

Quantum imaginary time evolution steered by reinforcement learning

Chenfeng Cao^{1,6}, Zheng An^{2,3,6}, Shi-Yao Hou¹, D. L. Zhou^{1,2,3,4,5} & Bei Zeng¹

The quantum imaginary time evolution is a powerful algorithm for preparing the ground and thermal states on near-term quantum devices. However, algorithmic errors induced by Trotterization and local approximation severely hinder its performance. Here we propose a deep reinforcement learning-based method to steer the evolution and mitigate these errors. In our scheme, the well-trained agent can find the subtle evolution path where most algorithmic errors cancel out, enhancing the fidelity significantly. We verified the method's validity with the transverse-field Ising model and the Sherrington-Kirkpatrick model. Numerical calculations and experiments on a nuclear magnetic resonance quantum computer illustrate the efficacy. The philosophy of our method, eliminating errors with errors, sheds light on error reduction on near-term quantum devices.

¹ Department of Physics, The Hong Kong University of Science and Technology, Hong Kong, China. ² Institute of Physics, Beijing National Laboratory for Condensed Matter Physics, Chinese Academy of Sciences, Beijing 100190, China. ³ School of Physical Sciences, University of Chinese Academy of Sciences, Beijing 100049, China. ⁴ Collaborative Innovation Center of Quantum Matter, Beijing 100190, China. ⁵ Songshan Lake Materials Laboratory, Dongguan, Guangdong 523808, China. ⁶These authors contributed equally: Chenfeng Cao, Zheng An.  email: zhoudl72@iphy.ac.cn; zengb@ust.hk

Quantum computers promise to solve some computational problems much faster than classical computers in the future. However, large-scale fault-tolerant quantum computers are still years away. On noisy intermediate-scale quantum (NISQ) devices¹, quantum noise strictly limits the depth of reliable circuits, which makes many quantum algorithms unrealistic, e.g., Shor algorithm for factorization², Harrow–Hassidim–Lloyd algorithm for solving linear systems of equations³. Nevertheless, there exist quantum algorithms that are well suited for NISQ devices and may achieve quantum advantage with practical applications, such as the variational quantum algorithms^{4–12}, the quantum imaginary time evolution^{13–15}, quantum annealing¹⁶.

The quantum imaginary time evolution (QITE) is a promising near-term algorithm to find the ground state of a given Hamiltonian. It has also been applied to prepare thermal states, simulate open quantum systems, and calculate finite temperature properties^{17–19}. A pure quantum state is said to be k -UGS if it is the unique ground state of a k -local Hamiltonian $\hat{H} = \sum_{j=1}^m \hat{h}[j]$, where each local term $\hat{h}[j]$ acts on at most k neighboring qubits. Any k -UGS state can be uniquely determined by its k -local reduced density matrices among pure states (which is called k -UDP) or even among all states (which is called k -UDA)^{20,21}. The QITE algorithm is well suited for preparing k -UGS states with a relatively small k . We start from an initial state $|\Psi_{\text{init}}\rangle$, which is non-orthogonal to the ground state of the target Hamiltonian. The final state after long-time imaginary time evolution

$$\lim_{\beta \rightarrow \infty} e^{-\beta \hat{H}} |\Psi_{\text{init}}\rangle \quad (1)$$

has very high fidelity with the k -UGS state. If the ground state of \hat{H} is degenerate, the final state still falls into the ground state space. Trotter–Suzuki decomposition can simulate the evolution,

$$e^{-\beta \hat{H}} = \left(e^{-\Delta \tau \hat{h}[1]} e^{-\Delta \tau \hat{h}[2]} \dots e^{-\Delta \tau \hat{h}[m]} \right)^n + \mathcal{O}(\Delta \tau^2), \quad (2)$$

where $\Delta \tau$ is the step interval, $n = \frac{\beta}{\Delta \tau}$ is the number of Trotter step. Trotter error subsumes terms of order $\Delta \tau^2$ and higher. On NISQ devices, Trotter error is difficult to reduce due to the circuit depth limits and Trotterized simulation cannot be implemented accurately²².

Since we can only implement unitary operations on a quantum computer, the main idea of the QITE is to replace each non-unitary step $e^{-\Delta \tau \hat{h}[j]}$ by a unitary evolution $e^{-i \Delta \tau \hat{A}[j]}$ such that

$$|\Psi'\rangle = \frac{e^{-\Delta \tau \hat{h}[j]}}{\sqrt{\langle \Psi | e^{-2 \Delta \tau \hat{h}[j]} | \Psi \rangle}} |\Psi\rangle \approx e^{-i \Delta \tau \hat{A}[j]} |\Psi\rangle, \quad (3)$$

where $|\Psi\rangle$ is the state before this step. $\hat{A}[j]$ acts on D neighboring qubits and can be determined by measurements on $|\Psi\rangle$. For details of the local approximation see Supplementary Note 1. If the domain size D equals the system size N , there always exists $\hat{A}[j]$, such that the approximation sign of Eq. (3) becomes an equal sign. However, $\exp(D)$ local gates are required to implement a generic D -qubit unitary, and we also need to measure $\exp(D)$ observables to determine $\hat{A}[j]$. The exponential resource of measurements and computation makes a large domain size D unfeasible, and we can only use a small one on real devices. This brings the local approximation (LA) error.

Trotter error and the LA error are two daunting challenges in the QITE. These algorithmic errors accumulate with the increase of steps n , which severely weakens the practicability of the QITE. On NISQ computers, a circuit with too many noisy gates is unreliable, and the final measurements give no helpful information. Therefore we cannot use a small step interval $\Delta \tau$ to reduce

Trotter errors since this would increase the circuit depth, and noise would dominate the final state. The number of Trotter steps is a tradeoff between quantum noise and Trotter error. For the QITE with large-size systems, we need more Trotter steps and larger domain sizes, which seems hopeless on current devices. There exist some techniques to alleviate the problem, refs. 23–25 illustrated some variants of the QITE algorithm with shallower circuits. Refs. 19,26 used Hamiltonian symmetries, error mitigation, and randomized compiling to reduce the required quantum resources and improve the fidelity.

Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents interact with an environment to perform well in a specific task. It achieved great success in classical games^{27–32} and has been employed in quantum computing problems, such as quantum control^{33–38}, quantum circuit optimization^{39–41}, the quantum approximate optimization^{42,43}, and quantum annealing⁴⁴. Quantum computing, in turn, can enhance classical reinforcement learning^{45,46}.

In this work, we propose a deep reinforcement learning-based method to steer the QITE and mitigate algorithmic errors. In our method, we regard the ordering of local terms in the QITE as the environment and train an intelligent agent to take actions (i.e., exchange adjacent terms) to minimize the final state energy. RL is well suited for this task since the state and action-space can be pretty large. We verified the validity of our method with the transverse-field Ising model and the Sherrington–Kirkpatrick model. The RL agent can mitigate most algorithmic errors and decrease the final state energy. Our work pushes the QITE algorithm to more practical applications in the NISQ era.

Results and discussion

In the following, we apply our method to the transverse-field Ising model and the Sherrington–Kirkpatrick (SK) model. A QITE circuit, the experimental setup, and a schematic of the SK model are given in Fig. 1a–c.

Transverse-field Ising model. We first consider the one-dimensional transverse-field Ising model. With no assumption about the underlying structure, we initialize all qubits in the product state $|\Psi_{\text{init}}\rangle = (|0\rangle + |1\rangle)^{\otimes N} / \sqrt{2^N}$. The Hamiltonian can be written as

$$\hat{H}^{\text{TFI}} = - \sum_j (J \hat{Z}_j \hat{Z}_{j+1} + h \hat{X}_j). \quad (4)$$

In the following, we choose $J = h = 1$. The system is in the gapless phase. For finite-size systems, the ground state of \hat{H}^{TFI} is 2-UGS, therefore 2-UDP and 2-UDA.

In the standard QITE, the ordering of local terms in each Trotter step is the same, e.g. we put commuting terms next to each other and repeat the ordering $\hat{X}_1, \dots, \hat{X}_N, \hat{Z}_0 \hat{Z}_1, \hat{Z}_1 \hat{Z}_2, \dots, \hat{Z}_{N-1} \hat{Z}_N$. The quantum circuit of the standard QITE with four qubits is shown in Fig. 1a. Inspired by the randomization technique to speed up quantum simulation^{47,48}, we also consider a randomized QITE scheme where we randomly shuffle the ordering in each Trotter step. There is no large quality difference between randomizations, and we pick a moderate one. In the RL-steered QITE, the reward is based on the expectation value of the output state $|\Psi_f\rangle$ on the target Hamiltonian

$$E = \langle \Psi_f | \hat{H}^{\text{TFI}} | \Psi_f \rangle. \quad (5)$$

The lower the energy, the higher the reward. The RL agent updates the orderings step by step.

For any given β , the RL agent can steer the QITE path and maximize the reward. We fix the system size $N = 4$, the number

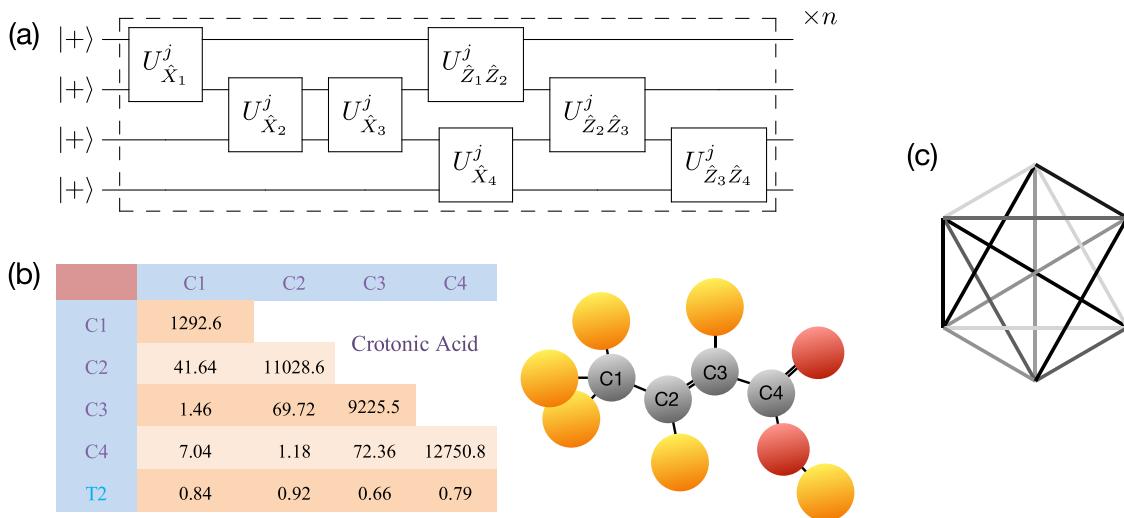


Fig. 1 Theoretical and experimental setup. a Quantum circuit of the quantum imaginary time evolution (QITE) for the transverse-field Ising model with 4 qubits and n Trotter steps. $U_{\hat{P}}^j$ represents the unitary operator that approximates $e^{-\Delta t \hat{P}}$ for the local Hamiltonian \hat{P} in the j -th Trotter step. **b** Molecule structure and nuclei parameters of the nuclear magnetic resonance processor. The molecule has four Carbon atoms C1, C2, C3, and C4. Diagonal entries of the table are the chemical shifts in Hz, off-diagonal entries of the table are the J -couplings between two corresponding nuclei. The T_2 row gives the relaxation time of each nucleus. **c** A six-vertex complete graph with weighted edges. Different shades of gray represent different couplings in the Sherrington-Kirkpatrick model.

of Trotter steps $n = 4$, and the domain size $D = 2$. A numerical comparison of energy/fidelity obtained by the standard, the randomized, and the RL-steered QITE schemes for different β values is shown in Fig. 2a, b. The RL-steered path here is β -dependent. Throughout this paper, we use the fidelity defined by $F(\rho, \sigma) = \text{Tr} \sqrt{\sigma^{1/2} \rho \sigma^{1/2}}$.

When β is small, RL cannot markedly decrease the energy since the total quantum resource is limited. With the increase of β , the imaginary time evolution target state $|\Psi'_k\rangle = e^{-\beta \hat{H}^{\text{TFI}}} |\Psi_{\text{init}}\rangle$ approaches the ground state, therefore the obtained energy of all paths decrease in the beginning. However, algorithmic errors increase with β , and this factor becomes dominant after a critical point, the energy of the standard/randomized QITE increases when $\beta > 1/3$. Accordingly, the fidelity increases first, then decreases. The RL-steered QITE outperforms the standard/randomized QITE for all β values. Algorithmic errors in this path are canceled out. The fidelity between the output state and the ground state constantly grows to $F > 0.996$. The gap between the ground state energy and the minimum achievable energy of the standard QITE is 0.053, and that of the RL-QITE is only 0.016. For a detailed optimized path see Supplementary Note 2 (Supplementary Table 1).

Further, we implement the same unitary evolutions on a 4-qubit liquid state nuclear magnetic resonance (NMR) quantum processor⁴⁹. We carry out the experiments with a 300-MHz Bruker AVANCE III spectrometer. The processor is ^{13}C -labeled trans-crotonic acid in a 7-T magnetic field. The resonance frequency for ^{13}C nuclei is about 75 MHz. The Hamiltonian of the system in a rotating frame is

$$\hat{H}_S = \sum_{j=1}^4 \pi \nu_j \hat{Z}_j + \sum_{1 \leq i < j \leq 4} \frac{\pi}{2} J_{ij} \hat{Z}_i \hat{Z}_j, \quad (6)$$

where ν_j is the chemical shift of the j -th nuclei, J_{ij} is the J -coupling between the i -th and j -th nuclei, \hat{Z}_j is the Pauli matrix σ_z acting on the j -th nuclei. All the parameters can be found in Fig. 1b. The quantum operations are realized by irradiating radiofrequency pulses on the system. We optimize the pulses over the fluctuation of the chemical shifts of the nuclei with the technique of gradient

ascent pulse engineering⁵⁰. The experiment is divided into three steps: (i) preparing the system into a pseudo-pure state using the temporal average technique⁵¹; (ii) applying the quantum operations; (iii) performing measurements⁵².

Denote the NMR output state as ρ , whose density matrix can be obtained through quantum state tomography. ρ is a highly mixed state since quantum noise is inevitable. We use the virtual distillation technique to suppress the noise^{53,54}. The dominant eigenvector of ρ , $\lim_{M \rightarrow \infty} \rho^M / \text{Tr}(\rho^M)$, can be extracted numerically. Its expectation value on \hat{H}^{TFI} and its fidelity with the ground state are shown in Fig. 2a, b with unfilled markers. Consistent with our numerical results, the RL-steered path significantly outperforms the other two for large β . For postprocessing of NMR data see Supplementary Note 3.

In our simulation, we have four orderings to optimize and 28 local unitary operations to implement. Denote $|\Psi_k\rangle$ as the state after the k -th operation, $|\Psi'_k\rangle$ as the temporal target state with the ideal imaginary time evolution. The instantaneous algorithmic error during the evolution can be characterized by the squared Euclidean distance between $|\Psi_k\rangle$ and $|\Psi'_k\rangle$,

$$\epsilon_{\text{alg}} = \| |\Psi_k\rangle - |\Psi'_k\rangle \|^2. \quad (7)$$

For $\beta = 0.9$, Fig. 2c shows ϵ_{alg} as a function of evolution step k . Although ϵ_{alg} fluctuates in all paths, it accumulates obviously in the standard QITE and eventually climbs to $\epsilon_{\text{alg}} = 0.082$. The randomized QITE performs slightly better and ends with $\epsilon_{\text{alg}} = 0.030$. The RL-steered QITE is optimal, the trend of ϵ_{alg} shows no accumulation and drops to $\epsilon_{\text{alg}} = 0.007$ in the end. Although we cannot directly estimate ϵ_{alg} in experiments, we can minimize it via maximizing the reward function.

One question that arises is whether we can enhance the QITE algorithm with RL for larger systems? Now we apply our approach to the transverse-field Ising model with system sizes $N = 2, 3, \dots, 8$ to demonstrate the scalability. Still, we consider the QITE with 4 Trotter steps. Denote the target state with “evolution

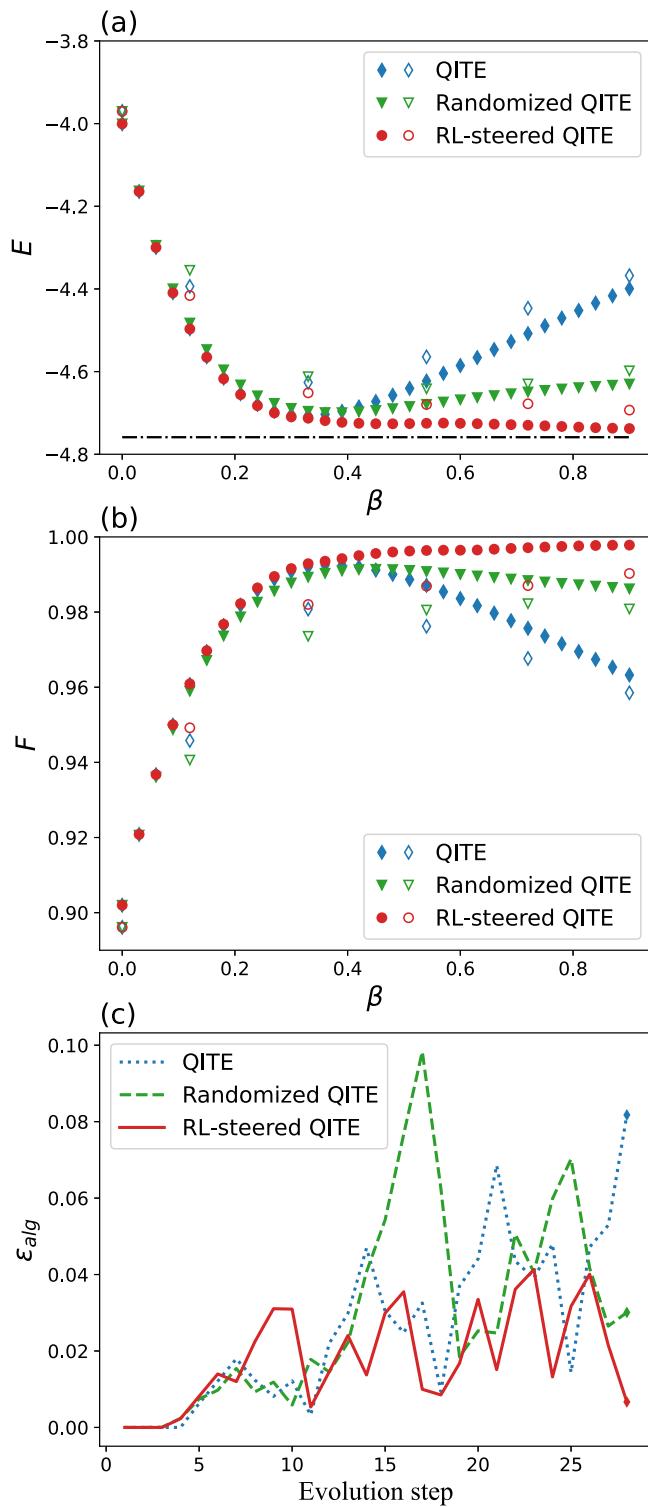


Fig. 2 Different QITE schemes for the transverse-field Ising model. Filled markers represent the numerical data; unfilled markers represent the experimental data. The experimental errors were mitigated by virtual distillation. Blue diamonds/dotted lines represent the standard quantum imaginary time evolution (QITE); green triangles/dashed lines represent the randomized QITE; red circles/lines represent the reinforcement learning (RL)-steered QITE. **a** Energy versus β , the black dashed line represents the ground state energy. **b** Fidelity versus β . **c** Algorithmic errors during the evolution.

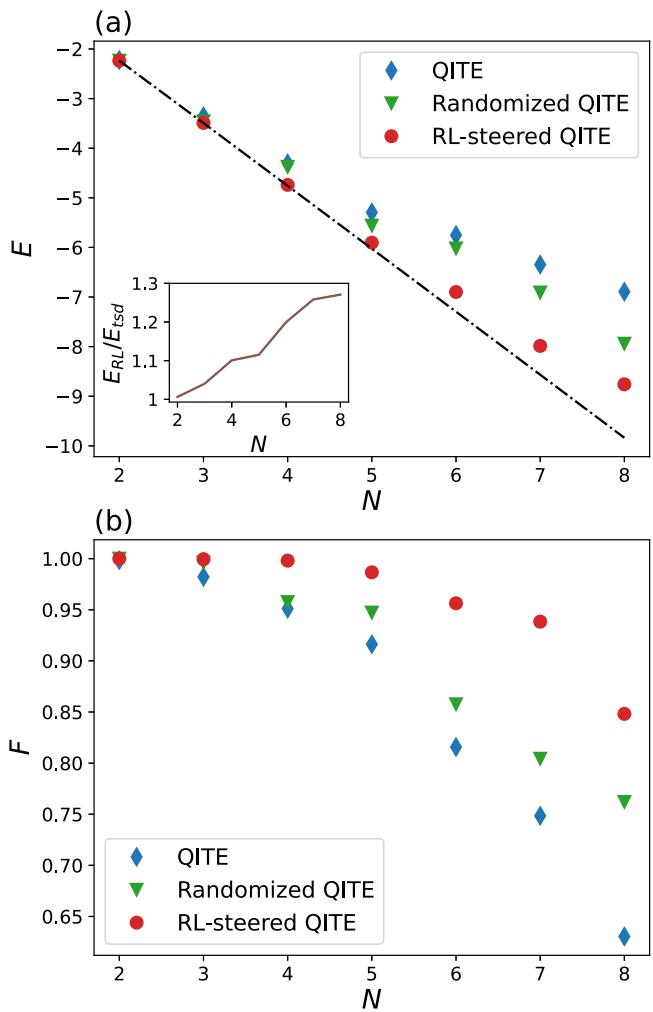


Fig. 3 Scaling of different QITE schemes. Blue diamonds represent the standard quantum imaginary time evolution (QITE); green triangles represent the randomized QITE; red circles represent the reinforcement learning (RL)-steered QITE. **a** Energy versus system size, and the energy ratio E_{RL}/E_{std} versus system size (inset). The black dash-dotted line represents the ground state energy. **b** Fidelity versus system size.

time” β as

$$|\Psi'_f(\beta)\rangle = \frac{e^{-\beta \hat{H}^{\text{TFI}}}}{\sqrt{\langle \Psi_{\text{init}} | e^{-2\beta \hat{H}^{\text{TFI}}} | \Psi_{\text{init}} \rangle}} |\Psi_{\text{init}}\rangle. \quad (8)$$

In the following, we use an adaptive β for different N such that the expectation value $\langle \Psi'_f(\beta) | \hat{H}^{\text{TFI}} t | \Psi'_f(\beta) \rangle$ is always higher than the ground state energy of \hat{H}^{TFI} by 1×10^{-3} . The results are illustrated in Fig. 3, the RL agent can efficiently decrease the energy and increase the fidelity between the final state and the ground state for all system sizes. The ratio of the RL-steered energy (E_{RL}) to the standard QITE energy (E_{std}) is also given. This ratio increases steadily with the number of qubits. Note that the neural networks we use here only contain four hidden layers. The hyperparameters were tuned for the $N=4$ case. We apply the same neural networks to larger N , the required number of training epochs does not increase obviously. If we want to increase the fidelity further for $N > 4$, we can use more Trotter steps and tune the hyperparameters accordingly. There is little

doubt, however, the training process will be more time-consuming. For comparison between the QITE and quantum annealing see Supplementary Note 4.

Sherrington–Kirkpatrick model. The second model we apply our method to is the Sherrington–Kirkpatrick (SK) model⁵⁵, a spin-glass model with long-range frustrated ferromagnetic and antiferromagnetic couplings. Finding a ground state of the SK model is NP-hard⁵⁶. On NISQ devices, solving the SK model can be regarded as a special Max-Cut problem and dealt with by the quantum approximate optimization algorithm^{5,57}. Here we use the QITE to prepare the ground state of the SK model. Compared with the quantum approximate optimization algorithm, the QITE does not need to sample a bunch of initial points and implement classical optimization with exponentially small gradients⁵⁸.

Consider a six-vertex complete graph shown in Fig. 1c. The SK model Hamiltonian can be written as

$$\hat{H}^{\text{SK}} = \sum_{i,j} J_{ij} \hat{Z}_i \hat{Z}_j, \quad (9)$$

we independently sample J_{ij} are from a uniform distribution $J_{ij} \sim U(-1, 1)$.

Since $\hat{Z}\hat{Z}$ -terms commute, there is no Trotter error in Eq. (2) for the SK model. The ground state of \hat{H}^{SK} is twofold degenerate. The QITE algorithm can prepare one of the ground states. We fix $\beta = 5$, $n = 6$, $D = 2$, sample J_{ij} and train the agent to steer the QITE path. Define the probability of finding a ground state of \hat{H}^{SK} through measurements as P_{gs} . Energy and P_{gs} as functions of β are shown in Fig. 4. Remember that the RL-steered path here was only optimized for a specific β value (i.e., $\beta = 5$) since we want to verify the dependence of the ordering on β .

For each path, we observe a sudden switch from a high probability of success to a low one when $4 < \beta < 5$. The reason is that for some states when the step interval exceeds a critical value, there is an explosion of LA error which utterly ruins QITE. After the critical value, the QITE algorithm loses its stability, and the energy E fluctuates violently. The randomized QITE performs even worse than the standard one, where the highest success probability is only 0.60. In the RL-steered QITE, P_{gs} falls down a deep “gorge” but can recover soon. When $\beta = 5$, most algorithmic errors disappeared and $P_{\text{gs}} = 0.9964$. In comparison, the standard QITE ends with $P_{\text{gs}} = 0.0002$ and the randomized QITE ends with $P_{\text{gs}} = 0.0568$, they dropped sharply and cannot be recovered even if we further improve β . For detailed couplings and QITE path see Supplementary Note 2 (Supplementary Tables 2 and 3). For additional numerical results of the SK model see Supplementary Note 5. For the control landscape see Supplementary Note 6.

Conclusions

We have proposed an RL-based framework to steer the QITE algorithm for preparing a k -UGS state. The RL agent can find the subtle evolution path to avoid error accumulation. We compared our method with the standard and the randomized QITE; numerical and experimental results demonstrate a clear improvement. The RL-designed path requires a smaller domain size and fewer Trotter steps to achieve satisfying performance for both the transverse-field Ising model and the SK model. We also noticed that randomization cannot enhance the QITE consistently, although it plays a positive role in quantum simulation. For the SK model, with the increase of the total imaginary time β , a switch from a high success probability to almost 0 exists. The accumulated error may ruin the QITE algorithm all of a sudden instead of gradually, which indicates the importance of an appropriate β for high-dimensional systems. The RL-based

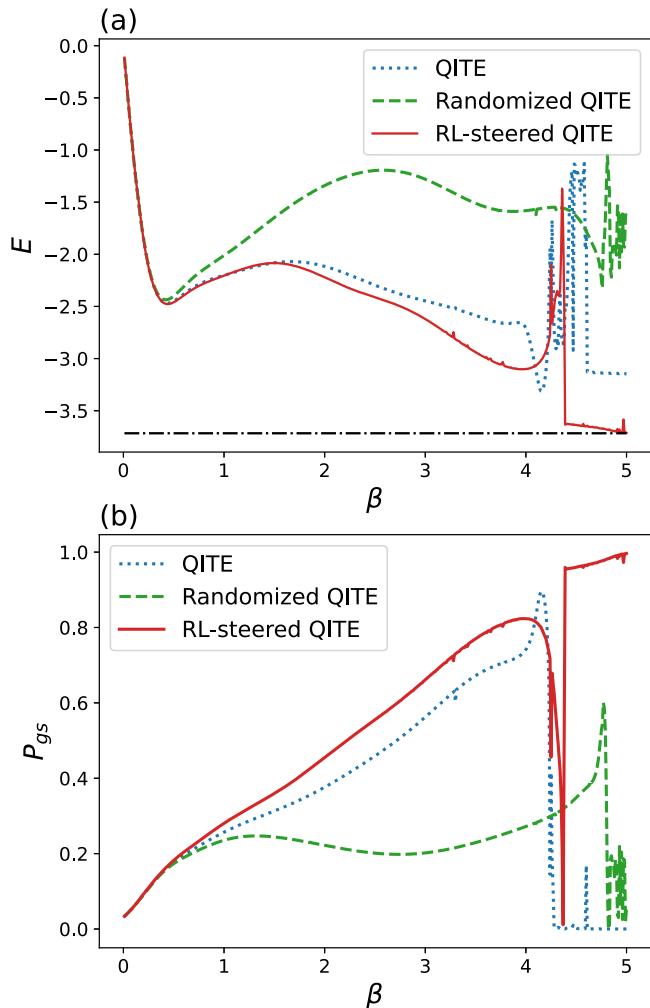


Fig. 4 Different QITE schemes for the Sherrington–Kirkpatrick model. Blue dotted lines represent the standard quantum imaginary time evolution (QITE); green dashed lines represent the randomized QITE; red lines represent the reinforcement learning (RL)-steered QITE. **a** Energy versus the imaginary time β . The black dash-dotted line represents the ground state energy. **b** Success probability versus the imaginary time β .

method is a winning combination of machine learning and quantum computing. Even though we investigated only relatively small systems, the scheme can be directly extended to larger systems. The number of neurons in the output layer of the RL agent only grows linearly with system size N . A relevant problem worth considering is how to apply the QITE (or the RL-steered QITE) for preparing a quantum state that is k -UDP or k -UDA but not k -UGS.

RL has a bright prospect in the NISQ era. In the future, one may use RL to enhance Trotterized/variational quantum simulation^{59–62} similarly, but the reward function design will be more challenging. Near-term quantum computing and classical machine learning methods may benefit each other in many ways. Their interplay is worth studying further.

Methods

The RL process is essentially a finite Markov decision process⁶³. This process is described as a state-action-reward sequence, 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 $p(s_{t+1}; R_{t+1}|s_t; a_t)$. In a finite Markov decision process, the state set, the action set, and the reward sets are finite. The total

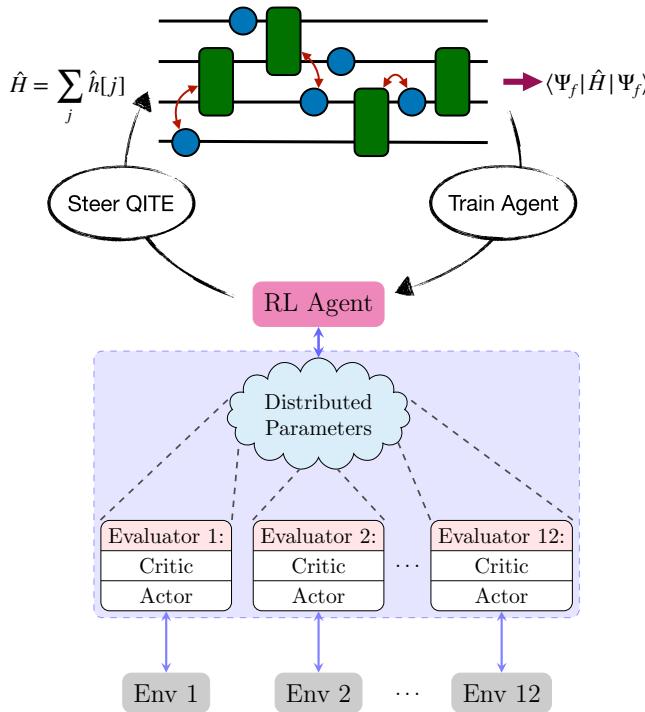


Fig. 5 Schematic of steering the quantum imaginary time evolution with reinforcement learning. The colored symbols represent single-qubit (blue dots) and two-qubit (green rectangles) non-unitary operations $\{e^{-i\Delta\tau\hat{h}[j]}\}$. The reinforcement learning (RL) agent, realized by neural networks, interacts with different environments (Env 1, Env 2, ..., Env 12) and optimizes the operation order to minimize the output state energy.

discounted return at time t is

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{t+k+1}, \quad (10)$$

where γ is the discount rate and $0 \leq \gamma \leq 1$.

The goal of RL is to maximize the total discounted return for each state and action selected by the policy π , which is specified by a conditional probability of action a for each state s , denoted as $\pi(a|s)$.

In this work, we use distributed proximal policy optimization (DPPO)^{64,65}, a model-free reinforcement learning algorithm with the actor-critic architecture. The agent has several distributed evaluators, and each evaluator consists of two components: an actor-network that computes a policy π , according to which the actions are probabilistically chosen; a critic-network that computes the state value $V(s)$, which is an estimate of the total discounted return from state s and the following policy π . Using multiple evaluators can break the unwanted correlations between data samples and make the training process more stable. The RL agent updates the neural network weights synchronously. For more details of DPPO see Supplementary Note 7.

The objective of the agent is to maximize the cumulative reward under a parameterized policy π_θ :

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

In our task, the environment state is the ordering of local terms in each Trotter step, and the state space size is $(m!)^n$. The agent observes the full state space at the learning stage, i.e., we deal with a fully observed Markov process. We define the action set by whether or not to exchange two adjacent operations in Eq. (2). Note that even if two local terms $h[j]$ and $h[j+1]$ commute, their unitary approximations $e^{-i\Delta\tau\hat{A}[j]}$ and $e^{-i\Delta\tau\hat{A}[j+1]}$ are state-dependent and may not commute. The ordering of commuting local terms still matters. For a local Hamiltonian with m terms, there are $m-1$ actions for each Trotter step. Any permutation on m elements can be decomposed to product of $\mathcal{O}(m^2)$ adjacent transpositions. Therefore our action set is universal, and a well-trained agent can rapidly steer the ordering to the optimum. The agent takes actions in sequence, the size of the action-space is $2^{n(n-1)}$. A deep neural network with $n(m-1)$ output neurons determines the action probabilities. We iteratively train the agent from measurement results on the output state $|\Psi_f\rangle$, the agent updates the path to maximize its total reward. Figure 5 shows the diagram of our method.

The reward of the agent received in each step is

$$R_t = \begin{cases} 0, & t \in \{0, 1, \dots, N_d - 1\} \\ \mathcal{R}, & t = N_d \end{cases} \quad (12)$$

where N_d is the time delay to get the reward, \mathcal{R} is the modified reward function. In particular, we define \mathcal{R} as

$$\mathcal{R} = \begin{cases} -1, & \text{if } E \leq E_{\text{std}} \\ -1/\log[\text{clip}((E/E_{\text{std}} - 1), 0.01, 1.99)], & \text{otherwise} \end{cases} \quad (13)$$

where E is the output energy given by our RL-steered path, E_{std} is the energy given by the standard repetition path without optimization. In order to avoid divergence of the reward, we use a clip function to clip the value of $(E/E_{\text{std}} - 1)$ within a range (0.01, 1.99).

Data availability

Generated data that support the plots are available at <https://github.com/Plmono/RL-qite>.

Code availability

Source codes are available at <https://github.com/Plmono/RL-qite>.

Received: 28 September 2021; Accepted: 17 February 2022;

Published online: 14 March 2022

References

1. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
2. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Rev.* **41**, 303–332 (1999).
3. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009).
4. Kandala, A. et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **549**, 242–246 (2017).
5. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. Preprint at <https://arxiv.org/abs/1411.4028> (2014).
6. Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
7. Bharti, K. et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **94**, 015004 (2022).
8. Romero, J., Olson, J. P. & Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quantum Sci. Technol.* **2**, 045001 (2017).
9. Bondarenko, D. & Feldmann, P. Quantum autoencoders to denoise quantum data. *Phys. Rev. Lett.* **124**, 130502 (2020).
10. Cao, C. & Wang, X. Noise-assisted quantum autoencoder. *Phys. Rev. Appl.* **15**, 054012 (2021).
11. Wiersema, R. et al. Exploring entanglement and optimization within the hamiltonian variational ansatz. *PRX Quantum* **1**, 020319 (2020).
12. Cao, C. et al. Energy extrapolation in quantum optimization algorithms. Preprint at <https://arxiv.org/abs/2109.08132> (2021).
13. McArdle, S. et al. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Inf.* **5**, 75 (2019).
14. Motta, M. et al. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nat. Phys.* **16**, 205–210 (2020).
15. Yeter-Aydeniz, K., Pooser, R. C. & Siopsis, G. Practical quantum computation of chemical and nuclear energy levels using quantum imaginary time evolution and lanczos algorithms. *npj Quantum Inf.* **6**, 63 (2020).
16. Albash, T. & Lidar, D. A. Adiabatic quantum computation. *Rev. Mod. Phys.* **90**, 015002 (2018).
17. Zeng, J., Cao, C., Zhang, C., Xu, P. & Zeng, B. A variational quantum algorithm for hamiltonian diagonalization. *Quantum Sci. Technol.* **6**, 045009 (2021).
18. Kamakari, H., Sun, S.-N., Motta, M. & Minnich, A. J. Digital quantum simulation of open quantum systems using quantum imaginary time evolution. *PRX Quantum* **3**, 010320 (2022).
19. Sun, S.-N. et al. Quantum computation of finite-temperature static and dynamical properties of spin systems using quantum imaginary time evolution. *PRX Quantum* **2**, 010317 (2021).
20. Aharonov, D. & Touati, Y. Quantum circuit depth lower bounds for homological codes. Preprint at <https://arxiv.org/abs/1810.03912> (2018).
21. Chen, J., Ji, Z., Zeng, B. & Zhou, D. L. From ground states to local hamiltonians. *Phys. Rev. A* **86**, 022339 (2012).
22. Smith, A., Kim, M. S., Pollmann, F. & Knolle, J. Simulating quantum many-body dynamics on a current digital quantum computer. *npj Quantum Inf.* **5**, 106 (2019).

23. Nishi, H., Kosugi, T. & Matsushita, Y.-i. Implementation of quantum imaginary-time evolution method on nisq devices by introducing nonlocal approximation. *npj Quantum Inf.* **7**, 85 (2021).

24. Gomes, N. et al. Efficient step-merged quantum imaginary time evolution algorithm for quantum chemistry. *J. Chem. Theory Comput.* **16**, 6256–6266 (2020).

25. Gomes, N. et al. Adaptive variational quantum imaginary time evolution approach for ground state preparation. *Adv. Quantum Technol.* **4**, 2100114 (2021).

26. Ville, J.-L. et al. Leveraging randomized compiling for the QITE algorithm. Preprint at <https://arxiv.org/abs/2104.08785> (2021).

27. Silver, D. et al. Mastering the game of go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).

28. Silver, D. et al. Mastering the game of go without human knowledge. *Nature* **550**, 354 (2017).

29. Silver, D. et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science* **362**, 1140–1144 (2018).

30. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529 (2015).

31. Vinyals, O. et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature* **575**, 350–354 (2019).

32. Agostinelli, F., McAleer, S., Shmakov, A. & Baldi, P. Solving the rubik's cube with deep reinforcement learning and search. *Nat. Mach. Intell.* **1**, 356–363 (2019).

33. Bukov, M. et al. Reinforcement learning in different phases of quantum control. *Phys. Rev. X* **8**, 031086 (2018).

34. Niu, M. Y., Boixo, S., Smelyanskiy, V. N. & Neven, H. Universal quantum control through deep reinforcement learning. *npj Quantum Inf.* **5**, 33 (2019).

35. 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.* **5**, 85 (2019).

36. An, Z. & Zhou, D. L. Deep reinforcement learning for quantum gate control. *EPL* **126**, 60002 (2019).

37. An, Z., Song, H.-J., He, Q.-K. & Zhou, D. L. Quantum optimal control of multilevel dissipative quantum systems with reinforcement learning. *Phys. Rev. A* **103**, 012404 (2021).

38. Yao, J., Kötterling, P., Gundlach, H., Lin, L. & Bukov, M. Noise-robust end-to-end quantum control using deep autoregressive policy networks. Preprint at <https://arxiv.org/abs/2012.06701> (2020).

39. Khairy, S., Shaydulin, R., Cincio, L., Alexeev, Y. & Balaprakash, P. Learning to optimize variational quantum circuits to solve combinatorial problems. In *Proc. AAAI Conference on Artificial Intelligence* 2367–2375 (2020).

40. Fösel, T., Niu, M. Y., Marquardt, F. & Li, L. Quantum circuit optimization with deep reinforcement learning. Preprint at <https://arxiv.org/abs/2103.07585> (2021).

41. Ostaszewski, M., Trenkwalder, L., Masarczyk, W., Scerri, E. & Dunjko, V. Reinforcement learning for optimization of variational quantum circuit architectures. In *Advances in Neural Information Processing Systems* <https://arxiv.org/abs/2103.16089> (2021).

42. Wauters, M. M., Panizon, E., Mbeng, G. B. & Santoro, G. E. Reinforcement-learning-assisted quantum optimization. *Phys. Rev. Res.* **2**, 033446 (2020).

43. Yao, J., Lin, L. & Bukov, M. Reinforcement learning for many-body ground-state preparation inspired by counterdiabatic driving. *Phys. Rev. X* **11**, 031070 (2021).

44. Lin, J., Lai, Z. Y. & Li, X. Quantum adiabatic algorithm design using reinforcement learning. *Phys. Rev. A* **101**, 052327 (2020).

45. Jerbi, S., Trenkwalder, L. M., Poulsen Nautrup, H., Briegel, H. J. & Dunjko, V. Quantum enhancements for deep reinforcement learning in large spaces. *PRX Quantum* **2**, 010328 (2021).

46. Saggio, V. et al. Experimental quantum speed-up in reinforcement learning agents. *Nature* **591**, 229–233 (2021).

47. Childs, A. M., Ostrander, A. & Su, Y. Faster quantum simulation by randomization. *Quantum* **3**, 182 (2019).

48. Campbell, E. Random compiler for fast hamiltonian simulation. *Phys. Rev. Lett.* **123**, 070503 (2019).

49. Vandersypen, L. M. K. & Chuang, I. L. NMR techniques for quantum control and computation. *Rev. Mod. Phys.* **76**, 1037–1069 (2005).

50. Khaneja, N., Reiss, T., Kehlet, C., Schulte-Herbrüggen, T. & Glaser, S. J. Optimal control of coupled spin dynamics: design of NMR pulse sequences by gradient ascent algorithms. *J. Magn. Reson.* **172**, 296–305 (2005).

51. Cory, D. G., Fahmy, A. F. & Havel, T. F. Ensemble quantum computing by NMR spectroscopy. *Proc. Natl Acad. Sci. USA* **94**, 1634–1639 (1997).

52. Lee, J.-S. The quantum state tomography on an NMR system. *Phys. Lett. A* **305**, 349–353 (2002).

53. Koczor, B. Exponential error suppression for near-term quantum devices. *Phys. Rev. X* **11**, 031057 (2021).

54. Huggins, W. J. et al. Virtual distillation for quantum error mitigation. *Phys. Rev. X* **11**, 041036 (2021).

55. Sherrington, D. & Kirkpatrick, S. Solvable model of a spin-glass. *Phys. Rev. Lett.* **35**, 1792–1796 (1975).

56. Mézard, M., Parisi, G. & Virasoro, M. A. *Spin Glass Theory and Beyond: An Introduction to the Replica Method and Its Applications* (World Scientific Publishing Company, 1987).

57. Farhi, E., Goldstone, J., Gutmann, S. & Zhou, L. The quantum approximate optimization algorithm and the sherrington-kirkpatrick model at infinite size. Preprint at <https://arxiv.org/abs/1910.08187> (2021).

58. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 4812 (2018).

59. Sieberer, L. M. et al. Digital quantum simulation, trotter errors, and quantum chaos of the kicked top. *npj Quantum Inf.* **5**, 78 (2019).

60. Benedetti, M., Fiorentini, M. & Lubasch, M. Hardware-efficient variational quantum algorithms for time evolution. *Phys. Rev. Res.* **3**, 033083 (2021).

61. Barison, S., Vicentini, F. & Carleo, G. An efficient quantum algorithm for the time evolution of parameterized circuits. *Quantum* **5**, 512 (2021).

62. Bolens, A. & Heyl, M. Reinforcement learning for digital quantum simulation. *Phys. Rev. Lett.* **127**, 110502 (2021).

63. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT Press, 1998).

64. Heess, N. et al. Emergence of locomotion behaviours in rich environments. Preprint at <https://arxiv.org/abs/1707.02286> (2017).

65. Schulman, J., Wolski, F., Dhariwal, P., Radford, A. & Klimov, O. Proximal policy optimization algorithms. Preprint at <https://arxiv.org/abs/1707.06347> (2017).

Acknowledgements

The authors thank Marin Bukov, Jiahao Yao, Qihao Guo, and Stefano Barison for helpful suggestions. The authors would also like to thank the anonymous reviewers for their constructive feedback. C.C. and B.Z. are supported by General Research Fund (No. GRF/16300220). Z.A. and D.L.Z. are supported by NSF of China (Grants No. 11775300 and No. 12075310), the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB28000000), and the National Key Research and Development Program of China (Grant No. 2016YFA0300603).

Author contributions

C.C. initiated the idea, wrote part of the code, analyzed the numerical and experimental data; Z.A. wrote the code of reinforcement learning and trained the neural networks; S.-Y.H. carried out the NMR experiments; D.L.Z. and B.Z. directed the project; C.C. and Z.A. wrote the manuscript with feedback from all authors.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at <https://doi.org/10.1038/s42005-022-00837-y>.

Correspondence and requests for materials should be addressed to D. L. Zhou or Bei Zeng.

Peer review information *Communications Physics* thanks Evert van Nieuwenburg and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Reprints and permission information is available at <http://www.nature.com/reprints>

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.