

Neural network agent playing spin Hamiltonian games on a quantum computer

Oleg M. Sotnikov^{1,*} and Vladimir V. Mazurenko¹

¹*Theoretical Physics and Applied Mathematics Department,
Ural Federal University, 620002 Ekaterinburg, Russia*

(Dated: October 25, 2019)

Quantum computing is expected to provide new promising approaches for solving the most challenging problems in material science, communication, search, machine learning and other domains. However, due to the decoherence and gate imperfection errors modern quantum computer systems are characterized by a very complex, dynamical, uncertain and fluctuating computational environment. We develop an autonomous agent effectively interacting with such an environment to solve magnetism problems. By using the reinforcement learning the agent is trained to find the best-possible approximation of a spin Hamiltonian ground state from self-play on quantum devices. We show that the agent can learn the entanglement to imitate the ground state of the quantum spin dimer. The experiments were conducted on quantum computers provided by IBM. To compensate the decoherence we use local spin correction procedure derived from a general sum rule for spin-spin correlation functions of a quantum system with even number of antiferromagnetically-coupled spins in the ground state. Our study paves a way to create a new family of the neural network eigensolvers for quantum computers.

Reinforcement machine learning techniques were initially developed for creating autonomous intelligent robotic systems [1]. Currently, they are being successfully applied in completely different decision making domains such as games [2–4], traffic control systems [5], computer resources management [6], news recommendation systems [7], optimization of chemical reactions [8] and others. Within a reinforcement learning approach an agent taking some actions interacts with environment, receives feedback, estimates rewards and corrects its actions to increase a future reward. This idea is very attractive, since in such a formulation the agent is fully autonomous and can develop different strategies to gain more. However, a practical realization of a reinforcement learning technique is problem specific and requires additional innovations providing the stability and convergence of numerical schemes.

Reinforcement learning techniques have been actively developed and implemented in such a new field of research as quantum computing. It includes quantum-error-correction systems in complex quantum devices [9–12], design and implementation of quantum communication technologies [13], quantum gate control [14–17], quantum gate design [18], quantum algorithms for reducing computational error [19]. It is important to note that only a few of these algorithms were tested on real quantum devices.

Motivated by recent results of Google DeepMind team [2] obtained for classic Atari games in this work we develop and practically realize a reinforcement learning scheme for approximating the ground states of spin Hamiltonians on quantum computers. In this field of quantum computing there are two approaches widely used to simulate magnetic systems. The first one is an

adiabatic simulation method [20–23] that is based on the discretization (Trotterization) of the time evolution operator. Another one is a variational quantum eigensolver proposed in Refs. 24–26 that uses the Ritz’s variational principle to prepare approximations of the ground state of a magnetic model. Being different by the construction both approaches assume to use some fixed sequences of quantum gates.

Practical applications of these methods for the simplest quantum spin Hamiltonian on real home-made and public quantum devices [21, 23, 27, 28] have revealed problems that are mainly related to the decoherence and gate errors. For instance, the experiments aimed to spin Hamiltonian dynamics [23] have demonstrated that such errors become more and more significant as the length of the quantum program increases. As a result, a few Trotter steps lead to a considerable error in the experimental data in comparison with exact results. On the other hand, the variational procedure [24] requires a calibration of gradient-descent method parameters to probe the energy landscape in the vicinity of the state defined with current set of parameters. These additional measurements are also source for errors.

In our study we follow a distinct logic and consider a spin Hamiltonian problem as a game with the following rules. Starting with a random quantum state a player performs several quantum actions and measurements to get the best score that means the lowest energy and, as a result, the best approximation of the spin Hamiltonian ground state. To play this game we develop a multi-neural-network agent that determines a sequence of quantum gates for a short quantum circuit. In contrast to previous approaches we do not use a fixed sequence of quantum gates, and at each iteration the agent chooses a new gate for quantum circuit depending on the current state of a quantum device on the basis of the calculated correlation functions. During the training process the agent writes short quantum programs and runs them on

* E-mail for correspondence: oleg.sotn@gmail.com

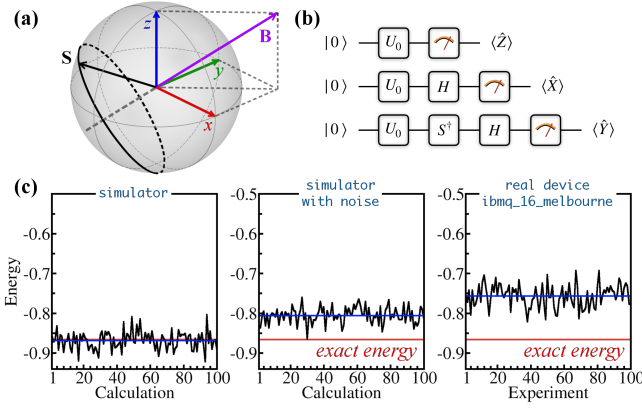


FIG. 1. (a) Schematic visualization of a quantum spin in the magnetic field $\mathbf{B} = (1,1,1)$. (b) Quantum circuits used to calculate the correlation functions and energy of the system. (c) The energies obtained from 100 independent calculations on the quantum simulator (left and central panels) and 100 experiments on the real quantum device (right panel). The number of measurement shots was equal to 1024. Blue lines correspond to the average energies. The energy of the exact solution depicted with red line is equal to $E_0 = -0.866$.

a simulator with noise. Here we apply the best latest expertise in the field of the reinforcement learning [2]. For instance, we use the experience replay mechanism repeatedly presenting the past experiences to its learning algorithm and an iterative update procedure adjusting the action-values (Q) towards target values. They provide the stability of the whole scheme running on the noisy quantum systems that are simulator and real device. Having trained the agent on the quantum simulator by using the developed reinforcement learning technique we demonstrate its performance on real IBM Quantum Experience devices.

1-QUBIT PROBLEM

We start with description of a single-spin Hamiltonian problem to explain details of our approach. The Hamiltonian is given by $\hat{H} = \mathbf{B}\hat{\mathbf{S}}$, where \mathbf{B} is the external magnetic field and $\hat{\mathbf{S}}$ is the spin- $\frac{1}{2}$ operator. The components of the external magnetic field were chosen to be $\mathbf{B} = (1,1,1)$. The solution of the problem can be obtained with universal $U_3(\theta, \phi, 0)$ gate that acts on a qubit in the initial state, $|0\rangle$. Here $\theta \approx 2.186276$ rad and $\phi = -\frac{3\pi}{4}$. It gives the ground state $|\Psi_0\rangle = U_3|0\rangle$. The definitions of the gates we use are given in the Supplementary Material [29].

For the considered single spin problem the energy is given by

$$E = \frac{1}{2}(B^x \langle \hat{X} \rangle + B^y \langle \hat{Y} \rangle + B^z \langle \hat{Z} \rangle), \quad (1)$$

where $\langle \hat{Z} \rangle$, $\langle \hat{X} \rangle$ and $\langle \hat{Y} \rangle$ are the correlation functions calculated by using the probabilities of the basis states,

which is a standard output of a quantum computer. It is important to note that these correlators estimated with real quantum computer are subjected by decoherence and

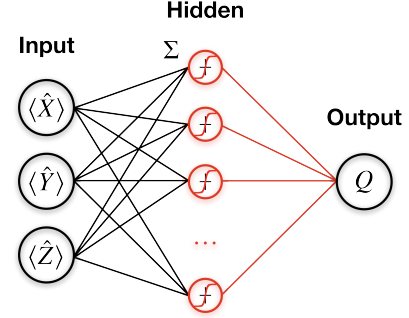


FIG. 2. Schematic representation of the neural network we use to define the action-value for each gate.

gate imperfection errors. The simulations of these errors is an active field of research [30–32]. To take these effects into account we used a noise generator as implemented in Qiskit Aer module [33]. It includes gate and readout errors imitating the real quantum device noise approximated as a relaxation process of qubits involved in the experiment. Despite the noise model of the local simulator is simple and takes into account only local errors occurring on each gate, we will show that inclusion of device noise even on this level significantly improves the agreement between results of simulations and experiments obtained with real device. We used a basic device noise model with parameters reported in official Qiskit tutorials for IBM Q 16 Melbourne device [34].

With circuits presented in Fig. 1 we have performed 100 independent calculations on simple quantum simulator, 100 calculations on the simulator with noise and 100 experiments on the real quantum device. In all the cases we observe fluctuations of the measurements results. For instance, for the simple simulator the energy fluctuates around exact value. The account of decoherence and gate imperfections with the noise model leads to a higher average energy that is about -0.8. Real experimental results obtained with IBM Q 16 Melbourne device are characterized by very strong fluctuations of the energy around -0.75.

These experiments clearly demonstrate that even for the simplest problem one deals with complex, dynamical, uncertain and fluctuating quantum computing environment. It motivates to develop an autonomous agent effectively interacting with such an environment.

Neural network agent

The agent we develop is multi-network one in accordance with a one-action-one-network concept proposed in Ref. 1. There is a separate network for each action, but the structures of all the networks are the same (Fig. 2). They contain input, one hidden and output layers. The

network takes spin correlation functions obtained with quantum computer or simulator as an input. The number of the input neurons depends on the problem we solve. For instance, in the case of the one-qubit problem (single spin in an external magnetic field) there are three correlation functions that form a state vector $\mathbf{s} = \{\langle \hat{X} \rangle, \langle \hat{Y} \rangle, \langle \hat{Z} \rangle\}$ characterizing the quantum system in question. In turn, there will be 6 single-spin and 9 spin-spin correlation functions (15 in total) in the case of the dimer spin Hamiltonian problem.

The number of the neurons in the hidden layer also depends on the size of the problem. In the one- and two-qubit cases we consider the number of the hidden neurons is equal to 32 and 64, respectively. The output layer contains only one neuron that represents the predicted action-value Q for the particular quantum gate (action). Having compared the calculated Q -values among all the networks the agent chooses the gate for which action-value function has the maximal value. In this way one

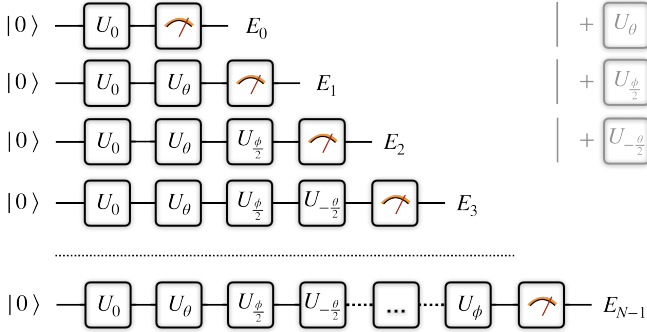


FIG. 3. Step-by-step construction of a quantum circuit of N gates. $U_0 = U_3^{rand}$ denotes preparation of a random initial state. Each line corresponds to adding new gate chosen by the agent from the action list described in the text. E_t represents the energy estimated at the t th step of the circuit formation.

defines an optimal Q -function satisfying [1]

$$Q(\mathbf{s}, a) = r + \gamma \max\{Q(\mathbf{s}', k), k \in \text{actions}\}. \quad (2)$$

According to this expression the utility of an action a in response to a state \mathbf{s} equals to immediate reward r plus the best utility that can be obtained from the next state \mathbf{s}' discounted by factor γ . During reinforcement learning, the difference between the two sides of Eq. (2) is to be minimized using a back-propagation algorithm [29].

Quantum computer programming

The developed agent is aimed to define a quantum circuit according to the predicted action values Q . In the case of the one-qubit problem we use the following set of gates: identity gate, U_θ , U_ϕ , $U_{-\theta}$, $U_{-\phi}$, $U_{\theta/2}$, $U_{\phi/2}$, $U_{-\theta/2}$ and $U_{-\phi/2}$ that are defined with the universal rotation gate U_3 as described in Ref. 29.

The process of the quantum circuit construction can be demonstrated by the example of Fig. 3. We start with an universal U_3 gate with random angles. Having added a new gate to the quantum circuit the measurements of the correlation functions are performed. It means that one can trace the energy evolution as the length of the quantum circuit increases.

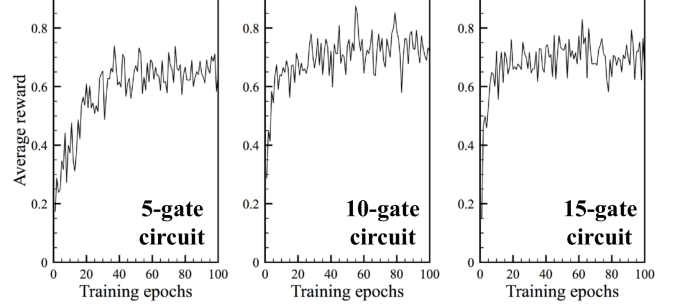


FIG. 4. Training results for the simplest problem, single spin in the magnetic field, $\mathbf{B}=(1, 1, 1)$. Average reward achieved per episode after the agent is run with ϵ -greedy policy ($\epsilon=0.05$).

Agent's training

The agent was trained with the IBM QE simulator including the noise model. Each training iteration contains three main steps. (i) The agent takes some action following ϵ -greedy policy. Having added a new gate to the quantum circuit the agent estimates the reward from the observation, $r = E_t - E_{t+1}$. (ii) The sequence (state, action, reward, new state) explored by the agent is stored in the replay memory. (iii) A sequence randomly chosen from the memory is used to optimize the weights of the neural network. A complete description of technical details on the agent's training is presented in Ref. 29. From Fig. 4 it follows that the 10-gate agent has demonstrated the best average reward during the training.

Experiments

To examine the performance of the trained agent we performed experiments on real quantum device provided by IBM. Fig. 5 demonstrates these results on the level of the individual real quantum device experiments. One can see that the agent decreases the energy of the system starting from different random states and approaches to the $E = -0.75$ defined as the average energy for the circuit (Fig. 1b) simulating the exact solution of the problem. If the energy of the initial random state is low enough and close to the ground state the agent follows a passive strategy trying to keep such a winning position. On the other hand, a high-energy initial state is a signal to the

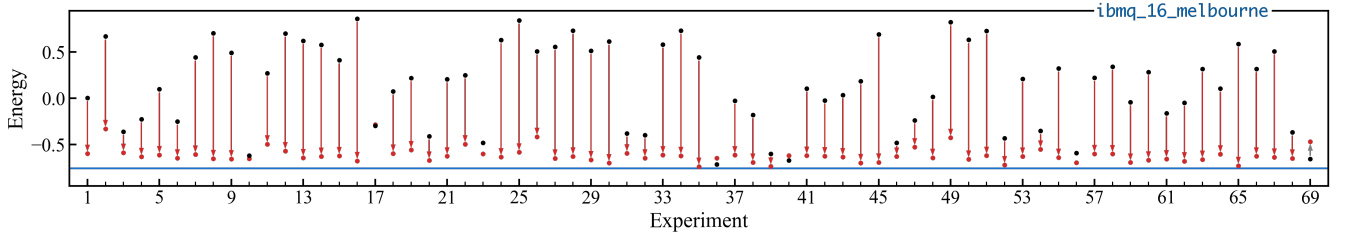


FIG. 5. Performance demonstrated by the trained agent in experiments on the IBM Q 16 Melbourne. Black and red circles denote initial and final energies for each 10-gate circuit constructed by the agent. Red and grey arrows indicate lowering and rising of energy with the corresponding circuits. Blue line shows the average energy obtained with exact solution quantum circuits presented in Fig. 1 on the real device. The number of measurement shots was equal to 1024.

agent to decrease the energy as much as possible from the current position. More results and discussions can be found in Supplementary Material [29].

SPIN DIMER

Having discussed the single-qubit problem we are in position to consider the agent's training in the two-qubit case. The corresponding spin model contains two antiferromagnetically-coupled spins, $\hat{H} = J\hat{\mathbf{S}}_1\hat{\mathbf{S}}_2$. Here J is isotropic exchange interaction. The ground state of this model is the singlet state $|\Psi_0\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$. Such an entanglement state can be realized on a quantum computer with the circuit presented in Fig. 6 (a). However, the experiments on the IBM Q Vigo (Fig. 6 (b)) have shown significant contributions of the triplet states, $|\uparrow\uparrow\rangle$ and $|\downarrow\downarrow\rangle$. As a result the experimental ground state energy of -0.51 is considerably higher than the exact value of -0.75. It also explains disagreement between experiment and theory on ground state of the Heisenberg model defined on single square reported in Ref. 24.

Another important observation is that an experiment conducted on the real IBM Q Vigo device gives the correlation functions that differ from the data of other experiments, which means that there are fluctuations of the correlation functions and energies within the series of independent experiments (Fig. 6 (c)). Moreover, sets of the experiments conducted with the same circuits but at different periods of time can give different average energies [29].

In the two-qubit case the action list for the agent includes the same gates as for single-qubit agent and additional Controlled NOT (CNOT) gate that is responsible for the entanglement [29]. The agent was trained starting with different random classical ground states. Such a non-entangled state is formed in the following way. The initial state of the first qubit is changed with random U_3 gate. Then the state of the second qubit is set to be antiparallel to the first one on the level of the Bloch sphere. Interestingly, the agent has learned from self-play the possibility to overcome the classical energy limit by using the CNOT gate. A specific structure of the CNOT

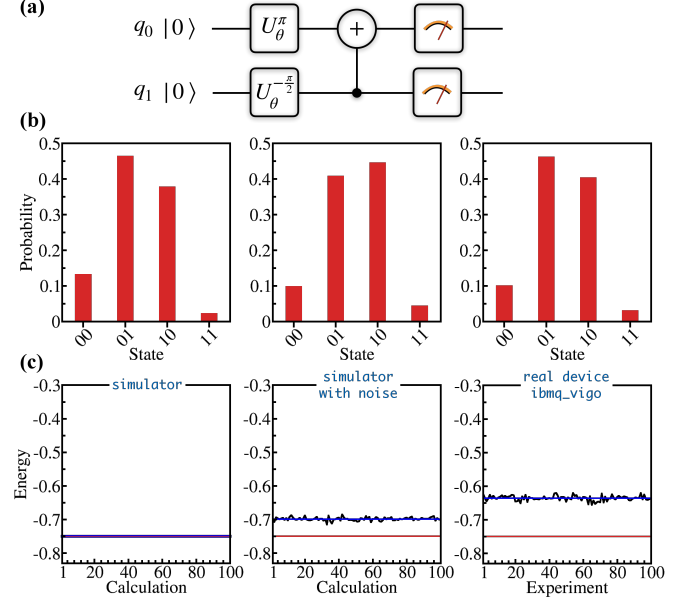


FIG. 6. (a) Quantum circuit realizing singlet ground state for spin- $\frac{1}{2}$ dimer. Here $U_\theta^\pi = U_3(\pi, 0, 0)$ and $U_\theta^{-\frac{\pi}{2}} = U_3(-\frac{\pi}{2}, 0, 0)$. (b) Probabilities of the basis states obtained from three independent experiments conducted on the real quantum device. (c) The energies obtained from 100 independent simulations on the quantum simulator (left and central panels) and 100 experiments on the real quantum device, IBM Q Vigo (right). The energy of exact solution denoted with red line is -0.75.

matrix implies an initial preparation by using single qubit rotations to decrease further the total energy with the corresponding entanglement gate.

In Fig. 7 (a) we compare the rewards obtained within the training processes performed with different values of the elementary rotation angle, $\delta = 0.5, 1.0$ and 1.5 rad (see Supplementary Material for further details on the rotation angle). One can see that the best and more stable results were obtained with $\delta = 1$. In general the choice of the particular angle value can be also considered as a part of the reinforcement learning algorithm we propose. We left a practical realization of such an option for a future investigation.

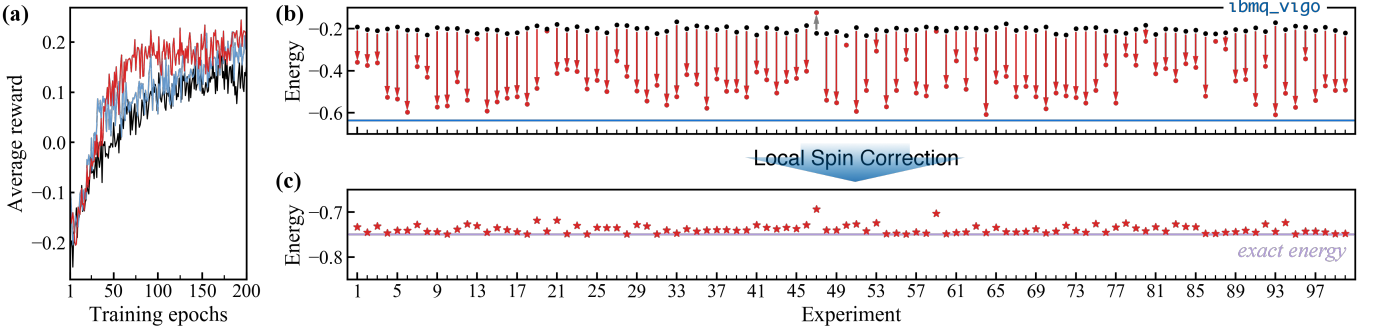


FIG. 7. (a) The average reward achieved per circuit after the two-qubit agent is run with ϵ -greedy policy ($\epsilon=0.05$) on the quantum simulator with noise. Black, red and blue curves denote training results obtained with elementary rotational angles of 0.5, 1.0 and 1.5 rad, respectively. (b) Performance of the trained agent demonstrated in experiments on the IBM Q Vigo device. Red arrows indicate lowering of energy with the corresponding circuits. Blue line shows the average energy (Fig. 6(c)) obtained with singlet state quantum circuit on the real device. The number of measurement shots was equal to 1024. (c) Ground state energies obtained on the real device after local moment correction.

Figure 7(b) shows that the classical energy of the spin dimer estimated with quantum device is about -0.2, which is higher than the exact solution of -0.25. The trained two-qubit agent decreases the energy of the system to the level of -0.6 (Fig. 7(b)) that was obtained with the singlet state circuit.

The average values denoted with blue lines in Fig. 5 (Fig. 1(c)) and Fig. 7(b) (Fig. 6(c)) were obtained with the shortest quantum circuits (one-gate circuit for single-spin problem and three-gate circuit for dimer problem) imitating the exact solution. In the case of the reinforcement learning technique the agent constructs circuits of the 10-gate length. Thus, the neural network results obtained with longer circuits closely approach to the level of the short circuits data, which is an additional demonstration of the performance of our method.

It is also important to analyze the evolution of the energy obtained with neural network during individual experiments. Some examples given in Fig. 8 demonstrate that the largest energy decreases are mainly achieved with the CNOT gate which can be implemented not only in the beginning of the circuit construction. To keep a minimal energy the agent uses the rotation gates from the action list with smallest angles, such as $\frac{\phi}{2}$. It can be seen by the examples of the experiments 6, 18 and 21.

Local spin correction

As it was shown above, the direct imitation of the single ground state as well as its neural network approximation on the real quantum device are characterized by the energies which are considerably larger than the exact ground state energy. Such a disagreement between experiment and theory is mainly related to decoherence and gate errors. In this situation it is important to find a way to compensate such hardware imperfections. The different strategies can be used for that. The authors of Ref. 35 have simulated time-dependent spin-spin cor-

relation functions of the Heisenberg-type magnetic systems in high magnetic fields. At these magnetic fields the quantum ground state of the spin system is fully polarized. To fit the experimental results obtained with an IBM quantum computer to exact ones the authors introduced a phase-and-scale procedure that is based on an artificial phase correction of the local spin-spin correlation function at the zero time and application of the same correction to the whole time domain. Below we show that for the quantum systems in the singlet ground state a similar correction procedure can be derived from a general sum rule for the spin-spin correlation functions.

For the quantum systems characterized by a singlet-type ground state we propose a local moment correction procedure that systematically improves the agreement between experiment and theory on the ground state energy. The ground state spin-spin correlation functions of a quantum spin system with even number of spins and antiferromagnetic interaction between them satisfies to the following sum rule

$$\sum_{ij} \langle \hat{\mathbf{S}}_i \hat{\mathbf{S}}_j \rangle = 0. \quad (3)$$

The sum in this equation contains local and non-local contributions that can be decomposed. It gives

$$\sum_i \langle (\hat{\mathbf{S}}_i)^2 \rangle = - \sum_{i \neq j} \langle \hat{\mathbf{S}}_i \hat{\mathbf{S}}_j \rangle. \quad (4)$$

The sum in the right part contains all possible pair correlation functions in the system. Namely these non-local correlation functions are calculated at each iteration of our algorithm with quantum computer and used as input for neural network. Within the correction procedure we use them to estimate the local correlator in the left part of Eq. (4) that should be compared with the exact value $S(S+1)$. Namely, this exact value is important insider information for us on unknown quantum system. The

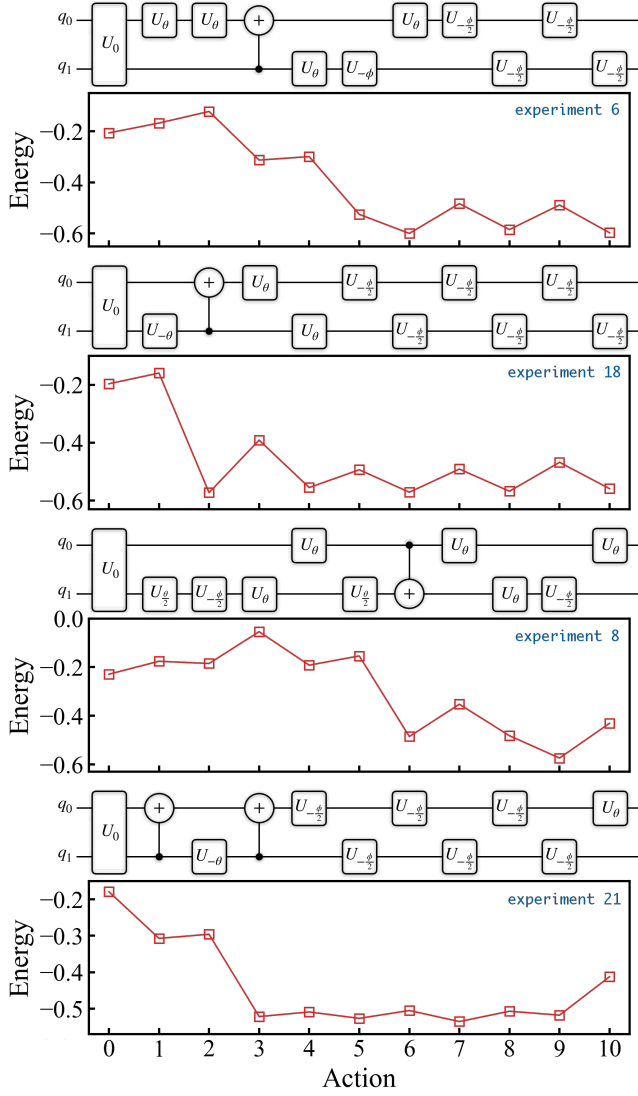


FIG. 8. Examples of quantum system energy evolution during some experiments presented in Fig. 7. These 10-gate circuits were constructed gate-by-gate by the neural network agent trained to approximate the spin Hamiltonian ground state.

ratio

$$X = \frac{S(S+1)}{\frac{1}{N} \sum_i \langle \hat{\mathbf{S}}_i^2 \rangle} \quad (5)$$

can be used to correct the non-local correlation functions $X \langle \hat{\mathbf{S}}_i \hat{\mathbf{S}}_j \rangle$ (N is the number of spins). It gives opportunity to correct the estimated energy of the Heisenberg system. In the case of the dimer system we have a trivial situation since the nonlocal spin-spin correlation functions $\langle \hat{\mathbf{S}}_1 \hat{\mathbf{S}}_2 \rangle$ and $\langle \hat{\mathbf{S}}_2 \hat{\mathbf{S}}_1 \rangle$ fully define the on-site spin-spin correlators $\langle \hat{\mathbf{S}}_1^2 \rangle$ and $\langle \hat{\mathbf{S}}_2^2 \rangle$. The modified experimental results are presented in Fig. 7(c). One can see that the most part of the experiments gave very accurate estimation of the exact energy of the spin dimer.

The procedure we propose is of general nature and can be used not only in the case of the neural network solver

presented in this work but also in conjunction with other quantum computer eigensolvers.

Comparison with variational quantum eigensolver

Since we propose a new neural network eigensolver it is important to implement previously developed quantum computer approaches to the same ground state problem of the Heisenberg dimer and compare the performance of these methods. As we have discussed in the introduction there are two standard quantum computer eigensolvers, they are variational quantum algorithm and phase estimation method. The comparison of these methods can be found in Ref. 26 in which the authors performed quantum computer experiments and computed the energy of hydrogen molecule. For the quantum computer experiments this electronic structure problem is formulated in a form of a 2-spin Hamiltonian similar to that we consider in this work. It was demonstrated that the variational quantum approach outperforms the phase estimation algorithm. That is why we perform the comparison the former method with our neural network solver by the example of the spin dimer model. For that a realization of the variational approach reported in Ref. 24 was used. Within this approach the wave function of a quantum Heisenberg model is stored on a quantum device and represented in the following form

$$|\Psi(\theta)\rangle = U_{\text{CNOT}} U_3(\theta) |00\rangle, \quad (6)$$

where $|00\rangle$ is the initial state of the 2-qubit system, U_3 is a set of rotation gates, U_{CNOT} is the CNOT gate that is responsible for the entanglement and θ is the set of angles we vary to approximate the ground state of the spin Hamiltonian. The angles at the $(k+1)$ th iteration are defined as

$$\theta_{k+1} = \theta_k - \alpha_k \mathbf{g}_k(\theta_k), \quad (7)$$

here $\mathbf{g}_k(\theta_k)$ is the gradient at θ_k and α_k is the step size parameter. This variational scheme contains a number of parameters for which we use the same values as it was proposed in Ref. 24. Every 20 steps we perform a calibration procedure to probe the energy landscape at the quantum state defined by the current set of the rotation angles and to renew the gradient-descent parameters.

The results obtained by using the variational approach with simulator, simulator with noise and real quantum device are presented in Fig. 9. In the case of the ideal quantum simulator we observe excellent agreement between the calculated energies and exact results. The average energy estimated with noisy simulator is -0.7, which is larger than the exact solution of -0.75. In turn, the real quantum device IBM Q Vigo gives the energy of -0.6. It agrees with neural network results (Fig. 7(b)). The real experiments are characterized by fluctuations of the energy after 200 iterations. They are due to large values of the step size α in Eq. (7) obtained with calibration

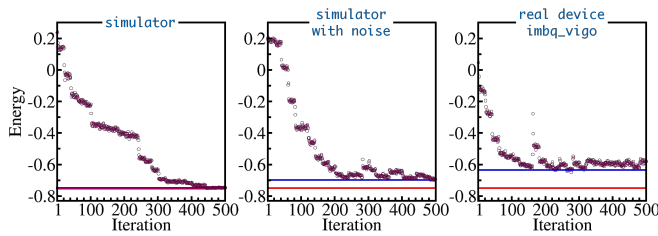


FIG. 9. Results of calculations and experiments obtained variational quantum eigensolver realized on the basis of work [24].

procedure. Here, one of the possible solutions is to limit the upper bound of α to 2.

Despite the agreement on real device results of variational and neural network solvers there are important differences between them. First of all the neural network results were obtained with circuits of 10 gates. At the same time there are 3 gates (two universal U_3 gates and one CNOT gate) in the variational scheme. Taking into account that gate errors accumulate as circuit length increases, our neural network approach seems to be stable to such errors. Another important difference is the initial quantum state. For neural network solver we always start with a random initial quantum state, which suggest the way to avoid local energy minima for more complicated spin Hamiltonian problems.

CONCLUSION

We have developed a neural-network agent approximating the ground state of a spin model on a quantum device. The agent was trained by reinforcement learning from self-play on quantum simulator with noise. Here the general objective of our reinforcement learning ap-

proach is to obtain the agent that can play moderately well in all the cases even on noisy real quantum devices. The agent performance was demonstrated on quantum simulator and real quantum devices. In the case of the dimer problem we found that the agent can learn entanglement by applying the CNOT gate. In combination with local spin correction our neural network approach provides excellent agreement with exact dimer solution on the ground state energy.

To consider the systems with more than two qubits, a new strategy should be adopted. For instance, one can use a single deep neural network for all the gates from the action list. In the current formulation of the neural network approach the set of observables will increase exponentially with the system size. Moreover, to compensate the decoherence we need to define all the nonlocal spin-spin correlation functions to get an accurate estimation of the ground state energy. This is the problem if we use classical computers to simulate the neural network. The problem could be solved if the network is realized with another quantum device. Recent studies [36, 37] have demonstrated that the equivalent of m -dimensional classical input and weight vectors can be encoded on the quantum hardware by using N qubits, where $m = 2^N$. Similar approach allows to exploit the exponential advantage of quantum information storage.

Another important problem is a smart selection of the implemented gates, since their number substantially grows as the system size increases. Such a selection can be also fulfilled with machine learning approach [38].

ACKNOWLEDGMENTS

This work was supported by the Russian Science Foundation, Grant No. 18-12-00185.

-
- [1] L. J. Lin, Reinforcement learning for robots using neural networks. *Technical Report, DTIC Document* (1993).
 - [2] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, *Nature* **518**, 529 (2015).
 - [3] 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).
 - [4] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, *Nature* **550**, 354 (2017).
 - [5] I. Arel, C. Liu, T. Urbanik, A.G. Kohls, *IET Intell. Transp. Syst.* **4**, 128 (2010).
 - [6] H. Mao, M. Alizadeh, I. Menache, S. Kandula, *Proceedings of the 15th ACM Workshop on Hot Topics in Networks*, <http://doi.acm.org/10.1145/3005745.3005750>.
 - [7] G. Zheng, F. Zhang, Z. Zheng, Y. Xiang, N.J. Yuan, X. Xie, Z. Li, *Proceedings of the 2018 World Wide Web Conference*, <https://doi.org/10.1145/3178876.3185994>.
 - [8] Zhenpeng Zhou, Xiaocheng Li, Richard N. Zare, *ACS Cent. Sci.* **3**, 1337 (2017).
 - [9] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, *Phys. Rev. X* **8**, 031084 (2018).
 - [10] R. Sweke, M. S. Kesselring, E. P. L. van Nieuwenburg, and J. Eisert, “Reinforcement Learning Decoders for Fault-Tolerant Quantum Computation,” (2018), arXiv:1810.07207.
 - [11] Y. H. Liu, and D. Poulin, “Neural Belief-Propagation Decoders for Quantum Error-Correcting Codes,” (2018), arXiv:1811.07835.
 - [12] H. P. Nautrup, N. Delfosse, V. Dunjko, H. J. Briegel, and N. Friis, “Optimizing Quantum Error Correction Codes with Reinforcement Learning,” (2018),

- arXiv:1812.08451.
- [13] Julius Walln fer, Alexey A. Melnikov, Wolfgang D r, Hans J. Briegel, arXiv:1904.10797.
 - [14] Zheng An, D. L. Zhou, EPL, 126, 60002 (2019).
 - [15] Murphy Yuezhen Niu, Sergio Boixo, Vadim Smelyanskiy, Hartmut Neven, arXiv:1803.01857.
 - [16] Moritz August, Jos  Miguel Hern ndez-Lobato, arXiv:1802.04063.
 - [17] Marin Bukov, Alexandre G.R. Day, Dries Sels, Phillip Weinberg, Anatoli Polkovnikov, Pankaj Mehta, Phys. Rev. X 8, 031086 (2018).
 - [18] Luca Innocenti, Leonardo Banchi, Alessandro Ferraro, Sougato Bose, Mauro Paternostro, arXiv:1803.07119.
 - [19] Lukasz Cincio, Yiit Suba, Andrew T. Sornborger, Patrick J. Coles, New J. Phys. 20, 113022 (2018).
 - [20] S. Lloyd, Science **273**, 1073 (1996).
 - [21] Z. Li, H. Zhou, C. Ju, H. Chen, W. Zheng, D. Lu, X. Rong, C. Duan, X. Peng, and J. Du, Phys. Rev. Lett. **112**, 220501 (2014).
 - [22] M. Steffen, W. van Dam, T. Hogg, G. Breyta, and I. Chuang, Phys. Rev. Lett. **90**, 067903 (2003).
 - [23] A. A. Zhukov, S. V. Remizov, W. V. Pogosov, and Y. E. Lozovik, Quantum Information Processing **17**, 223 (2018).
 - [24] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, Nature **549**, 242 (2017).
 - [25] D. Wecker, M. B. Hastings, and M. Troyer, Phys. Rev. A **92**, 042303 (2015).
 - [26] P. J. J. OMalley, R. Babbush, I. D. Kivlichan, J. Romero, J. R. McClean, R. Barends, J. Kelly, P. Roushan, A. Tranter, N. Ding, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, A. G. Fowler, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, C. Quintana, D. Sank, A. Vainsencher, J. Wenner, T. C. White, P. V. Coveney, P. J. Love, H. Neven, A. Aspuru-Guzik, J. M. Martinis, Phys. Rev. X 6, 031007 (2016).
 - [27] IBM Quantum Experience.
 - [28] M. Hebenstreit, D. Alsina, J. I. Latorre, and B. Kraus, Phys. Rev. A **95**, 052339 (2017).
 - [29] Supplementary Material.
 - [30] Christina Knapp, Michael Beverland, Dmitry I. Pikulin, Torsten Karzig, Quantum 2, 88 (2018).
 - [31] A. Aharony, O. Entin-Wohlman, D. Chowdhury, S. Dattagupta, J. Stat. Phys. 175, 704 (2019).
 - [32] Nairi Usher, Dan E. Browne, arXiv:1704.07298.
 - [33] https://quantum-computing.ibm.com/jupyter/tutorial/advanced/aer/2_device_noise_simulation.ipynb
 - [34] Qiskit tutorials: <https://nbviewer.jupyter.org/github/Qiskit/qiskit-tutorial/blob/master/index.ipynb>
 - [35] A. Chiesa, F. Tacchino, M. Grossi, P. Santini, I. Tavernelli, D. Gerace, S. Carretta, Nature Physics **15**, 455 (2019).
 - [36] Francesco Tacchino, Chiara Macchiavello, Dario Gerace, Daniele Bajoni, npj Quantum Information 5, 26 (2019).
 - [37] M. Schuld, M. Fingerhuth, F. Petruccione, EPL 119, 60002 (2017).
 - [38] A. A. Melnikov, H. P. Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, Proceedings of the National Academy of Sciences **115**(6), 1221-1226 (2018).

Supplemental Material: Neural network agent playing spin Hamiltonian games on a quantum computer

TRAINING ALGORITHM

In this section we present technical details concerning the reinforcement learning approach. The basic steps of the algorithm we use is given in the listing below

```

Initialize replay memory  $D$  with capacity  $M$ 
Initialize action-value function  $Q$  with random weights  $\xi_0(a_k)$  for each action gate  $a_k$ 
Initialize target action-value function  $\hat{Q}$  with weights  $\xi^-(a_k) = \xi_0(a_k)$ 
For circuit = 1, Num_circuits do
  Add random  $U_3$  gate, perform the measurement and initialize  $s_1$ 
  For t = 1, Num_gates do
    With probability of  $\epsilon$  select a random gate  $a_t$ 
    otherwise select  $a_t = \operatorname{argmax}_a Q(s_t, a; \xi(a))$ 
    Add gate  $a_t$  to the circuit, perform the measurement, form  $s_{t+1}$  and estimate the reward  $r_t$ 
    Store the transition  $(s_t, a_t, s_{t+1}, r_t)$  in  $D$ 
    Sample random minibatch storing a transition  $(s_\tau, a_\tau, s_{\tau+1}, r_\tau)$  from  $D$ 


$$y_\tau = \begin{cases} r_\tau & \text{if circuit terminates at step } \tau+1 \text{ (Num_gates}=\tau+1) \\ r_\tau + \gamma \max_{a'} \hat{Q}(s_{\tau+1}, a'; \xi^-(a')) & \text{otherwise} \end{cases}$$


    Perform gradient descent step on  $(y_\tau - Q(s_\tau, a_\tau; \xi_t(a_\tau)))^2$  with respect to  $\xi_t(a_\tau)$ 
    Every  $C$  steps reset  $\xi^-(a_k) = \xi_t(a_k)$ 
  End for
End for

```

The parameters used for training the neural networks are described in Table I

TABLE I. Parameters of the reinforcement learning used to train the agent.

Parameter	Value	Description
replay memory size, M	32	Number of minibatches storing the transitions (s_t, a_t, s_{t+1}, r_t)
global learning rate, α	0.05	the learning rate used in (S5)
target network update frequency, C	500	Number of parameters updates with which target state is updated
number of hidden neurons	32 (64)	Neuron number for one-qubit (two-qubit) problems
discount factor, γ	0.99	discount factor in (S1)
Num_circuits	100	Number of the constructed circuits for one epoch
Num_gates	5, 10, 15	Maximal number of gates in one circuit
initial exploration	1	Initial value of ϵ in ϵ -greedy exploration
final exploration	0.05	Final value of ϵ in ϵ -greedy exploration
final exploration step	$10 \times \text{Num_gates}$	Number of measurements over which the initial value of ϵ is annealed to its final value

The ϵ -greedy policy described in Ref. 1 was used in our study. With probability ϵ the agent selects a random gate and with probability $1-\epsilon$ it follows greedy policy $a_t = \operatorname{argmax}_a Q(s_t, a; \xi_t(a))$. The latter expression means that for specific system state s_t the agent will choose a gate for which the neural network gives the largest action value function Q .

Following the work 1 we introduce two sets of weights for each action, ξ and ξ^- at each iteration. Thus, for specific action a_i and specific iteration t we use the notation $\xi_t(a_k) = \{W_{ij}^h(a_k), W_i^o(a_k)\}_t$. The loss function at t th iteration is given by the following expression

$$L_t(\xi_t(a_k)) = [r + \gamma \max_{a'} \hat{Q}(s_{\tau+1}, a'; \xi^-(a')) - Q(s_\tau, a_k, \xi_t(a_k))]^2 \quad (\text{S1})$$

Since the values of the output neurons are in the range from -1 to 1, we clip y_τ to be between -1 and 1.

NEURAL NETWORK TRAINING

As an input for neural networks we use the correlation functions calculated with quantum simulator with noise emulation. The input and output of the hidden layer neurons are given by

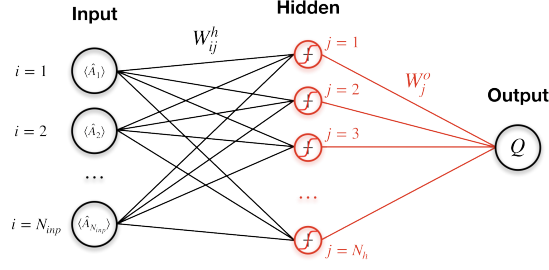


FIG. S1. Schematic representation of constructed neural network with single hidden layer. We used sigmoid as an activation function of hidden and output neurons. All the notations are described in the text below.

$$h_j^{out} = \frac{1}{1 + e^{-h_j^{inp}}}, \quad h_j^{inp} = \sum_{i=1}^{N_{inp}} s_i W_{ij}^h, \quad (S2)$$

where s_i is the value of i th input neuron (i th correlation function), W_{ij}^h — weight between the i th input neuron and j th hidden neuron, N_{inp} — number of the input neurons which is equal to number of the correlation functions calculated with quantum computer. The value of the output neuron is calculated in a standard way by using the following equations

$$o^{out} = 2 \left(\frac{1}{1 + e^{-o^{inp}}} - 0.5 \right), \quad o^{inp} = \sum_{j=1}^{N_h} h_j^{out} W_j^o, \quad (S3)$$

where N_h is the number of hidden neurons, W_j^o — weight between the j th hidden neuron and output neuron. Here we use symmetric version of the sigmoid function.

The error function is given by

$$E[\xi_t(a_k)] = (o^{target} - o^{actual}[\xi_t(a_k)])^2, \quad (S4)$$

where o^{target} and o^{actual} are the target and actual output value of the tailing neuron, (S3).

Due to the fact that we optimized our network through back-propagation method by means of the gradient descent we used the following expression for new weights

$$W(t+1) = W(t) + \alpha \cdot g \cdot \nabla_{\xi_t} E, \quad (S5)$$

where α and g are the global and individual learning rates, respectively. With the gain parameter g we control the individual learning rates for each weight of each network. The gain are constrained to lie in the range $[0.1, 2]$. The local gain is increased, $g(t) = g(t-1) + 0.05$ if the gradient for that weight does not change the sign, $\nabla_{\xi_t} E(t) \nabla_{\xi_t} E(t-1) > 0$. Otherwise, $g(t) = g(t-1) \cdot 0.95$.

The weights gradients were calculated by the following equations

$$\frac{\partial E}{\partial W_j^o} = \delta o \cdot h_j^{out} \quad (S6)$$

and

$$\frac{\partial E}{\partial W_{ij}^h} = \delta h_j^{out} \cdot s_i, \quad (S7)$$

where

$$\delta o = (o^{target} - o) \cdot (o + 1) \cdot \left(\frac{1}{2} - \frac{o}{2} \right), \quad (S8)$$

$$\delta h_j^{out} = h_j^{out} (1 - h_j^{out}) W_j^o \delta o. \quad (S9)$$

CALCULATION OF THE ENERGY

To define the energy of the system in question we use the correlation functions that can be calculated through the probabilities of the different basis states, P . The latter is a standard output of a quantum computer. In a one-qubit case the correlation function $\langle Z \rangle = P(0) - P(1)$, where $P(0)$ and $P(1)$ are the probabilities of the state 0 and 1, respectively. To measure the $\langle X \rangle$ and $\langle Y \rangle$ one is to rotate the standard basis frame to lie along the corresponding axis, and to make a measurement in the standard basis [2]. For instance, for the $\langle X \rangle$ correlator we implement Hadamard (Fig. 1, main text) gate before the measurement gate.

In the case of the dimer problem we are interested in the $\langle Z_1 Z_2 \rangle$, $\langle X_1 X_2 \rangle$, $\langle Y_1 Y_2 \rangle$, $\langle X_1 Y_2 \rangle$, $\langle Y_1 X_2 \rangle$, $\langle X_1 Z_2 \rangle$, $\langle Z_1 X_2 \rangle$, $\langle Y_1 Z_2 \rangle$, $\langle Z_1 Y_2 \rangle$ correlation functions, since we use them as input for our neural network agent. The zz correlator is to be defined by using the following combination of the basis states probabilities: $\langle Z_1 Z_2 \rangle = P(00) + P(11) - P(01) - P(10)$. To determine other correlation functions one needs to perform the rotations of the standard basis frame as described above.

COMPARISON OF THE SIMULATOR WITH NOISE AND REAL QUANTUM DEVICE

To examine the performance of the trained one-qubit agent we performed experiments on quantum simulator and real quantum device provided by IBM Quantum Experience. During these experiments the agent was approximating ground state of the single-spin Hamiltonian. Here we used the neural networks weights obtained for 10-gate agent demonstrating the best average reward. The results obtained with 5, 10 and 15 gates quantum circuits are presented in Fig. S2. In analogy with the Google's agent playing classical Atari games, we detect the best score that is the minimal energy in our case (Fig. S2 right panel). Remarkably, in all the cases the energies obtained by the agent on the real device and simulator are close to that estimated with quantum circuit simulating the exact solution. It confirms importance of the noise model we used for realistic simulation of the experimental data.

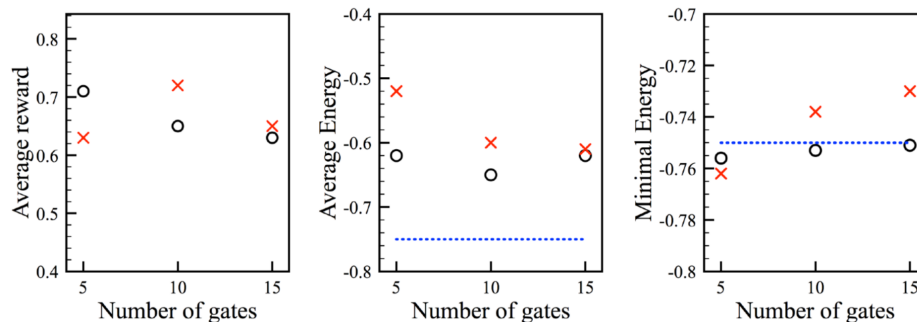


FIG. S2. Performance of the trained one-qubit agent estimated with noisy quantum simulator (black circles) and real quantum device (red crosses). Left, center and right panels represent the average reward, average energy and minimal energy obtained with quantum circuits of 5, 10 and 15 gates. The dashed blue line denotes the average energy obtained with circuits corresponding to the exact solution of the single spin problem.

STABILITY OF THE QUANTUM COMPUTER MEASUREMENTS OVER TIME

Here, by the example of the dimer problem we give an additional demonstration that the agent explores a very complex and dynamical computational environment. The results of the real experiments conducted with singlet state circuit (Fig. 6, main text) are presented in Fig. S3. One can see that the average energies estimated on different days vary from -0.55 to -0.4.

GATE DESCRIPTION

Assuming that the state of n th qubit is written in form:

$$|\phi_n\rangle = \cos \frac{\theta_n}{2} |0\rangle + \cos \frac{\theta_n}{2} e^{-i\phi_n} |1\rangle \quad (\text{S10})$$

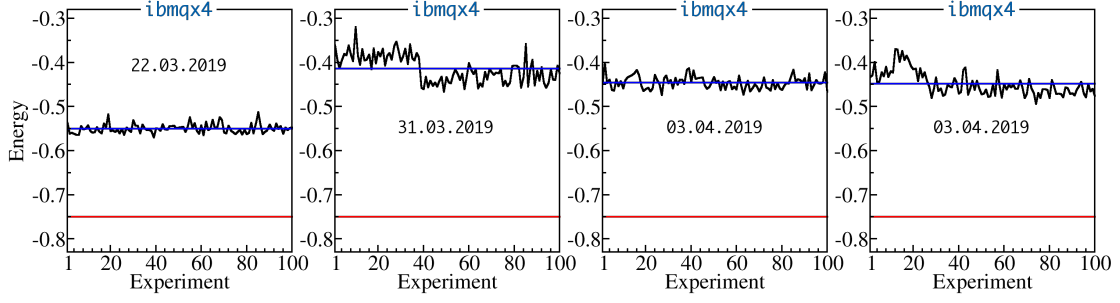


FIG. S3. Energies obtained from the real device experiments conducted on different days. In all the experiments the same quantum circuit simulating singlet state (Fig. 6, main text) was used. The blue lines denote the average energy for each set of the experiments. The red line corresponds to the exact ground state energy for the spin dimer.

TABLE II. List of one- and two-qubit gates used in this work.

Action	Gate	Description	Action	Gate	Description
U_0	$U_3(\delta^{rand}, \phi^{rand}, \lambda^{rand})$	rotation by random angle	\oplus	$CNOT$	exclusive NOT (creation of entanglement)
I	$IDLE$	do nothing			
U_ϕ	$U_3(0, \delta, 0)$	rotation around Z axis by δ	U_θ	$U_3(\delta, 0, 0)$	rotation around Y axis by δ
$U_{-\phi}$	$U_3(0, -\delta, 0)$	rotation around Z axis by $-\delta$	$U_{-\theta}$	$U_3(-\delta, 0, 0)$	rotation around Y axis by $-\delta$
$U_{\frac{\phi}{2}}$	$U_3(0, \frac{\delta}{2}, 0)$	rotation around Z axis by $\frac{\delta}{2}$	$U_{\frac{\theta}{2}}$	$U_3(\frac{\delta}{2}, 0, 0)$	rotation around Y axis by $\frac{\delta}{2}$
$U_{-\frac{\phi}{2}}$	$U_3(0, -\frac{\delta}{2}, 0)$	rotation around Z axis by $-\frac{\delta}{2}$	$U_{-\frac{\theta}{2}}$	$U_3(-\frac{\delta}{2}, 0, 0)$	rotation around Y axis by $-\frac{\delta}{2}$

one can represent the generic operator U_3 as the following:

$$U_3(\theta, \phi, \lambda) = \begin{pmatrix} \cos \frac{\theta}{2} & e^{-i\lambda} \sin \frac{\theta}{2} \\ e^{i\phi} \sin \frac{\theta}{2} & e^{i\lambda+i\phi} \cos \frac{\theta}{2} \end{pmatrix} \quad (S11)$$

The gates that can be used by the one- and two-qubit agents are listed in Table II. We have found that the optimal value of δ that yields the best reward during calculation is equal to 0.5 and 1 rad for single spin and dimer problems respectively.

-
- [1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, Nature **518**, 529 (2015).
[2] IBM Quantum Experience.