Deep Reinforcement Learning for Control Design of Quantum Gates

Shouliang Hu, Chunlin Chen
Department of Control and Systems Engineering
Nanjing University
Nanjing, 210093, China
clchen@nju.edu.cn

Daoyi Dong
School of Engineering and Information Technology
University of New South Wales
Canberra, ACT 2600, Australia
daoyidong@gmail.com

***Abstract*—This paper investigates quantum gate control problems using the deep reinforcement learning algorithm, i.e., a model-free machine learning method. We implement the twin delayed deep deterministic policy gradient (TD3) algorithm to search for piece-wise constant control pulses for quantum gates through the trail interaction with the quantum system. Simulation results on four typical gates, including three one-qubit gates and a two-qubit CNOT gate, demonstrate that deep reinforcement learning exhibits improved performance for quantum gate control tasks. By setting punishment for steps in the reward function, DRL can automatically find a shorter control sequence than the traditional gradient-based algorithm (e.g., GRAPE algorithm) and the evolutionary algorithm (e.g., DE algorithm) while maintaining high control precision.**

***Index Terms*—Deep reinforcement learning, quantum control, quantum gate**

I. Introduction

Realizing high precision quantum control is crucial for practical quantum information processing and quantum simulation technology. One fundamental issue in quantum control is how to design a control law to manipulate the system for achieving objective goals like fast state preparation or high fidelity gate operation [1]. The traditional gradient-based method (e.g., GRAPE) [2–4] can achieve high precision in the control law design working with accurate gradient information. Nevertheless, the practical quantum devices working under noises cannot provide accurate gradient information. Evolutionary algorithms such as the genetic algorithm (GA) and differential evolution (DE) are model-independent and have successfully solved various quantum control tasks [5–11]. However, evolutionary algorithms are time-consuming and might not find optimal solutions for problems with high dimensions.

Reinforcement learning (RL) is a model-free machine learning approach exploring optimal control strategies by interacting with the environment [12]. Due to the development of deep learning techniques, deep reinforcement learning (DRL) algorithms have successfully solved various tasks such as Atari games and robot control problems. Recently, the use of RL for various quantum control tasks has attracted much attention [13–16]. For example, the tabular based Q-learning algorithm and modified deep RL algorithms have been applied to quantum states preparation tasks [17–23]. The dueling double deep Q-learning neural network (DDDQN) algorithm has been applied to two-qubit quantum gate control problem [24]. A universal cost function for quantum gate control is proposed and the trusted-region-policy-optimization is used to optimize the speed and fidelity of various one-qubit and two-qubit quantum gates [25]. In the quantum state preparation problem, by performing a comparative study on RL algorithms and the traditional GRAPE and Krotov [26] algorithm, it is revealed that RL could adaptively reduce the complexity of the control pulses [27].

High precision and fast gate operation control technology is crucial for near-term practical quantum computation technology. In this paper, we explore the use of deep reinforcement learning for solving quantum gate control problems. In particular, we implement the twin delayed deep deterministic policy gradient (TD3) [28] algorithm to control a universal set of quantum gates including several one-qubit gates and the two-qubit controlled-not gate. Numerical results demonstrated that DRL could optimize gate control precision and evolution time simultaneously. Compared to the traditional gradient-based GRAPE and evolutionary algorithms, DRL stands out as it provides a shorter control-pulse sequence that significantly lowers the control resources.

The rest of the paper is organized as follows. Section II presents a general formulation of quantum gate control for closed quantum systems. Section III introduces the TD3 algorithm and presents the control framework with detailed implementations. Numerical simulations for high precision control of a universal set of quantum gates are illustrated in Section IV. Conclusions are drawn in Section V.

II. Problem formulation

The evolution of a unitary transformation U in a closed n-level quantum system is governed by the Schrödinger equation:

$$\frac{d}{dt}U(t) = -iH(t)U(t), \quad U(0) = \mathbb{I}_n, \tag{1}$$

This work was supported in part by the National Key Research and Development Program of China (No.2018AAA0101100) and the National Natural Science Foundation of China (No.62073160).

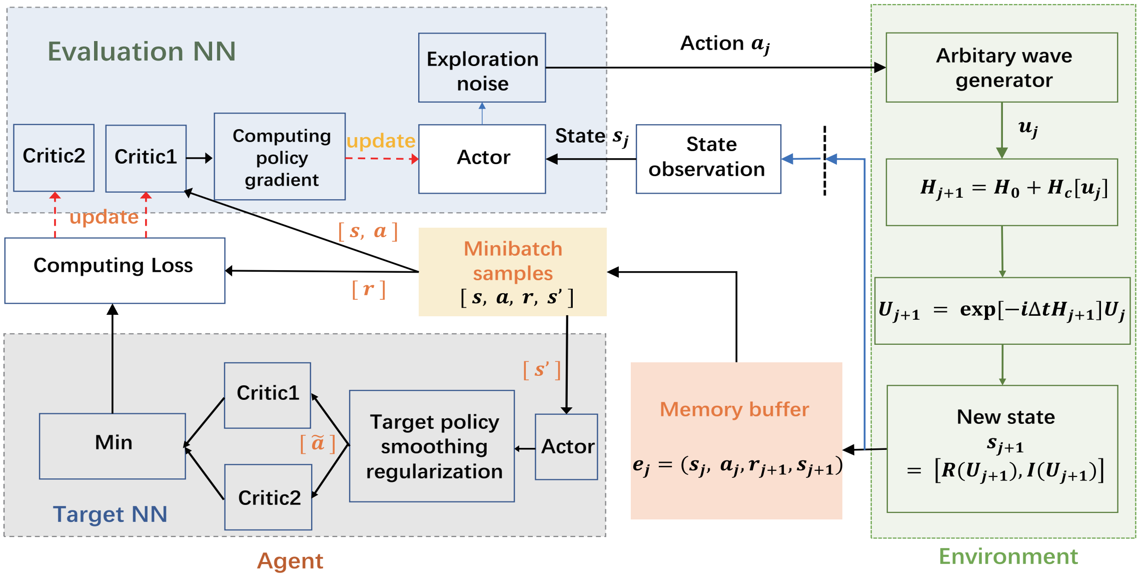

Evaluation NN
Exploration noise
Action a_j
Arbitary wave generator
u_j
Critic2
Critic1
Computing policy gradient
update
Actor
State s_j
State observation
$H_{j+1} = H_0 + H_c[u_j]$
update
[s, a]
Computing Loss
[r]
Minibatch samples [s, a, r, s']
$U_{j+1} = \exp[-i\Delta t H_{j+1}]U_j$
[s']
Critic1
Target policy smoothing regularization
Actor
Min
[$\tilde{a}$]
Critic2
Memory buffer
$e_j = (s_j, a_j, r_{j+1}, s_{j+1})$
New state $s_{j+1} = [R(U_{j+1}), I(U_{j+1})]$
Target NN
Agent
Environment

Fig. 1: Framework of quantum gate control with the TD3 algorithm

where $\mathbb{I}_N$ is the $N \times N$ identity matrix and $H(t)$ is the system Hamiltonian having the following form:

$$H(t) = H_0 + H_c[u(t)]. \tag{2}$$

Here H_0 represents the time-independent free Hamiltonian and $H_c[u(t)]$ is the control Hamiltonian that describes the interaction between the external control field $u(t)$ and the quantum system. We adopt the piece-wise constant control method that keeps the control function $u(t)$ constant over the jth evolution time duration $\triangle t$ giving the constant Hamiltonian H_j. During the jth transfer time interval, i.e., $[(j-1)\triangle t, j\triangle t]$, the unitary transformation U_j satisfies

$$U_j = \exp[-iH_j\triangle t]U_{j-1}, \tag{3}$$

and the final gate $U(T)$ after control of K steps is given by

$$U(T) = \exp[-iH_K\triangle t] \cdots \exp[-iH_1\triangle t]\mathbb{I}_n. \tag{4}$$

Given the target gate U_f and the final propagator $U(T)$, this paper uses the logarithmic infidelity L to assess the control precision, which is defined as

$$L = \log_{10}(1 - \frac{1}{n}|\text{trace}\{U_f^\dagger U(T)\}|), \tag{5}$$

where $L \to -\infty$ when the control is perfect. The task is to search for the optimal control sequence to minimize the final gate infidelity.

III. Methods

A. TD3 Algorithm

Reinforcement learning utilizes the idea of an intelligent agent interacting with the environment to search for optimal strategies.

At time j, the agent chooses an action a_j based on the observation of environment state s_j, following a strategy $\pi: s \to a$. Then, at next time $j+1$, the environment evolves to state s_{j+1} and gives the agent a reward r_{j+1}. The goal of RL is to learn an optimal policy π that maximizes the expected return $J = E_{s_i \sim p_\pi, a_i \sim \pi}[R_0]$. Here, the return at time j is defined as the sum of discounted future reward $R_j = \sum_{i=j}^{T} \gamma^{i-j} r(s_i, a_i)$, where s_T is the terminal state and γ is the discount factor.

This paper explores the control design of quantum gates using the twin delayed deep deterministic policy gradient (TD3) algorithm, which utilizes two kinds of neural networks including critic network $Q_\theta(s,a)$ and actor network $\pi_\phi(s)$ [29]. The actor network is used to generate actions. The evaluation critic network $Q_\theta(s,a)$ is used to evaluate the state-action value and learned by the Bellman equation.

After each interaction between the agent and environment, the transition $(s_j, a_j, r_{j+1}, s_{j+1})$ is stored in a memory replay buffer. TD3 samples a mini-batch of transition tuples from the memory buffer to update the evaluation critic network and calculates the sampled policy gradient to update the evaluation actor-network.

As a value-based reinforcement learning algorithm, TD3 makes several modifications to avoid overestimating value estimates. First, for critic network updating, the TD3 algorithm calculates a pair of critic values and then uses the minimum value to limit network updating. Second, TD3 suggests delaying policy updates to reduce per-update error. These improvements significantly enhance learning speed and performance [28].

B. Quantum Gate Control with TD3

Concept mapping is needed to implement the reinforcement learning algorithm for quantum gate control tasks. Considering

Algorithm 1 TD3 for quantum gate control

```
Input: Set the total learning episode $m$, the infidelity-based termination condition $L_c$ and the maximum step limit $j_{max}$.
Initialization: Initialize critic networks $Q_{\theta_1}, Q_{\theta_2}$ and actor network $\pi_\phi$ with random parameters $\theta_1, \theta_2, \phi$. Initialize target networks $\theta_1' \leftarrow \theta_1'$,$\theta_2' \leftarrow \theta_2'$, $\phi' \leftarrow \phi'$. Initialize an empty replay buffer $\mathcal{B}$.
training:
for episode = 1:$m$ do
    Initialize $j = 0, U_j = U_0 = I, s_j = s_0 = [\mathcal{R}(I), \mathcal{I}(I)]$.
    repeat
        Select an action $a_j$ with exploration noise $a_j \sim \pi_\phi(s_j) + \epsilon_j,\ \epsilon_j \sim \mathcal{N}(0, \sigma)$.
        New operator $U_{j+1}$ is given as
            $U_{j+1} = \exp[-iH_0 + H_c[a_j]\triangle t]U_j$
        and gives new state $s_{j+1} = [\mathcal{R}(U_{j+1}), \mathcal{I}(U_{j+1})]$.
        Calculate the instantaneous infidelity $L_{j+1}$ and gives the reward $r_{j+1}$ using equation (10).
        Store transition tuple $(s_j, a_j, r_{j+1}, s_{j+1})$ in $\mathcal{B}$.
        Sample mini-batch of N transitions $(s, a, r, s')$ from $\mathcal{B}$.
        $\widetilde{a} \leftarrow \pi_\phi'(s') + \epsilon', \epsilon' \in \text{clip}(\mathcal{N}(0, \sigma), -c, c)$.
        $y \leftarrow r + \gamma \text{min}_{i=1,2} Q_{\theta_i'}(s', \widetilde{a})$.
        Update critics $\theta_i \leftarrow \text{argmin}_{\theta_i} N^{-1} \sum (y - Q_{\theta_i}(s, a))^2$.
        if t mod d then
            Update $\phi$ by the deterministic policy gradient:
                $\nabla_\phi J(\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s, a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$.
            Update target networks:
                $\theta_i' \leftarrow \tau\theta_i + (1 - \tau)\theta_i',\ i = 1, 2;$
                $\phi' \leftarrow \tau\phi + (1 - \tau)\phi'.$
        end if
        $j \leftarrow j + 1$
    until $j = j_{max}$ or $L_{j+1} < L_c$
end for
Output: Use the actor network to generate the optimal control sequence $\mathbf{u}^*$.
```

a unitary transformation U_j at time j, the observed state s_j is defined as

$$s_j = [\mathcal{R}(U_j), \mathcal{I}(U_j)] \tag{6}$$

where $\mathcal{R}(U_j)$ and $\mathcal{I}(U_j)$ are the real and imaginary part of the matrix elements of U_j. Based on the state s_j, the agent chooses an action a_j and the system Hamiltonian H_{j+1} is given as

$$H_{j+1} = H_0 + H_c[a_j]. \tag{7}$$

The quantum system evolves following the Schrödinger equation and the propagator U_{j+1} is given as

$$U_{j+1} = \exp[-iH_{j+1}\triangle t]U_j, \tag{8}$$

The quantum environment state becomes $s_{j+1} = [\mathcal{R}(U_{j+1}), \mathcal{I}(U_{j+1})]$. At the end of an exploitation step, we evaluate the gate infidelity L_{j+1} defined as

$$L_{j+1} = \log_{10}(1 - \frac{1}{n}|\text{trace}\{U_f^\dagger U_{j+1}\}|), \tag{9}$$

and give the agent a reward r_{j+1} as

$$r_{j+1} = k_1|L_{j+1}| - k_2, \tag{10}$$

where k_1, k_2 are positive real numbers. Here, $k_1|L|$ motivates the agent to achieve higher fidelity while k_2 gives step punishment motivating the agent to find a shorter control sequence. The detailed algorithm is shown as in **Algorithm 1**.

In the beginning, the actor and critic networks, π_ϕ and $Q_{\theta_1}/Q_{\theta_2}$, are initialized with random parameters, respectively. Then these networks are copied to create $\pi_{\phi'}$ and $Q_{\theta_1'}/Q_{\theta_2'}$, that are used to calculate the target values. The replay buffer is established with a finite-sized cache $\mathcal{B}$, which abandons old data to store the new.

At each exploration step, the evaluation actor network $\pi_\phi(s)$ generates a continuous action a_j according to the input state s_j, i.e., $a_j = \pi_\phi(s_j) + \epsilon_j$, where ϵ_j is the additional exploration noise that follows the normal distribution $\mathcal{N}(0, \sigma)$ and is constrained as $\epsilon_j \sim [-c, c]$. Then, the transition tuple $(s_j, a_j, r_{j+1}, s_{j+1})$ is stored in $\mathcal{B}$.

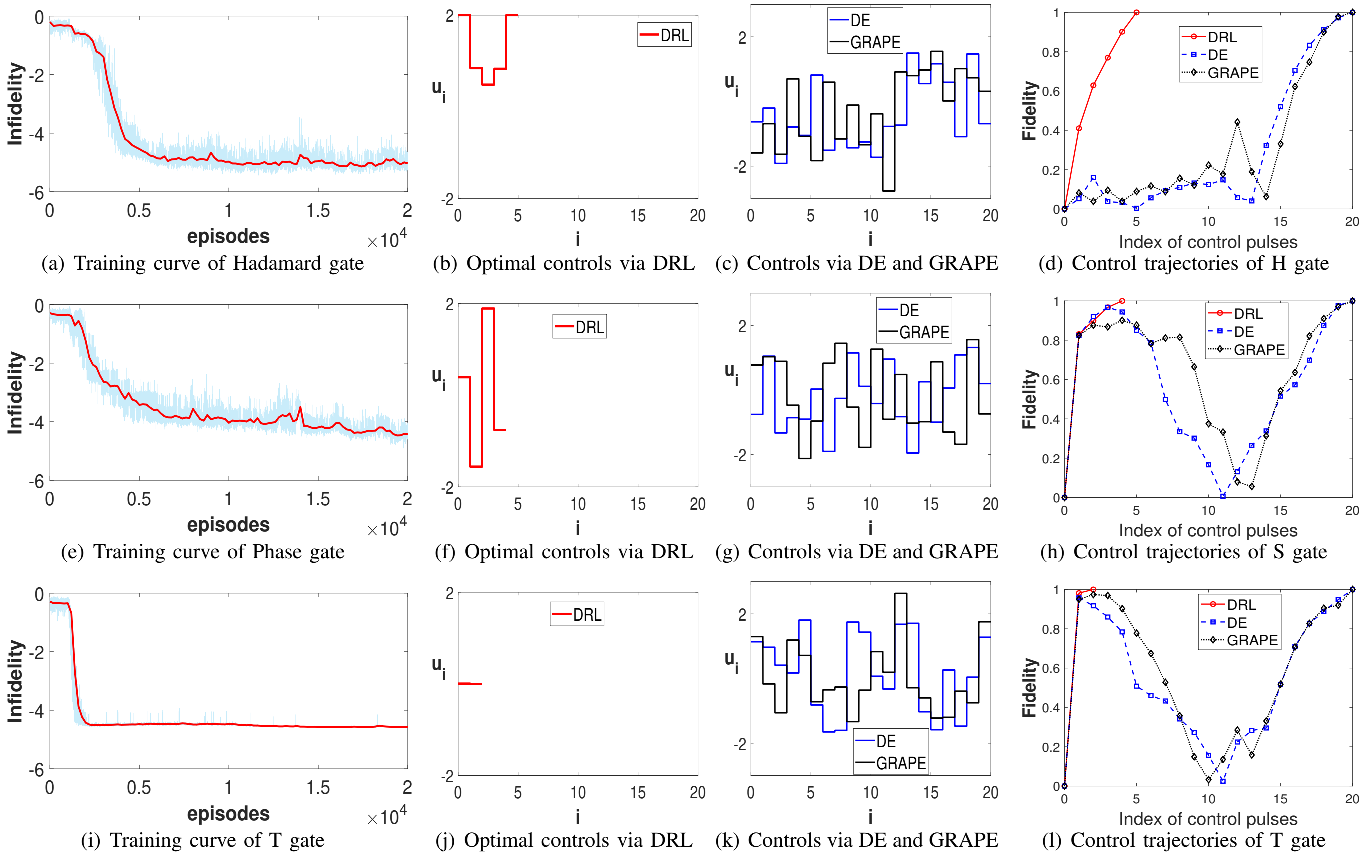
For each step, a mini-batch of transition tuples (s, a, r, s') is sampled from the memory buffer. The states s' are input to the target actor network to generate target actions a'. Particularly, TD3 smooths the value estimate by fitting the value of a small area around the target action, i.e., $\widetilde{a} \leftarrow \pi_\phi'(s') + \epsilon'$. As TD3 maintaining a pair of critic values along with a single action, the target value of action is chosen as the minimum $y = r + \gamma \min_{i,=1,2} Q_{\theta_i'}(s', \widetilde{a})$. And the pair of critic networks is updated by minimizing $loss(\theta_i) = N^{-1} \sum (y - Q_{\theta_i}(s, a))^2,\ i = 1, 2$.

With the idea of delaying the policy updates, the actor-network π_ϕ is updated with regard to Q_{θ_1} following the deterministic policy gradient algorithm every d iterations. And the target networks are softly updated.

At the end of a step, we calculate the infidelity of the obtained unitary operator. Once the infidelity L_{j+1} satisfies the termination condition or the step j reaches the maximum limit, the current exploitation episode ends. The state is reset and the algorithm turns to a new learning iteration. It should be noted that the infidelity-based break-out setting, along with the reward function setting, helps to find a shorter control sequence and accelerate the training.

IV. High Precision Control of the Universal Quantum Gates

This section presents the simulation results on control of three one-qubit gates and the CNOT gate. Both the actor and



(a) Training curve of Hadamard gate (b) Optimal controls via DRL (c) Controls via DE and GRAPE (d) Control trajectories of H gate

(e) Training curve of Phase gate (f) Optimal controls via DRL (g) Controls via DE and GRAPE (h) Control trajectories of S gate

(i) Training curve of T gate (j) Optimal controls via DRL (k) Controls via DE and GRAPE (l) Control trajectories of T gate

Fig. 2: Numerical results of one-qubit gates.

policy networks are fully connected networks. The detailed hyper-parameter settings are presented in TABLE I. All the experiments are implemented with Python 3.7 and run on a 6-core 3.70 GHz CPU with 16 GB memory.

TABLE I: Parameter settings for one-qubit gates and CNOT gate.

Hyper-parameters	**Values (one-qubit)**	**Values (CNOT)**
Actor network	{8,120,120,1}	{32,256,256,1}
Critic network	{9,120,120,1}	{36,256,256,1}
Mini-batch size	64	256
Replay memory	20000	50000
Factor of softly updating τ	0.004	0.004
Learning rate of actor NN	0.001	0.001
Learning rate of critic NN	0.002	0.002
Reward decay γ	0.9	0.9
Total episode	20000	40000

A. Control of One-qubit Gates

In this section, we consider two-level systems and focus on the control design of one-qubit gates, including Hadamard gate, Phase gate and $T_{\frac{\pi}{8}}$ gate:

$$H=\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}, S=\begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}, T_{\frac{\pi}{8}}=\begin{pmatrix}1 & 0\\ 0 & e^{\frac{\pi}{4}i}\end{pmatrix}. \quad (11)$$

The system Hamiltonian is considered to be

$$H(t)=\sigma_z+u_x(t)\sigma_x, \quad (12)$$

where σ_x, σ_z are Pauli matrices given as

$$\sigma_x=\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}, \sigma_z=\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}. \quad (13)$$

The termination infidelity is chosen as $L=-4$. We bound the control search space as $u\in[-2,2]$ and the reward function is set as

$$r=\begin{cases}|L|-1, \text{ if } |L|\in[0,4)\\ 5|L|, \text{ else.}\end{cases} \quad (14)$$

The average training curves, each calculated by 10 ten trail runs, are presented in Fig. 2. The duration time of a control step is set as $\Delta t=0.2$. The red lines show the average infidelity calculated by hundreds of episodes (500 for H and S gate and 200 for T gate). Average finally achieved gate infidelities are listed in TABLE II. DRL could achieve very high control precision ($L\to-4$).

In addition, we compare DRL with traditional methods, including GRAPE and DE. As the maximal exploitation step is set as 20, DRL could obtain shorter optimal control pulses with the step of five, four and two for Hadamard, Phase and T gate, respectively. In particular, for the T gate control problem, DRL has searched control pulses with tinny amplitudes,

TABLE II: The average gate infidelity calculated by ten trail runs.

Gate	Average Infidelity
Hadamard	-5.25
Phase	-4.49
T	-4.57

which means excellent energy-saving. The control trajectories demonstrate that the control process via DRL is superior to DE and GRAPE.

B. Control of CNOT Gate

CNOT gate acts on two-qubit and its matrix representation is given as:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{15}$$

The system Hamiltonian is considered to be

$$\begin{aligned} H(t) =& H_0 + \sum_{i=1}^{m} u_m(t) H_m \\ =& \sigma_z^{(1)} \otimes \sigma_z^{(2)} + u_1(t)\sigma_x^{(1)} \otimes I^{(2)} + u_2(t) I^{(1)} \otimes \sigma_x^{(2)} \\ &+ u_3(t)\sigma_y^{(1)} \otimes I^{(2)} + u_4(t) I^{(1)} \otimes \sigma_y^{(2)}. \end{aligned} \tag{16}$$

There are four external control fields to manipulate the gate evolution, which are bounded as $u_m \in [-2, 2], m = 1, \cdots, 4$. The duration time Δt of a control step is 0.2. The maximum step limit is 20. We choose the termination infidelity as $L = -3$ and set the reward function as

$$r = \begin{cases} |L| - 1, & \text{if } |L| \in [0, 2) \\ 2|L|, & \text{if } |L| \in [2, 3) \\ 4|L|, & \text{else.} \end{cases} \tag{17}$$

The average training curve by 17 trail runs is presented in Fig. 3. In the best cases, DRL can explore optimal control sequence with ten steps and achieve high precision with infidelity less than -5. Detailed results are listed in TABLE III. The obtained optimal control pulses are presented in Fig. 4.

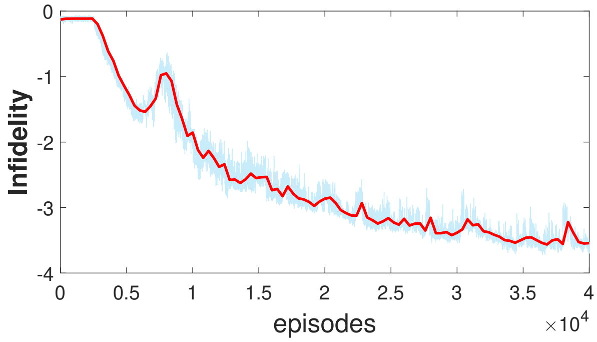


Fig. 3: Training curves of CNOT gate. The blue line presents average training infidelity via 17 trail runs. The red line represents the average infidelity calculated by 400 episodes.

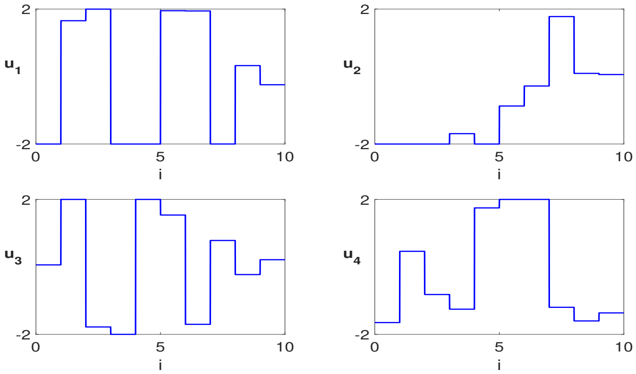


Fig. 4: Optimal control strategies for CNOT gate via TD3

TABLE III: Results of CNOT gate with 17 trail runs.

Trail runs	Infidelity
Best case	-5.07
median case	-4.59
Worst case	-0.10

V. Conclusion

This paper applied the DRL algorithm to quantum gate control tasks. Numerical results show that DRL could explore fast and high precision control pulses for the universal quantum gates including three one-qubit gates and a CNOT gate through the action-state interactions. Traditional algorithms like GRAPE or evolutionary algorithms usually work with fixed control time. Nevertheless, our DRL-based method may automatically optimize the control time by setting punishment for steps in the reward function.

Our future work will focus on the extension of DRL for quantum control tasks with complicated parameter fluctuations and high control dimensions. The exploration space grows exponentially with the rise of the system dimension and control noises, which challenges the efficiency of DRL algorithms. We will also try other advanced DRL algorithms (e.g., PPO) with transfer learning or incremental learning to further improve the learning performance for quantum control tasks.

References

[1] D. Dong, "Learning control of quantum systems," *Encyclopedia of Systems and Control, J. Baillieul, T. Samad (eds.), Springer-Verlag London Ltd*, pp. 1090–1096, 2021.

[2] 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," *Journal of Magnetic Resonance*, vol. 172, no. 2, pp. 296–305, 2005.

[3] R.-B. Wu, H. Ding, D. Dong, and X. Wang, "Learning robust and high-precision quantum controls," *Physical Review A*, vol. 99, no. 4, p. 042327, 2019.

[4] D. Dong, C. Wu, C. Chen, B. Qi, I. R. Petersen, and F. Nori, "Learning robust pulses for generating universal quantum gates," *Scientific Reports*, vol. 6, no. 1, pp. 1–9, 2016.

[5] R. S. Judson and H. Rabitz, "Teaching lasers to control molecules," *Physical Review Letters*, vol. 68, no. 10, p. 1500, 1992.

[6] D. Dong, X. Xing, H. Ma, C. Chen, Z. Liu, and H. Rabitz, "Learning-based quantum robust control: Algorithm, applications, and experiments," *IEEE Transactions on Cybernetics*, vol. 50, pp. 3581–3593, 2020.

[7] E. Zahedinejad, S. Schirmer, and B. C. Sanders, "Evolutionary algorithms for hard quantum control," *Physical Review A*, vol. 90, no. 3, p. 032310, 2014.

[8] H. Ma, C. Chen, and D. Dong, "Differential evolution with equally-mixed strategies for robust control of open quantum systems," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 2015, pp. 2055–2060.

[9] Y. Sun, H. Ma, C. Wu, C. Chen, and D. Dong, "Ensemble control of open quantum systems using differential evolution," in *2015 10th Asian Control Conference (ASCC)*. IEEE, 2015, pp. 1–6.

[10] H. Ma, D. Dong, C.-C. Shu, Z. Zhu, and C. Chen, "Quantum learning control using differential evolution with equally-mixed strategies," *Control Theory and Technology*, vol. 15, no. 3, pp. 226–241, 2017.

[11] D. Dong, C.-C. Shu, J. Chen, X. Xing, H. Ma, Y. Guo, and H. Rabitz, "Learning control of quantum systems using frequency-domain optimization algorithms," *IEEE Transactions on Control Systems Technology*, vol. 29, no. 4, pp. 1791–1798, 2021.

[12] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[13] D. Dong, C. Chen, T.-J. Tarn, A. Pechen, and H. Rabitz, "Incoherent control of quantum systems with wavefunction-controllable subspaces via quantum reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 4, pp. 957–962, 2008.

[14] H. Ma, D. Dong, S. X. Ding, and C. Chen, "Curriculum-based deep reinforcement learning for quantum control," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[15] H. Ma and C. Chen, "Several developments in learning control of quantum systems," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 2020, pp. 4165–4172.

[16] H. Yu, X. Xu, H. Ma, Z. Zhu, and C. Chen, "Control design of two-level quantum systems with reinforcement learning," in *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*. IEEE, 2018, pp. 922–927.

[17] M. Bukov, "Reinforcement learning for autonomous preparation of floquet-engineered states: Inverting the quantum kapitza oscillator," *Physical Review B*, vol. 98, no. 22, p. 224305, 2018.

[18] O. Shindi, Q. Yu, D. Dong, and J. Tang, "A modified Q-learning algorithm for control of two-qubit systems," in *2020 International Conference on Machine Learning and Cybernetics (ICMLC)*. IEEE, 2020, pp. 194–200.

[19] J. Mackeprang, D. B. R. Dasari, and J. Wrachtrup, "A reinforcement learning approach for quantum state engineering," *Quantum Machine Intelligence*, vol. 2, pp. 1–14, 2020.

[20] S. Borah, B. Sarma, M. Kewming, G. J. Milburn, and J. Twamley, "Measurement based feedback quantum control with deep reinforcement learning," *arXiv preprint arXiv:2104.11856*, 2021.

[21] S.-F. Guo, F. Chen, Q. Liu, M. Xue, J.-J. Chen, J.-H. Cao, T.-W. Mao, M. K. Tey, and L. You, "Faster state preparation across quantum phase transition assisted by reinforcement learning," *Physical Review Letters*, vol. 126, no. 6, p. 060401, 2021.

[22] R. Porotti, A. Essig, B. Huard, and F. Marquardt, "Deep reinforcement learning for quantum state preparation with weak nonlinear measurements," *arXiv preprint arXiv:2107.08816*, 2021.

[23] C. Chen, D. Dong, H.-X. Li, J. Chu, and T.-J. Tarn, "Fidelity-based probabilistic Q-learning for control of quantum systems," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 5, pp. 920–933, 2013.

[24] Z. An and D. Zhou, "Deep reinforcement learning for quantum gate control," *EPL (Europhysics Letters)*, vol. 126, no. 6, p. 60002, 2019.

[25] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, "Universal quantum control through deep reinforcement learning," *npj Quantum Information*, vol. 5, no. 1, pp. 1–8, 2019.

[26] O. V. Morzhin and A. N. Pechen, "Krotov method for optimal control of closed quantum systems," *Russian Mathematical Surveys*, vol. 74, no. 5, p. 851, 2019.

[27] 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 Information*, vol. 5, no. 1, pp. 1–7, 2019.

[28] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International Conference on Machine Learning*. PMLR, 2018, pp. 1587–1596.

[29] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.