He et al. *EPJ Quantum Technology* (2021) 8:29
https://doi.org/10.1140/epjqt/s40507-021-00119-6

EPJ Quantum Technology
a SpringerOpen Journal

EPJ.org

RESEARCH Open Access

Deep reinforcement learning for universal quantum state preparation via dynamic pulse control

Run-Hong He[1], Rui Wang[1], Shen-Shuang Nie[1], Jing Wu[1], Jia-Hui Zhang[1] and Zhao-Ming Wang[1*]

*Correspondence:
mingmoon78@126.com
[1]College of Physics and Optoelectronic Engineering, Ocean University of China, Qingdao, China

Abstract

Accurate and efficient preparation of quantum state is a core issue in building a quantum computer. In this paper, we investigate how to prepare a certain single- or two-qubit target state from arbitrary initial states in semiconductor double quantum dots with only a few discrete control pulses by leveraging the deep reinforcement learning. Our method is based on the training of the network over numerous preparing tasks. The results show that once the network is well trained, it works for any initial states in the continuous Hilbert space. Thus repeated training for new preparation tasks is avoided. Our scheme outperforms the traditional optimization approaches based on gradient with both the higher efficiency and the preparation quality in discrete control space. Moreover, we find that the control trajectories designed by our scheme are robust against stochastic fluctuations within certain thresholds, such as the charge and nuclear noises.

Keywords: Quantum control; Quantum state preparation; Semiconductor double quantum dots; Deep reinforcement learning

1 Introduction

Future quantum computers promise exponential speed-ups over their classical counterparts in solving certain problems like search and simulation [1]. A wide variety of promising modalities emerges in the race to realize the quantum computer, such as trapped ions [2, 3], photonic system [4–7], nitrogen-vacancy centers [8], nuclear magnetic resonance [9], superconducting circuits [10, 11] and semiconductor quantum dots [12–18]. Among these the semiconductor quantum dots is a powerful competitor for potential scalability, integrability with existing classical electronics and well-established fabrication technology. Spins of electrons trapped in quantum dots structure based on Coulomb effect can serve as spin-qubits for quantum information [19]. Spin qubits can be encoded in many ways, such as spin-1/2, singlet-triplet (S-T_0) and hybrid systems [20]. In particular, the spin S-T_0 qubit in double quantum dots (DQDs) attracts much attention for the merit that it can be manipulated solely with electrical pulses [21–23].

It has been proved that several arbitrary single-qubit gates plus an entangling two-qubit gate are the prototypes of all other logic gates in quantum algorithm implemented on a

Springer

circuit-model quantum computer [1, 24]. In an authentic sense, the implementation of any single- and two-qubit gates can be reduced to the state preparation problems. Arbitrary manipulations of a single-qubit can be achieved by successive rotations around the x- and z-axes on the Bloch sphere. In the context of S-T_0 single-qubit in semiconductor QDQs, the only tunable parameter J is the rotation rate around the z-axis, while the rotation rate h around the x-axis is difficult to be changed [25].

Various schemes have been proposed to add proper pulses on J to control the qubits [26–28]. It is typically required to iteratively solve a set of nonlinear equations [29, 30] for analytically tailoring the control trajectory, so it is a computationally exorbitant and time-consuming task in practice. There are also several traditional optimal methods based on gradient that can be used to design the control trajectory, such as stochastic gradient descent (SGD) [31], chopped random-basis optimization (CRAB) [32, 33] and gradient ascent pulse engineering (GRAPE) [34, 35]. However, the intensities of their pulses are nearly continuous, which may leave challenges to the experimental implementation. While the requirement of discrete pulses will inevitably reduce their performance [36]. In addition, their efficiency is limited by their iterative nature, which makes the task of designing pulses a big burden especially when there exist a large number of states waiting to be processed. Except for these traditional routes, recently the deep reinforcement learning (RL) [37] shows a wide applicability in quantum control problems [38–51]. For example, how to drive a qubit from a fixed initial state to another fixed target state with discrete pulses by leveraging the deep RL [52] has been investigated [36].

Recently, the generation of arbitrary states from a specific state [53] in nitrogen-vacancy center has been realized with the aid of the deep RL. Then it is intriguing to check if the deep RL can be used to realize a contrary problem: preparing a certain target state from arbitrary initial states, i.e., universal state preparation (USP). In practical quantum computation, it is often required to reset an arbitrary state to a specific target state [54–56]. For example, the initial state of the system always needs to be set to the ground state when transferring a quantum state through a spin chain [54, 55]. In the realization of quantum Toffoli or Fredkin gate, the ancilla state must be preprepared to the standard state $|0\rangle$ in certain cases [57–59]. Generation of two-qubit entangled state is also required [1] in completing quantum information processing tasks, such as the teleportation [60, 61]. Note that the network typically requires being trained again once the preparing task changes [36, 46]. Thus, the designing task of control pulses could be an exhausting work when there are lots of different states waiting to be prepared to a certain target state. In this paper, we investigate this USP problem with the deep RL in such a constrained driving parameters system. Benefited from a more sufficient learning on numerous preparing tasks, we find that the USP can be achieved with a single training of the network. Evaluation results show that our scheme outperforms the alternative optimization approaches both in terms of the efficiency of pulses designing and preparation quality in discrete control space. In addition, we find that the average step of control trajectories designed by our USP algorithm is obviously less than that of alternatives. Moreover, we discuss the robustness of the control trajectories designed by our USP algorithm against various noises and explore the major source of errors in control accuracy. We point out that by combining our scheme with Ref. [53], the driving between arbitrary states can be realized.

2 Models and methods

At first, we present the models of electrically controlled S-T_0 single- and two-qubit in semiconductor DQDs in Sects. 2.1 and 2.2, respectively. Then we present our USP algorithm in Sect. 2.3.

2.1 Voltage-controlled single-qubit in semiconductor DQDs

The effective control Hamiltonian of a single-qubit encoded by S-T_0 states in semiconductor DQDs can be written as [62–65],

$$H(t) = J(t)\sigma_z + h\sigma_x. \tag{1}$$

It is written under the computational basis states: the spin singlet state $|0\rangle = |S\rangle = (|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)/\sqrt{2}$ and the spin triplet state $|1\rangle = |T_0\rangle = (|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle)/\sqrt{2}$. Here the arrows indicate the spin projections of the electron in the left and right dots, respectively. σ_z and σ_x are the Pauli matrices. h accounts for the Zeeman energy spacing of two spins. Considering h is difficult to be changed experimentally [20], here we assume it is a constant $h = 1$ and set it to be the unit of pulse intensity. We also take the reduced Planck constant $\hbar = 1$ and the $1/\hbar$ as the time-scale throughout. Physically the exchange coupling $J(t)$ is tunable and non-negative [20]. In addition, if the $J(t)$ is limited to a finite range, so that not to destroy the charge configuration of the DQDs, the leakage of population to the non-computational space will be suppressed and we can study the evolution of the system safely within the Hilbert space spanned by the two bases [29, 30, 38].

Arbitrary single-qubit states can be written as

$$|s\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle, \tag{2}$$

where θ and φ are real numbers that define points on the Bloch sphere. For an initial state $|s_{\text{ini}}\rangle$ on the Bloch sphere, any target state $|s_{\text{tar}}\rangle$ can be achieved by successive rotations around the x- and z-axes of the Bloch sphere. In the context of semiconductor DQDs, h and $J(t)$ cause rotations around the x-axis and z-axis of the Bloch sphere, respectively.

2.2 Capacitively coupled S-T_0 qubits in semiconductor DQDs

Operations on two entangled qubits are often required in quantum information processing. In semiconductor DQDs, interqubit operations can be performed on two adjacent and capacitively coupled S-T_0 qubits. In the basis of $\{|SS\rangle, |ST_0\rangle, |T_0S\rangle, |T_0T_0\rangle\}$, the Hamiltonian can be written as [21, 23, 28, 63, 66, 67],

$$H_{2-qubit} = \frac{\hbar}{2}\begin{pmatrix} J_1+J_2 & h_2 & h_1 & 0 \\ h_2 & J_1-J_2 & 0 & h_1 \\ h_1 & 0 & J_2-J_1 & h_2 \\ 0 & h_1 & h_2 & -J_1-J_2+2J_{12} \end{pmatrix}, \tag{3}$$

where h_i and J_i are the Zeeman energy spacing and exchange coupling of the ith qubit respectively. $J_{12} \propto J_1J_2$ refers to the strength of Coulomb coupling between two qubits. $J_i > 0$ is required to maintain the interqubit coupling all the time. For simplicity, we take $h_1 = h_2 = 1$ and $J_{12} = J_1J_2/2$ here.

2.3 Universal state preparation via deep reinforcement learning

Our target is to drive arbitrary initial states to a certain target state with discrete pulses. The control trajectory is discretized as a piece-wise constant function, i.e., the pulses have rectangular shapes [34]. While, the conclusion still holds if one take into account the finite rise time of the pulses that can be available with an arbitrary wave generator [23, 28, 63, 66, 68] in actual experiments: we need just alter the parameters of the pulses generated by our algorithm slightly as demonstrated in [26] and [29]. So, it is a reasonable simplification to perform the optimization with ideal, zero rise time pulses.

The strategy used here is to generate this control trajectory with the deep Q network algorithm (DQN) [69, 70], which is an important member of deep RL. The details of the DQN are described in *Appendix*. Here we just refer it as a neural network, i.e., the Main Net θ.

Our scheme of obtaining a competent Main Net goes as follows: *Firstly*, a database comprised of numerous potential initial quantum states is divided randomly into the training set, the validation set and the test set. The states in the training set will be used to train the Main Net in turn. The validation set will be utilized to estimate the generalization error of the Main Net during the training process. The test set will be employed to evaluate the Main Net's final performance after training. *Secondly*, the random-initialized Main Net is initially fed with a sampled initial state s from the training set at step $k = 1$ and then outputs the predicted "best action" a_k (i.e., the pulse intensity $J(t)$). According to the current state s and the action a_k, calculate the next state $|s'\rangle = \exp(-iH(a_k)\,\mathrm{d}t)|s\rangle$ and the corresponding fidelity $F = |\langle s_{\mathrm{tar}}|s'\rangle|^2$. The fidelity F indicates how close the next state is to the target state. Then the next state s' is fed to the Main Net as the new current state with the step $k \leftarrow k+1$. The reward will envelopes the fidelity $r = r(F)$ and be used to train the Main Net. Then repeat the above operations until the episode terminates when k reaches the maximum step or the fidelity excesses a certain satisfactory threshold. Correspondingly, the control trajectory is constructed by these predicted actions orderly. After more than enough episodes of training over different preparing tasks, the Main Net learns to assign an action-value (also named Q-value) to each state and action pair gradually according to the correspondence between them and the target state. With accurate Q-values, it is easily to determinate which action should be chosen in a given state. So that the Main Net can match every potential state with a reasonable action towards the target state. *Finally*, the well-trained Main Net can be used to tailor the appropriate control trajectories for these initial states databased in the test set and even all other states in the continuous Hilbert space.

The overview of this training and pulses designing process is pictured in Fig. 1. And a full description of the training process is given in Algorithm 1.

3 Results

In this section, we compare and contrast the performance of our scheme with two sophisticated optimization approaches based on gradient for the USP problem. As demonstration, we consider the preparation of a single-qubit state $|0\rangle$ and a two-qubit Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$. We stress that our USP scheme is applicable to any other target states as long as it is trained specifically.

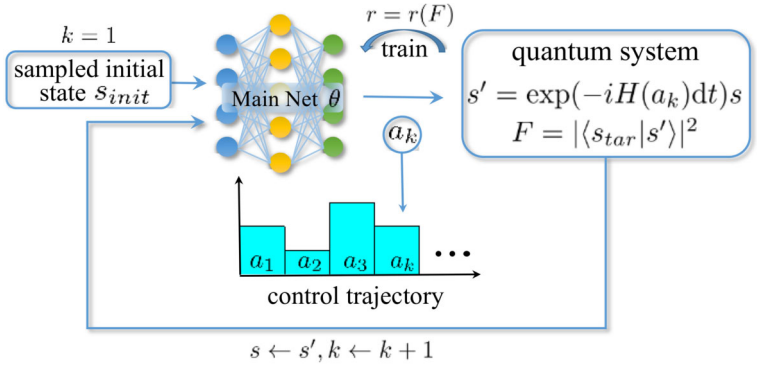


Figure 1 Overview of the USP algorithm to learn control trajectory designing. The details of this algorithm is described in the main text of Sect. 2.3 and in the pseudocode Algorithm 1

Algorithm 1 The pseudocode for training the USP algorithm

Initialize the Experience Memory D to empty.
Randomly initialize the Main-network θ.
Initialize the Target-network θ^- by: $\theta^- \leftarrow \theta$.
Set the $\epsilon = 0$.
for episode = 0, episode$_{\max}$ **do**
  Initialize the state $s = s_{\text{ini}}$ according to the training point selected randomly from the training set.
  while True **do**
    With probability $1 - \epsilon$ select a random action a_i, otherwise $a_i = \text{argmax}_a\, Q(s, a; \theta)$.
    Set the $\epsilon = \epsilon + \delta\epsilon$, except $\epsilon = \epsilon_{\max}$.
    Execute a_i and observe the reward r, and the next state s'.
    Store experience unit = (s, a_i, r, s') in D.
    Select batch size N_{bs} of experiences units randomly from D.
    Update θ by minimizing the *Loss* function.
    Every C steps, set $\theta^- \leftarrow \theta$.
    break if $F > F_{\text{threshold}}$ or step $\geq T/\mathrm{d}t$.
  end while
end for

3.1 Universal single-qubit state preparation

Now we consider the preparation of the single-qubit state $|0\rangle$ by using our USP algorithm. Considering the challenges to implement pulses with continuous intensity, our scheme takes only several discrete allowed actions on $J(t)$: 0, 1, 2 or 3 with duration $\mathrm{d}t = \pi/10$. We stress that these settings are made experimentally and can be further tailored as required. The maximum total operation time is limited to be 2π, which is uniformly discretized into 20 slices. The Main Net consists of two hidden layers with 32 neurons each. The reward function should be set to allow a growth in itself as the fidelity increases, thus the Main Net can be inspirited to pursue a higher fidelity. In practice, we find that the function $r = F$ works well. For training the Main Net and evaluating the performance of our algorithm, we sample 128 points on the Bloch sphere uniformly with respective to the θ and the φ as the initial states. Both the training and validation sets contain 32 points, while the test set

Table 1 List of hyperparameters for USP

Parameters\Target state	$\|0\rangle$	Bell state
Allowed actions a ($J(t)$)	0, 1, 2, 3	[a]
Size of the training set	32	256
Size of the validation set	32	256
Size of the test set	64	6400
Batch size N_{bs}	32	32
Memory size M	20,000	40,000
Learning rate α	0.01	0.0001
Replace period C	200	200
Reward discount factor γ	0.9	0.9
Number of hidden layers	2	3
Neurons per hidden layer	32/32	256/256/128
Activation function	Relu	Relu
ϵ-greedy increment $\delta\epsilon$	0.001	0.0001
Maximal ϵ in training $\epsilon_{\max}$	0.95	0.95
ϵ in validation and testing	1	1
$F_{\mathbf{threshold}}$ per episode	0.999	0.999
$\mathbf{episode}_{\max}$ for training	33	731
Total time T	2π	20π
Action duration $\mathbf{d}t$	$\pi/10$	$\pi/2$
Maximum steps per episode	20	40

[a] The allowed actions of two-qubit operations satisfy $\{(J_1, J_2)|J_1, J_2 \in \{1,2,3,4,5\}\}$.

consists of the remaining 64 points. The details of all hyperparameters for this algorithm has been listed in Table 1.

As shown in Fig. 2(a), after about 33 episodes of training, the average fidelity and total reward over the validation set have no obvious increase as the episode grows up, which implies the Main Net converges and can be used to implement the USP task.

To evaluate the performance of our algorithm, we compare it with two alternatives: the GRAPE and the CRAB. Considering that the efficiency of an algorithm is also an important metric when facing a large number of different preparation tasks, we plot their preparation fidelities of state $|0\rangle$ versus the corresponding runtime of designing the control trajectories in Fig. 2(b). The average fidelities $\bar{F}$ = 0.9968, 0.9721, 0.9655 and the average pulses designing time $\bar{t}$ = 0.0120, 0.0268, 0.7504 with USP, GRAPE and CRAB, respectively. The fidelities of the three algorithms are the maximums that can be achieved within the maximum step. To satisfy the limitation of discrete pulses, for the GRAPE and the CRAB, their continuous control strengths are discretized into the nearest allowed actions when the designing process is completed [36]. Although the two traditional algorithms can achieve high average fidelities after convergence with continuous control pulses, $\bar{F}$ = 0.9997 for GRAPE and $\bar{F}$ = 0.9995 for CRAB, they do not perform well in discrete control space. Figure 2(b) shows that our USP algorithm outperforms the alternative optimization approaches both in terms of preparation quality and pulses designing efficiency in discrete control space. Clearly, CRAB algorithm performs the worst, and GRAPE algorithm is in the middle. The average steps to achieve the maximum fidelities are 12.297, 14.109, 13.375 with USP, GRAPE and CRAB, respectively. A trajectory with fewer steps required for a given state preparation task corresponds to a faster control scheme in experiment. Overall, the control trajectories generated by our USP algorithm are better than that of the alternatives.

To show the control trajectory designed by our USP algorithm visually, as an example we plot one in Fig. 3(a), where the position of the initial state on the Bloch sphere is $\theta = 2\pi/7$,

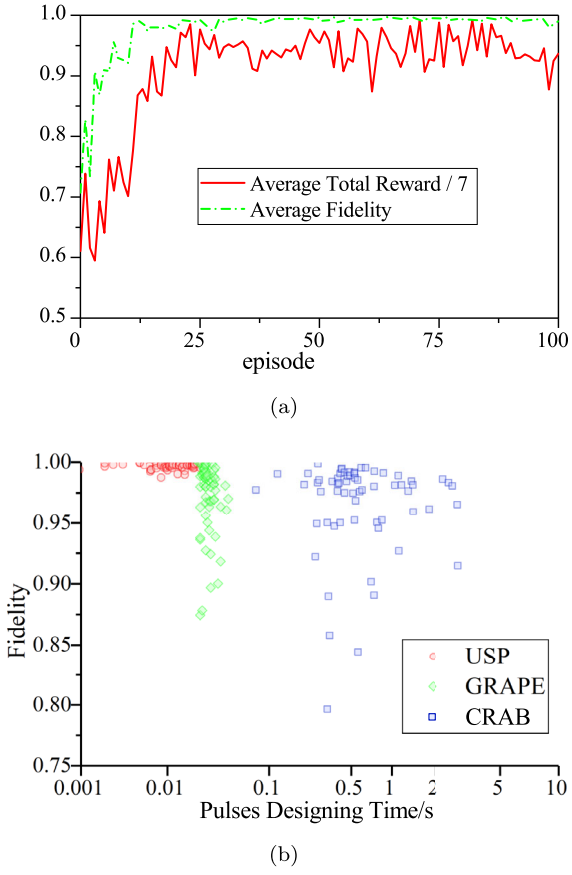
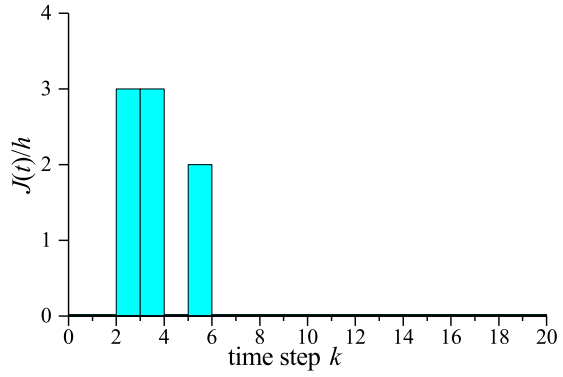


Figure 2 (**a**) The average fidelity and total reward over the validation set as functions of the number of episodes in the training process for the single-qubit $|0\rangle$ USP. (**b**) The distribution of state preparation fidelities F versus pulses designing time for preparing single-qubit $|0\rangle$ over 128 sampled tasks in the test set with different optimization algorithms. The average fidelities $\overline{F} = 0.9968, 0.9721, 0.9655$ and the average pulses designing time $\overline{t} = 0.0120, 0.0268, 0.7504$ with USP, GRAPE and CRAB, respectively

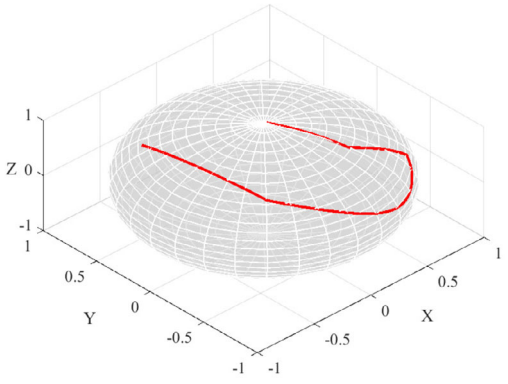
$\varphi = 3\pi/7$ and the target state is $|0\rangle$. It shows that the USP algorithm takes only 6 steps to complete this task. The reason is that the DQN algorithm favors the policy with fewer steps due to the discounted reward (See the details of the DQN described in *Appendix*). In Fig. 3(b), we plot the corresponding motion trail of the quantum state on the Bloch sphere during operations. It shows that the final quantum state reaches a position that is very close to the target state $|0\rangle$ on the Bloch sphere and the final fidelity $F = 0.9999$.

3.2 Universal two coupled S-T_0 qubits state preparation

Now we consider the preparation of the Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$ [1] from arbitrary initial states. The allowed pulse strengths on each qubit are defined as $\{(J_1, J_2)|J_1, J_2 \in \{1, 2, 3, 4, 5\}\}$. The reward function is set to be $r = F$. The architecture of the Main Net employed in this task is different from the one used for the manipulation of single-qubit and the detailed hyper-parameters are captured in Table 1. The database used to train and to test the Main Net contains 6912 points that are defined as $\{[a_1, a_2, a_3, a_4]^T\}$. $a_j = bc_j$ refers to the probability amplitude corresponding to the jth basis state. $b \in \{1, i, -1, -i\}$. c_js



(a)



(b)

Figure 3 (**a**) Control trajectory designed by our USP algorithm. The task is to reset the point $\theta = 2/7\pi$, $\varphi = 3/7\pi$ on the Bloch sphere to the target state $|0\rangle$. The pulses only take discrete values 0, 2 and 3. This reset task is completed at the sixth step. (**b**) The corresponding motion trail for the reset task on the Bloch sphere with the final fidelity $F = 0.9999$

together define points on a four-dimensional unit hypersphere,

$$\begin{cases} c_1 = \cos\theta_1, \\ c_2 = \sin\theta_1 \cos\theta_2, \\ c_3 = \sin\theta_1 \sin\theta_2 \cos\theta_3, \\ c_4 = \sin\theta_1 \sin\theta_2 \sin\theta_3, \end{cases} \tag{4}$$

where $\theta_i \in \{\pi/8, \pi/4, 3\pi/8\}$. The normalization condition is satisfied for each quantum state represented by these points. The database is divided randomly into the training set, the validation set and the test set with 256, 256 and 6400 points, respectively. As depicted in Fig. 4(a), the Main Net converges after about 700 episodes of training. With 731 episodes of training, the average fidelity of the Bell state preparation over all the test points $\bar{F} = 0.9695$. The maximum total operation time is taken as 20π and be discretized into 40 slices with pulse duration $dt = \pi/2$. In Fig. 4(b), we plot the distribution of the fidelities of the test points under control trajectories designed by our USP scheme in this two-qubit preparing

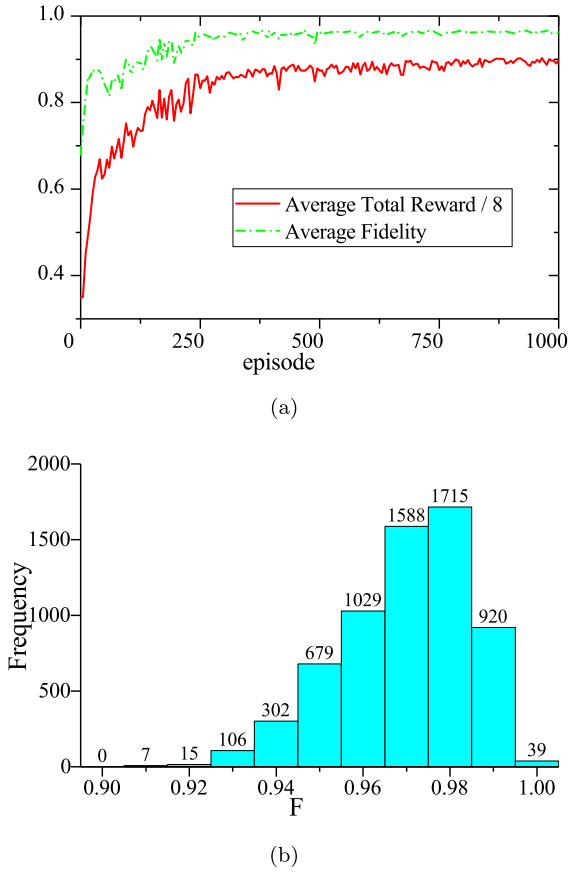


Figure 4 (**a**) The average fidelity and total reward over the validation set as functions of the number of training episodes for two-qubit Bell state USP. (**b**) The fidelities distribution of 6400 sampled preparation tasks in the test set for the two-qubit Bell state USP with $\bar{F} = 0.9695$

task. The average pulses designing time $\bar{t} = 0.0477$ and the average step to complete the preparation tasks is 24.014. It shows that although the fidelities are distributed unevenly between the interval $[0.91, 1]$, the overall performance is good.

3.3 USP in noisy environments

In the preceding section, we have studied the USP problem without considering the surrounding environment. However the qubits will suffer from a variety of fluctuations in a practical experiment, which prevents the accessibility of high precision control over the system. There exist works studied the corrected gate operations that employ the additional pulses to counteract the impact of various noises, such as the SUPCODE [26, 29]. However they treat different noises equally resulting the designed control trajectories are too long to implement in actual experiment (about 300π of rotation for a single quantum gate). Thus it is worth exploring which noise will lead to the most serious threat to the control accuracy. Then designing the compensating pulses to shorten the total control trajectory using the SUPCODE as well as to improve the physical platform accordingly.

Next we will study the performance of the control trajectories designed by our USP algorithm under two main noises leading to stochastic errors in the system Hamiltonian: the charge noise and nuclear noise. Considering that they vary on a typical time scale

($\sim$100 μs) much longer than a gate duration ($\sim$10 ns), we take them constants during the preparation task. We point out that these noises are integrated into the system's evolution after the control trajectories have been designed by our Main Net, which is trained on a clean model. This is a reasonable assumption since normally the environment is unpredictable.

The charge noise stems from the imperfection of the external voltage field, while the nuclear noise comes from the uncontrolled hyperfine coupling with spinful nuclei of the host material [63, 71, 72]. They can be represented by an additional term $\delta\sigma_z$ (or $\delta\sigma_x$) in the Hamiltonian (1) for the single-qubit case or by additional terms $\delta_i\sigma_z$ (or $\delta_i\sigma_x$) in the Hamiltonian (3) for the two-qubit case. Where $i \in \{1, 2\}$ indicate the corresponding qubit and δ (δ_i) are the amplitudes of the noises. In addition, for the two-qubit Bell state preparation, we assume that the amplitudes of the noises on the two qubits are identical.

Average fidelities of the target states $|0\rangle$ and Bell state preparation with control trajectories generated by our USP over all test points versus amplitudes of two noises are plotted in Fig. 5(a) and (b), respectively. It can be seen from Fig. 5, the average fidelities do not change significantly and the control trajectories exhibit robustness against considered imperfections within certain thresholds. We also find that in the analyzed parameter windows the $\overline{F}$ in nuclear noise are always higher than that in charge noise with the same amplitudes for

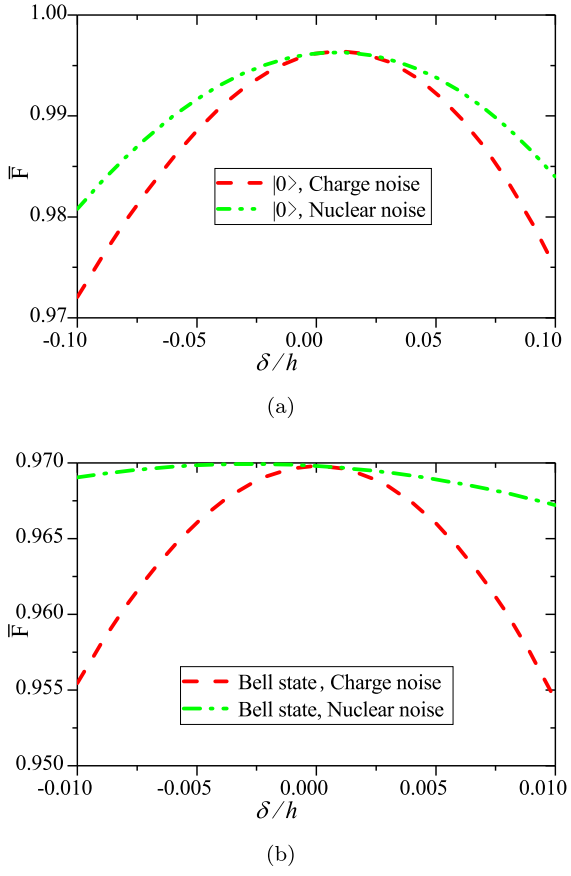


Figure 5 Average fidelities of the target state (**a**) $|0\rangle$ and (**b**) Bell state preparation with USP over all test points versus amplitudes of charge and nuclear noises

both single- and two-qubit cases. It reveals that the charge noise leaves the most impact to the preparation tasks.

A meaningful point worth stating is that the best average fidelity can even be obtained in non-zero nuclear noise from Fig. 5(b). That is to say, certain noises can be helpful to boost the fidelity due to subtle adjustments on parameters. A possible explanation may be the limitation of the discrete value in our calculation. We believe that there is still a room for the achievement of better performance by employing more allowed actions and more deliberate Zeeman energy spacing, just as what these noises do. Of course, more sufficient training on Main Net is also helpful for the enhancement of the fidelity.

Given the limitations of quantum computing hardwares presently accessible, we simulate quantum computing on a classical computer and generate data to train the network. Our algorithm is implemented with PYTHON 3.7.9, TensorFlow 2.6.0 and QuTip 4.6.2 and have been run on an 4-core 1.80 GHz CPU with 8 GB memory. Details of the running environment of the algorithm can be found in *Availability of data and materials*. The runtime for the training process of USP algorithms are about tens of seconds in the single-qubit case and about an hour in the two-qubit case.

4 Conclusions

Precise and efficient quantum state preparation is crucial for quantum information processing. In this paper, we proposed an efficient scheme to generate appropriate control trajectories to reset arbitrary single- or two-qubit states to a certain target state with the aid of deep RL. Our scheme has the advantage that once the network is well trained, it works for arbitrary initial states and does not require training again. Taking the control of spin S-T_0 qubits in semiconductor DQDs as an example, the evaluation results show that our scheme outperforms traditional optimization approaches with both preparation quality and pulses designing efficiency. In addition, the average step required to complete the preparation tasks of our USP algorithm is obviously less than that of the alternatives, which implies faster control schemes in experiment. Moreover, we found that the control trajectories designed by our scheme exhibit robustness against two main noises within certain thresholds and discovered the charge noise leaves the most impact to the control precision. Although we only considered the single and two-qubit state preparation in semiconductor DQDs, this scheme can be extended to a wide variety of quantum control problems.

Appendix

A.1 Deep reinforcement learning and deep Q network

In this section, we will introduce the deep RL and DQN algorithm, which underlie our USP scheme.

The deep RL combines the deep learning algorithm that is good at nonlinear fitting and the RL algorithm that is expert in dynamic programming problems [37, 52, 73]. In RL, an Agent is generally used to represent an object with decision-making and action capability, such as a robot. We consider a Markov decision process in which the next state depends only on the current state as well as the action performed by the Agent and has no relation with the past states [52]. In the interaction between the Agent and the Environment, the current state s of the Environment will be changed to another next state s',

after the Agent selecting and performing an action a_i chose from the set of allowed actions $a = \{a_1, a_2, \ldots, a_n\}$ at time t. In return, the Environment also gives a feedback, or immediate reward r to the Agent. A Policy π represents which action the Agent will be chose in a given state, i.e., $a_i = \pi(s)$. The process is defined as an episode in which the Agent starts from an initial state until it completes the task or terminates in halfway.

The total discounted reward R gained in an N-steps episode can be written as [52]:

$$R = r_1 + \gamma r_2 + \gamma^2 r_3 \cdots + \gamma^{N-1} r_N = \sum_{t=1}^{N} \gamma^{t-1} r_t, \tag{5}$$

where γ is a discount factor within the interval $[0, 1]$, which indicates that the immediate reward r discounts with the steps increasing. The goal of the Agent is to maximize R, because a greater R implies a better performance of the Agent. Because the discounted r, the Agent tends naturally to get a bigger reward as quickly as possible to ensure a considerable R. To determine which action should to be chose in a given state, we introduce the action-value function, which is also named Q-value [74]:

$$Q^{\pi}(s, a_i) = E[r_t + \gamma r_{t+1} + \cdots | s, a_i] = E\big[r_t + \gamma Q^{\pi}\left(s', a'\right) | s, a_i\big]. \tag{6}$$

The Q-value indicates the expectation of R, which the Agent will get after it executing an action a_i in a given state s under the policy π, and this value can be obtained iteratively according to the Q-values of the next state. Because there are multiple allowed actions can be chosen in each state, and different actions will lead to different next states, it is a time-consuming task to calculate Q-values in a multi-step process. To reduce the overhead, there are various algorithms used to calculate approximations of that expectation, such as Q-learning [74] and SARSA [52].

In Q-learning, the current $Q(s, a_i)$ value is obtained by the Q-value of the next state's "best action" [74]:

$$Q(s, a_i) \leftarrow Q(s, a_i) + \alpha[r_t + \gamma \max_{a'} Q(s', a') - Q(s, a_i)], \tag{7}$$

where α is the learning rate, which affects the convergence of this function. The part of $Q_{\text{target}}(s', a') = r_t + \gamma \max_{a'} Q(s', a')$ is called the Q_{target} value. All the Q-values of different states and actions can be recorded in a so-called Q-Table. With a precise Q-Table, it is easily to identify which action should be chose in a given state. However, on the one hand, we need the best action to calculate iteratively the Q-value; on the other hand, we must know all the Q-values to determine which action is the best. To solve this dilemma of "exploitation" and "exploration", we adopt the ϵ-greedy strategy in choosing action to execute, i.e., choose the action corresponding to the current maximum Q-value with a probability of ϵ to calculate Q-value efficiently, or choose an action randomly with a probability of $1 - \epsilon$ to expand the range of consideration. At the beginning, since it is not known that which action is the best one in a certain state, the ϵ is set to be 0 to explore as many states and actions as possible. When sufficient states and actions are explored, that parameter gradually increases with the amplitude of $\delta\epsilon$ until to $\epsilon_{\max}$, which is slightly smaller than 1, to calculate the Q-values efficiently.

For an Environment with a large number or even an infinite number of states, the Q-Table would be prohibitively large. To solve this "dimensional disaster", we can substitute this table with a multi-layer neural network. After learning, the network will be capable to match a suited Q-value to each action after be fed with a certain state. The deep Q network algorithm (DQN) [69, 70] are based on the Equation (7). A network, the Main Net θ, is used to predict the term $Q(s, a_i)$, and another network, the Target Net θ^- is used to predict the term $\max_{a'} Q(s', a')$ in Equation (7) respectively. In order to ensure the ability of generalization, the data used to train the Main Net must meet the assumption of independent and identically distributed, i.e. each sample of the dataset is independent of another while the training and test set are identically distributed. So we adopt the experience memory replay strategy [70]: the Agent could get an experience unit (s, a, r, s') at each step. After numerous steps, the Agent will collect a lot of such units that can be stored in an Experience Memory D with capacity of Memory size M. In the process of training, the Agent randomly samples batch size N_{bs} of experience units from the Experience Memory to train the Main Net at each time step. Notice that to ensure the stability of the algorithm only the Main Net is trained in every time step by minimizing the *Loss* function:

$$Loss = \frac{1}{N_{bs}} \sum_{i=1}^{N_{bs}} \left(\left[r + \gamma \max_{a'} Q(s', a') \right]_i - Q(s, a)_i \right)^2, \tag{8}$$

where N_{bs} is the sample batch size through mini-batch gradient descent (MBGD) algorithm [37, 69, 70]. While the Target Net θ^- is not updated in real time, instead, it copies the parameters from the Main Net θ every C steps. A schematic of this DQN algorithm is shown in Fig. 6.

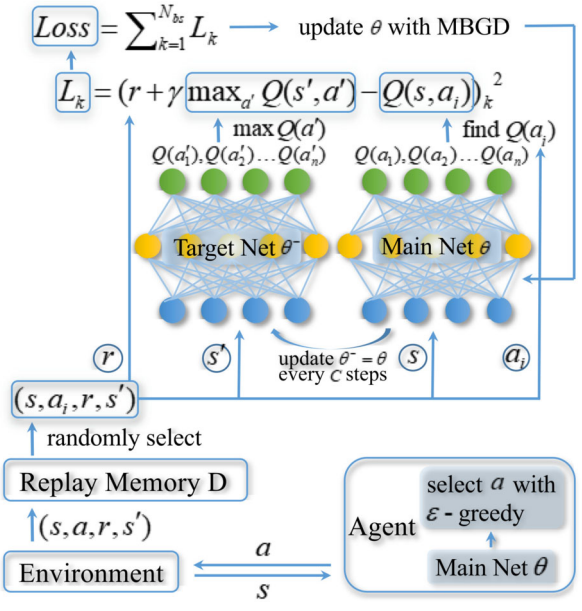


Figure 6 Schematic for the DQN algorithm. See the main text of the Appendix for details of the algorithm

Acknowledgements
We would like to thank Xin-Hong Han, Jing-Hao Sun and Chen Chen for useful discussions.

Funding
This work was supported by the Shandong Provincial Natural Science Foundation (Grant. Nos. ZR2021LLZ004 and ZR2014AM023) and the Natural Science Foundation of China (Grant No. 11475160).

Abbreviations
DQD, double quantum dot; SGD, stochastic gradient descent; GRAPE, gradient ascent pulse engineering; CRAB, chopped random-basis optimization; RL, reinforcement learning; USP, universal state preparation; SUPCODE, soft uniaxial positive control for orthogonal drift error; DQN, deep Q network; SARSA, State action reward state action; MBGD, mini-batch grade descent.

Availability of data and materials
The code, running environment of algorithm and all data supporting the conclusions of this article are available from the corresponding author on reasonable request or on Gitee repository under MIT License (https://gitee.com/herunhong/DL_for_USP).

Declarations

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
Z-MW and R-HH conceived the project, R-HH carried out the numerical simulations and prepared the first version of the manuscript. All authors participated in the discussions and approved the final manuscript.

Publisher's Note

Received: 27 May 2021 Accepted: 13 December 2021 Published online: 20 December 2021

References
1. Nielsen MA, Chuang I. Quantum computation and quantum information. American Association of Physics Teachers. 2002.
2. Richerme P, Gong Z-X, Lee A, Senko C, Smith J, Foss-Feig M, Michalakis S, Gorshkov AV, Monroe C. Non-local propagation of correlations in quantum systems with long-range interactions. Nature. 2014;511(7508):198–201.
3. Casanova J, Mezzacapo A, McClean JR, Lamata L, Aspuru-Guzik A, Solano E, et al. From transistor to trapped-ion computers for quantum chemistry. Sci Rep. 2014.
4. Bellec M, Nikolopoulos GM, Tzortzakis S. Faithful communication Hamiltonian in photonic lattices. Opt Lett. 2012;37(21):4504–6.
5. Perez-Leija A, Keil R, Moya-Cessa H, Szameit A, Christodoulides DN. Perfect transfer of path-entangled photons in j x photonic lattices. Phys Rev A. 2013;87(2):022303.
6. Peruzzo A, McClean J, Shadbolt P, Yung M-H, Zhou X-Q, Love PJ, Aspuru-Guzik A, O'brien JL. A variational eigenvalue solver on a photonic quantum processor. Nat Commun. 2014;5:4213.
7. Chapman RJ, Santandrea M, Huang Z, Corrielli G, Crespi A, Yung M-H, Osellame R, Peruzzo A. Experimental perfect state transfer of an entangled photonic qubit. Nat Commun. 2016;7:11339.
8. Childress L, Hanson R. Diamond nv centers for quantum computing and quantum networks. Mater Res Soc Bull. 2013;38(2):134–8.
9. Vandersypen LM, Chuang IL. Nmr techniques for quantum control and computation. Rev Mod Phys. 2005;76(4):1037.
10. Devoret MH, Schoelkopf RJ. Superconducting circuits for quantum information: an outlook. Science. 2013;339(6124):1169–74.
11. Wendin G. Quantum information processing with superconducting circuits: a review. Rep Prog Phys. 2017;80(10):106001.
12. Zajac DM, Sigillito AJ, Russ M, Borjans F, Taylor JM, Burkard G, Petta JR. Resonantly driven cnot gate for electron spins. Science. 2018;359(6374):439–42.
13. Huang W, Yang C, Chan K, Tanttu T, Hensen B, Leon R, Fogarty M, Hwang J, Hudson F, Itoh KM et al. Fidelity benchmarks for two-qubit gates in silicon. Nature. 2019;569(7757):532–6.
14. Watson T, Philips S, Kawakami E, Ward D, Scarlino P, Veldhorst M, Savage D, Lagally M, Friesen M, Coppersmith S et al. A programmable two-qubit quantum processor in silicon. Nature. 2018;555(7698):633–7.
15. Jang W, Cho M-K, Kim J, Chung H, Umansky V, Kim D. Three individual two-axis control of singlet-triplet qubits in a micromagnet integrated quantum dot array. 2020. arXiv preprint. 2009.13182.
16. Hanson R, Kouwenhoven LP, Petta JR, Tarucha S, Vandersypen LM. Spins in few-electron quantum dots. Rev Mod Phys. 2007;79(4):1217.
17. Eriksson MA, Friesen M, Coppersmith SN, Joynt R, Klein LJ, Slinker K, Tahan C, Mooney P, Chu J, Koester S. Spin-based quantum dot quantum computing in silicon. Quantum Inf Process. 2004;3(1–5):133–46.
18. Zwanenburg FA, Dzurak AS, Morello A, Simmons MY, Hollenberg LC, Klimeck G, Rogge S, Coppersmith SN, Eriksson MA. Silicon quantum electronics. Rev Mod Phys. 2013;85(3):961.
19. Loss D, DiVincenzo DP. Quantum computation with quantum dots. Phys Rev A. 1998;57(1):120.

20. Zhang X, Li H-O, Cao G, Xiao M, Guo G-C, Guo G-P. Semiconductor quantum computation. Nat Sci Rev. 2019;6(1):32–54.
21. Taylor J, Engel H-A, Dür W, Yacoby A, Marcus C, Zoller P, Lukin M. Fault-tolerant architecture for quantum computation using electrically controlled semiconductor spins. Nat Phys. 2005;1(3):177–83.
22. Wu X, Ward DR, Prance J, Kim D, Gamble JK, Mohr R, Shi Z, Savage D, Lagally M, Friesen M et al. Two-axis control of a singlet–triplet qubit with an integrated micromagnet. Proc Natl Acad Sci. 2014;111(33):11938–42.
23. Nichol JM, Orona LA, Harvey SP, Fallahi S, Gardner GC, Manfra MJ, Yacoby A. High-fidelity entangling gate for double-quantum-dot spin qubits. npj Quantum Inf. 2017;3(1):1–5.
24. Bishnoi B. Quantum-computation and applications. 2020.
25. Throckmorton RE, Zhang C, Yang X-C, Wang X, Barnes E, Sarma SD. Fast pulse sequences for dynamically corrected gates in singlet-triplet qubits. Phys Rev B. 2017;96(19):195424.
26. Wang X, Bishop LS, Kestner J, Barnes E, Sun K, Sarma SD. Composite pulses for robust universal control of singlet–triplet qubits. Nat Commun. 2012;3(1):1–7.
27. Kestner J, Wang X, Bishop LS, Barnes E, Sarma SD. Noise-resistant control for a spin qubit array. Phys Rev Lett. 2013;110(14):140502.
28. Wang X, Barnes E, Sarma SD. Improving the gate fidelity of capacitively coupled spin qubits. npj Quantum Inf. 2015;1(1):1–7.
29. Wang X, Bishop LS, Barnes E, Kestner J, Sarma SD. Robust quantum gates for singlet-triplet spin qubits using composite pulses. Phys Rev A. 2014;89(2):022310.
30. Yang X-C, Yung M-H, Wang X. Neural-network-designed pulse sequences for robust control of singlet-triplet qubits. Phys Rev A. 2018;97(4):042324.
31. Ferrie C. Self-guided quantum tomography. Phys Rev Lett. 2014;**113**(19).
32. Doria P, Calarco T, Montangero S. Optimal control technique for many body quantum systems dynamics. Phys Rev Lett. 2010;106(19):237.
33. Caneva T, Calarco T, Montangero S. Chopped random-basis quantum optimization. Phys Rev A. 2011;84(2):17864.
34. Khaneja N, Reiss T, Kehlet C, Schulte-Herbrüggen T, Glaser SJ. Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms – sciencedirect. J Magn Res. 2005;172(2):296–305.
35. Rowland B, Jones JA. Implementing quantum logic gates with gradient ascent pulse engineering: principles and practicalities. Philos Trans A Math Phys Eng. 2012;370(1976):4636–50.
36. Zhang X-M, Wei Z, Asad R, Yang X-C, Wang X. When does reinforcement learning stand out in quantum control? A comparative study on state preparation. npj Quantum Inf. 2019;5(1):1–7.
37. Goodfellow I, Bengio Y, Courville A. Deep learning. Cambridge: MIT Press; 2016.
38. Zhang X-M, Cui Z-W, Wang X, Yung M-H. Automatic spin-chain learning to explore the quantum speed limit. Phys Rev A. 2018;97(5):052333.
39. Yang X, Liu R, Li J, Peng X. Optimizing adiabatic quantum pathways via a learning algorithm. Phys Rev A. 2020;102(1):012614.
40. Lin J, Lai ZY, Li X. Quantum adiabatic algorithm design using reinforcement learning. Phys Rev A. 2020;101(5):052327.
41. Bukov M. Reinforcement learning for autonomous preparation of Floquet-engineered states: inverting the quantum kapitza oscillator. Phys Rev B. 2018;98(22):224305.
42. Bukov M, Day AG, Sels D, Weinberg P, Polkovnikov A, Mehta P. Reinforcement learning in different phases of quantum control. Phys Rev X. 2018;8(3):031086.
43. Kong X, Zhou L, Li Z, Yang Z, Qiu B, Wu X, Shi F, Du J. Artificial intelligence enhanced two-dimensional nanoscale nuclear magnetic resonance spectroscopy. npj Quantum Inf. 2020;6(1):1–10.
44. Palmieri AM, Kovlakov E, Bianchi F, Yudin D, Straupe S, Biamonte JD, Kulik S. Experimental neural network enhanced quantum tomography. npj Quantum Inf. 2020;6(1):1–5.
45. Wang ZT, Ashida Y, Ueda M. Deep reinforcement learning control of quantum cartpoles. Phys Rev Lett. 2020;**125**(10).
46. Zheng A, Zhou DL. Deep reinforcement learning for quantum gate control. Europhys Lett. 2019;126(6):60002.
47. Niu MY, Boixo S, Smelyanskiy V, Neven H. Universal quantum control through deep reinforcement learning. npj Quantum Inf. 2019;**5**(33).
48. Gratsea A, Metz F, Busch T. Universal and optimal coin sequences for high entanglement generation in 1d discrete time quantum walks. J Phys A, Math Theor. 2020.
49. Lin J, Lai ZY, Li X. Quantum adiabatic algorithm design using reinforcement learning. Phys Rev A. 2020;101:052327. https://doi.org/10.1103/PhysRevA.101.052327.
50. Ma H, Dong D, Ding SX, Chen C. Curriculum-based deep reinforcement learning for quantum control. 2021. 2012.15427.
51. Bukov M, Day AGR, Sels D, Weinberg P, Polkovnikov A, Mehta P. Reinforcement learning in different phases of quantum control. Phys Rev X. 2018;8:031086. https://doi.org/10.1103/PhysRevX.8.031086.
52. Sutton RS, Barto AG. Reinforcement learning: an introduction. IEEE Trans Neural Netw. 1998;9(5):1054.
53. Haug T, Mok WK, You JB, Zhang W, Png CE, Kwek LC. Classifying global state preparation via deep reinforcement learning. Mach Learn Sci Technol. 2021;2(1):01.
54. Wang Z-M, Sarandy MS, Wu L-A. Almost exact state transfer in a spin chain via pulse control. Phys Rev A. 2020;102:022601. https://doi.org/10.1103/PhysRevA.102.022601.
55. Wang Z-M, Ren F-H, Luo D-W, Yan Z-Y, Wu L-A. Almost-exact state transfer by leakage-elimination-operator control in a non-Markovian environment. Phys Rev A. 2020;102:042406. https://doi.org/10.1103/PhysRevA.102.042406.
56. Ren F-H, Wang Z-M, Wu L-A. Accelerated adiabatic quantum search algorithm via pulse control in a non-Markovian environment. Phys Rev A. 2020;102:062603. https://doi.org/10.1103/PhysRevA.102.062603.
57. DiVincenzo DP. Two-bit gates are universal for quantum computation. Phys Rev A. 1995;51(2):1015.
58. Feynman RP. Simulating physics with computers. Int J Theor Phys. 1982;**21**(6/7).
59. Smolin JA, DiVincenzo DP. Five two-bit quantum gates are sufficient to implement the quantum Fredkin gate. Phys Rev A. 1996;53(4):2855.
60. Bennett CH, Brassard G, Crépeau C, Jozsa R, Peres A, Wootters WK. Teleporting an unknown quantum state via dual classical and Einstein–Podolsky–Rosen channels. Phys Rev Lett. 1993;70(13):1895.

61. Bouwmeester D, Pan J-W, Mattle K, Eibl M, Weinfurter H, Zeilinger A. Experimental quantum teleportation. Nature. 1997;390(6660):575–9.
62. Malinowski FK, Martins F, Nissen PD, Barnes E, Cywiński Ł, Rudner MS, Fallahi S, Gardner GC, Manfra MJ, Marcus CM et al. Notch filtering the nuclear environment of a spin qubit. Nat Nanotechnol. 2017;12(1):16–20.
63. Petta JR, Johnson AC, Taylor JM, Laird EA, Yacoby A, Lukin MD, Marcus CM, Hanson MP, Gossard AC. Coherent manipulation of coupled electron spins in semiconductor quantum dots. Science. 2005;309(5744):2180–4.
64. Bluhm H, Foletti S, Mahalu D, Umansky V, Yacoby A. Universal quantum control of two electron spin qubits via dynamic nuclear polarization. APS. 2009. 17–008.
65. Maune BM, Borselli MG, Huang B, Ladd TD, Deelman PW, Holabird KS, Kiselev AA, Alvarado-Rodriguez I, Ross RS, Schmitz AE et al. Coherent singlet-triplet oscillations in a silicon-based double quantum dot. Nature. 2012;481(7381):344–7.
66. Shulman MD, Dial OE, Harvey SP, Bluhm H, Umansky V, Yacoby A. Demonstration of entanglement of electrostatically coupled singlet-triplet qubits. Science. 2012;336(6078):202–5.
67. Van Weperen I, Armstrong B, Laird E, Medford J, Marcus C, Hanson M, Gossard A. Charge-state conditional operation of a spin qubit. Phys Rev Lett. 2011;107(3):030506.
68. Krantz P, Kjaergaard M, Yan F, Orlando TP, Oliver WD. A quantum engineer's guide to superconducting qubits. Appl Phys Rev. 2019;6(2):021318.
69. Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller M. Playing atari with deep reinforcement learning. 2013. arXiv preprint. 1312.5602.
70. Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G et al. Human-level control through deep reinforcement learning. Nature. 2015;518(7540):529–33.
71. Barnes E, Cywiński Ł, Sarma SD. Nonperturbative master equation solution of central spin dephasing dynamics. Phys Rev Lett. 2012;109(14):140403.
72. Nguyen NT, Sarma SD. Impurity effects on semiconductor quantum bits in coupled quantum dots. Phys Rev B. 2011;83(23):235322.
73. Shalev-Shwartz S, Ben-David S. Understanding machine learning: from theory to algorithms. Cambridge: Cambridge University Press; 2014.
74. Watkins CJ, Dayan P. Q-learning. Mach Learn. 1992;8(3–4):279–92.