may have a larger variance than the traditional DRL methods which do not utilize intermediate tasks. It is worthy to note that the steps in an episode actually mean the number of control pulses for a control strategy. Although the performance of CDRL may fluctuate in the early stage [such as shown in Figs. 5(a) and 6(a)], an efficient use of the past experiences contributes to an optimal control pulse with a high fidelity. The convergence with shorter steps in an episode [as shown in Figs. 5(b) and 6(b)] reveals that CDRL has the potential to search for shorter control pulses compared with DRL-1 and DRL-2.

2) Three-Qubit Quantum Systems: For a three-qubit system, its Hamiltonian is assumed to be \( H(t) = H_0 + \sum_{m=1}^{6} u_m(t) H_m \). Denote \( \sigma_{x}^{(12)} = \sigma_x \otimes \sigma_x \otimes I_z, \sigma_{x}^{(23)} = I_z \otimes \sigma_x \otimes \sigma_x, \sigma_{x}^{(13)} = \sigma_x \otimes I_z \otimes \sigma_x, \sigma_{x}^{(2)} = I_z \otimes \sigma_x \otimes \sigma_x \). The free Hamiltonian is \( H_0 = 0.1 \sigma_{x}^{(12)} + 0.1 \sigma_{x}^{(23)} + 0.1 \sigma_{x}^{(13)} \). The control Hamiltonian operators are \( H_1 = \sigma_x \otimes I_z \otimes I_z, H_2 = I_z \otimes \sigma_x \otimes I_z, H_3 = I_z \otimes I_z \otimes \sigma_x, \) and \( H_4 = \sigma_x \otimes I_z \otimes I_z, H_5 = I_z \otimes \sigma_x \otimes I_z, H_6 = I_z \otimes I_z \otimes \sigma_x \). The maximum control step is set as \( N_{\text{max}} = 100 \) and the control bounds are set as \( G_1 = G_2 = G_3 = G_4 = G_5 = G_6 = 1 \).
and the control Hamiltonian operators are chosen as

\[ H = \sum_{j=1}^{2} j (j+1) |j\rangle \langle j| \]

and the control Hamiltonian operators are chosen as

\[ H_1 = \sum_{i=1}^{3} (i| i+1 \rangle + |i+1 \rangle i) \]

and \( H_2 = |1\rangle \langle 3| + |3\rangle \langle 1| \).

The Lindblad operators are given as [50]

\[ L_1 = \tau_{12} |1\rangle \langle 2|, \quad L_2 = \tau_{13} |1\rangle \langle 3|, \quad L_3 = \tau_{14} |1\rangle \langle 4| \]

\[ L_4 = \tau_{12} |2\rangle \langle 3|, \quad L_5 = \tau_{13} |2\rangle \langle 4|, \quad L_6 = \tau_{12} |3\rangle \langle 4| \]

with \( \tau_{12} = 0.4, \tau_{13} = 0.3, \tau_{14} = 0.2 \). Consider a control task from an initial state \( |\psi_0\rangle = |1\rangle \) to the target state \( |\psi_f\rangle = |3\rangle \), we define a maximum step \( N_{\text{max}} = 100 \).

Fig. 9(a) reveals three trajectories with the optimal control pulses under \( dt = 0.05 \). A closer look at the inset of Fig. 9(a), CDRL achieves the best fidelity, and GA achieves a slightly better result than GD. The curves in Fig. 9(b) demonstrate that CDRL obtains better results than GA and GD under different time durations. In addition, the curve of CDRL is close to a horizontal straight line, while GA and GD exhibit similar performance, with the fidelity decreasing with the time duration. The comparison of training performance using the three DRL variants is summarized as in Fig. 10, where CDRL exhibits a better performance than DRL-2 regarding fidelity. Since CDRL might explore a few steps to achieve a tiny increment of fidelity, the average steps of CDRL are a little higher than DRL-2. While DRL-1 falls far away from CDRL and DRL-2 regarding fidelity, steps, and average reward.

2) Four-Level Open Quantum Systems: For a four-level open quantum system, we assume that the four energy levels are \( |1\rangle = (1, 0, 0, 0)^T, |2\rangle = (0, 1, 0, 0)^T, |3\rangle = (0, 0, 1, 0)^T, |4\rangle = (0, 0, 0, 1)^T \). Let the free Hamiltonian be \( H_0 = 0.25 \sum_{j=1}^{4} j (j+1) |j\rangle \langle j| \) and the control Hamiltonian operators be \( H_1 = \sum_{i=1}^{3} (i| i+1 \rangle + |i+1 \rangle i) \) and \( H_2 = |1\rangle \langle 3| + |3\rangle \langle 1| \). The Lindblad operators are given as [50]

\[ L_1 = \tau_{12} |1\rangle \langle 2|, \quad L_2 = \tau_{13} |1\rangle \langle 3|, \quad L_3 = \tau_{14} |1\rangle \langle 4| \]

\[ L_4 = \tau_{12} |2\rangle \langle 3|, \quad L_5 = \tau_{13} |2\rangle \langle 4|, \quad L_6 = \tau_{12} |3\rangle \langle 4| \]

with \( \tau_{12} = 0.4, \tau_{13} = 0.3, \tau_{14} = 0.2 \). Consider a control task from an initial state \( |\psi_0\rangle = |1\rangle \) to the target state \( |\psi_f\rangle = |3\rangle \), we set the maximum step as \( N_{\text{max}} = 100 \).

The comparison result of the three DRL methods under \( dt = 0.06 \) is summarized as in Fig. 11, where CDRL converges to a better fidelity compared with the other two DRL variants in Fig. 11(a). In addition, CDRL achieves the best performance regarding both steps in Fig. 11(b) and average reward in Fig. 11(c). The steps in an episode mean the number of control pulses for a control strategy, suggesting that CDRL has the potential to search for shorter control pulses with higher fidelity within for four-level open quantum systems. The results under different time durations are summarized as
in Fig. 12, where the performance of CDRL is higher than that of DRL-1. In addition, the curve of CDRL is close to a horizontal straight line, while the curves of DRL-1 and DRL-2 decrease with the time duration.

D. Additional Comparison

We now take DQN and policy gradient (PG) in [22] and apply proximal policy optimization (PPO) [51] for comparison. In particular, these methods are applied to state preparation on three quantum systems and each control task is implemented ten times with different seeds. The numerical results are summarized in Fig. 13, where each dot represents the best fidelity of each running. As we can see, CDRL achieves the best performance, followed by DQN. In addition, the superiority of CDRL over these existing DRL methods is clear for four-level open quantum systems.

In addition, the performance of CDRL with manually predefined curriculums (CDRL-static) and CDRL with automatically created curriculums (CDRL-dynamic) is also compared. The results in Fig. 14 suggest that they achieve similar performance regarding average fidelities, but CDRL with dynamic curriculums has the potential to search for shorter control pulses for four-level open quantum systems.

V. Conclusion

In this article, a CDRL approach is presented for the learning control design of quantum systems. In CDRL, a curriculum is constructed using a set of intermediate tasks. In particular, fidelity is utilized to indicate the difficulty of one task. By sequencing a set of tasks, the agent learns to grasp easy knowledge and gradually increases the difficulty until the final task is achieved. The numerical results show that CDRL achieves comparative performance to GD, and achieves better performance than GA. Moreover, CDRL exhibits superior performance over two traditional DRL methods regarding fidelity and tends to find shorter control pulses. For our future work, we will extend this work to a sample-level curriculum to achieve more accurate manipulation of quantum systems. In addition, we will focus on the investigation of curriculum-based learning approaches for continuous control fields and integrate curriculum learning into other DRL methods such as deep deterministic policy gradient (DDPG) [52] and PPO [51].

REFERENCES

[1] M. A. Nielsen and I. L. Chuang, Quantum Computation and Quantum Information. Cambridge, U.K.: Cambridge Univ. Press, 2010.
[2] H. M. Wiseman and G. J. Milburn, Quantum Measurement and Control. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[3] D. Dong and I. R. Petersen, “Quantum control theory and applications: A survey,” IET Control Theory Appl., vol. 4, no. 12, pp. 2651–2671, Dec. 2010.
[4] V. Giovannetti, S. Lloyd, and L. Maccone, “Advances in quantum metrology,” Nature Photon., vol. 5, no. 4, p. 222, 2011.
[5] H. Rabitz, R. de Vivie-Riedle, M. Motzku, and K. Kompa, “Whither the future of controlling quantum phenomena?” Science, vol. 288, no. 5467, pp. 824–828, 2000.
[6] A. C. Doherty, S. Habib, K. Jacobs, H. Mabuchi, and S. M. Tan, “Quantum feedback control and classical control theory,” Phys. Rev. A, Gen. Phys., vol. 62, no. 1, Jun. 2000, Art. no. 012105.
[7] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, “Optimal control of coupled spin dynamics: Design of NMR pulse sequences by gradient ascent algorithms,” J. Magn. Reson., vol. 172, no. 2, pp. 296–305, 2005.
[8] H. Ma, C. Chen, and D. Dong, “Differential evolution with equally-mixed strategies for robust control of open quantum systems,” in Proc. IEEE Int. Conf. Syst., Man Cybern., Oct. 2015, pp. 2055–2060.
[9] H. Ma, D. Dong, C.-C. Shu, Z. Zhu, and C. Chen, “Quantum learning control using differential evolution with equally-mixed strategies,” Control Theory Technol., vol. 15, no. 3, pp. 226–241, Aug. 2017.
[10] D. Dong, X. Xing, H. Ma, C. Chen, Z. Liu, and H. Rabitz, “Learning-based quantum robust control: Algorithm, applications, and experiments,” IEEE Trans. Cybern., vol. 50, no. 8, pp. 3581–3593, Aug. 2020.
[11] D. Dong, M. A. Mabrok, I. R. Petersen, B. Qi, C. Chen, and H. Rabitz, “Sampling-based learning control for quantum systems with uncertainties,” IEEE Trans. Control Syst. Technol., vol. 23, no. 6, pp. 2155–2166, Nov. 2015.
[12] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning,” Nature, vol. 549, no. 7671, p. 195, 2017.
[13] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2018.
[14] C. Chen, D. Dong, H.-X. Li, J. Chu, and T.-J. Tarn, “Fidelity-based probabilistic Q-learning for control of quantum systems,” IEEE Trans. Neural Netw. Learn. Syst., vol. 25, no. 5, pp. 920–933, May 2014.
[15] 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, vol. 8, no. 3, Sep. 2018, Art. no. 031086.
[16] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, pp. 529–533, 2015.
[17] H. Li, Z. Qichao, and D. Zhao, “Deep reinforcement learning-based automatic exploration for navigation in unknown environment,” IEEE Trans. Neural Netw. Learn. Syst., vol. 51, no. 6, pp. 2064–2076, Jun. 2020.
[18] D. Zhao, D. Liu, F. L. Lewis, J. C. Principe, and S. Squartini, “Special issue on deep reinforcement learning and adaptive dynamic programming,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 6, pp. 2038–2041, Jun. 2018.
[19] Z. Wang, H. He, Z. Wan, and Y. Sun, “Coordinated topology attacks in smart grid using deep reinforcement learning,” IEEE Trans. Ind. Informat., vol. 17, no. 2, pp. 1407–1415, Feb. 2021.
[20] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, Feb. 2015.
[21] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. Cambridge, MA, USA: MIT Press, 2016.
[22] 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., vol. 5, no. 1, pp. 1–7, Dec. 2019.
[23] J. Mackeprang, D. B. R. Dasari, and J. Wachtrop, “A reinforcement learning approach for quantum state engineering,” Quantum Mach. Intell., vol. 2, no. 1, pp. 1–14, Jun. 2020.
[24] Z. An and D. L. Zhou, “Deep reinforcement learning for quantum gate control,” Europhys. Lett., vol. 126, no. 6, p. 60002, Jul. 2019.
[25] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, “Universal quantum control through deep reinforcement learning,” npj Quantum Inf., vol. 5, no. 1, pp. 1–8, Dec. 2019.
[26] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, “Reinforcement learning with neural networks for quantum feedback,” Phys. Rev. X, vol. 8, no. 3, Sep. 2018, Art. no. 031084.
[27] H. Xu, J. Li, L. Liu, Y. Wang, H. Yuan, and X. Wang, “Generalizable control for quantum parameter estimation through reinforcement learning,” npj Quantum Inf., vol. 5, no. 1, pp. 1–8, Dec. 2019.
[28] K. Bharati, T. Haug, V. Vedral, and L.-C. Kwek, “How to teach AI to play bell non-local games: Reinforcement learning,” 2019, arXiv:1912.10783.
[29] F. Chen, J.-J. Chen, L.-N. Wu, Y.-C. Liu, and L. You, “Extreme spin squeezing from deep reinforcement learning,” Phys. Rev. A, Gen. Phys., vol. 100, no. 4, Oct. 2019, Art. no. 041801.
[30] R. Porotti, D. Tamascelli, M. Restelli, and E. Prati, “Coherent transport of quantum states by deep reinforcement learning,” Commun. Phys., vol. 2, no. 1, pp. 1–9, Dec. 2019.
[31] T. Haug, W.-K. Mok, J.-B. You, W. Zhang, C. E. Pog, and L.-C. Kwek, “Classifying global state preparation via deep reinforcement learning,” Mach. Learn., Sci. Technol., vol. 2, no. 1, 2020, Art. no. 01LT02.
[32] J. Schuff, L. J. Fidler, and D. Braun, “Improving the dynamics of quantum sensors with reinforcement learning,” New J. Phys., vol. 22, no. 3, Mar. 2020, Art. no. 035001.

[33] S. Narvekar, B. Peng, M. Leonetti, J. Sinapov, M. E. Taylor, and P. Stone, “Curriculum learning for reinforcement learning domains: A framework and survey,” 2020, arXiv:2003.04960.

[34] C. Altafani, “Coherent control of open quantum dynamical systems,” Phys. Rev. A, Gen. Phys., vol. 70, no. 6, Dec. 2004, Art. no. 062321.

[35] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning;” in Proc. 26th Annu. Int. Conf. Mach. Learn., 2009, pp. 41–48.

[36] Z. Ren, D. Dong, H. Li, and C. Chen, “Self-paced prioritized curriculum learning with coverage penalty in deep reinforcement learning,” IEEE Trans. Neural Netw. Learn. Syst., vol. 29, no. 6, pp. 2216–2226, Jun. 2018.

[37] K. Shao, Y. Zhu, and D. Zhao, “StarCraft micromanagement with reinforcement learning and curriculum transfer learning,” IEEE Trans. Emerg. Topics Comput. Intell., vol. 3, no. 1, pp. 73–84, Feb. 2019.

[38] C. Lin, D. Sels, and Y. Wang, “Time-optimal control of a dissipative qubit,” Phys. Rev. A, Gen. Phys., vol. 101, no. 2, Feb. 2020, Art. no. 022320.

[39] M. V. Berry, “Transitionless quantum driving,” J. Phys. A, Math. Theor., vol. 42, no. 36, Sep. 2009, Art. no. 365303.

[40] C. J. C. H. Watkins and P. Dayan, “Q-learning,” in Proc. 22nd Int. Conf. Mach. Learn. (ICML), 2005, pp. 279–286.

[41] L. Baird, “Residual algorithms: Reinforcement learning with function approximation,” in Proc. Mach. Learn., 1995, pp. 30–37.

[42] A. Lazaric, M. Restelli, and A. Bonarini, “Transfer of samples in batch reinforcement learning,” in Proc. 25th Int. Conf. Mach. Learn. (ICML), 2008, pp. 544–551.

[43] M. E. Taylor and P. Stone, “Transfer learning for reinforcement learning domains: A survey,” J. Mach. Learn. Res., vol. 10, no. 7, pp. 1633–1685, 2009.

[44] F. Yang, S. Cong, R. Long, T.-S. Ho, R. Wu, and H. Rabitz, “Exploring the transition-probability-control landscape of open quantum systems: Application to a two-level case,” Phys. Rev. A, Gen. Phys., vol. 88, no. 3, Sep. 2013, Art. no. 033420.

[45] S. Narvekar, J. Sinapov, M. Leonetti, and P. Stone, “Source task creation for curriculum learning,” in Proc. Int. Conf. Auton. Agents Multiagent Syst., 2016, pp. 566–574.

[46] S. Narvekar, J. Sinapov, and P. Stone, “Autonomous task sequencing for customized curriculum design in reinforcement learning,” in Proc. 26th Int. Joint Conf. Artif. Intell., Aug. 2017, pp. 2536–2542.

[47] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, “Prioritized experience replay,” in Proc. Int. Conf. Learn. Represent., 2016, pp. 1–21.

[48] R. S. Judson and H. Rabitz, “Teaching lasers to control molecules,” Phys. Rev. Lett., vol. 68, no. 10, p. 1500, 1992.

[49] T. D. Kulkarni, K. Narasimhan, A. Saeedi, and J. Tenenbaum, “Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation,” in Proc. Adv. Neural Inf. Process. Syst., vol. 29, 2016, pp. 3675–3683.

[50] H. Jirari and W. Pötz, “Optimal coherent control of dissipative N-level systems,” Phys. Rev. A, Gen. Phys., vol. 72, no. 1, 2005, Art. no. 013409.

[51] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” 2017, arXiv:1707.06347.

[52] T. P. Lillicrap et al., “Continuous control with deep reinforcement learning;” in Proc. Int. Conf. Learn. Represent., 2016, pp. 1–14.

Hailan Ma received the B.E. degree in automation and the M.A.Sc. degree in control science and engineering from Nanjing University, Nanjing, China, in 2014 and 2017, respectively. She is currently pursuing the Ph.D. degree with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT, Australia. She was a Research Assistant with Nanjing University from 2019 to 2020. Her research interests include learning control and optimization of quantum systems, and quantum machine learning. Ms. Ma is a member of the Technical Committee on Quantum Cybernetics, IEEE Systems, and Man Cybernetics Society.

Daoyi Dong (Senior Member, IEEE) received the B.E. degree in automatic control and the Ph.D. degree in engineering from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively. He was with the Institute of Systems Science, Chinese Academy of Sciences, Beijing, China, and with Zhejiang University, Hangzhou, China. He had visiting positions at Princeton University, NJ, USA; RIKEN, Wako-Shi, Japan; and The University of Hong Kong, Hong Kong. He is currently a Scientia Associate Professor at the University of New South Wales, Canberra, ACT, Australia and an Alexander von Humboldt Fellow at AKS, University of Duisburg-Essen, Duisburg, Germany. His research interests include quantum control and machine learning.

Dr. Dong was awarded the ACA Temasek Young Educator Award by the Asian Control Association and was a recipient of the International Collaboration Award and the Australian Post-Doctoral Fellowship from the Australian Research Council, and a Humboldt Research Fellowship from the Alexander von Humboldt Foundation of Germany. He is a Member-at-Large, Board of Governors, and the Associate Vice President for Conferences and Meetings, IEEE Systems, Man and Cybernetics Society. He served as an Associate Editor for the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS from 2015 to 2021. He is currently an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS and a Technical Editor of the IEEE/ASME TRANSACTIONS ON MECHATRONICS.

Steven X. Ding received the Ph.D. degree in electrical engineering from the Gerhard-Mercator University of Duisburg, Duisburg, Germany, in 1992. From 1992 to 1994, he was a Research and Development Engineer at Rheinmetall GmbH, Unterluess, Germany. From 1995 to 2001, he was a Professor of control engineering at the University of Applied Science Lausitz, Senftenberg, Germany, where he served as the Vice President from 1998 to 2000. Since 2001, he has been a Chair Professor of control engineering and the Head of the Institute for Automatic Control and Complex Systems (AKS), University of Duisburg-Essen, Duisburg. His research interests include model-based and data-driven fault diagnosis, control, and fault-tolerant systems as well as their applications in industry with a focus on automotive systems, chemical processes, and renewable energy systems.

Chunlin Chen (Senior Member, IEEE) received the B.E. degree in automatic control and the Ph.D. degree in control science and engineering from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively. He was with the Department of Chemistry, Princeton University, Princeton, NJ, USA, from September 2012 to September 2013. He had visiting positions at the University of New South Wales, Canberra, ACT, Australia, and the City University of Hong Kong, Hong Kong. He is currently a Professor and the Head of the Department of Control and Systems Engineering, School of Management and Engineering, Nanjing University, Nanjing, China. His current research interests include machine learning, intelligent control, and quantum control and machine learning.

Dr. Chen serves as the Chair for the Technical Committee on Quantum Cybernetics, IEEE Systems, Man and Cybernetics Society.