



**Quantum optimal control of multilevel dissipative quantum systems with reinforcement learning**Zheng An <sup>1,2</sup> Hai-Jing Song,<sup>1,2</sup> Qi-Kai He <sup>3</sup> and D. L. Zhou<sup>1,2,4,5,\*</sup><sup>1</sup>*Institute of Physics, Beijing National Laboratory for Condensed Matter Physics, Chinese Academy of Sciences, Beijing 100190, China*<sup>2</sup>*School of Physical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China*<sup>3</sup>*Hangzhou Tuya Information Technology Co., Ltd., Hangzhou, Zhejiang 310000, China*<sup>4</sup>*Collaborative Innovation Center of Quantum Matter, Beijing 100190, China*<sup>5</sup>*Songshan Lake Materials Laboratory, Dongguan, Guangdong 523808, China*

(Received 28 August 2020; revised 30 November 2020; accepted 21 December 2020; published 11 January 2021)

Manipulation and control of the complex quantum system with high precision are essential for achieving universal fault-tolerant quantum computing. For a physical system with restricted control resources, it is a challenge to control the dynamics of the target system efficiently and precisely under disturbances. Here we propose a multilevel dissipative quantum control framework and show that deep reinforcement learning provides an efficient way to identify the optimal strategies with restricted control parameters of the complex quantum system. This framework can be generalized to be applied to other quantum control models. Compared with the traditional optimal control method, this deep reinforcement learning algorithm can realize efficient and precise control for multilevel quantum systems with different types of disturbances.

DOI: [10.1103/PhysRevA.103.012404](https://doi.org/10.1103/PhysRevA.103.012404)**I. INTRODUCTION**

Precise and complete control of complex quantum systems is the core to achieve quantum computation and quantum information processing. The quantum control (QC) theory provides a powerful tool to achieve high-precision control of quantum dynamics. A QC problem can be phrased as finding strategies of inducing complete transfer of population from an arbitrary initial quantum state to the desired target state. An optimal strategy to get a selected state of a finite-energy-level quantum system is of primary importance for the control of quantum dynamics. The theory for the design of such an optimal strategy has been studied widely, such as in Lyapunov quantum control [1,2], geometric control theory [3], and the Pontryagin maximum principle [4]. Also, robust and optimal strategies of QC are essential for many areas of physical systems from nitrogen-vacancy center experiments [5], to optical systems [6], to superconducting qubits [7]. However, it is hard to get a convincing result with traditional control theory if there are some restricted conditions in the control system. To manipulate more complicated systems, there have been developed several numerical algorithms, like the gradient ascent pulse engineering (GRAPE) algorithm [8,9] and chopped random basis (CRAB) [10,11]. Furthermore, the disturbance of quantum dynamics is the main obstacle in implementing scalable quantum computing [12]. To deal with the spin or qubit decoherence, various strategies have been developed, including quantum error correction [13–16], dynamical decoupling (DD) [17–19], and optimized control in protecting quantum coherence [20–22]. One way to achieve optimal control is to use an arbitrarily slow change of the dynamical

parameters and the adiabatic theorem [23]. However, for a multilevel system, these require several resources that also increase exponentially with the size of the system. On the other hand, when application to a typical realistic condition of an open quantum system is considered, there are few analytical or ansatz solutions available. To simplify those constraints, in this paper we introduce a switch-on-off control problem with dissipative dynamics. In particular, we discuss the dynamics that are affected by dephasing and energy decay. These two effects exist, to different degrees, in any practical attempt to implement quantum control tasks in real physical systems [24–27]. Those disturbance effects emerge from the interaction of the system with the surrounding environment [28].

Quantum control theory has been recently applied with success to the optimization of the dynamics of simple systems [5,29–31] and quantum many-body systems [10,11,32,33]. With the progress of quantum control techniques and computer science, the numerical algorithm gives us a robust and efficient way to implement high-fidelity quantum control. Among various control algorithms, reinforcement learning (RL) has been attracting much focus. Reinforcement learning has demonstrated remarkable abilities in board games [34–36] and video games [37–39]. Recently it has also been widely applied to a wide array of physics problems, such as quantum state preparation [32,40], quantum gate control [41,42], quantum error correction [43], and quantum metrology [44]. Those successes naturally raise the question of how much quantum control might benefit from the application of reinforcement learning.

In this paper, we study a general quantum control model of a finite-level system under disturbances. To explore the optimal strategy of the control problem in this scenario, we use the distributed proximal policy optimization (DPPO) algorithm [45,46] to study this problem in this paper. The proximal

\*zhoudl72@iphy.ac.cn

policy optimization (PPO) algorithm has been successfully used in robotics [47] and aircraft control [48]. Recently, it has been applied in QC problems [49,50].

The rest of this paper is structured as follows. In Sec. II, we briefly introduce the basic description of our quantum control model. In Sec. III, we present the actor-critic model of reinforcement learning and the DPPO algorithm used in our paper. In Sec. IV, we present the methodology of our method, the architecture of the neural network for our agent, the interactive interface, as well as numerical results of tested examples. Finally, in Sec. V, we draw our conclusions.

## II. MODEL

We study a quantum system with a finite number of distinct energy levels driven by a time-dependent external field whose Hamiltonian reads

$$H = H_0 + V \quad (1)$$

with

$$H_0 = \sum_{i=1}^n E_i |i\rangle\langle i|, \quad (2)$$

$$V(t) = \sum_{i=1}^{n-1} \gamma(t) (|i\rangle\langle i+1| + |i+1\rangle\langle i|), \quad (3)$$

where  $H_0$  is called the drift Hamiltonian and  $V(t)$  is called the control Hamiltonian in quantum control theory. The state  $|i\rangle$  is the  $i$ th eigenstate of  $H_0$  with eigenenergy  $E_i$ ,  $n$  is the number of the energy levels, and the time-dependent real parameter  $\gamma(t)$  is the coupling strength between  $|i\rangle$  and  $|i+1\rangle$  for  $1 \leq i \leq n-1$ . Without loss of generality, we assume that  $E_1 \leq E_2 \leq \dots \leq E_n$ . In particular, we assume  $H_0$  is regular, where the energy levels  $E_i = i$  ( $i = 1, \dots, n$ ). However, a different distribution of eigenenergies may affect the performance of control algorithms. So we present the effect of the different distribution of eigenenergies on two examples in Appendix D.

When our system weakly interacts with its environment, its dynamics is described by the master equation of the Lindblad type:

$$\dot{\rho} = -\frac{i}{\hbar} [H, \rho] + \sum_k \Gamma_{k,n} \left( A_{k,n} \rho A_{k,n}^\dagger - \frac{1}{2} \{A_{k,n}^\dagger A_{k,n}, \rho\} \right), \quad (4)$$

where  $A_{k,n}$  is the Lindblad operator associated with some dissipative process with a decay rate  $\Gamma_{k,n}$  for each  $k$ , and the subscript  $n$  labels the type of dissipative process.  $\{A, B\} = AB + BA$  denotes the anticommutator. Here we consider two typical dissipative processes. One is the dephasing process, whose Lindblad operator  $A_{k,d} = |k\rangle\langle k|$  with an identical dephasing rate  $\Gamma_{k,d} = \Gamma_d$  for  $1 \leq k \leq n$ . The other is the energy decay process, whose Lindblad operator  $A_{k,l} = |1\rangle\langle k|$  with an identical energy decay rate  $\Gamma_{k,l} = \Gamma_l$  for  $2 \leq k \leq n$ .

Our central task can be stated as follows. Initially, our system is prepared in the ground state  $|1\rangle$  of  $H_0$ . By controlling the time dependence of the parameter  $\gamma(t)$ , we aim to maximize the probability for our system to be in the highest excited state  $n$  of  $H_0$  at a fixed time  $T$ .

For simplicity, we adopt the bang-bang control protocol. We divide the total control time  $T$  into  $N$  periods with the same duration  $\delta t = T/N$ . In the  $i$ th period with  $(i-1)\delta t \leq t \leq i\delta t$  ( $1 \leq i \leq N$ ), the coupling is either switched on or switched off, i.e.,  $\gamma(t) = a_i \gamma$  with  $a_i \in \{0, 1\}$ . Then a control strategy is specified by a series of binary numbers  $\{a_1, a_2, \dots, a_N\}$ . We aim to find out an optimal strategy to maximize the fidelity

$$\mathcal{F}(\rho(T), |n\rangle\langle n|) = \langle n | \rho(T) | n \rangle. \quad (5)$$

It is worth pointing out that, since the size of the set of the strategy space is  $2^N$ , it is impossible to get the optimal strategy by exhaustively searching in the strategy space for a large  $N$ .

Note that we focus on a regime where  $\gamma$  is much smaller than the energy gap  $E_n - E_1$ , which implies that the probability of arriving at the state  $|n\rangle$  at any time is very small with the coupling always on. However, the optimal strategy to improve the probability of arriving at the highest-energy eigenstate of  $H_0$  with switching on and off the coupling  $V$  can be understood as follows. First, we switch on the coupling  $V$  for a short period from a lower-energy eigenstate to a higher-energy eigenstate, then we switch off the coupling  $V$  to avoid the effect of  $|i\rangle\langle i+1|$ . Furthermore, when the coupling  $V$  is switched off, the free Hamiltonian  $H_0$  changes the state of the system while keeping the energy invariant. Thus the energy of the system can be increased by suitable arrangements of switching the coupling on and off.

In fact, we study the cases where the dimension of the Hilbert space is 4, 6, 8, and 10 while we do not increase the number of the control parameters, which brings a great challenge to get an optimal strategy to arrive at the highest eigenenergy state by a sequence of jumps  $|1\rangle \leftrightarrow |2\rangle \leftrightarrow \dots \leftrightarrow |n\rangle$ .

## III. REINFORCEMENT LEARNING: ACTOR-CRITIC MODEL

To find out the optimal strategy in our multilevel quantum control problem, we adopt a modern reinforcement learning method called the actor-critic model. In this section, we give a short review of the actor-critic reinforcement learning model.

In traditional reinforcement learning, there are two different types of methods to implement artificial intelligence. One includes the value-based methods (such as Q learning [51]), where the agent learns the value function that maps each state-action pair to a value. According to the value function, the agent will take the action with the largest return value for each state. It works well when the set of actions is finite. The other type includes the policy-based methods (such as policy gradients [52]), where we directly optimize the policy without using a value function. It is efficient when the action space is continuous or stochastic.

The reinforcement learning process is a finite Markov decision process [52]. As shown in Fig. 1, a state  $S_t$  at time  $t$  is transmitted into a new state  $S_{t+1}$  together with giving a scalar reward  $R_{t+1}$  at time  $t+1$  by the action  $A_t$  with the transmission probability  $p(S_{t+1}, R_{t+1} | S_t, A_t)$ .

For a finite Markov decision process, the sets of the states, the actions, and the rewards are finite. In the value-based methods, the goal is to maximize the total discounted return

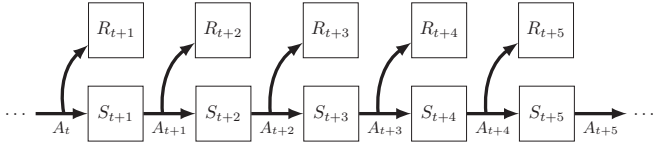


FIG. 1. A schematic diagram of a Markov decision process.

at time  $t$ :

$$G_t = \sum_{k=0}^{\infty} \Gamma^k R_{t+k+1}, \quad (6)$$

where  $\Gamma$  is the discount rate and  $0 \leq \Gamma \leq 1$ . The policy  $\pi$  is defined by the conditional probability  $\pi(a|s)$  of selecting an action  $a$  for each state  $s$ . To estimate how good a policy  $\pi$  is, two value functions are introduced:

$$q_{\pi}(s, a) \doteq \mathbb{E}_{\pi}[G_t | S_t = s, A_t = a], \quad (7)$$

$$v_{\pi}(s) \doteq \mathbb{E}_{\pi}[G_t | S_t = s], \quad (8)$$

where  $q_{\pi}(s, a)$  is called the state-action value function and  $v_{\pi}(s)$  is called the state value function;  $E_{\pi}$  denotes the probability expectation for all the actions in the process taken following the policy  $\pi$ . Note that we have the following relations:

$$q_{\pi}(s, a) = \sum_R R p(R|s, a) + \Gamma \sum_{s'} v_{\pi}(s') p(s'|s, a), \quad (9)$$

$$v_{\pi}(s) = \sum_{R,a} R p(R|s, a) \pi(a|s) + \Gamma \sum_{s',a} v_{\pi}(s') p(s'|s, a) \pi(a|s). \quad (10)$$

In addition, the advantage function is defined as  $A_{\pi}(s, a) = q_{\pi}(s, a) - v_{\pi}(s)$ , which measures the advantage of an action  $a$  with respect to the state  $s$  under the policy  $\pi$ .

In the policy gradient scheme, the objective is to maximize the cumulant reward under a parametrized policy  $\pi_{\theta}$ :

$$J(\pi_{\theta}) = \mathbb{E}_{\pi_{\theta}} \left[ \sum_{t=0}^{\infty} \Gamma^t R(s_t) \right]. \quad (11)$$

The model-free policy gradient of the cumulant reward is given by [53]

$$\nabla_{\theta} J(\pi_{\theta}) \propto \sum_s \mu(s) \sum_a A_{\pi_{\theta}}(s, a) \nabla_{\theta} \pi_{\theta}(a|s), \quad (12)$$

where  $\mu(s)$  is the probability of state  $s$  appearing in the Markov process under the policy  $\pi$ . The above gradient can be estimated by the score function estimator [54].

In this paper, we use a hybrid type of reinforcement learning method, called the actor-critic, whose protocol is shown in Fig. 2. The agent has two parts: a critic that measures how good the action taken is and an actor that controls how our agent behaves. The actor builds a network to evaluate the policy  $\pi_{\theta}$ , and takes an action for the current state of the environment following the policy  $\pi_{\theta}$ . The critic builds a network to evaluate the state value function  $v_{\phi}(s)$ , which is used to approximate  $A_{\pi_{\theta}}(s, a)$  in Eq. (12). The critic improves the value network according to the reward from the environment, and the actor improves the policy network according to

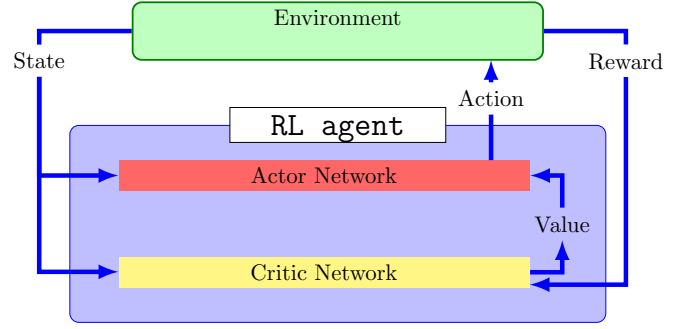


FIG. 2. A schematic diagram of the actor-critic model: at each time step of training, the actor network of the agent proposes a control action of  $A_t$ , and the environment takes the proposed action and evaluates the quantum state for time duration  $\delta t$  to obtain the reward, both of which are fed into the RL agent. The critic network of the agent receives the reward and estimates the action's value based on the state.

a modified version of Eq. (12):

$$\nabla_{\theta} J(\pi_{\theta}) \propto \sum_s \mu(s) \sum_a A_{\phi}(s, a) \nabla_{\theta} \pi_{\theta}(a|s). \quad (13)$$

In the actor-critic model, we get the advantage by building a network, which is more efficient than by directly simulating following the policy  $\pi_{\theta}$ . Besides, it improves the convergence significantly to use the advantage function to replace the state-action value function in evaluating the policy gradient [55].

In this work, we use the DPPO algorithm [46] to learn an optimal policy under the policy gradient framework. The loss function of DPPO reads

$$L(\theta, \phi) = \hat{\mathbb{E}}_{\pi_{\theta_{\text{old}}}} \left[ \min(r_{\theta_{\text{old}}}(a|s_0, \theta) A_{\phi}(s_0, a), \right. \\ \left. \times \text{clip}(r_{\theta_{\text{old}}}(a|s_0, \theta), 1 - \epsilon, 1 + \epsilon) A_{\phi}(s_0, a) \right], \quad (14)$$

where  $\epsilon$  is a hyperparameter ( $\epsilon = 0.2$  in this paper). The expectation  $\hat{\mathbb{E}}_{\pi_{\theta_{\text{old}}}}$  indicates the empirical average over a finite batch of samples under the policy  $\pi_{\theta_{\text{old}}}$ . The term  $r_{\theta_{\text{old}}}(a|s, \theta)$  is defined as the ratio of likelihoods:

$$r_{\theta_{\text{old}}}(a|s, \theta) = \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{\text{old}}}(a|s)}. \quad (15)$$

The clip function for  $c \leq d$  is defined as

$$\text{clip}(f(x), c, d) = \begin{cases} d & \text{if } f(x) > d \\ f(x) & \text{if } c \leq f(x) \leq d \\ c & \text{if } f(x) < c. \end{cases} \quad (16)$$

The clip function for  $r_{\theta_{\text{old}}}(a|s, \theta)$  penalizes large changes between nearest updates, which corresponds to the trust region of the first-order policy gradient. Based on the first-order trust region search gradient descent, DPPO has a robust learning process and can handle both discrete and continuous action spaces. A detailed description of the DPPO can be found in the Appendix C.

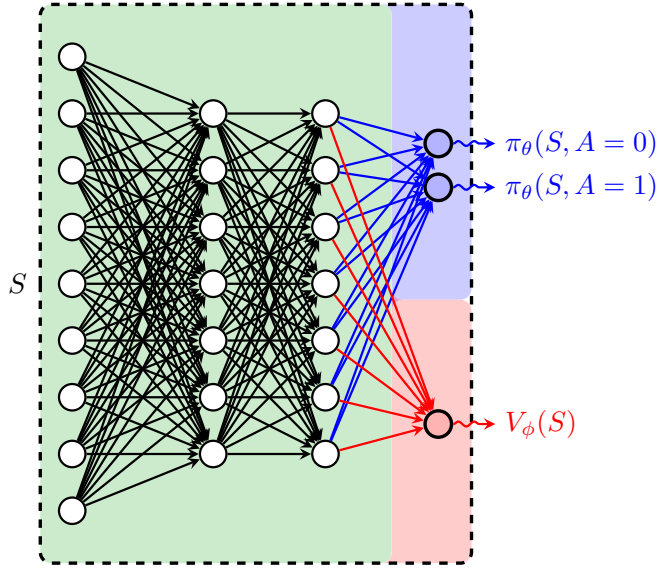


FIG. 3. The architecture of the actor-critic neural network for the agent. The actor and the critic share the same architecture of hidden layers (green). The actor network has an action head (blue) to output the possible policy. The critic network has a value head (red) to output the value of the given state.

#### IV. QUANTUM STATE CONTROL WITH ACTOR-CRITIC LEARNING

##### A. Agent-environment interface

To implement the RL agent for our problem, we propose an interactive interface between the RL agent and the physical environment (Fig. 2) adapted to OpenAI Gym [56]. We have used TENSORFLOW [57] and BASELINES [58] to implement the learning algorithms with QUTIP [59,60] simulating the dynamics of our control problem. The architecture of deep neural network in our RL agent is shown in Fig. 3. In our quantum control problem, the state at time  $t$  in the reinforcement learning is the state  $\rho(t)$ , which is expressed by its components:

$$s_t = \{\text{Re}(\rho_{11}(t)), \text{Im}(\rho_{11}(t)), \\ \text{Re}(\rho_{12}(t)), \text{Im}(\rho_{12}(t)), \dots, \\ \text{Re}(\rho_{nn}(t)), \text{Im}(\rho_{nn}(t))\}, \quad (17)$$

where  $\text{Re}(\rho_{ij}(t))$  and  $\text{Im}(\rho_{ij}(t))$  are the real and the imaginary parts of the component  $\rho_{ij}(t)$ , respectively. Our action space is formed by a switchable control field  $a_t \in \{0, 1\}$ , which steers our quantum state  $\rho(t)$  to  $\rho(t + \delta t)$  according to Eq. (4). After evaluating the new state  $\rho(t + \delta t)$  the agent obtains the single step reward

$$R_{t+1} = \mathcal{F}(\rho(t + \delta t), |n\rangle\langle n|) - \mathcal{F}(\rho(t), |n\rangle\langle n|), \quad (18)$$

where  $\mathcal{F}$  is the fidelity defined by Eq. (5).

##### B. Numerical results

We now apply the actor-critic RL approach to our quantum state control problem with different settings, illustrating the flexibility and efficiency of our RL agent. Here the different settings include different numbers of energy levels for our system, and different types of environments affecting our system.

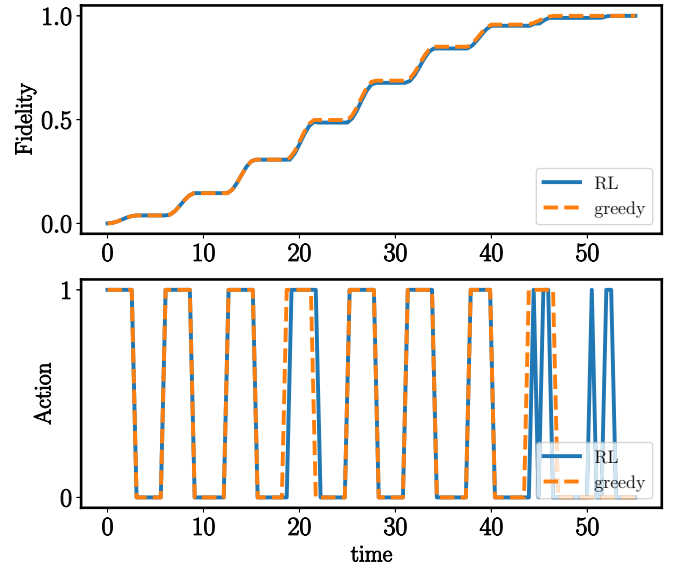


FIG. 4. The best fidelities (up) and strategy (down) of preparing an excited state for the two-level control model with  $\gamma = 0.1$ . The markers correspond to the algorithms RL (blue line) and greedy (orange line). The time step is  $N = 110$ .

We give the numerical results of the best fidelity  $\mathcal{F}$  in our quantum state control problem from the deep reinforcement learning. To show the effectiveness of our deep RL method, we also calculate the fidelity with the greedy method and the GRAPE algorithm. The greedy algorithm is used for finding successful policies by performing local searches. The GRAPE method looks at the direct gradient of the fidelity function. In particular, to get better results, the GRAPE algorithm allows for the coupling strength  $\gamma(t)$  to take any value in the interval  $[0, \gamma]$ . We then present our analysis of the performance of our deep RL algorithm against the two algorithms. Details of the greedy algorithm can be found in the Appendixes.

##### 1. Quantum state control without environments

In this section, we consider our quantum state control problem with a quantum system with negligible environments. In other words, we assume that all the coefficients  $\Gamma_{k,n} = 0$ .

In Fig. 4 we show the results of the optimal fidelity and the corresponding strategy on our quantum state control problem with parameters  $\{n = 2, \gamma = 0.1, T = 55, N = 110\}$  in Fig. 4. With 1500 episodes, our RL agent gets the optimal fidelity  $\mathcal{F}_{RL}(T) \approx 0.999998$ , which is a little larger than the fidelity  $\mathcal{F}_{Greedy}(T) \approx 0.999815$  from the direct greedy algorithm. While the difference of the fidelities between those two methods is very small, the strategy in Fig. 4 is different for about  $T > 45$ , which shows that our RL agent has learned a globally optimized protocol in this task. Notice that for all control tasks discussed in our paper, the time scale  $\delta t$  is always 0.5. A detailed optimal strategy of the greedy method can be found in Appendix B.

We further apply our RL agent to the quantum state control problem in the multilevel Hilbert space. We give the optimal fidelities in the cases with the dimension of Hilbert space equal to 4, 6, 8, and 10 by the RL algorithm (red

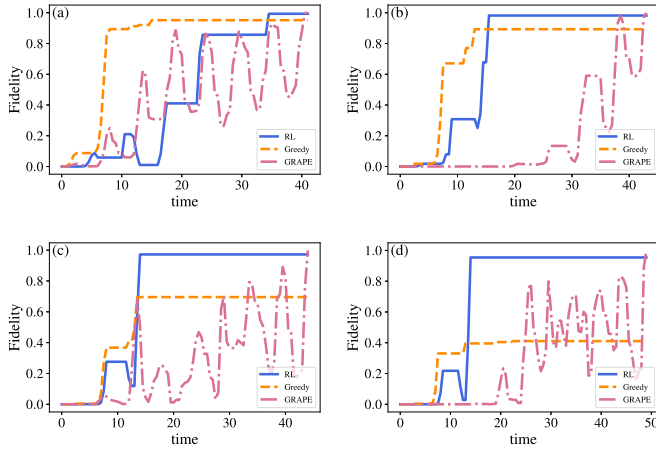


FIG. 5. Results from the three algorithms for different level control models. The horizontal and vertical axes of each panel denote evolution time  $t$  and fidelity  $\mathcal{F}$ . The fidelities for three different methods with (a) 4, (b) 6, (c) 8, and (d) 10 level control models. The corresponding coupling strengths with the different models are  $\gamma = 0.8, 1.1, 1.4, \text{ and } 1.9$ . The time steps with different control tasks are  $N = 82, 86, 88, \text{ and } 98$ .

dashed line), the greedy algorithm (blue solid line), and the GRAPE algorithm (violet dot-dashed line), which are shown in Figs. 5(a)–5(d). We find that the greedy algorithm becomes less effective with the increase of the dimension of the Hilbert space, but the RL algorithm and the GRAPE algorithm perform well in all cases. For example, when the dimension of the Hilbert space varies from 4 to 10, the optimal fidelity from the greedy algorithm varies from about 0.954 to about 0.411, but the fidelity from the RL algorithm varies from about 0.993 to 0.954. While the GRAPE algorithm has the best performance out of the three methods, the algorithm requires the fidelity gradients at all times.

## 2. Quantum state control with environments

We now turn our attention to the behavior of our learning strategy when applied to a nonideal scenario in which typical realistic conditions are considered. In particular, we discuss the results produced by the RL agent when the system is affected by dephasing and energy decay.

In Fig. 6 we present our numerical results on the control problem under dephasing dynamics. Figures 6(a)–6(d) show the results for dephasing rate  $\sqrt{\Gamma_d} = \{0.05, 0.06, 0.07, 0.08, 0.09, 0.1\}$ . In both cases, our best results from the RL agent outperform the greedy algorithm and even GRAPE. Also, with the energy-level number getting higher, the differences of fidelities between the three methods get larger.

Figure 7 shows the superior performance of the RL agent versus the greedy and GRAPE algorithms during the time evolution under the disturbance of energy decay. Similar to the dephasing cases, the RL agent has successfully conquered the control problem under energy decay dynamics. However, for the greedy algorithm, it is impossible to get a convincing result with a large energy decay rate in high-dimensional control problems. In this scenario, we find that the RL agent

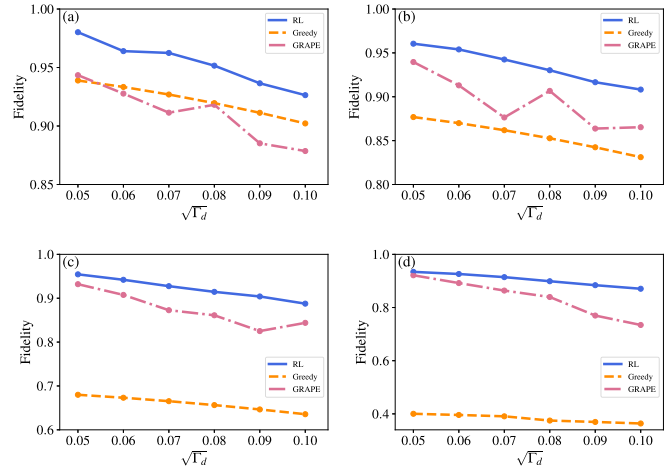


FIG. 6. Results from the three algorithms for different level control models under dephasing dynamics. The horizontal and vertical axes of each panel denote dephasing rate  $\sqrt{\Gamma_d}$  and fidelity  $\mathcal{F}$ . Best fidelity for three different methods of (a) 4, (b) 6, (c) 8, and (d) 10 level control models. Note that the Hamiltonian is the same as shown in Fig. 5.

successfully learns to adapt to overcome the disturbance of energy decay in multilevel control problems.

To further understand the results shown in Figs. 6 and 7, we take examples from  $\sqrt{\Gamma_d}, \sqrt{\Gamma_l} = 0.1$  and plot the corresponding trajectories of the fidelity in Figs. 8 and 9. We realized that the RL agent yields different policies according to the types of environments: one only has to learn how to quickly control the state to the target and decide whether to place the control sequence at the beginning or the end of the control. As shown in Fig. 8, the best strategy is to quickly drive the initial state to the final state at the start of the control, since the environment cannot change the energy of the system. While in Fig. 9 the strategy becomes opposed as previously

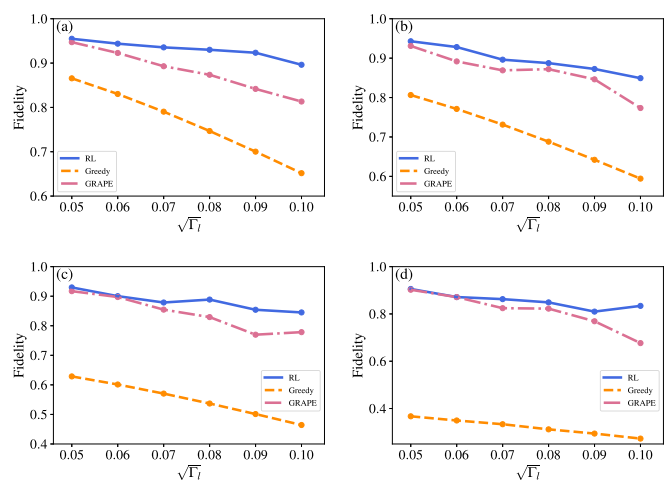


FIG. 7. Results from the three algorithms for different level control models under energy decay dynamics. The horizontal and vertical axes of each panel denote energy decay rate  $\sqrt{\Gamma_l}$  and fidelity  $\mathcal{F}$ . Best fidelity for three different methods of (a) 4, (b) 6, (c) 8, and (d) 10 level control models. Note that the Hamiltonian is the same as shown in Fig. 5.

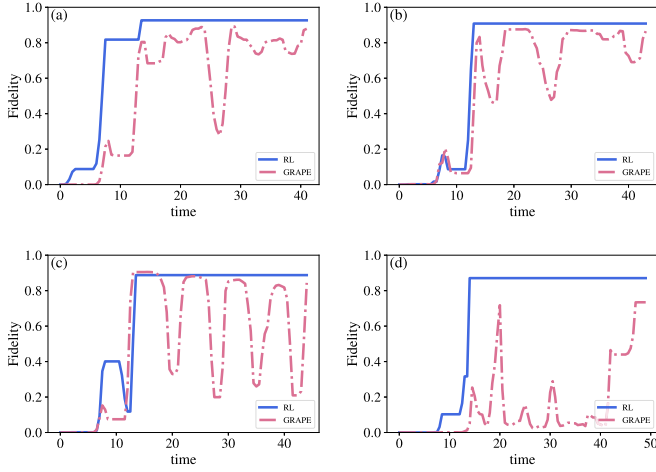


FIG. 8. Results from RL agent and GRAPE strategies for different high-level control models with  $\sqrt{\Gamma_d} = 0.1$ . The horizontal and vertical axes of each panel denote evolution time  $t$  and fidelity  $\mathcal{F}$ . The evolution of fidelity with the RL agent and GRAPE control of (a) 4, (b) 6, (c) 8, and (d) 10 level control models.

shown, the agent learns to avoid a complex control strategy to maintain the target state but to get at the end of the control, because the energy of the system is decaying. The trajectory of GRAPE shows there indeed are many local minima in the control landscape. However, the RL agent can use those local minima to find optimal strategies.

## V. CONCLUSION

We propose a quantum control framework for multilevel dissipative quantum control optimization. The RL method is capable of finding the control protocol that has high-fidelity of a finite-dimensional quantum control problem under disturbances and is superior to the traditional greedy method and GRAPE algorithm. Moreover, the RL method can accom-

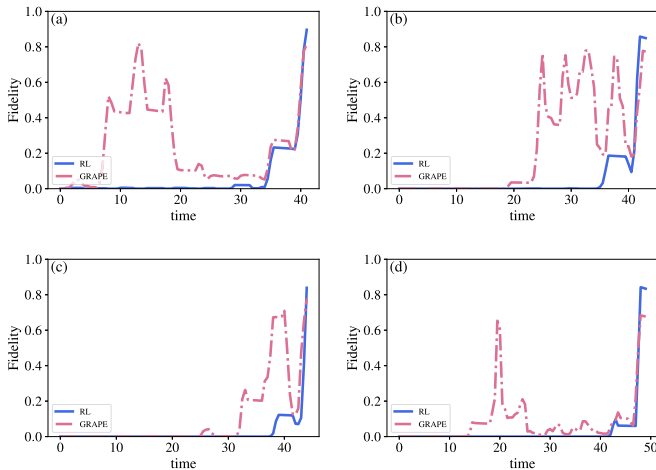


FIG. 9. Results from RL agent and GRAPE strategies for different high-level control models with  $\sqrt{\Gamma_l} = 0.1$ . The horizontal and vertical axes of each panel denote evolution time  $t$  and fidelity  $\mathcal{F}$ . The evolution of fidelity with RL agent and GRAPE control of (a) 4, (b) 6, (c) 8, and (d) 10 level control models.

modate switch-on-off pulse shapes, which would be hard for traditional gradient methods.

Although the control problems dealt with the different dynamics optimization tasks, the RL agent can find high-fidelity solutions with a single set of algorithmic hyperparameters. This suggests that learning the control landscape can be performed with minimal expert knowledge about the physical problem.

Our results, therefore, suggest that the RL-based methods can be powerful alternatives to commonly used algorithms, capable of finding control protocols that could be more efficient in practical complex quantum control problems. Also, the RL agent can be used to control experimental quantum devices. The present approach is flexible enough to be applied to different physical systems, such as qubit-cavity systems, weak measurements, and quantum error correction. We expect that our work would extend the deep learning techniques to deal with more practical quantum control problems in the near future.

## ACKNOWLEDGMENTS

This work is supported by NSF of China (Grants No. 11775300 and No. 12075310), the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDB28000000), and the National Key Research and Development Program of China (Grant No. 2016YFA0300603).

## APPENDIX A: OPTIMAL LYAPUNOV QUANTUM CONTROL (GREEDY) METHOD

As the first trial, we consider a greedy way to get the optimal strategy. Greedy algorithms are used for finding successful policies because the algorithms are fast in converging on successful solutions when performing local searches. To describe the greedy method more intuitively, we use the optimal Lyapunov quantum control theory [1,2,61] to analyze the relationship between the strength of the control field and the control fidelity.

In Lyapunov quantum control, the control field is determined by a Lyapunov function  $f$ , which will decrease with time. The evolution of the control protocol is determined by Eq. (4). Furthermore, we assume the system satisfies the requirement for a Lyapunov function,  $f \geq 0$  [62]. The Lyapunov function can be defined as

$$f = \text{Tr}(|n\rangle\langle n|\rho). \quad (\text{A1})$$

The time derivative of the Lyapunov function is given by (with  $[H_0, |n\rangle\langle n|] = 0$ )

$$\begin{aligned} \dot{f} &= \text{Tr}\left(|n\rangle\langle n|\left(-\frac{i}{\hbar}[H_0 + V, \rho] + \mathcal{L}(\rho)\right)\right) \\ &= \text{Tr}(\mathcal{L}(\rho)|n\rangle\langle n|) - \frac{i}{\hbar}\text{Tr}(\rho[|n\rangle\langle n|, \gamma(t)H_c]), \end{aligned} \quad (\text{A2})$$

where  $\mathcal{L}(\rho) = \sum_k \Gamma_k (A_k \rho A_k^\dagger - \frac{1}{2}\{A_k^\dagger A_k, \rho\})$  and  $H_c = \sum_{i=1}^{n-1} (|i\rangle\langle i+1| + |i+1\rangle\langle i|)$ . It is clear that  $\dot{f} \leq 0$ , which ensures the decreasing of the Lyapunov function. So the

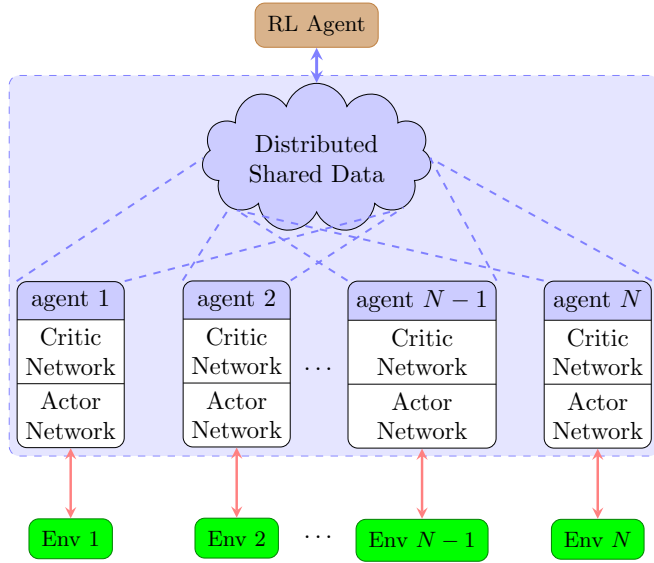


FIG. 10. Schematics of the DPPO algorithm. Data collection and gradient calculation are distributed over workers, labeled as “agent  $i$ .” Then the weights of the RL agent update synchronously. The environments are labeled as “Env  $i$ .”

control function  $\gamma(t)$  satisfies

$$\text{Tr}(\mathcal{L}(\rho)|n\rangle\langle n|) \leq \gamma(t) \frac{i}{\hbar} \text{Tr}(\rho[|n\rangle\langle n|, H_c]). \quad (\text{A3})$$

Let

$$\begin{aligned} C &= \text{Tr}(\mathcal{L}(\rho)|n\rangle\langle n|), \\ D &= \frac{i}{\hbar} \text{Tr}(\rho[|n\rangle\langle n|, H_c]). \end{aligned} \quad (\text{A4})$$

In our problem, the control function  $\gamma(t)$  always switches between two values, so the mathematical expressions of the control fields are as follows:

$$\gamma(t) = \begin{cases} \gamma & \text{if } D \geq 0, C > 0 \\ 0 & \text{if } D \geq 0, C \leq 0 \\ 0 & \text{if } D < 0, C > 0 \\ \gamma & \text{if } D < 0, C \leq 0. \end{cases} \quad (\text{A5})$$

## APPENDIX B: TWO-LEVEL CASE WITHOUT DISSIPATIVE DYNAMICS

Consider a two-level system governed by the following Hamiltonian:

$$H = -\frac{\omega}{2}\sigma_z + \gamma\sigma_x, \quad (\text{B1})$$

where we set  $\hbar = 1$ .  $\omega$  is the level spacing of the system, and  $\gamma = \gamma(t)$  denotes the control field. Assume that the aim is to steer the system from an arbitrary state  $|\psi_0\rangle = \cos(\frac{\gamma_0}{2})|0\rangle + e^{i\phi} \sin(\frac{\gamma_0}{2})|1\rangle$  to state  $|1\rangle$  (target state), where  $|1\rangle$  is the excited

state of the system, and  $|0\rangle$  is the excited state. Define a positive operator

$$P_e = \mathbf{I} - |0\rangle\langle 0| = |1\rangle\langle 1|. \quad (\text{B2})$$

The Lyapunov function can be written as

$$f_e = \text{Tr}(P_e\rho) \quad (\text{B3})$$

with

$$\rho = |\psi\rangle\langle\psi|, \quad |\psi\rangle = a(t)|0\rangle + b(t)|1\rangle. \quad (\text{B4})$$

The Lyapunov function  $f_e$  represents the overlapping between the function  $\mathbf{I} - |0\rangle\langle 0|$  of target state  $|1\rangle\langle 1|$  and the actual state of the system. The time derivative of the Lyapunov function can be calculated as follows [with the abbreviations  $a = a(t)$ ,  $b = b(t)$ ]:

$$\begin{aligned} \dot{f}_e &= \text{Tr}(P_e\dot{\rho}) = \text{Tr}\left(-iP_e\left[-\frac{\omega}{2}\sigma_z + \gamma\sigma_x, \rho\right]\right) \\ &= \text{Tr}\left(-iP_e\left[-\frac{\omega}{2}\sigma_z, \rho\right]\right) + \text{Tr}\left(-iP_e[\gamma\sigma_x, \rho]\right) \\ &= 2\gamma\text{Im}(-ab^*). \end{aligned} \quad (\text{B5})$$

If  $f_e \leq 0$  for all times,  $f_e$  would monotonically decrease with time under the control; meanwhile, the system is asymptotically steered into the target state  $|1\rangle$ . Using the method of the greedy algorithm, the control field  $\gamma(t)$  takes values

$$\gamma(t) = \begin{cases} \gamma & \text{Im}(-ab^*) < 0 \\ 0 & \text{Im}(-ab^*) \geq 0. \end{cases} \quad (\text{B6})$$

With the optimal Lyapunov control, the time evolution of the two-level system can be analytically calculated. In a basis spanned by  $\{|0\rangle, |1\rangle\}$ , the total Hamiltonian can be expressed as

$$H = \sqrt{\frac{\omega^2}{4} + \gamma^2} \begin{pmatrix} -\cos(\theta) & \sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix} \quad (\text{B7})$$

with  $\theta$  defined by

$$\tan \theta = \frac{2\gamma}{\omega}.$$

The eigenvalues of the Hamiltonian  $H$  are

$$E_{\pm} = \pm \sqrt{\frac{\omega^2}{4} + \gamma^2} \quad (\text{B8})$$

and the corresponding eigenvectors are given by

$$|E_+\rangle = -\cos\frac{\theta}{2}|0\rangle + \sin\frac{\theta}{2}|1\rangle,$$

$$|E_-\rangle = \sin\frac{\theta}{2}|0\rangle + \cos\frac{\theta}{2}|1\rangle.$$

The time-evolution operator can be calculated to be

$$U = \exp(-iHt) = \begin{pmatrix} e^{-iE_-t} \cos^2 \frac{\theta}{2} + e^{-iE_+t} \sin^2 \frac{\theta}{2} & \frac{1}{2}(e^{-iE_+t} - e^{-iE_-t}) \sin \theta \\ \frac{1}{2}(e^{-iE_+t} - e^{-iE_-t}) \sin \theta & e^{-iE_-t} \sin^2 \frac{\theta}{2} + e^{-iE_+t} \cos^2 \frac{\theta}{2} \end{pmatrix}. \quad (\text{B9})$$

In the absence of a control field [i.e.,  $\gamma(t) = 0$ ], we have  $\theta = 0$ . The time evolution operator reduces to a diagonal form,

$$U = \begin{pmatrix} e^{\frac{i\omega t}{2}} & 0 \\ 0 & e^{-\frac{i\omega t}{2}} \end{pmatrix}.$$

Assume that the initial state of a two-level system is

$$|\psi_0\rangle = \cos\left(\frac{\gamma_0}{2}\right)|0\rangle + e^{i\phi} \sin\left(\frac{\gamma_0}{2}\right)|1\rangle = a_0|0\rangle + b_0|1\rangle.$$

With different parameters  $\gamma_0$  and  $\psi$ ,  $|\psi_0\rangle$  can represent an arbitrary pure state. Let the target state  $|1\rangle$  correspond to the south pole on the Bloch sphere. Since  $\text{Im}(-a_0b_0^*) = -\frac{\sin\phi \sin\gamma_0}{2}$ , the first control field is calculated as

$$\gamma(t) = \begin{cases} \gamma & (\text{Im}(-ab^*) < 0), (0 < \theta < \pi) \\ 0 & (\text{Im}(-ab^*) \geq 0), (\pi \leq \theta < 2\pi, \theta = 0). \end{cases} \quad (\text{B10})$$

Assume that this control would last until time  $\tau$ ; i.e., the duration of this control is  $\tau$ . With this control, the state evolves to

$$\begin{aligned} |\psi_\tau\rangle &= \left[ \left( e^{-iE_-t} \cos^2 \frac{\theta}{2} + e^{-iE_+t} \sin^2 \frac{\theta}{2} \right) \cos \frac{\gamma_0}{2} \right. \\ &\quad \left. + \frac{1}{2} (e^{-iE_+t} - e^{-iE_-t}) \sin \theta e^{i\phi} \sin \frac{\gamma_0}{2} \right] |0\rangle \\ &\quad + \left[ \frac{1}{2} (e^{-iE_+t} - e^{-iE_-t}) \sin \theta \cos \frac{\gamma_0}{2} \right. \\ &\quad \left. + \left( e^{-iE_-t} \sin^2 \frac{\theta}{2} + e^{-iE_+t} \cos^2 \frac{\theta}{2} \right) e^{i\phi} \sin \frac{\gamma_0}{2} \right] |1\rangle \\ &\equiv a_\tau |0\rangle + b_\tau |1\rangle. \end{aligned} \quad (\text{B11})$$

From the design of the control law, we find that a control field would last until  $\text{Im}(-a_\tau b_\tau a u^*)$  changes sign. Then  $\tau$  can be given by solving  $\text{Im}(-a_\tau b_\tau a u^*) = 0$ . Meanwhile, the sign of  $\text{Im}(-a_\tau b_\tau a u^*)$  determines the next control field. Simple algebra shows that

$$\begin{aligned} \text{Im}(-a_\tau b_\tau^*) &= \frac{1}{2} (\sin(2E_- \tau) (\cos \theta \sin \gamma_0 \cos \phi \\ &\quad + \sin \theta \cos \gamma_0) + \sin \gamma_0 \sin \phi \cos(2E_- \tau)). \end{aligned} \quad (\text{B12})$$

### APPENDIX C: DISTRIBUTED PROXIMAL POLICY OPTIMIZATION

The actor-critic algorithm combines the advantages of policy-based and value-based methods, while the PPO algorithm [45,46] based on the actor-critic algorithm aims to optimize policy update. The central idea of proximal policy optimization is to avoid having too large a policy update which is proposed by trust region policy optimization (TRPO) [63]. The underlying idea of such improvements thereby is limiting the magnitude of updates to  $\theta$  by imposing constraints on the difference between  $\pi_{\theta_{\text{old}}}$  and  $\pi_\theta$  in order to prevent catastrophic jumps out of optima and achieve a better convergence behavior.

One main novelty hereby lies in the introduced loss of DPPO:

$$L^{CLIP}(\theta) = \mathbb{E}_t [\min(r_t(\theta)A_t(s, a), \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t(s, a))], \quad (\text{C1})$$

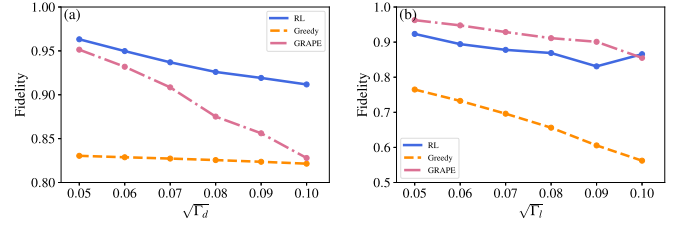


FIG. 11. Results from the three algorithms for four-level control model with the Hamiltonian of case (i) under disturbances. The horizontal and vertical axes of each panel denote noise rate  $\sqrt{\Gamma_{k,n}}$  and fidelity  $\mathcal{F}$ . (a) Dephasing dynamics and (b) energy decay dynamics.

where  $\mathbb{E}_t$  and  $A_t(s, a)$  are the expectation over time steps and the advantage at time  $t$ , respectively. If  $r_t(\theta) > 1$ , the action is more probable in the current policy than the old policy; if  $r_t(\theta) < 1$ , the action is less probable for the current policy than for the old one.

As consequence, a new objective function from Eq. (11) could be

$$L^{CPI}(\theta) = \hat{\mathbb{E}}_t \left[ \frac{\pi_\theta(a|s)}{\pi_{\theta_{\text{old}}}(a|s)} A_t(s, a) \right] = \hat{\mathbb{E}}_t [r_t(\theta) A_t(s, a)]. \quad (\text{C2})$$

However, without a constraint, if the action taken is much more probable in our current policy than in our former, this would lead to a large policy gradient step and as a consequence an excessive policy update.

So the PPO algorithm took a new objective function to clip the estimated advantage function if the new policy is far away from the old policy [Eq. (C1)]. The loss function poses a lower bound on the improvement induced by an update and hence establishes a trust region around  $\pi_{\theta_{\text{old}}}$ . The hyperparameter  $\theta$  controls the maximal improvement and thus the size of the trust region.

#### Algorithm 1: Distributed Proximal Policy Optimization

---

Randomly initialize critic network  $V_\phi(s)$  and actor  $\pi_\theta(a|s)$  with weights  $\phi$  and  $\theta$ ;

---

**for** iteration  $\in \{1, 2, \dots, C\}$  **do**

**for** actor = 0, ...,  $N$  **do**

    Initialize  $s_0$ ;

    Run policy  $\pi_\theta \frac{T}{\delta t}$  times, collecting  $\{s_t, a_t, R_{t+1}\}$ ;

    Estimate advantages  $A_t = \sum_{t' > t} \gamma^{t'-t} R_{t'} - V_\phi(s_t)$ ;

    Estimate  $\hat{V}_t = A_t + V_\phi(s_t)$ ;

**end**

$\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$

**for**  $j \in \{1, \dots, M\}$  **do**

$J_{PPO}(\theta) = [\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1 - \epsilon, 1 + \epsilon)A_t)]$ ;

    Update  $\theta$  by a gradient method with respect to  $J_{PPO}$ ;

$J_{\text{critic}}(\phi) = -\mathbb{E}_t (\hat{V}_t - V_\phi(s_t))^2$ ;

    Update  $\phi$  by a gradient method with respect to  $J_{\text{critic}}(\phi)$

**end**

**end**

---

In order to improve the efficiency of the learning process, a distributed version of the PPO algorithm (DPPO) [45] is implemented in our calculation (Fig. 10). Algorithm 1 shows the pseudocode for the DPPO.



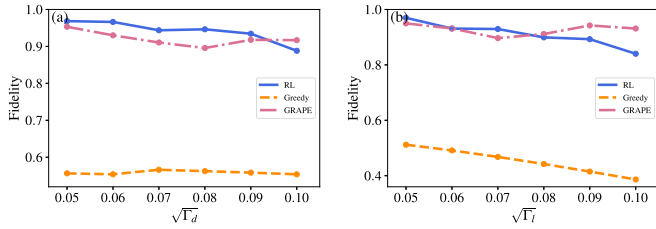


FIG. 12. Results from the three algorithms for four-level control model with the Hamiltonian of case (ii) under disturbances. The horizontal and vertical axes of each subfigure denote noise rate  $\sqrt{\Gamma_{k,n}}$  and fidelity  $\mathcal{F}$ . (A): dephasing dynamics (B): energy decay dynamics.

**APPENDIX D: THE EFFECT OF DISTRIBUTION OF EIGENERGIES**

In the main text we use a regular distribution of eigenenergies to test our algorithm. However, a different distribution of eigenenergy would affect the performance of the algorithm. The effect of the distribution of eigenenergies was examined for two example cases: (i) the eigenenergy  $E_i$  is extracted from the uniform distribution with  $E_1 = 0.40252154$ ,  $E_2 = 0.68846289$ ,  $E_3 = 0.8557115$ , and  $E_4 = 0.25471114$  and (ii) the eigenenergy  $E_i$  has degenerated in the middle with  $E_1 = 1$ ,  $E_2 = E_3 = 2$ , and  $E_4 = 3$ .

As Figs. 11 and 12 show, the performance of the three algorithms is affected by the different energy distributions. However, the GRAPE algorithm and our algorithm still maintain superiority over the greedy algorithm. This is consistent with what we discussed in the main text.

TABLE I. Training hyperparameters.

| Hyperparameter            | Values                   |
|---------------------------|--------------------------|
| Neurons in actor network  | {1024, 1024, 1024, 1024} |
| Neurons in critic network | {1024, 1024, 1024, 1024} |
| Actor numbers, $N$        | 12                       |
| Batch size                | $a$                      |
| PPO clipping $\epsilon$   | 0.2                      |
| Learning rate             | 0.0001 <sup>b</sup>      |
| Update steps, $M$         | 15                       |
| Reward decay, $\Gamma$    | 0.85                     |
| Total episode, $C$        | $c$                      |

<sup>a</sup>The same as the time steps.

<sup>b</sup>With the Adam algorithm.

<sup>c</sup>Different for various tasks.

**APPENDIX E: HYPERPARAMETERS AND LEARNING CURVES**

Our RL agent makes use of two deep neural networks to approximate the values for the possible actions of each state and the optimal policy. Each network consists of four layers. All layers have rectified linear unit (ReLU) activation functions except the output layer which has linear activation. The hyperparameters of the network are summarized in Table I.

All algorithms are implemented with PYTHON 3.6, and have been run on two 14-core 2.60-GHz CPUs with 188 GB memory and four GPUs.

[1] P. Vettori, *In Proceedings of the Mathematical Theory of Networks and Systems Conference CD ROM* (South Bend, Indiana, USA, 2002), pp. 1–6.

[2] S. Grivopoulos and B. Bamieh, in *42nd IEEE International Conference on Decision and Control*, Vol. 1 (IEEE, Piscataway, NJ, 2003), pp. 434–438.

[3] V. Jurdjevic, *Geometric Control Theory*, Cambridge Studies in Advanced Mathematics (Cambridge University Press, Cambridge, 1996).

[4] D. V. Zhdanov and T. Seideman, [arXiv:1709.09423](https://arxiv.org/abs/1709.09423).

[5] C. Avinadav, R. Fischer, P. London, and D. Gershoni, *Phys. Rev. B* **89**, 245311 (2014).

[6] K. Xia and J. Twamley, *Phys. Rev. X* **3**, 031013 (2013).

[7] J. Plantenberg, P. De Groot, C. Harmans, and J. Mooij, *Nature* **447**, 836 (2007).

[8] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, *J. Magn. Reson.* **172**, 296 (2005).

[9] T. Schulte-Herbrüggen, A. Spörl, N. Khaneja, and S. J. Glaser, *Phys. Rev. A* **72**, 042331 (2005).

[10] P. Doria, T. Calarco, and S. Montangero, *Phys. Rev. Lett.* **106**, 190501 (2011).

[11] T. Caneva, T. Calarco, and S. Montangero, *Phys. Rev. A* **84**, 022326 (2011).

[12] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2010).

[13] P. W. Shor, *Phys. Rev. A* **52**, R2493 (1995).

[14] A. M. Steane, *Phys. Rev. Lett.* **77**, 793 (1996).

[15] E. Knill and R. Laflamme, *Phys. Rev. A* **55**, 900 (1997).

[16] P. Zanardi and M. Rasetti, *Phys. Rev. Lett.* **79**, 3306 (1997).

[17] L. Viola, E. Knill, and S. Lloyd, *Phys. Rev. Lett.* **82**, 2417 (1999).

[18] W. Yang and R.-B. Liu, *Phys. Rev. Lett.* **101**, 180403 (2008).

[19] X. Xu, Z. Wang, C. Duan, P. Huang, P. Wang, Y. Wang, N. Xu, X. Kong, F. Shi, X. Rong, and J. Du, *Phys. Rev. Lett.* **109**, 070502 (2012).

[20] G. S. Uhrig, *Phys. Rev. Lett.* **98**, 100504 (2007).

[21] A. G. Kofman and G. Kurizki, *Phys. Rev. Lett.* **87**, 270405 (2001).

[22] P. Palittapongarnpim, P. Wittek, E. Zahedinejad, S. Vedaie, and B. C. Sanders, *Neurocomputing* **268**, 116 (2017).

[23] M. Born and V. Fock, *Z. Phys.* **51**, 165 (1928).

[24] O. Astafiev, Y. A. Pashkin, Y. Nakamura, T. Yamamoto, and J. S. Tsai, *Phys. Rev. Lett.* **93**, 267007 (2004).

[25] A. Shimizu and M. Ueda, *Phys. Rev. Lett.* **69**, 1403 (1992).

[26] M. R. Delbecq, T. Nakajima, P. Stano, T. Otsuka, S. Amaha, J. Yoneda, K. Takeda, G. Allison, A. Ludwig, A. D. Wieck, and S. Tarucha, *Phys. Rev. Lett.* **116**, 046802 (2016).

[27] C. H. Henry and R. F. Kazarinov, *Rev. Mod. Phys.* **68**, 801 (1996).

[28] F. Petruccione and H.-P. Breuer, *The Theory of Open Quantum Systems* (Oxford University Press, Oxford, 2002).

- [29] K. Børkje, A. Nunnenkamp, J. D. Teufel, and S. M. Girvin, *Phys. Rev. Lett.* **111**, 053603 (2013).
- [30] U. Boscain, *J. Math. Phys.* **43**, 2107 (2002).
- [31] M. Hirose and P. Cappellaro, *Quantum Inf. Process.* **17**, 88 (2018).
- [32] M. Bukov, A. G. R. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, *Phys. Rev. X* **8**, 031086 (2018).
- [33] T. Caneva, T. Calarco, R. Fazio, G. E. Santoro, and S. Montangero, *Phys. Rev. A* **84**, 012312 (2011).
- [34] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, *Nature* **529**, 484 (2016).
- [35] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, *Nature* **550**, 354 (2017).
- [36] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. Lillicrap, K. Simonyan, and D. Hassabis, *Science* **362**, 1140 (2018).
- [37] V. Mnih *et al.*, *Nature* **518**, 529 (2015).
- [38] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vechnyevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver, *Nature* **575**, 350 (2019).
- [39] M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castañeda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman, N. Sonnerat, T. Green, L. Deason, J. Z. Leibo, D. Silver, D. Hassabis, K. Kavukcuoglu, and T. Graepel, *Science* **364**, 859 (2019).
- [40] H. Yu, X. Xu, H. Ma, Z. Zhu, and C. Chen, in *2018 33rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)* (IEEE, New York, 2018), pp. 922–927.
- [41] Z. An and D. L. Zhou, *Europhys. Lett.* **126**, 60002 (2019).
- [42] M. Y. Niu, S. Boixo, V. N. Smelyanskiy, and H. Neven, *npj Quantum Inf.* **5**, 33 (2019).
- [43] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, *Phys. Rev. X* **8**, 031084 (2018).
- [44] H. Xu, J. Li, L. Liu, Y. Wang, H. Yuan, and X. Wang, *npj Quantum Inf.* **5**, 82 (2019).
- [45] N. Heess, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. Eslami, M. Riedmiller *et al.*, [arXiv:1707.02286](https://arxiv.org/abs/1707.02286).
- [46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, [arXiv:1707.06347](https://arxiv.org/abs/1707.06347).
- [47] J. C. V. Tieck, M. V. Pogančić, J. Kaiser, A. Roennau, M.-O. Gewaltig, and R. Dillmann, in *Artificial Neural Networks and Machine Learning—ICANN 2018*, edited by V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis (Springer International, Cham, 2018), pp. 211–221.
- [48] E. Bøhn, E. M. Coates, S. Moe, and T. A. Johansen, in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, Atlanta, GA, USA (IEEE, Piscataway, NJ, 2019), pp. 523–533.
- [49] M. August and J. M. Hernandez-Lobato, in *International Conference on High Performance Computing, ISC High Performance 2018*, Lecture Notes in Computer Science, Vol. 11203, edited by R. Yokota, M. Weiland, J. Shalf, and S. Alam (Springer, Cham, 2018), pp. 591–613.
- [50] J.-J. Chen and M. Xue, [arXiv:1901.08748](https://arxiv.org/abs/1901.08748).
- [51] C. Watkins, Ph.D. thesis, Department of Psychology, Cambridge University, 1989.
- [52] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, MA, 1998).
- [53] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, [arXiv:1506.02438](https://arxiv.org/abs/1506.02438).
- [54] A. Shapiro, *Handb. Oper. Res. Manage. Sci.* **10**, 353 (2003).
- [55] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, in *International Conference on Machine Learning*, Vol. 48 (JMLR: W&CP, New York, NY, USA, 2016), pp. 1928–1937.
- [56] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, OpenAI gym, [arXiv:1606.01540](https://arxiv.org/abs/1606.01540).
- [57] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TENSORFLOW: Large-scale machine learning on heterogeneous systems (2015), software available from tensorflow.org.
- [58] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov, OpenAI BASELINES, 2017, <https://github.com/openai/baselines>.
- [59] J. Johansson, P. Nation, and F. Nori, *Comput. Phys. Commun.* **183**, 1760 (2012).
- [60] J. Johansson, P. Nation, and F. Nori, *Comput. Phys. Commun.* **184**, 1234 (2013).
- [61] L. Wang, S. Hou, X. Yi, D. Dong, and I. R. Petersen, *Phys. Lett. A* **378**, 1074 (2014).
- [62] D. d’Alessandro, *Introduction to Quantum Control and Dynamics* (Chapman and Hall/CRC, London, 2007).
- [63] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, [arXiv:1502.05477](https://arxiv.org/abs/1502.05477).