Deep Reinforcement Learning With Quantum-Inspired Experience Replay

Qing Wei, *Graduate Student Member, IEEE*, Hailan Ma, *Graduate Student Member, IEEE*, Chunlin Chen, *Member, IEEE*, and Daoyi Dong, *Senior Member, IEEE*

***Abstract*—In this article, a novel training paradigm inspired by quantum computation is proposed for deep reinforcement learning (DRL) with experience replay. In contrast to the traditional experience replay mechanism in DRL, the proposed DRL with quantum-inspired experience replay (DRL-QER) adaptively chooses experiences from the replay buffer according to the complexity and the replayed times of each experience (also called transition), to achieve a balance between exploration and exploitation. In DRL-QER, transitions are first formulated in quantum representations and then the preparation operation and depreciation operation are performed on the transitions. In this process, the preparation operation reflects the relationship between the temporal-difference errors (TD-errors) and the importance of the experiences, while the depreciation operation is taken into account to ensure the diversity of the transitions. The experimental results on Atari 2600 games show that DRL-QER outperforms state-of-the-art algorithms, such as DRL-PER and DCRL on most of these games with improved training efficiency and is also applicable to such memory-based DRL approaches as double network and dueling network.**

***Index Terms*—Deep reinforcement learning (DRL), quantum computation, quantum-inspired experience replay (QER), quantum reinforcement learning.**

NOMENCLATURE

$\bigotimes$	Tensor product.
δ	Temporal-difference error.
$\delta_{\tilde{i},i}$	Kronecker delta.
ϵ	ϵ-greedy factor.
γ	Discount factor.
ι	Angle of the uniform state.
$\|\psi\rangle$	State vector (quantum state).
$\mathscr{H}$	Hilbert space.
μ	Hyperparameters for the rotation time m_k.
$\boldsymbol{\omega}$	Depreciation factor.
Φ	Rotation angle.
π	Policy function.
i	$i=\sqrt{-1}$.
σ	Preparation factor.
$\tau_{1,2}$	Hyperparameters for the depreciation factor $\boldsymbol{\omega}$.
θ	Parameters of the evaluation network.
θ^-	Parameters of the target network.
$\zeta_{1,2}$	Hyperparameters for the preparation factor δ.
A	Action space.
a_t	Selected action according to policy $\pi(s_t)$ at time t.
e_t	t-th experience.
h	Eigenvalue.
k	Label of experience in buffer.
M	Size of experience replay buffer.
P	State transition probability.
$P(h)$	Probability of obtaining h.
$Q(s,a)$	State–action values.
R	Reward function.
r_t	Scalar reward at time step t.
$RT_{\max}$	Maximum value of times of being replayed.
S	State space.
s_t	State at time step t.
SK	Label of the selected experience in experience replay.
T	Terminal time of one episode.
U	Unitary operator.
$U^\dagger$	Transposed conjugate matrix of U.
X_h	Projector onto the eigenspace of H with h.
Y	Pauli Y matrix.

Manuscript received October 20, 2020; accepted January 12, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 71732003, Grant 62073160, and Grant 61828303; in part by the Australian Research Council's Discovery Projects Funding Scheme under Project DP190101566; and in part by the National Key Research and Development Program of China under Grant 2018AAA0101100. This article was recommended by Associate Editor D. Zhao. *(Qing Wei and Hailan Ma contributed equally to this work.) (Corresponding author: Chunlin Chen.)*

Qing Wei and Chunlin Chen are with the Department of Control and Systems Engineering, School of Management and Engineering, Nanjing University, Nanjing 210093, China (e-mail: clchen@nju.edu.cn).

Hailan Ma is with the Department of Control and Systems Engineering, School of Management and Engineering, Nanjing University, Nanjing 210093, China, and also with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia (e-mail: hailanma0413@gmail.com).

Daoyi Dong is with the School of Engineering and Information Technology, University of New South Wales, Canberra, ACT 2600, Australia (e-mail: daoyidong@gmail.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2021.3053414.

Digital Object Identifier 10.1109/TCYB.2021.3053414

I. INTRODUCTION

REINFORCEMENT learning (RL) is an intelligent paradigm that learns through the interaction with the environment. During the training process of this interaction-based algorithm, similar to human behaviors [1]–[3], the agent adjusts its behavior to maximize the cumulative rewards for the entire control task according to the retroaction it receives from the environment. When it comes to the general situation of the real-world environment, most control tasks often come

with high-dimensional inputs, where traditional RL approaches cannot work well. Fortunately, deep learning provides a new approach to handle the complex input information and has achieved a huge breakthrough in various fields [4]–[10]. In particular, by combining deep learning with RL, a new framework of deep RL (DRL) arises, where the deep Q network (DQN) becomes one of the most famous DRL methods [11].

DQN was employed to estimate the action values to help the agent make sequential decisions, where raw images were fed into the convolutional neural networks followed by the fully connected networks to output the action values that estimated the future rewards. In order to improve the utilization of the state–action transitions, an experience replay mechanism was deployed in the DQN framework [12], where experiences were stored in a finite-size buffer and were retrieved from the buffer. This mechanism of experience replay effectively sped up the processing of the experiences during training, but it ignored the difference in the importance between experiences. A series of experience replay variants has been developed to further improve the learning process, such as prioritized experience replay (PER) [13], deep curriculum RL (DCRL) [14], remember and forget for experience replay (ReF-ER) [15], attentive experience replay (AER) [16], and competitive experience replay (CER) [17].

In PER, temporal difference errors (TD-errors) determine the priorities of the experiences and influence the probabilities of those experiences' being replayed. This method further enhances the utilization of experiences compared to the original experience replay, but the data generated by the agent are noisy. Since DRL-PER gives higher priorities to transitions with larger TD-errors, there might exist some experiences, whose large TD-errors would not decrease even after many times of replay. From this respective, DRL-PER may cause some experiences to be overused, which might result in oscillations of the neural network [18]. To improve the sample efficiency of DQN, DCRL proposed a criterion for the samples' importance based on the difficulties and diversities of the experiences, where the difficulties are positively correlated with TD-errors and the diversities are related to the number of replaying times [14]. However, it introduced a number of parameters that required more prior knowledge to tune accurately. In ReF-ER [15], policy updates are penalized according to the Kullback–Leibler divergence to accelerate convergence, AER selects experiences according to the similarities between their states and the agent's current state [16], and CER sets up two agents for competitive exploration between a pair of agents [17]. These three methods may rely on high computing resources and it is desirable to design a more effective and general approach to enhance experience replaying for DRL.

At the same time, quantum physics has been employed to dramatically enhance information-processing capability [19]–[26] and has a positive influence on specific algorithmic tasks of applied artificial intelligence [27]–[30]. In particular, there has been much interest in the quantum enhancement of RL and its applications. The idea of quantum RL first originated from introducing the characteristics of quantum parallelism into classical RL algorithms [31], which achieved a better tradeoff between exploration and exploitation and sped up the learning as well. Quantum mechanics was found to be able to bring an overall quadratic speedup for intelligent agents [32]. The general agent-environment framework was also extended to the quantum domain [33]. In addition, quantum RL with multiqubits was evaluated on superconducting circuits [34] and was extended to other cases, such as multilevel and open quantum systems [35]. Multiple value functions using the Grover algorithm were proved to converge in fewer iterations than their classical counterparts [36]. Recent research also demonstrated the advantage of RL using the quantum Boltzmann machines over the classical one [37].

Inspired by quantum machine learning, we may produce atypical patterns in data. For example, the quantum superposition state provides an exponential scale of computation space in the n-qubits linear physical space [31], [38]. In this article, we propose a quantum-inspired experience replay (QER) approach for DRL (DRL-QER) to improve the training performance of DRL in a natural way without deliberate hyperparameter tuning. In DRL-QER, the experiences are expressed in quantum representations and the probability amplitudes of the quantum representations of experiences are iteratively manipulated by quantum operations, including the preparation operation and depreciation operation. In particular, the preparation operation is designed according to the importance of the experiences, and the depreciation operation is associated with the replaying times for the selected experiences. With the two operations, the importance of the experiences is distinguished and the diversity of experiences is guaranteed. To test the proposed DRL-QER algorithm, experiments are carried out on the Gym-Atari platform with a comparison to DRL-PER and DCRL. In addition, DRL-QER is implemented with double DQN and dueling DQN, and the DRL-QER variants are compared with their classical counterparts.

The remainder of this article is organized as follows. Section II introduces DRL, experience replay, and the basic concepts of quantum computation as well. In Section III, the framework of DRL-QER is introduced and quantum representations and quantum operations are presented, followed by the algorithm description of DRL-QER with specific implementation details. In Section IV, experimental results are demonstrated to verify the performance of the proposed DRL-QER algorithm. The conclusion is drawn in Section V.

II. Preliminaries

A. Deep Reinforcement Learning and Experience Replay

1) Markov Decision Process: The training process of RL is based on the model of the *Markov decision process*, whose basic components can be described by a tuple of $\langle S, A, P, R\rangle$ [1], where S is the state space, A is the action space, $P : S \times A \times S \to [0, 1]$ is the state transition probability, and $R : S \times A \to \mathbb{R}$ is the reward function.

In the process of interaction with the environment, the agent forms state $s_t \in S$ at the time step $t \in [0, T]$ and chooses an action $a_t = \pi(s_t)$, $a_t \in A$, where T is the terminal time and policy π is a mapping from state space S to action space A. After carrying out the action a_t, the agent transits to the next state s_{t+1} and receives a scalar reward signal r_t. Thus, we

obtain a transition of $e_t = (s_t, a_t, r_t, s_{t+1})$ at time step t. RL aims at determining an optimal policy π^* so as to maximize the cumulative discounted future rewards $R_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k}$, where $\gamma \in [0, 1]$ is a discount factor to balance the importance of the current rewards and the future rewards. As a widely used RL algorithm, Q-learning defines $Q(s, a)$ as the expected discounted reward for executing action a at state s following the policy π, and a lookup Q table is established to store the Q-values [39].

2) Deep Q Network: In high-dimensional environments, it is a general and effective approach to approximate $Q(s, a)$ using a neural network with parameter θ, that is, $Q(s, a; \theta) \approx Q(s, a)$, instead of a lookup table that stores all state–action values $Q(s, a)$ [39]. In order to update the parameters of the neural network with a gradient descent method, the "true values" $y(s, a)$ of the state–action values $Q(s, a)$ are estimated from the maximum of the next state–action values $Q(s', a')$, that is, $y(s, a) = r + \gamma \max_{a'} Q(s', a'; \theta^-)$, where θ^- denotes the parameters of the target network, which are fixed during the computation of $y(s, a)$ and are updated after some training steps.

The TD-errors δ can be measured by the deviation between $y(s, a)$ and $Q(s, a)$ as

$$\delta = y(s, a) - Q(s, a) = r + \gamma \max_{a'} Q\big(s', a'; \theta^-\big) - Q(s, a; \theta). \tag{1}$$

Accordingly, the loss function $\text{Loss}(\theta; Q, y)$ to be optimized is

$$\text{Loss}(\theta; Q, y) = \frac{1}{2}\left(r + \gamma \max_{a'} Q\big(s', a'; \theta^-\big) - Q(s, a; \theta)\right)^2. \tag{2}$$

Differentiate the loss function $\text{Loss}(\theta; Q, y)$ with respect to parameter θ, and we obtain the gradient as

$$\nabla_\theta \text{Loss} = \left[r + \gamma \max_{a'} Q\big(s', a'; \theta^-\big) - Q(s, a; \theta)\right] \nabla_\theta Q(s, a; \theta). \tag{3}$$

3) Experience Replay: In most RL frameworks, agents incrementally update their parameters while they observe a stream of experiences. In the simplest form, the incoming data are used for a single update and discarded immediately, which brings two disadvantages: 1) strongly correlated transitions break the independent identically distributed (i.i.d.) assumption that is necessary for many popular stochastic gradient-based algorithms and 2) the rapid forgetting of possibly rare experiences that are potentially useful in the future leads to sampling inefficiency. A natural solution would be to put the past experiences into a large buffer and select a batch of samples from them for training [12], [40], [41]. Such a process is called *experience replay*.

In experience replay, how to choose the experiences (transitions) to be replayed plays a vital role to improve the training performance of DRL. When putting the transition e_t into a fixed experience replay buffer with size M, a new index label $k \in \{1, \ldots, M\}$ is assigned to it, with its priority denoted as P_k. As such, the entire experience buffer can be regarded as a collection of transitions as $\{\langle e_t, P_k \rangle\}$. A complete process of experience replay is actually a store-and-sample process, and the learning process works by selecting a minibatch sample from the entire buffer to update the parameters of the RL agent. The key of experience replay lies in the criterion by which the importance of each transition is measured, that is, to determine P_k for each transition.

B. Quantum Computation

In quantum computation, the basic unit that carries information is a quantum bit (also called qubit) and a qubit can be in a superposition state of its eigenstates $|0\rangle$ and $|1\rangle$ [38], [42], which can be written as the following form of:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{4}$$

where α and β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$. Quantum mechanics reveals that measuring a qubit in the superposition state $|\psi\rangle$ leads it to collapse into one of its eigenstates of $|0\rangle$ with probability $|\alpha|^2$ or $|1\rangle$ with probability $|\beta|^2$. In particular, the coefficients can be written as $\alpha = \langle 0|\psi\rangle$ and $\beta = \langle 1|\psi\rangle$, where $\langle a|b\rangle$ represents the inner product between $|a\rangle$ and $|b\rangle$.

In quantum computing, unitary transformation is an essential operation on quantum systems and can transform an initial state $|\psi\rangle$ to another state $|\psi'\rangle$:

$$\big|\psi'\big\rangle = U|\psi\rangle \tag{5}$$

where U satisfies $U^\dagger U = UU^\dagger \equiv I$. For example, a Hadamard gate that transforms $|0\rangle$ to $(|0\rangle + |1\rangle)/\sqrt{2}$ and $|1\rangle$ to $(|0\rangle - |1\rangle)/\sqrt{2}$ can be formulated as

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \tag{6}$$

Another significant quantum gate is the phase gate, which is an important element to carry out the Grover iteration [20] for reinforcing the amplitude of the "target" item. More discussions about quantum operations and quantum gates can be found in [38].

The Grover algorithm is one of the most important quantum algorithms. It has been widely used in the problem of large-scale database searching and is able to locate items with the complexity of $O(\sqrt{N})$ in the unstructured database with high probabilities. Its core idea is to represent items as a quantum system and manipulate its state using a unitary operator in an iterative way [20]. As one of the main operations in the Grover algorithm, the Grover iteration has been successfully applied to RL methods [31], where the action is represented in the superposition of its possible eigen actions. Then, unitary transformation is iteratively performed on the superposition states to change the probability amplitudes of the "good" actions.

The state space of a composite quantum system is represented by the tensor product, denoted as $\bigotimes$, of the state space of each component system. For example, the composite quantum system of two subsystems A and B can be defined on a Hilbert space $\mathscr{H} = \mathscr{H}_A \otimes \mathscr{H}_B$, where $\mathscr{H}_A$ and $\mathscr{H}_B$ correspond to the Hilbert space of the subsystems A and B, respectively. Furthermore, its state $|\psi_{AB}\rangle$ may be described by the tensor product of the states of its subsystems, that is, $|\psi_{AB}\rangle = |\psi\rangle_A \otimes |\psi\rangle_B$.

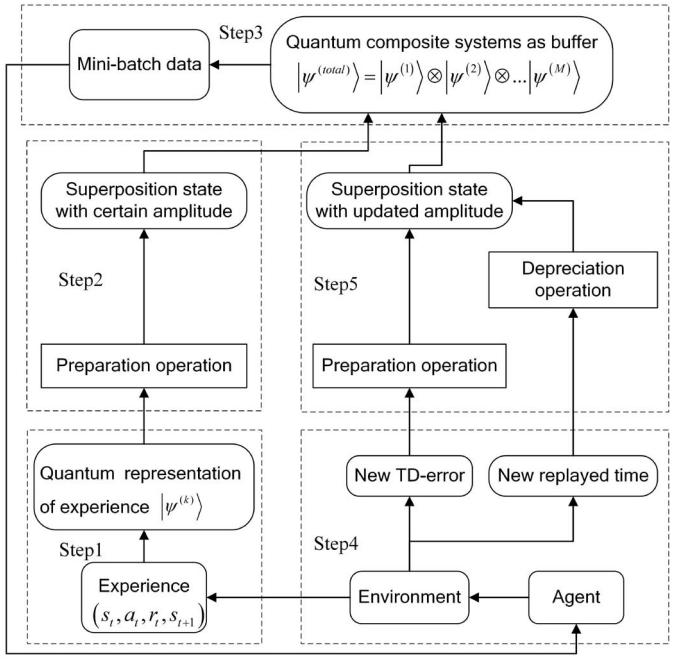

Fig. 1. Framework of DRL-QER. Step 1: Representing the newly generated experience using a qubit; Step 2: Performing the preparation operation on the quantum representation with Grover iteration; Step 3: Sampling experiences to compose minibatch data; Step 4: Training the DRL agent with the minibatch data; and Step 5: Updating the quantum representation of the experience by the new TD-error (the preparation operation) and the new number of replayed times (the depreciation operation) and then put it back into the replay buffer.

To obtain information by measuring or observing a quantum system, positive-operator-valued measure (POVM) can be applied [38]. For an observable H, there exists a complete set of orthogonal projectors $\{X_h : \sum_h X_h = I, X_h = X_h^\dagger, X_{\tilde{h}} X_h = \delta_{\tilde{h},h} X_h\}$, where $\delta_{\tilde{i},i}$ is the Kronecker delta, X_h is the projector onto the eigenspace of H with eigenvalue h, and we have $H = \sum_h hX_h$. The probability of obtaining outcome h can be calculated by $P(h) = \langle\psi|X_h|\psi\rangle$.

III. Deep Reinforcement Learning With Quantum-Inspired Experience Replay

In this section, the framework of DRL-QER is first introduced. Then, quantum representations and quantum operations using the Grover iteration are designed to provide a natural and appropriate experience replay mechanism. Finally, the implementation of the integrated DRL-QER algorithm is presented.

A. Framework of DRL-QER

In DRL-QER, quantum characteristics are borrowed to design new manipulation methods to improve the experience replay mechanism, which aims at providing a natural and easy-to-use experience replay approach using quantum representations and unitary transformation for the experiences and their importance, respectively.

The framework of DRL-QER is described as in Fig. 1. During each learning iteration, the agent interacts with the environment and obtains the transition e_t at time step t. Such a transition is first expressed in the quantum representation, or more precisely, the kth qubit, where k is its index in the buffer. Second, the state of the qubit evolves to a superposition state through the preparation operation. Then, transitions are sampled with probabilities proportional to their importance and those selected samples compose the minibatch data for training the neural network. In addition, after each training step, the amplitudes of the selected quantum representations are manipulated by the combined unitary transformation, including the preparation operation to adapt to the new TD-errors and the depreciation operation to deal with the replaying times of the transitions. This procedure is carried out iteratively until the algorithm converges, whose specific details are implemented in the following sections.

B. Quantum Representation of Experiences

In the quantum theory, a qubit can be realized by a two-level atom, a spin system, or a photon. For two-level atoms, $|0\rangle$ can be the ground state, while $|1\rangle$ represents the excited state. For spin systems, $|0\rangle$ can be the state of *spin up*, while $|1\rangle$ represents the state of *spin down*. For photons, $|0\rangle$ can be the state of horizontal polarization, while $|1\rangle$ represents the state of vertical polarization. Here, in experience replay, one experience can be regarded as a qubit system, and its two eigenstates $|0\rangle$ and $|1\rangle$ represent the actions of *rejecting* and *accepting* this experience, respectively.

During the learning process, the agent tries to interact with its environment, which can be modeled as an MDP. For each step t, with the current state s_t, the agent selects an action a_t under a certain exploration policy (such as ϵ-greedy), and then transfers to the next state s_{t+1} and obtains a reward r_t. Finally, four elements together compose a transition (s_t, a_t, r_t, s_{t+1}), which is assigned a new index k to denote its order in the experience buffer. In transforming the transition into quantum representation, we define the action of *accepting* and *rejecting* the transition as two eigenstates. Then, the transition is regarded as a qubit (as shown in Fig. 2) with its state as

$$\left|\psi^{(k)}\right\rangle = b_0^{(k)}|0\rangle + b_1^{(k)}|1\rangle \tag{7}$$

where the coefficients $b_0^{(k)}$ and $b_1^{(k)}$ have probability amplitude meanings and satisfy $|b_0^{(k)}|^2 + |b_1^{(k)}|^2 = 1$. In particular, the probability of rejecting this transition is $|b_0^{(k)}|^2 = |\langle 0|\psi^{(k)}\rangle|^2$ and the probability of accepting it is $|b_1^{(k)}|^2 = |\langle 1|\psi^{(k)}\rangle|^2$. It is worth noting that the coefficients of the qubit are related with the significance of the experience. Before determining its importance, it is practical to first set an initial state and let the qubit evolve from the initial state to a desired state.

In quantum computing, a uniform state is one significant superposition state and has the form as

$$|\psi_0\rangle = \frac{\sqrt{2}}{2}(|0\rangle + |1\rangle). \tag{8}$$

It has equal probabilities for two eigenstates and means that the least knowledge is given about the state with maximum entropy, which makes it feasible to adopt the uniform state as the initial state for each experience.

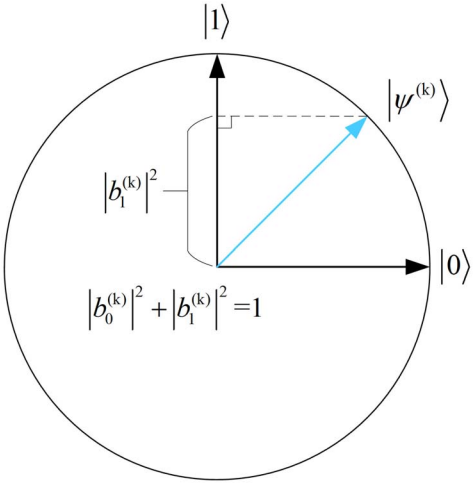

Fig. 2. Experience represented in a qubit system. Here, $|0\rangle$ and $|1\rangle$ correspond to *rejecting* or *accepting* the transition. The state of such a transition can be formulated as $|\psi\rangle = b_0^{(k)}|0\rangle + b_1^{(k)}|1\rangle$, where $|b_1^{(k)}|^2$ is the probability of accepting and $|b_0^{(k)}|^2$ corresponds to the probability of rejecting.

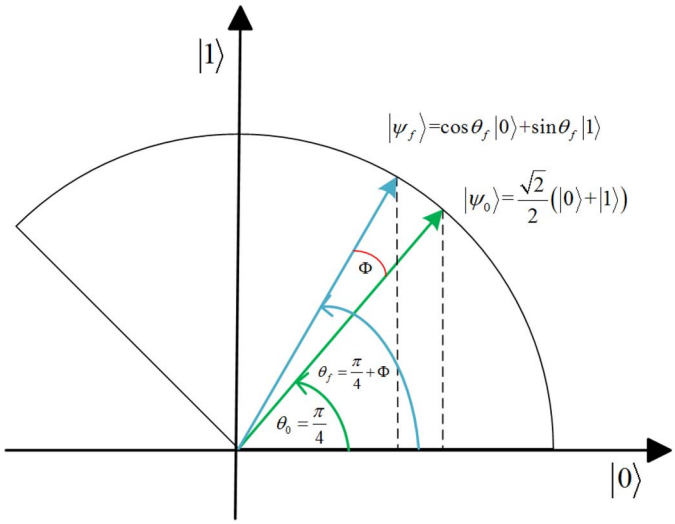

Fig. 3. State transition of a qubit system under unitary rotation U_Φ, where $|\psi_0\rangle$ is the initial state and $|\psi_f\rangle$ is the final state.

To adjust the probability amplitudes of qubit state in (7), a rotation operator, which is the basic element of the Grover iteration [20], [31], is applied with

$$U_\Phi = e^{-i\Phi Y} = \begin{bmatrix} \cos(\Phi) & -\sin(\Phi) \\ \sin(\Phi) & \cos(\Phi) \end{bmatrix} \tag{9}$$

where Φ is a real number and has the physical meaning of the rotation angle. Pauli Y operator is given as

$$Y \equiv \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}. \tag{10}$$

The operation of performing a rotation operator U_Φ on a qubit of experience is visualized in Fig. 3. The quantum system evolves from the initial state $|\psi_0\rangle$ (the green one) to the final state $|\psi_f\rangle$ (the blue one), under the unitary transformation U_Φ. Projecting $|\psi_0\rangle$ and $|\psi_f\rangle$ to the y-axis, the amplitude of observing $|1\rangle$ increases, which reveals that the probability of accepting the transition is slightly increased.

Since the kth experience in the buffer has the quantum representation form of $|\psi^{(k)}\rangle$, the state of a memory buffer, which is composed of M experiences, is the tensor product of M subsystems:

$$\left|\psi^{\text{total}}\right\rangle = \left|\psi^{(1)}\right\rangle \otimes \left|\psi^{(2)}\right\rangle \otimes \cdots \left|\psi^{(\text{M})}\right\rangle. \tag{11}$$

C. Quantum Operations on Experiences

To deal with the quantum representations of experiences, three subprocesses are involved, that is: 1) preparation operation; 2) depreciation operation; and 3) experience selection by quantum observation. First, the preparation operation is introduced to steer the quantum systems toward the target states, whose amplitudes are related to the TD-errors of the experiences. In fact, whenever the TD-errors of the experiences have changed, the preparation operation is performed to update their probability amplitudes. From this respect, every time when a suitable priority is determined, the quantum systems are to be transferred to a new target state, which can be regarded as a process of quantum state preparation. Hence, we call this special operation the preparation operation. In addition, the depreciation operation is utilized to make sure that the significance of the experiences is adapted to the experience relaying process, such as the times of the experiences' being visited. Another significant operation is to select experiences by quantum observation, to compose minibatch data for training.

To adjust the amplitudes of quantum systems in a natural and appropriate way, a Grover iteration method is adopted for both the preparation operation and depreciation operation. The Grover iteration is a significant operation for dealing with quantum states originated from classical information and it aims at intensifying the probabilities of the target eigenstates, with others at equal probabilities [20], [31]. Considering that the probabilities of experiences' being extracted from an experience buffer vary, we do not use the conventional method, that is, performing the unitary transformation on the composite system (the entire experience buffer). Instead, the Grover iteration is conducted on a single experience with its quantum representation. This strategy helps to adaptively adjust the probability amplitude of each transition and, therefore, to circumvent the neglect of the differences between experiences.

1) Preparation Operation: To better optimize the process of experience replay in DRL-QER, the importance of experiences needs to be distinguished first. Since a single rotation changes the probability amplitude of a qubit system, we define a basic rotation operator as

$$U_\sigma = e^{-i\sigma Y} = \begin{bmatrix} \cos(\sigma) & -\sin(\sigma) \\ \sin(\sigma) & \cos(\sigma) \end{bmatrix} \tag{12}$$

where $\sigma \in R$ is a tiny rotation angle. Based on the exponential approximation formula, that is, $U_\Sigma = (U_\sigma)^m$ with an integer m, several iterations of unitary rotations amount to an overall rotation on the qubits. In addition, due to $e^{-i\sigma Y}e^{-i(-\sigma)Y} = I$, the rotation in the reverse direction can be conducted with U_σ^{-1} (or $U_\sigma^\dagger$). Hence, different rotations can be achieved by

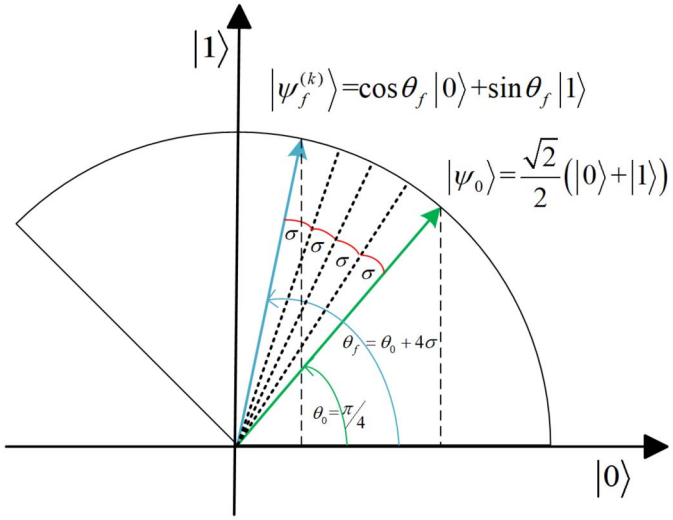

Fig. 4. Procedure of the preparation process using Grover iterations for the kth qubit. $|\psi_0\rangle$ is the uniform quantum state and $|\psi_f^{(k)}\rangle$ is the quantum state after conducting the transformation $U_\Sigma^{(k)}$.

performing multiple times of basic rotations in clockwise or counterclockwise directions.

The preparation operation for a single experience (the kth transition in the buffer) is described in Fig. 4, where four times of basic rotations in the counterclockwise direction are performed on the qubit to intensify the "accepting" amplitude of the good experience or equivalently to strength the "rejecting" amplitude of the bad experience. Generally, the state evolution of such a quantum system can be expressed as

$$U_\Sigma^{(k)} = (U_\sigma)^{m_k}, \quad \left|\psi_f^{(k)}\right\rangle = U_\Sigma^{(k)}|\psi_0\rangle \tag{13}$$

where m_k represents the number of rotation times of the kth qubit. Considering that TD-error reveals the importance of the transition, we associate the value of m_k with its TD-error. For transition e_t with TD-error δ_t, the priority for the kth qubit is given as $P_k = |\delta_t| + \epsilon$ and the maximum priority of all the experiences is $P_{\max}$. To convert the priority into the probability amplitude of the qubit, we try to map priority P_k to a rotation angle, which corresponds to a unitary transformation of quantum states. In particular, $P_{\max}$ is mapped to a rotation angle $\Sigma_{\max}$ and then the angle of P_k can be recorded as $\Sigma_k = \Sigma_{\max} \times (P_k/P_{\max})$. Since the Grover iterations aim at iteratively performing unitary transformations until a desired state is achieved, $\Sigma_{\max}$ is split into μ pieces, and each piece is assigned with σ. The initial state of the qubit is assigned with rotation angle ι, so the angle of rotation can be defined as the target angle minus the initial angle. Finally, the value of m_k reads as

$$m_k = \text{Floor}(\mu \times P_k/P_{\max} - \iota/\sigma) \tag{14}$$

where $\mu, \iota \in R$ are two hyperparameters and Floor(x) takes the largest integer not greater than x. In particular, the sign of m_k reflects the rotation direction relative to the angle of the uniform state, that is, $(\pi/4)$. For example, when m_k is a positive integer, the Grover iteration works in the counterclockwise direction; otherwise, it is conducted in the clockwise direction.

The value of σ in (13) is usually carefully set since it plays an important role in the quantum representation of experiences. From a convenient point of view, it is the most appropriate to set a fixed value. While from the perspective of adaptation to different environments, associating it with the training process, such as the TD-errors, the maximum times of experiences being visited, and the training steps are more preferable. In this work, we describe it with a function associated with the training episode TE

$$\sigma = \frac{\zeta_1}{1 + e^{\frac{\zeta_2}{\text{TE}}}} \tag{15}$$

where $\zeta_1, \zeta_2 \in R$ are two hyperparameters.

By performing the same procedure to each transition, all experiences will end up in their target quantum representations. For example, for most of those valuable transitions, performing the preparation operation in the counterclockwise direction makes them approach $|1\rangle$, while for those less important experiences, the preparation operation in the clockwise direction can be deployed on their quantum representations to make them closer to $|0\rangle$.

2) Depreciation Operation: After the process of preparation, the probabilities of selecting the experiences are closely associated with their TD-errors. However, in actual training, some experiences are replayed at high frequencies and may result in poor learning performance, which is called overtraining, and the limited size of the replay buffer may aggravate this situation [43]. In RL, overtraining reveals the issue of exploration–exploitation tradeoff [44]–[49]. Sufficient exploration in the state–action space helps prevent the algorithm from being trapped in locally optimal solutions while exploiting the current policy helps the algorithm converge as fast as possible. To achieve a balance between exploration and exploitation, the sample diversity is considered to enhance the learning performance of the agent. As such, the depreciation operation is developed for the experiences according to the replaying process. This is achieved by iteratively modifying their probabilities once the transitions are selected, whose effect contains and is greater than the utilization of the importance-sampling correction, which is demonstrated in the ablation experiments in the supplementary material.

Once the experiences are selected and put back in the memory buffer for training, their importance to the agent is unavoidably changed, not only because their TD-errors have been changed but also in that they are no longer brand new to the agent. Therefore, their probability amplitudes need to be modified. From this perspective, another unitary transformation

$$U_{\boldsymbol{\omega}} = \begin{bmatrix} \cos(\boldsymbol{\omega}) & -\sin(\boldsymbol{\omega}) \\ \sin(\boldsymbol{\omega}) & \cos(\boldsymbol{\omega}) \end{bmatrix} \tag{16}$$

is used for the depreciation operation, with $\boldsymbol{\omega} \in R$. In particular, it is implemented on the selected experiences using the Grover iteration. Every time the experiences have been accepted, their quantum representations go through a unitary transformation as follows:

$$\left|\psi_f^{(k)}\right\rangle \leftarrow U_{\boldsymbol{\omega}}\left|\psi_f^{(k)}\right\rangle. \tag{17}$$

The value of $U_{\boldsymbol{\omega}}$, or more precisely $\boldsymbol{\omega}$, should be adapted to the specific scenario. In experience replay, when the buffer is full, new transitions are orderly put in the buffer, with the old ones replaced. Besides, the period of the experiences being replaced is a fixed number of steps. Hence, a transition will be kept in the buffer for fixed-time steps, before it is replaced. In that case, during fixed training steps, the total replaying times of all the experiences are fixed. A large value of the maximum number of replaying times among all the experiences (denoted as $\mathrm{RT}_{\max}$) reveals an uneven replaying distribution, which means that some experiences have outstanding priorities compared to other experiences. To weaken this phenomenon, a smaller $\boldsymbol{\omega}$ helps to retain those less important experiences; otherwise, a large depreciation factor might result in sharp declines in the accepting probabilities of those experiences. Hence, the value of $\boldsymbol{\omega}$ is decreased with $\mathrm{RT}_{\max}$.

In addition, $\boldsymbol{\omega}$ should be adapted to the training episode TE. In the early training stage, the importance of experiences is ambiguous. After a period of training, the TD-errors of some experiences tend to remain in large values, regardless of how many times they have been selected to update the network. Therefore, it is feasible to "intensify" the accepting probabilities of the experiences that have been replayed with more times compared with others at the early training stage and to "cool" down their accepting probabilities to avoid overtraining at the later stage. This is realized by increasing $\boldsymbol{\omega}$ with the training episode TE. Finally, the depreciation factor $\boldsymbol{\omega}$ is given as

$$\boldsymbol{\omega} = \frac{\tau_1}{\mathrm{RT}_{\max}\left(1 + e^{\tau_2/\mathrm{TE}}\right)} \tag{18}$$

where $\tau_1, \tau_2 \in R$ are two hyperparameters.

3) Experience Selection by Quantum Observation: To accomplish the training process, samples are chosen from the buffer and fed into the network for learning. Here, we draw from the quantum measurement principle and determine the probabilities of experiences based on quantum observation. For the kth qubit in state $|\psi_f^{(k)}\rangle$, observing its probability of being accepted is $|\langle 1|\psi_f^{(k)}\rangle|^2$, which is actually the probability of measuring $|1\rangle$. Then, by normalizing the probability based on all transitions, we obtain its replaying probability as

$$b_k = \frac{\left|\left\langle 1|\psi_f^{(k)}\right\rangle\right|^2}{\sum_i \left|\left\langle 1|\psi_f^{(i)}\right\rangle\right|^2}. \tag{19}$$

The process of experience selection is summarized as in Fig. 5, where each transition has its own probability in the buffer. Inspired by the quantum observation principle, this process determines the probabilities of being selected among the buffer. During the sampling process, several times of sampling one transition from the experience buffer are performed under fixed probabilities. The sampling times are consistent with the size of the minibatch, which is set as 32 in the simulations.

Remark 1: The process of obtaining minibatch data from the buffer with fixed probabilities is a sampling process with replacement. For each sampling process, the selected sample is still retained in the buffer and is reset to the uniform state after being sampled. This idea is inspired by the phenomena that observing a quantum system makes its state collapses. In that case, the quantum operation (i.e., the preparation operation and depreciation operation) on the selected quantum experience starts from the uniform state, rather than its previous state.

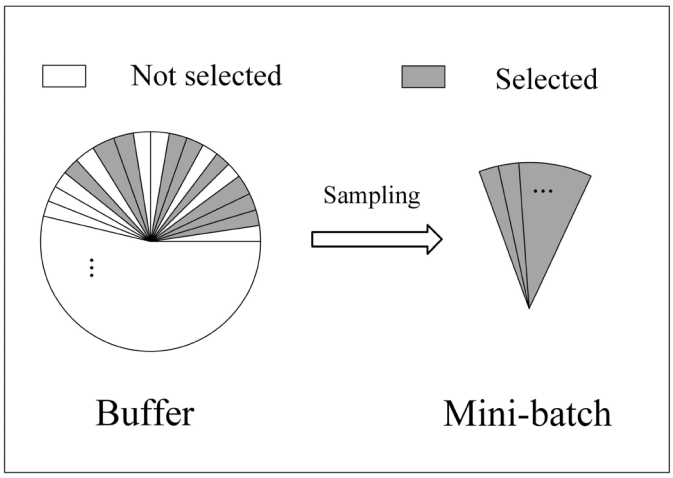

Fig. 5. Observation process for experience replay. The buffer is composed of a number of transitions, where each one is accompanied with its probability drawn from the quantum observation principle. The transitions are sampled out from buffer according to their replaying probabilities to compose the minibatch data.

D. Implementation

An integrated DRL-QER algorithm is shown as in Algorithm 1. During each step, the agent encounters transition e_t. Considering that the newly generated transition does not have a TD-error, we assign the maximum TD-error for it, that is, $\delta_t = \delta_{\max}$ to give it a high priority. This guarantees that every new experience is sampled with a high priority. Then, the transition is represented as a qubit, with its initial state as $|\psi_0\rangle$. The preparation operation using the Grover iteration is performed on the experience until it reaches the final state $|\psi_f^{(k)}\rangle$. After the buffer is full, transitions are sampled with probabilities proportional to their amplitudes of quantum states, and those selected samples compose the minibatch data for training the network. For the selected transitions, after being reset to the uniform state, their corresponding quantum representations are manipulated through the preparation operation to adapt to new priorities, and the depreciation operation to adapt to the replaying times. This procedure is carried out iteratively until the algorithm converges.

Remark 2: The proposed QRL-QER method works by representing the classical information (experiences) into quantum forms and performing quantum operations. Although it is inspired by quantum laws, the process can be simulated on a classical device. Hence, it is a quantum-inspired algorithm and does not need to be implemented on a quantum device.

Remark 3: In DRL, the buffer is utilized to store the past experiences, which can be used to update the parameters of the agent. During this process, the agent interacts with the environment under the new network parameters. Hence, the experiences in the buffer should be updated after some training

Algorithm 1: DRL-QER Algorithm

Input: size of experience buffer M, size of mini batch N.
Initialize the preparation factor σ, the depreciation factor $\boldsymbol{\omega}$, the maximum TD-error $\delta_{\max}$, the replayed time vector $cn = [cn_1, cn_2, \ldots, cn_M] = \vec{0}$, the index in the buffer $k = 1$, a variable $LF = False$;
for $TE = 1 \rightarrow TrainingFrames$ **do**
  Observe s_1 and choose $a_1 \sim \pi(s_1)$;
  for $t = 1 \rightarrow T$ **do**
    Observe r_t, s_{t+1} and then obtain a transition e_t;
    if s_{t+1} *is terminal* **then**
      break;
    end
    Initialize the kth qubit as the uniform state $|\psi_0\rangle$;
    Set $P_k = |\delta_{max}|$ and obtain m_k according to (14);
    Perform the preparation operation on kth qubit using Grover iteration, and obtain its final state $|\psi_f^{(k)}\rangle = (U_\sigma)^{m_k}|\psi_0\rangle$;
    Store the transition e_t with its quantum representation $|\psi_f^{(k)}\rangle$ in the buffer;
    if LF==*True* **then**
      Determine the probabilities of the experiences by quantum observations and obtain their replaying probabilities $[b_1, b_2, ..., b_M]$ according to (19);
      Update the preparation factor σ and the depreciation factor $\boldsymbol{\omega}$;
      for $j = 1 \rightarrow N$ **do**
        Sampling a transition with its index in the buffer as $d \in \{1, 2, ..., M\}$ based on $[b_1, b_2, ..., b_M]$;
        Reset the dth qubit back to the uniform state $|\psi_0\rangle$;
        Compute its TD-error $\delta_j = r_j + \eta \max_a Q_{target}(s_{j+1}, a) - Q(s_j, a_j)$;
        Obtain its priority $P_d = |\delta_j|$ and obtain m_d according to (14);
        Update the replaying time cn_d by $cn_d = cn_d + 1$;
        Conduct a complex Grover iteration process including both the preparation operation and depreciation operation on the experience's quantum representation $|\psi_f^{(d)}\rangle = (U_{\boldsymbol{\omega}})^{cn_d}(U_\sigma)^{m_d}|\psi_0\rangle$;
        Update $\delta_{max} = \max(\delta_{max}, |\delta_j|)$ and update $RT_{\max} = \max(cn_1, cn_2, ..., cn_M)$;
      end
      Update weights θ by stochastic gradient descent;
      Copy weights into target network $\theta_{target} \leftarrow \theta$;
      Remove the kth quantum representation of experience from the buffer and reset $cn_k = 0$;
    end
    $k \leftarrow k + 1$;
    if $k > M$ **then**
      Set $LF = True$ and set $k = 1$;
    end
    Choose action $a_{t+1} \sim \pi(s_{t+1})$;
  end
end

steps to gain a better training effect. To achieve this, the buffer is set as a fixed size and the oldest experience is discarded to make room for the newly produced experience (reset $k = 1$ in Algorithm 1) when the buffer is full ($k >= M$ in Algorithm 1). In addition, the procedure of updating the parameters of the network begins after the buffer is full, that is, the variable LF is set *True*. This technique is also applied to DRL-PER and DCRL to achieve a fair comparison in the following experiment section. In the implementation of Algorithm 1, we set a predefined value for the maximum value of TD-error, that is, $\delta_{\max}$. During the entire learning process, $\delta_{\max}$ should be updated once a larger TD-error is found. As such, new $\delta_{\max}$ is assigned to the future newly generated transitions to give them the highest priorities.

IV. Experiments

To test the proposed DRL-QER algorithm, several groups of experiments are carried out on Atari games with comparison to two benchmark algorithms (DRL-PER and DCRL). In addition, DRL-QER is combined with a double network and a dueling network and tested on additional experiments to verify its performance.

A. Setup

The experiments are carried out on the widely used platform OpenAI Gym to play Atari 2600 games [50], and the testing games can be divided into four categories, namely: 1) shooting games; 2) antagonistic games; 3) racing games; and 4) strategy

TABLE I
HYPERPARAMETERS ADJUSTMENTS IN NUMERICAL EXPERIMENTS

Altered Hyper-parameters		
Hyper-parameter	Original Value	Altered Value
Training Frames	5×10^7	5×10^6
Preparation sub-factor ζ_1	—	0.03π
Preparation sub-factor ζ_2	—	2×10^6
Depreciation sub-factor τ_1	—	π
Depreciation sub-factor τ_2	—	1×10^6
Parameter μ of m	—	100
Parameter ι of m	—	0.25π

games. For all games, the agent takes high-dimensional data (210×160 color video) as input to learn good policies. In order to win the games, the agent has to plan over the long term. All the experiments are deployed on a computer of ThinkStation P920 with 24xCPU@2.40 GHz, Nvidia Tesla p5000, Ubuntu 16.04.5 LTS, and Python.

To verify the effectiveness of DRL-QER, two baseline algorithms, including: 1) DRL-PER [13] and 2) DCRL [14] are also tested for comparison. The sampling method in DRL-QER can be regarded as a generalization of that in DRL-PER. DRL-PER sample experiences according to their TD-errors with proportional prioritization, and this can be regarded as a situation in which DRL-QER does not consider the influence of overtraining and discards the depreciation operation.

When deploying DRL-QER on Atari 2600 games, we adopt a similar neural-network architecture and the same hyperparameter setting to those in [11] and [13]. Considering the hardware limitation and the high computation cost, we make some fine tuning on the hyperparameters and train for five million frames instead of 50 million frames to avoid the expensive cost of training. The quantum operations are implemented on classical computers via necessary approximation for the simulation of DRL-QER, that is, the states of qubit systems are represented by 2-D complex vectors and the preparation operation and depreciation operation are performed in the form of unitary matrix transformation. In addition, the normalized probabilities are stored in a special binary heap called "sum tree," where the value of a parent node is the sum of all values of its children. Last but not least, for performing necessary operations on experiences, we introduce some hyperparameters for the preparation factor σ and the depreciation factor $\boldsymbol{\omega}$, and their values are provided in Table I. Specifically, ζ_1 and ζ_2 are tuned to make sure that $\mu\cdot\sigma$ approaches 0 at the early stage and approaches $(3\pi/4)$ at the ending stage. Similarly, τ_1 and τ_2 are tuned to ensure that $\boldsymbol{\omega}$ is close to 0 and $(\mu\sigma/\mathrm{RT}_{\max})$ in the beginning and the ending periods, respectively. The other hyperparameters of DRL-QER are selected by performing a grid search on the game breakout.

TABLE II
AVERAGE REWARDS PER EPISODE OF DRL-PER, DCRL, AND DRL-QER

Game Name	DRL-PER(±std)	DCRL(±std)	DRL-QER(±std)
Alien	1270.2(±341.3)	1223.9(±297.1)	**1309.3**(±348.8)
Beam Rider	1448.3(±290.2)	**1594.6**(±302.3)	1508.2(±330.4)
Breakout	5.7(±2.1)	5.2(±1.9)	**5.8**(±2.8)
Carnival	1142.5(±376.2)	**1235.1**(±331.5)	1214.0(±413.4)
Enduro	43.0(±20.6)	**44.3**(±20.5)	42.7(±19.2)
Freeway	60.6(±6.6)	60.2(±5.8)	**60.7**(±6.3)
Kangaroo	1129.3(±291.3)	**1237.3**(±303.2)	1143.3(±324.6)
Kung-Fu Master	521.3(±328.8)	670.0(±434.8)	**712.0**(±370.9)
Ms. Pacman	1862.2(±661.8)	**1918.9**(±741.2)	1903.5(±735.3)
River Raid	**2248.6**(±626.8)	1239.5(±251.6)	1479.5(±307.5)
Road Runner	2312.7(±897.7)	**3615.3**(±1543.8)	3208.7(±1298.7)
Space Invaders	679.2(±321.2)	735.1(±250.8)	**741.5**(±317.2)

B. Experimental Results

The experiments of the 12 games are deployed to compare the performance of DRL-PER, DCRL, and DRL-QER. Similar to DCRL [14], AER [16], and CER [17], each simulation is run three times to collect the average performance for a fair comparison. After the training process, we test the agents for 150 episodes and the average rewards with the standard deviation are summarized in Table II. It is clear that DRL-QER outperforms DRL-PER in most of the testing games. The statistical analysis also reveals that DRL-QER and DCRL achieve a comparative performance for the 12 games and they have different advantages in different games. Considering the total reward metric tends to be noisy because small changes to the weights of the DRL agent can lead to large changes in the distribution of states the agent visits [11], we take the estimated action value as the metric, which has been demonstrated to be more stable than the reward metric to reveal the training performance of DRL methods. In particular, we divide the training phase into 125 epochs and the average action values of the testing frames are recorded after each training epoch. The experimental results demonstrate that the learning progress of DRL-QER is faster and more robust than that of DRL-PER and is not worse than DCRL. It is worth noting that DRL-QER merely changes the priorities of experiences without affecting the convergence of the baseline DRL method [13]. However, the efficient use of samples contributes to a faster convergence under limited training epochs. Hence, the trends of the training curves (shown in Fig. 6) reflect the superiority of our method. What is more, DCRL involves many parameters that are difficult to fine tune for different games, while DRL-QER does not require prior knowledge to fine tune parameters. In fact, the parameter settings of DRL-QER are almost the same across the 12 games. From this respective, DRL-QER is an effective and general approach with enhanced performance.

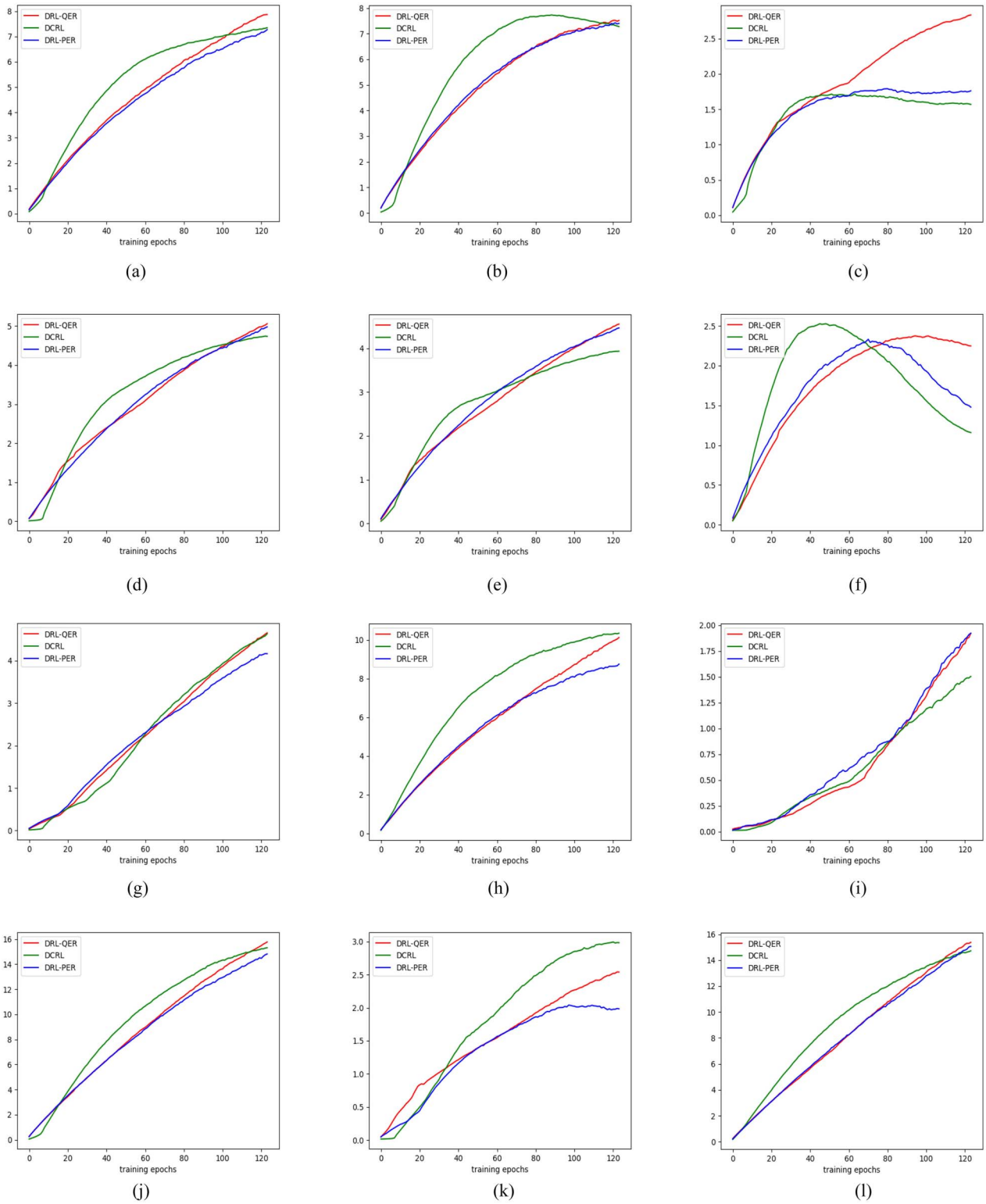

Fig. 6. Performance of DRL-QER with comparison to DRL-PER and DCRL regarding the average Q value. (a) Space invaders. (b) Carnival. (c) Breakout. (d) Freeway. (e) Beam rider. (f) Kung-Fu master. (g) Road runner. (h) River raid. (i) Enduro. (j) Ms. Pacman. (k) Kangaroo. (l) Alien.

C. Additional Exploratory Experiments

The proposed DRL-QER aims at taking advantage of quantum characteristics in the experience replay mechanism. To figure out whether this mechanism can be applied to other memory-based RL algorithms, we further apply DRL-QER to the double network [51] and the dueling network [52] and implement experimental simulations on randomly selected four games using the same hyperparameter setting in Table I. In Fig. 7, both double DQN and dueling DQN algorithms using the QER method show faster convergence regarding the average Q value compared with their classical counterparts. The

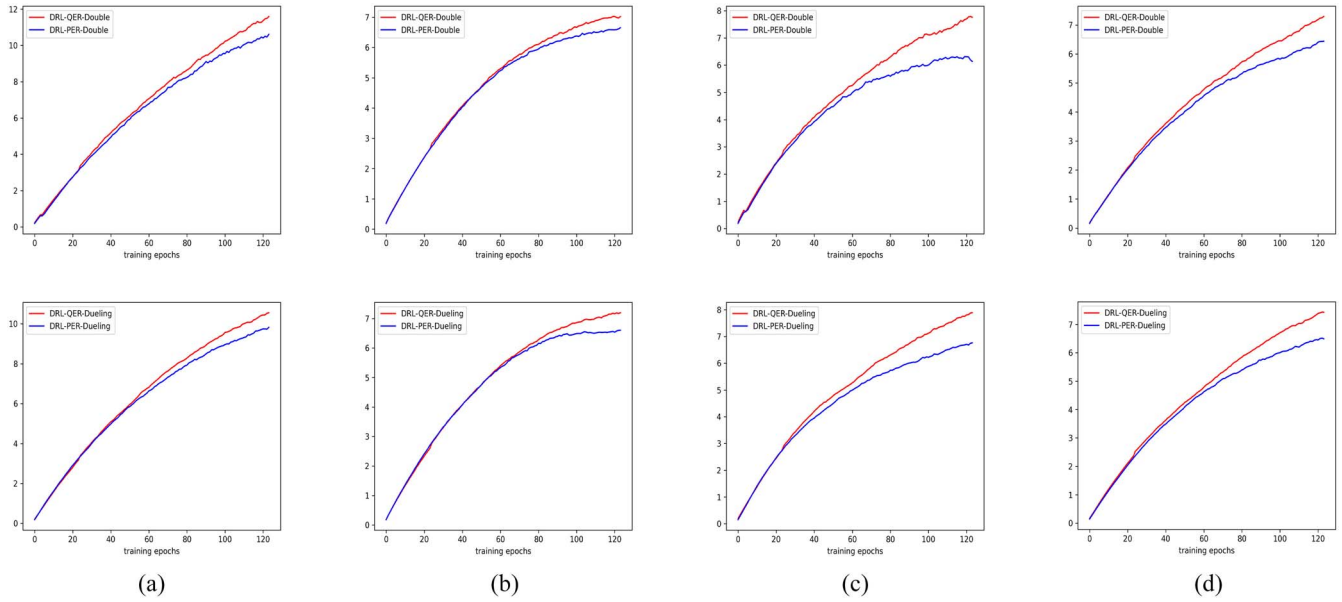

Fig. 7. Performance of DRL-QER-Double/DRL-PER-Double and DRL-QER-Dueling/DRL-PER-Dueling regarding the average Q value. (a) Alien. (b) Carnival. (c) River raid. (d) Space invaders.

TABLE III
AVERAGE REWARDS PER EPISODE OF DRL-PER WITH DOUBLE NETWORK AND DRL-QER WITH DOUBLE NETWORK

Game Name	DRL-PER(±std)	DRL-QER(±std)
Alien	1215.2(±392.0)	**1275.0**(±350.0)
Carnival	2420.5(±315.2)	**2480.3**(±319.2)
River Raid	**1906.2**(±522.6)	1870.9(±310.7)
Space Invaders	723.6(±229.9)	**750.3**(±262.8)

TABLE IV
AVERAGE REWARDS PER EPISODE OF DRL-PER WITH DUELING NETWORK AND DRL-QER WITH DUELING NETWORK

Game Name	DRL-PER(±std)	DRL-QER(±std)
Alien	801.4(±131.6)	**854.6**(±140.3)
Carnival	1317.0(±376.5)	**1367.7**(±380.2)
River Raid	**3249.2**(±522.0)	3007.3(±456.7)
Space Invaders	756.1(±229.5)	**758.8**(±262.2)

average rewards of DRL-QER-Double and DRL-PER-Double are summarized in Table III. Besides, the average rewards of DRL-QER-Dueling and DRL-PER-Dueling are summarized in Table IV. From these two tables, the average rewards per episode are also increased in the "double network" and "dueling network" for three games except for Riverraid.

V. CONCLUSION

In this article, the DRL-QER method was proposed by introducing quantum characteristics into the process of experience replay in DRL to guarantee that the learning scheme focuses on what the agent has learned from the interaction with the environment instead of the prior knowledge. In DRL-QER, the experiences are represented in quantum states, whose amplitudes are correlated with the TD-errors and the replaying times. In particular, the preparation operation and depreciation operation in DRL-QER help speed up the training progress and achieve an improved sampling efficiency. The experimental results demonstrated the superior performance of the proposed DRL-QER over DRL-PER and DCRL. Comparisons of DRL-PER and DRL-QER in dueling DQN and double DQN further showed that DRL-QER can also achieve improved performance for other memory-based DRL algorithms. Our future work will focus on in-depth theoretical research on the convergence of DRL-QER and quantum-enhanced RL along with its applications to other continuous control methods, such as deep deterministic policy gradient (DDPG) [53], [54] and soft actor critic [55].

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.
[2] M. L. Littman, "Reinforcement learning improves behaviour from evaluative feedback," *Nature*, vol. 521, no. 7553, pp. 445–451, May 2015. [Online]. Available: https://doi.org/10.1038/nature14540
[3] J.-A. Li *et al.*, "Quantum reinforcement learning during human decision-making," *Nat. Human Behav.*, vol. 4, no. 3, pp. 294–307, Mar. 2020.
[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[5] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 8, pp. 1915–1929, Aug. 2013.
[6] H. Goh, N. Thome, M. Cord, and J.-H. Lim, "Learning deep hierarchical visual feature coding," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 25, no. 12, pp. 2212–2225, Dec. 2014.
[7] G. Hinton *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–97, Nov. 2012.
[8] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[9] P. P. Brahma, D. Wu, and Y. She, "Why deep learning works: A manifold disentanglement perspective," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 10, pp. 1997–2008, Oct. 2016.

[10] H. Tembine, "Deep learning meets game theory: Bregman-based algorithms for interactive deep generative adversarial networks," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1132–1145, Mar. 2020.

[11] V. Mnih *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, Feb. 2015.

[12] L.-J. Lin, "Self-improving reactive agents based on reinforcement learning, planning and teaching," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 293–321, 1992.

[13] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proc. IEEE Int. Conf. Learn. Represent.*, 2016.

[14] 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.

[15] G. Novati and P. Koumoutsakos, "Remember and forget for experience replay," in *Proc. 36th Int. Conf. Mach. Learn.*, vol. 97. Long Beach, CA, USA, Jun. 2019, pp. 4851–4860. [Online]. Available: http://proceedings.mlr.press/v97/novati19a.html

[16] P. Sun, W. Zhou, and H. Li, "Attentive experience replay," in *Proc. 34th AAAI Conf. Artif. Intell., 32nd Innov. Appl. Artifi. Intell. Conf., 10th AAAI Symp. Educ. Adv. Artif. Intell.*, New York, NY, USA, Feb. 2020, pp. 5900–5907. [Online]. Available: https://aaai.org/ojs/index.php/AAAI/article/view/6049

[17] H. Liu, A. Trott, R. Socher, and C. Xiong, "Competitive experience replay," in *Proc. 7th Int. Conf. Learn. Represent. ICLR*, New Orleans, LA, USA, 2019. [Online]. Available: https://openreview.net/forum?id=Sklsm20ctX

[18] H.-S. Chang, E. Learned-Miller, and A. McCallum, "Active bias: Training more accurate neural networks by emphasizing high variance samples," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1002–1012.

[19] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124–134.

[20] L. K. Grover, "Quantum computers can search arbitrarily large databases by a single query," *Phys. Rev. Lett.*, vol. 79, pp. 4709–4712, Dec. 1997. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.79.4709

[21] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, Sep. 2017. [Online]. Available: https://doi.org/10.1038/nature23474

[22] X.-D. Cai *et al.*, "Entanglement-based machine learning on a quantum computer," *Phys. Rev. Lett.*, vol. 114, Mar. 2015, Art. no. 110504. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.114.110504

[23] Z. Li, X. Liu, N. Xu, and J. Du, "Experimental realization of a quantum support vector machine," *Phys. Rev. Lett.*, vol. 114, Apr. 2015, Art. no. 140504. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.114.140504

[24] K. Beer *et al.*, "Training deep quantum neural networks," *Nat. Commun.*, vol. 11, p. 808, Feb. 2020. [Online]. Available: https://doi.org/10.1038/s41467–020–14454–2

[25] L. Bai, L. Rossi, L. Cui, J. Cheng, and E. R. Hancock, "A quantum-inspired similarity measure for the analysis of complete weighted graphs," *IEEE Trans. Cybern.*, vol. 50, no. 3, pp. 1264–1277, Mar. 2020.

[26] W. Ding, C. Lin, and Z. Cao, "Deep neuro-cognitive co-evolution for fuzzy attribute reduction by quantum leaping PSO with nearest-neighbor memeplexes," *IEEE Trans. Cybern.*, vol. 49, no. 7, pp. 2744–2757, Jul. 2019.

[27] S. C. Kak, "Quantum neural computing," in *Advances in Imaging and Electron Physics*, vol. 94. Amsterdam, The Netherlands: Elsevier, 1995, pp. 259–313.

[28] V. Dunjko and H. J. Briegel, "Machine learning & artificial intelligence in the quantum domain: A review of recent progress," *Rep. Progr. Phys.*, vol. 80, no. 7, 2018, Art. no. 074001.

[29] G. Carleo and M. Troyer, "Solving the quantum many-body problem with artificial neural networks," *Science*, vol. 355, no. 6325, pp. 602–606, Feb. 2017. [Online]. Available: https://science.sciencemag.org/content/355/6325/602

[30] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum principal component analysis," *Nat. Phys.*, vol. 10, no. 9, pp. 631–633, Sep. 2014. [Online]. Available: https://doi.org/10.1038/nphys3029

[31] D. Dong, C. Chen, H. Li, and T.-J. Tarn, "Quantum reinforcement learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 38, no. 5, pp. 1207–1220, Oct. 2008. [Online]. Available: https://doi.org/10.1109/TSMCB.2008.925743

[32] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, "Quantum speedup for active learning agents," *Phys. Rev. X*, vol. 4, no. 3, 2014, Art. no. 031002.

[33] V. Dunjko, J. M. Taylor, and H. J. Briegel, "Quantum-enhanced machine learning," *Phys. Rev. Lett.*, vol. 117, Sep. 2016, Art. no. 130501. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.117.130501

[34] L. Lamata, "Basic protocols in quantum reinforcement learning with superconducting circuits," *Sci. Rep.*, vol. 7, p. 1609, May 2017.

[35] F. Cárdenas-López, L. Lamata, J. C. Retamal, and E. Solano, "Multiqubit and multilevel quantum reinforcement learning with quantum technologies," *PloS One*, vol. 13, no. 7, 2018, Art. no. e0200455.

[36] W. Hu and J. Hu, "Training a quantum neural network to solve the contextual multi-armed bandit problem," *Nat. Sci.*, vol. 11, no. 1, pp. 17–27, 2019.

[37] D. Crawford, A. Levit, N. Ghadermarzy, J. S. Oberoi, and P. Ronagh, "Reinforcement learning using quantum boltzmann machines," *Quantum Inf. Comput.*, vol. 18, nos. 1&2, pp. 51–74, 2018. [Online]. Available: http://www.rintonpress.com/xxqic18/qic-18–12/0051–0074.pdf

[38] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[39] L. C. Baird, "Residual algorithms: Reinforcement learning with function approximation," in *Proc. Mach. Learn.*, 1995, pp. 30–37.

[40] B. Luo, Y. Yang, and D. Liu, "Adaptive *Q*-learning for data-based optimal output regulation with experience replay," *IEEE Trans. Cybern.*, vol. 48, no. 12, pp. 3337–3348, Dec. 2018.

[41] Z. Ni, N. Malla, and X. Zhong, "Prioritizing useful experience replay for heuristic dynamic programming-based learning systems," *IEEE Trans. Cybern.*, vol. 49, no. 11, pp. 3911–3922, Nov. 2019.

[42] D. Dong and I. R. Petersen, "Quantum control theory and applications: A survey," *IET Control Theory Appl.*, vol. 4, no. 12, pp. 2651–2671, 2010.

[43] T. De Bruin, J. Kober, K. Tuyls, and R. Babuška, "The importance of experience replay database composition in deep reinforcement learning," in *Proc. Deep Reinforcement Learn. Workshop (NIPS)*, 2015.

[44] S. Ishii, W. Yoshida, and J. Yoshimoto, "Control of exploitation–exploration meta-parameter in reinforcement learning," *Neural Netw.*, vol. 15, nos. 4–6, pp. 665–687, 2002.

[45] P. Abbeel and A. Y. Ng, "Exploration and apprenticeship learning in reinforcement learning," in *Proc. 22nd Int. Conf. Mach. Learn.*, 2005, pp. 1–8.

[46] T. Mannucci, E.-J. van Kampen, C. de Visser, and Q. Chu, "Safe exploration algorithms for reinforcement learning controllers," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 4, pp. 1069–1081, Apr. 2018.

[47] D. Dong, C. Chen, J. Chu, and T.-J. Tarn, "Robust quantum-inspired reinforcement learning for robot navigation," *IEEE/ASME Trans. Mechatron.*, vol. 17, no. 1, pp. 86–97, Feb. 2012.

[48] C. Chen, D. Dong, H. 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.

[49] Q. Zhang and D. Zhao, "Data-based reinforcement learning for nonzero-sum games with unknown drift dynamics," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2874–2885, Aug. 2019.

[50] G. Brockman *et al.*, "OpenAI gym," CoRR, vol. abs/1606.01540, 2016. [Online]. Available: http://arxiv.org/abs/1606.01540.

[51] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc. 30th AAAI Conf. Artif. Intell.*, Phoenix, AZ, USA, Feb. 2016, pp. 2094–2100. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12389

[52] Z. Wang, T. Schaul, M. Hessel, H. van Hasselt, M. Lanctot, and N. de Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. 33rd Int. Conf. Mach. Learn.*, New York, NY, USA, Jun. 2016, pp. 1995–2003. [Online]. Available: http://proceedings.mlr.press/v48/wangf16.html

[53] T. P. Lillicrap *et al.*, "Continuous control with deep reinforcement learning," in *Proc. 4th Int. Conf. Learn. Represent.*, 2016. [Online]. Available: http://arxiv.org/abs/1509.02971

[54] D. Zhao and Y. Zhu, "MEC–A near-optimal online reinforcement learning algorithm for continuous deterministic systems," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 2, pp. 346–356, Feb. 2015.

[55] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn. ICML*, Stockholm, Sweden, Jul. 2018, pp. 1856–1865. [Online]. Available: http://proceedings.mlr.press/v80/haarnoja18b.html

Qing Wei (Graduate Student Member, IEEE) received the B.E. degree in automation from the Department of Control and Systems Engineering, School of Management and Engineering, Nanjing University, Nanjing, China, in 2018, where he is currently pursuing the M.S. degree in control science and engineering.

His current research interests include machine learning and deep reinforcement learning.

Hailan Ma (Graduate Student Member, IEEE) 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 Wale, 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, Man and Cybernetics Society.

Chunlin Chen (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 is currently a Professor and the Head of the Department of Control and Systems Engineering, School of Management and Engineering, Nanjing University, Nanjing, China. He was with the Department of Chemistry, Princeton University, from 2012 to 2013. He had visiting positions with the University of New South Wales, Canberra, ACT, Australia, and City University of Hong Kong, Hong Kong. His current research interests include machine learning, intelligent control, and quantum control.

Prof. Chen serves as the Chair of Technical Committee on Quantum Cybernetics, IEEE Systems, Man and 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 is currently a Scientia Associate Professor with the University of New South Wales, Canberra, ACT, Australia. He was with the Institute of Systems Science, Chinese Academy of Sciences, Beijing, China, and with the Institute of Cyber-Systems and Control, Zhejiang University, Zhejiang, China. He had visiting positions with Princeton University, Princeton, NJ, USA; RIKEN, Wako, Japan; University of Duisburg–Essen, Duisburg, Germany; and the University of Hong Kong, Hong Kong. His research interests include quantum control and machine learning.

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