quantum reports

CITESCORE 2.4

Article

Model-Free Deep Recurrent Q-Network Reinforcement Learning for Quantum Circuit Architectures Design

Tomah Sogabe, Tomoaki Kimura, Chih-Chieh Chen, Kodai Shiba, Nobuhiro Kasahara, Masaru Sogabe and Katsuyoshi Sakamoto

MDPI

https://doi.org/10.3390/quantum4040027

quantum reports

MDPI

Article

Model-Free Deep Recurrent Q-Network Reinforcement Learning for Quantum Circuit Architectures Design

Tomah Sogabe [1,2,3,*], Tomoaki Kimura [1], Chih-Chieh Chen [2,*], Kodai Shiba [1,2], Nobuhiro Kasahara [1], Masaru Sogabe [2] and Katsuyoshi Sakamoto [1,3]

[1] Engineering Department, The University of Electro-Communications, Tokyo 182-8585, Japan
[2] Grid Inc., Tokyo 107-0061, Japan
[3] i-Powered Energy Research Center (i-PERC), The University of Electro-Communications, Tokyo 182-8585, Japan
* Correspondence: sogabe@uec.ac.jp (T.S.); chen.chih.chieh@gridsolar.jp (C.-C.C.)

Abstract: Artificial intelligence (AI) technology leads to new insights into the manipulation of quantum systems in the Noisy Intermediate-Scale Quantum (NISQ) era. Classical agent-based artificial intelligence algorithms provide a framework for the design or control of quantum systems. Traditional reinforcement learning methods are designed for the Markov Decision Process (MDP) and, hence, have difficulty in dealing with partially observable or quantum observable decision processes. Due to the difficulty of building or inferring a model of a specified quantum system, a model-free-based control approach is more practical and feasible than its counterpart of a model-based approach. In this work, we apply a model-free deep recurrent Q-network (DRQN) reinforcement learning method for qubit-based quantum circuit architecture design problems. This paper is the first attempt to solve the quantum circuit design problem from the recurrent reinforcement learning algorithm, while using discrete policy. Simulation results suggest that our long short-term memory (LSTM)-based DRQN method is able to learn quantum circuits for entangled Bell–Greenberger–Horne–Zeilinger (Bell–GHZ) states. However, since we also observe unstable learning curves in experiments, suggesting that the DRQN could be a promising method for AI-based quantum circuit design application, more investigation on the stability issue would be required.

Keywords: quantum circuits; reinforcement learning; Q-learning; LSTM

check for updates

Citation: Sogabe, T.; Kimura, T.; Chen, C.-C.; Shiba, K.; Kasahara, N.; Sogabe, M.; Sakamoto, K. Model-Free Deep Recurrent Q-Network Reinforcement Learning for Quantum Circuit Architectures Design. *Quantum Rep.* **2022**, *4*, 380–389. https://doi.org/10.3390/quantum4040027

Academic Editor: Lucas Lamata

Received: 31 August 2022
Accepted: 19 September 2022
Published: 21 September 2022

1. Introduction

Recent advances in artificial intelligence (AI) and Noisy Intermediate-Scale Quantum (NISQ) technology produce new perspectives in quantum artificial intelligence [1,2]. The control of quantum system by a classical agent has been studied in various settings [3–5]. Reinforcement learning (RL) [6–12] was successfully applied to control problems [11,13] of classical systems and fully observable Markov Decision Process (MDP) environments [14]. However, the control and learning of Partially Observable Markov Decision Process (POMDP) [15–19] is more difficult due to indirect access to the state information. Both planning [20] and learning [21] of POMDP are proposed. For a POMDP system, the underlying state transition is classical Markovian and is different from quantum dynamics. The quantum counterpart of POMDP, Quantum Observable Markov Decision Process (QOMDP) [22–24], was theoretically studied. Implementation of a QOMDP planning method for quantum circuits [2,25–33] is studied in a previous work [34]. Comparing to state tomography-based methods, which require an exponentially large number of measurement shots with respect to the circuit width, QOMDP-based approaches have favorable sample complexity from quantum circuits. However, an exact QOMDP planning method requires exponentially expensive classical computing. It is desirable to explore approximation methods to reduce the cost of computational resources.

Applying deep artificial neural networks for function approximations in reinforcement learning is known as deep reinforcement learning (DRL) [6]. DRL can be applied to quantum control [35–46]. Deep Q-network (DQN) [7,11] learning is a reinforcement learning method using deep artificial neural networks for the Q value function approximation. The traditional DQN method uses deep neural networks for the state-action Q-function for fully observable MDP. The deep recurrent Q-network (DRQN) method is proposed to encode the history sequence to tackle POMDP problems [47–50].

In this work, we implement a deep recurrent Q-learning agent for model-free reinforcement learning [47–50] to design quantum circuits. The DRQN is based on long short-term memory (LSTM) [51–53] networks that encode the action-observation history time-series for partially observable environments [49,50]. The fidelity achieved by the DRQN learning agent is improved over learning episodes, showing the effectiveness of the proposed algorithm. However, we also observe unstable learning curves in experiments. These observations suggest that the DRQN could be a promising method for AI-based quantum circuit design application, but more investigation on the stability issue would be required.

Many previous works for quantum control using different approaches can be found in the literature [35–46]. Borah et al. and Baum et al. [35,42] use a policy gradient. Niu et al. [37] use an on-policy method. He et al. [38] use a DQN. Bukov et al. [39] use a Q-table. Mackeprang et al. [40] use a DQN and double DQN. Zhang et al. [41] provide comparative study of Q-table, DQL, and policy gradient methods. August and Hernández-Lobato [46] use LSTM for the policy gradient. All these works [35,37–42,46] are controlled at the Hamiltonian level instead of at the circuit architecture level [34,43–45]. Kuo et al. and Pirhooshyaran and Terlaky [43,44] use a policy gradient. Ostaszewski et al. [45] use a double DQN. We note that Sivak et al.'s model-free paper [36] has several similarities and differences compared to our work. Sivak et al. applied an actor–critic policy gradient method to a quantum optical system with a continuous action space. Our work applied deep recurrent Q-learning to a qubit system with discrete action set. Both Sivak et al.'s method and our method are model-free and use LSTM. Sivak et al. use LSTM for the policy network and the value network over a continuous action space. We use LSTM for a history-dependent Q-function over a discretize action space, which is more practical for field application.

This work is organized as follows. Section 2 introduces the LSTM-based DRQN reinforcement learning method for quantum circuit architecture. Section 3 presents the simulation results. Section 4 provides some discussion. Section 5 is the conclusion.

2. Methods

2.1. MDP, POMDP, and QOMDP

A POMDP problem instance is described by a set of states S, a set of actions A, a set of observations Ω, a state transition probability P, an observation probability O, a reward function R, and a discount rate $\gamma \in [\mathbf{0}, \mathbf{1}]$. At each time step t, the agent in state $s_t \in S$ takes an action $a_t \in A$ and moves to a new state $s_{t+1} \sim P(s'|s_t, a_t)$. The agent also receives an observation $o_t \sim O(o|s_t)$, $o_t \in \Omega$ and a reward $r_t = R(s_t, a_t, s_{t+1}) \in \mathbb{R}$. The action-observation history time series is $\hbar_t = \{a_1, o_1, a_2, o_2, \ldots a_t, o_t\}$. The goal is to find a policy $\pi(a|h)$ to optimize the expected future reward $\mathbb{E}_\pi\left[\sum_{i=t}^{T} \gamma^{i-t} r_i\right]$. In contrast to the situation of MDP, a POMDP agent does not have access to the time series $\{s_t\}$.

A QOMDP problem instance is described by a Hilbert space $\mathcal{S}$, a set of action super-operators $\mathcal{A}$, a set of observations Ω, a set of reward operators $\mathcal{R}$, a discount rate $\gamma \in [0, 1]$, and an initial quantum state $|s_0\rangle$. The set of actions consists of super-operators $\mathcal{A} = \{A^{a^1}, \ldots, A^{a^{|\mathcal{A}|}}\}$, where each super-operator $A^a = \{A^a_{o^1}, \ldots, A^a_{o^{|\mathcal{O}|}}\}$ has $|\mathcal{O}|$ Kraus matrices. At each time step t, the agent takes an action a_t, which introduces a change of the state of current quantum system

$$|s_t\rangle \longmapsto \frac{A^{a_t}_{o_t}|s_t\rangle}{\sqrt{\langle s_t|A^{a_t\dagger}_{o_t}A^{a_t}_{o_t}|s_t\rangle}}$$

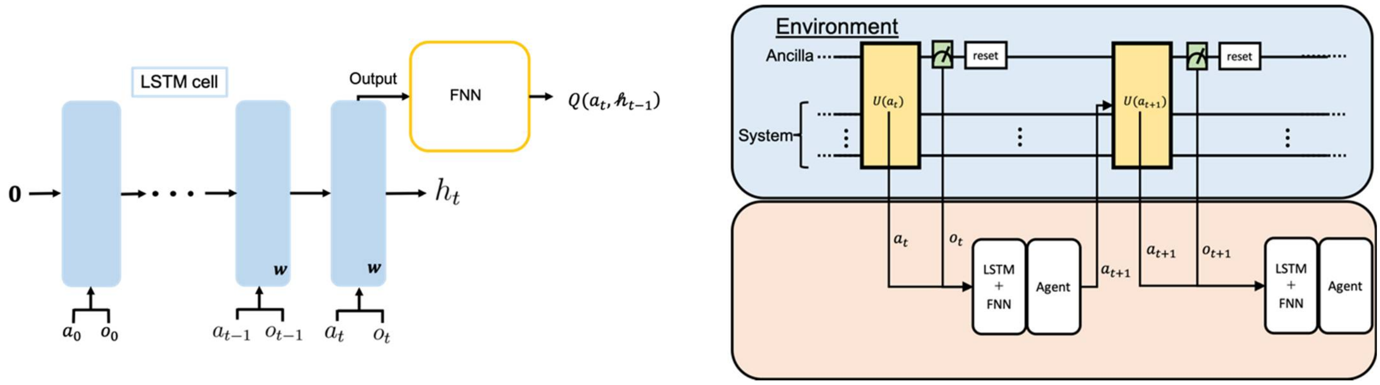
The agent also receives an observation $o_t \sim \Pr(o||s_t\rangle, a_t) = \langle s_t|A^{a_t\dagger}_{o}A^{a_t}_{o}|s_t\rangle$, $o_t \in \Omega$ and a reward $r_t = \langle s_t|R_{a_t}|s_t\rangle \in \mathbb{R}$, where $R_{a_t} \in \mathcal{R}$. Similar to POMDP and MDP, the goal is to find a policy to optimize the expected future reward.

2.2. LSTM-Based Deep Recurrent Q-Network

LSTM is a type recurrent neural network which can be used to model sequential data. The hidden state h_t and output c_t are computed by the recurrence $(h_t, c_t) = LSTM(h_{t-1}, c_{t-1}, x_{t-1})$ for time-dependent input signal x_t. Traditional Q-learning for observable MDP uses a state-action Q-function $Q(s_t, a_t)$ to represent the value of an action a_t at a known state s_t. To deal with partially observable environments in which s_t is unknown, a history-dependent Q-function $Q(a_t, \hbar_{t-1})$ is used instead of the state-action Q-function. By treating the action-observation pair as input $x_t = (a_t, o_t)$, LSTM enables the encoding of the history-dependent Q-function $Q(a_t, \hbar_{t-1})$. A feed-forward neural network (FNN) is concatenated with the LSTM output to represent the Q-function. The FNN is a simple linear transformation, and its output gives the Q-value $Q(:, \hbar_{t-1}) = Wc_{t-1} + b$, where $W \in \mathbb{R}^{|\mathcal{A}|\times|h|}$ is a trainable weight matrix, and $b \in \mathbb{R}^{|h|}$ is a bias vector. $|h|$ denotes the size of the LSTM hidden states. The LSTM–FNN structure is shown in Figure 1a. The update of the Q function is via the optimization of loss function

$$L = (Q(a_t, \hbar_{t-1}) - (r_{t-1} + \gamma \max_A Q(A, \hbar_t)))^2$$

which can be computed by back-propagation through time. The implementation is performed by using the package PyTorch [54]. The hyperparameters can be found in Table 1.



(**a**) (**b**)

Figure 1. The setting of the proposed learning algorithm. (**a**) A LSTM cell and a feed-forward neural network (FNN) are used for history Q-function approximation. (**b**) The RL environment–agent diagram.

Table 1. List of hyperparameters.

Hyperparameter	Value
Target state fidelity threshold	0.99
Maximum steps per episode	100
Number of episodes	30,000
Reply buffer size	1,000,000
Epsilon start	1.0

Table 1. *Cont.*

Hyperparameter	Value
Epsilon end	0.01
Epsilon decay rate	0.9997
LSTM sequence length	3
LSTM hidden states size	30
FNN hidden states size	30
FNN activation function	linear
Minibatch size	32
Learning rate	0.001
Soft update rate tau	0.001
Discount rate	0.95

2.3. RL Method

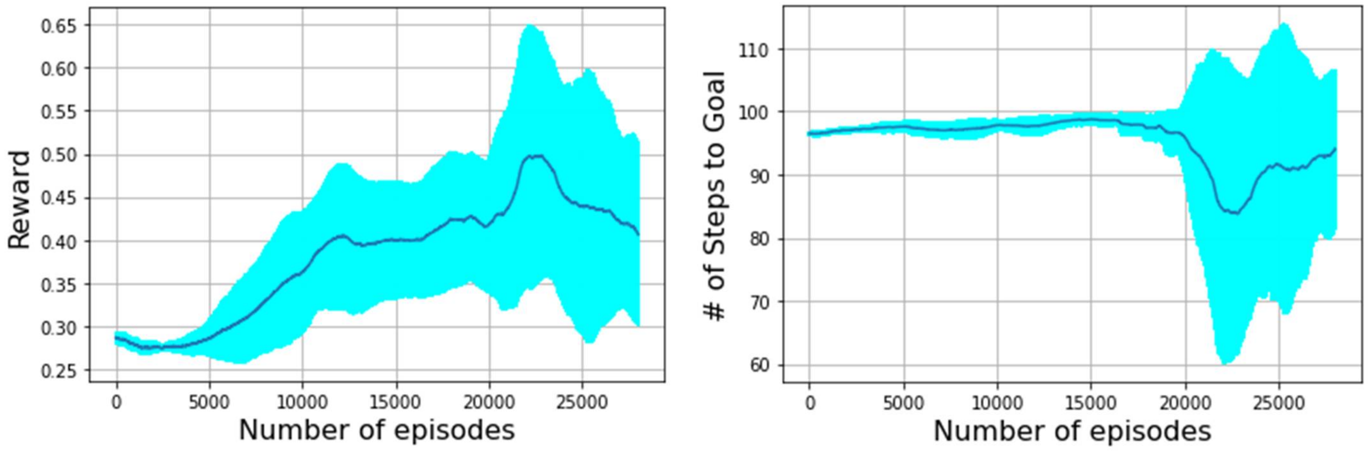
The proposed method is depicted in Figure 1b. The RL environment is the quantum circuit to be designed. The classical agent receives 0–1 observation from measurement result of the ancillary qubit. The action–observation pair is used to update the DRQN, and then the decision for the next action is made by the agent to control the circuit. The reward is the fidelity with respect to the target state $r_t = \langle s_t | s_{target} \rangle \langle s_{target} | s_t \rangle$. The policy is epsilon-greedy. Experience reply is used to stabilize the calculation. Using the convention that the Hilbert space is *ancilla* $\otimes$ *system*, and the operator in Figure 1b is $U(a_t) = U_{ent}(H \otimes U_{action})$, where H is the single qubit Hadamard gate acting on the ancillary qubit. The action unitary U_{action} is chosen from the action set $\{CNOT_{i,j} : i, j \in system\} \cup \{R_{d,i}(\theta) : i \in system, \theta \in \{\pm\frac{\pi}{9}\}, d \in \{X, Y, Z\}\}$. Here, $CNOT_{i,j}$ denotes the control-not gate, for which the i-th qubit is the control qubit and the j-th qubit is the target qubit. $R_{d,i}(\theta)$ denotes single qubit rotation of i-th qubit around d-axis. The system–ancilla entangler $U_{ent} = \prod_{i \in system} CNOT_{i,ancilla}$ computes the system parity function and outputs the result to an ancilla qubit. The setup is similar to that of [34], but the classical agent in this work is an RL agent instead of a planning agent.

3. Results

Numerical simulations are conducted to test the applicability of the proposed method. The simulation code is based on the packages Numpy [55], Matplotlib [56], PyTorch [54], and Qiskit [57]. We test the state generation task for the 2-qubit Bell state and 3-qubit Greenberger–Horne–Zeilinger (GHZ) state [58]. The target state is considered reached when the fidelity is larger than a threshold value 0.99. The maximum number of steps for each episode is set to be 100. The PyTorch hyperparameters are listed in Table 1.

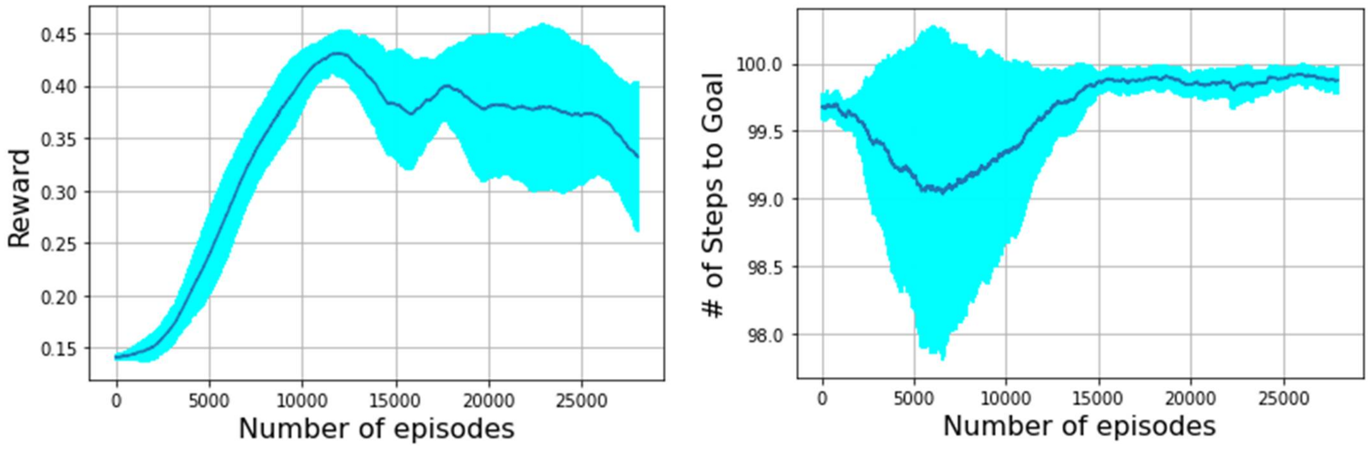
Figure 2 is the learning curves for the 2-qubit Bell state. The received reward and number of steps to reach the target state are plotted with respect to the number of learning episodes. Each curve is the moving average of 2000 episodes and 10 independent runs. The error bar denotes the one standard deviation over 10 independent runs. For 30,000 episodes, we observe that the average reward is increased from <0.3 to >0.4. The maximum of the one-sigma error bar is close to 0.65. The average number of steps to reach the goal is decreased from >95 to <90. The minimum of the one-sigma error bar is close to 60.

Figure 3 shows the learning curves for 3-qubit GHZ state. For 30,000 episodes, we observe that the reward is increased from <0.15 to >0.3. The maximum of one-sigma error bar can be larger than 0.45. The average number of steps to the goal is larger than 99 throughout the learning episodes.



(**a**) (**b**)

Figure 2. Learning curves for 2-qubit Bell state generation. Each data point is the moving average of 2000 episodes, and the average value (solid line) with one standard deviation error bar (cyan color) over 10 independent curves are reported. (**a**) Reward is plotted against number of episodes; (**b**) number of steps to reach the goal is plotted against number of episodes.



(**a**) (**b**)

Figure 3. Learning curves for 3-qubit GHZ state generation. Each data point is the moving average of 2000 episodes, and the average value (solid line) with one standard deviation error bar (cyan color) over 10 independent curves is reported. (**a**) Reward is plotted against number of episodes; (**b**) number of steps to reach the goal is plotted against number of episodes.

Figure 4 is the city diagram for the density matrix generated by the RL agent. The result is the highest fidelity result over 10 independent training runs and 100 test steps for each training obtained by the policy of the last (30,000th) training episode. The fidelity of the obtained density matrix is 0.9698 for the Bell state, and the fidelity of the obtained density matrix is 0.6710 for the GHZ state.

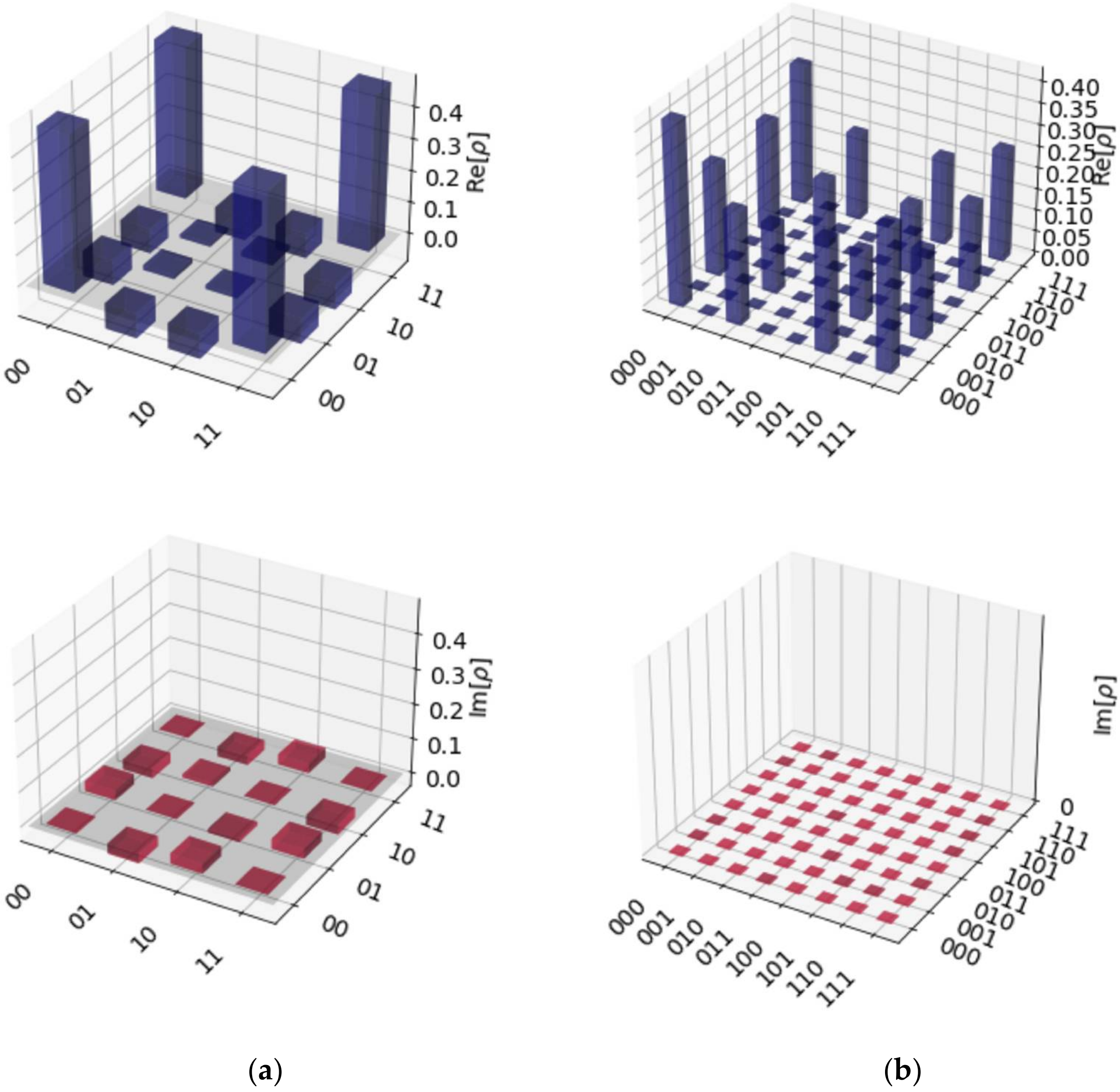


Figure 4. City diagrams for density matrices produced by the learning agent. The best result (highest fidelity) over 10 random seeds and 100 test steps of the policy obtained in the last episode is reported. (**a**) The 2-qubit Bell state experiment. The fidelity is 0.9698. (**b**) The 3-qubit GHZ state experiment. The fidelity is 0.6710.

4. Discussion

From the experimental data in Figures 2 and 3, we observe that the fidelity of the 2-qubit Bell state and 3-qubit GHZ state are improved by the proposed learning algorithm. However, since these values are mostly way below the stopping criteria 0.99, the number of steps is not improved significantly. The best output state has high fidelity with respect to the target for the 2-qubit case, while the 3-qubit case provides moderate fidelity. These results demonstrate that the learning algorithm is effective, but the performance within our experiments is not satisfactory. More learning episodes and fine-tuning of hyperparameters could potentially improve the performance. The fidelity achieved in the 2-qubit Bell experiments is generally better than that of the 3-qubit GHZ experiments. This is reasonable, since the possible action space for the 2-qubit system is smaller, and the required action sequence to produce a 2-qubit Bell state is shorter than that of a 3-qubit GHZ state.

The city diagram in Figure 4 allows us to visualize the states produced by the agent. The Bell–GHZ target state is $\frac{1}{\sqrt{2}}(|00+|11\rangle)$ for two qubits and $\frac{1}{\sqrt{2}}(|000+|111\rangle)$ for three qubits. The ideal city diagram has peaks at four corners of the real part. For the 2-qubit case, the experimental data resemble the ideal case, and, hence, the fidelity is higher. On the other hand, the 3-qubit city diagram has many sub-peaks, which implies low fidelity.

To further understand the reasons behind the limitation of our method, the test fidelity distribution histogram for 10 independent runs is plotted in Figure 5. It is observed that all samples lie in the region $Fidelity > 0.4$ for both the 2-qubit and 3-qubit cases. However, the 2-qubit result has the highest fidelity sample in the interval $Fidelity \in [0.9, 1.0)$, while the 3-qubit result has the highest fidelity sample in the interval $Fidelity \in [0.6, 0.7)$. The 2-qubit result not only has better best-case performance but also has distribution maximum at $Fidelity \in [0.6, 0.7)$. This is better than the peak location of the 3-qbuit result, which is $Fidelity \in [0.4, 0.5)$. The problem is that a learning method that is successful for small problem instances would not necessarily scale to larger problem instances. We are encountering an scalability issue that arises commonly in the application of machine learning methodologies to optimization problems [59]. To the best of our knowledge, this is still an unresolved issue in the community, so further investigation in this direction is desirable.

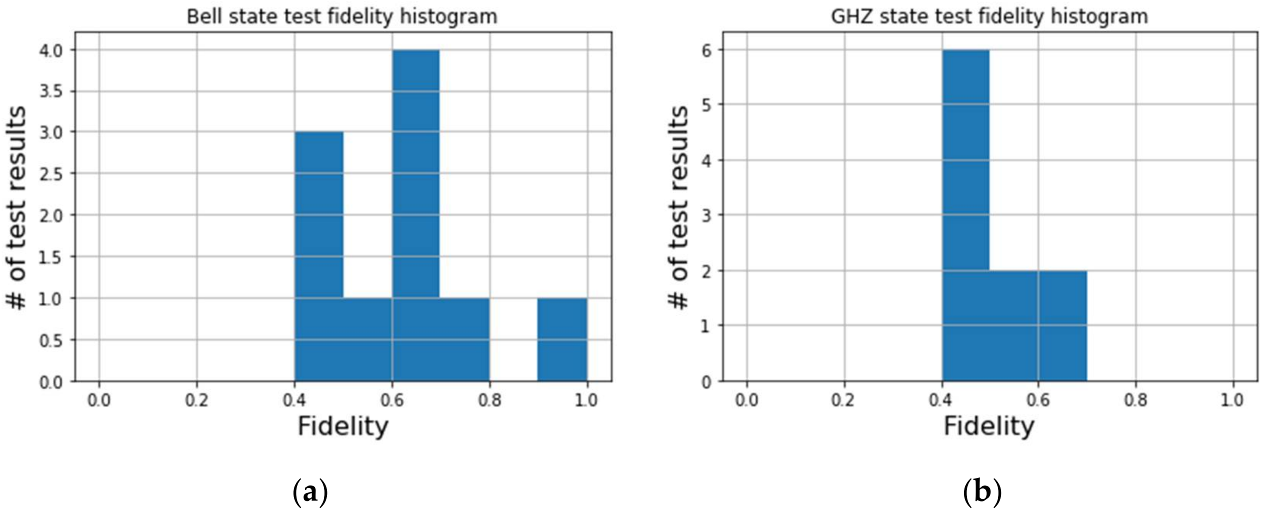


Figure 5. Histograms of maximum fidelity over 100 test steps for 10 independent samples. (**a**) The 2-qubit Bell state experiment. (**b**) The 3-qubit GHZ state experiment.

5. Conclusions

In this work, we propose and implement a deep recurrent Q-network algorithm for quantum circuit design. Experimental results show that the agent is able to learn to produce a better quantum circuit for entangled states' preparation. However, the learned fidelity is not satisfactory. Future research and development are required to improve the quality of the state-generation task. In particular, scalability to larger problem instances should be tackled. It would also be desirable to explore other applications, for example, the energy minimization task [26,34,60–62].

Author Contributions: Conceptualization, T.K., C.-C.C. and T.S.; methodology, T.K. and T.S.; software, T.K.; validation, T.K. and C.-C.C.; formal analysis, C.-C.C.; investigation, C.-C.C.; resources, K.S. (Kodai Shiba), N.K., M.S. and K.S. (Katsuyoshi Sakamoto); data curation, K.S. (Kodai Shiba), N.K. and C.-C.C.; writing, C.-C.C.; visualization, T.K. and C.-C.C.; supervision, T.S.; project administration, M.S., K.S. (Katsuyoshi Sakamoto) and T.S.; funding acquisition, M.S. and T.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data and scripts that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: We thank Naoki Yamamoto for the valuable discussions. We thank Takuma Yokoyama for the technical support.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dunjko, V.; Briegel, H.J. Machine learning & artificial intelligence in the quantum domain: A review of recent progress. *Rep. Prog. Phys.* **2018**, *81*, 074001. [CrossRef] [PubMed]
2. Preskill, J. Quantum Computing in the NISQ era and beyond. *Quantum* **2018**, *2*, 79. [CrossRef]
3. Wiseman, H.M.; Milburn, G.J. *Quantum Measurement and Control*; Cambridge University Press: Cambridge, UK, 2009; ISBN 978-0-521-80442-4.
4. Nurdin, H.I.; Yamamoto, N. *Linear Dynamical Quantum Systems: Analysis, Synthesis, and Control*, 1st ed; Springer: New York, NY, USA, 2017; ISBN 978-3-319-55199-9.
5. Johansson, J.R.; Nation, P.D.; Nori, F. QuTiP 2: A Python framework for the dynamics of open quantum systems. *Comput. Phys. Commun.* **2013**, *184*, 1234–1240. [CrossRef]
6. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; Adaptive Computation and Machine Learning Series; Bradford Books: Cambridge, MA, USA, 2018; ISBN 978-0-262-03924-6.
7. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 4th ed.; Pearson Education Limited: London, UK, 2021; ISBN 978-1-292-40113-3.
8. Szepesvari, C. *Algorithms for Reinforcement Learning*, 1st ed.; Morgan and Claypool Publishers: San Rafael, CA, USA, 2010; ISBN 978-1-60845-492-1.
9. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement Learning: A Survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [CrossRef]
10. Geramifard, A.; Walsh, T.J.; Tellex, S.; Chowdhary, G.; Roy, N.; How, J.P. A Tutorial on Linear Function Approximators for Dynamic Programming and Reinforcement Learning. *Found. Trends® Mach. Learn.* **2013**, *6*, 375–451. [CrossRef]
11. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef]
12. Watkins, C.J.C.H.; Dayan, P. Q-learning. *Mach. Learn.* **1992**, *8*, 279–292. [CrossRef]
13. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. Mastering the game of Go without human knowledge. *Nature* **2017**, *550*, 354–359. [CrossRef]
14. Bellman, R. *Dynamic Programming*; Reprint Edition; Dover Publications: Mineola, NY, USA, 2003; ISBN 978-0-486-42809-3.
15. Aoki, M. Optimal control of partially observable Markovian systems. *J. Frankl. Inst.* **1965**, *280*, 367–386. [CrossRef]
16. Åström, K.J. Optimal control of Markov processes with incomplete state information. *J. Math. Anal. Appl.* **1965**, *10*, 174–205. [CrossRef]
17. Papadimitriou, C.H.; Tsitsiklis, J.N. The Complexity of Markov Decision Processes. *Math. Oper. Res.* **1987**, *12*, 441–450. [CrossRef]
18. Xiang, X.; Foo, S. Recent Advances in Deep Reinforcement Learning Applications for Solving Partially Observable Markov Decision Processes (POMDP) Problems: Part 1—Fundamentals and Applications in Games, Robotics and Natural Language Processing. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 554–581. [CrossRef]
19. Kimura, T.; Shiba, K.; Chen, C.-C.; Sogabe, M.; Sakamoto, K.; Sogabe, T. Variational Quantum Circuit-Based Reinforcement Learning for POMDP and Experimental Implementation. *Math. Probl. Eng.* **2021**, *2021*, 3511029. [CrossRef]
20. Kaelbling, L.P.; Littman, M.L.; Cassandra, A.R. Planning and acting in partially observable stochastic domains. *Artif. Intell.* **1998**, *101*, 99–134. [CrossRef]
21. Singh, S.P.; Jaakkola, T.; Jordan, M.I. Learning without State-Estimation in Partially Observable Markovian Decision Processes. In *Machine Learning Proceedings 1994*; Cohen, W.W., Hirsh, H., Eds.; Morgan Kaufmann: San Francisco, CA, USA, 1994; pp. 284–292, ISBN 978-1-55860-335-6.
22. Barry, J.; Barry, D.T.; Aaronson, S. Quantum partially observable Markov decision processes. *Phys. Rev. A* **2014**, *90*, 032311. [CrossRef]
23. Ying, S.; Ying, M. Reachability analysis of quantum Markov decision processes. *Inf. Comput.* **2018**, *263*, 31–51. [CrossRef]
24. Ying, M.-S.; Feng, Y.; Ying, S.-G. Optimal Policies for Quantum Markov Decision Processes. *Int. J. Autom. Comput.* **2021**, *18*, 410–421. [CrossRef]
25. Abhijith, J.; Adedoyin, A.; Ambrosiano, J.; Anisimov, P.; Casper, W.; Chennupati, G.; Coffrin, C.; Djidjev, H.; Gunter, D.; Karra, S.; et al. Quantum Algorithm Implementations for Beginners. *ACM Trans. Quantum Comput.* **2022**, *3*, 18:1–18:92. [CrossRef]
26. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]

27. Nielsen, M.A.; Chuang, I.L. Quantum Computation and Quantum Information: 10th Anniversary Edition. Available online: https://www.cambridge.org/highereducation/books/quantum-computation-and-quantum-information/01E10196D0A682A6AEFFEA52D53BE9AE (accessed on 22 August 2022).
28. Barenco, A.; Bennett, C.H.; Cleve, R.; DiVincenzo, D.P.; Margolus, N.; Shor, P.; Sleator, T.; Smolin, J.A.; Weinfurter, H. Elementary gates for quantum computation. *Phys. Rev. A* **1995**, *52*, 3457–3467. [CrossRef]
29. Deutsch, D. Quantum theory, the Church–Turing principle and the universal quantum computer. *Proc. R. Soc. Lond. Math. Phys. Sci.* **1985**, *400*, 97–117. [CrossRef]
30. Feynman, R.P. Simulating physics with computers. *Int. J. Theor. Phys.* **1982**, *21*, 467–488. [CrossRef]
31. Mermin, N.D. *Quantum Computer Science: An Introduction*; Cambridge University Press: Cambridge, UK, 2007; ISBN 978-0-521-87658-2.
32. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef] [PubMed]
33. Chen, C.-C.; Shiau, S.-Y.; Wu, M.-F.; Wu, Y.-R. Hybrid classical-quantum linear solver using Noisy Intermediate-Scale Quantum machines. *Sci. Rep.* **2019**, *9*, 16251. [CrossRef] [PubMed]
34. Kimura, T.; Shiba, K.; Chen, C.-C.; Sogabe, M.; Sakamoto, K.; Sogabe, T. Quantum circuit architectures via quantum observable Markov decision process planning. *J. Phys. Commun.* **2022**, *6*, 075006. [CrossRef]
35. Borah, S.; Sarma, B.; Kewming, M.; Milburn, G.J.; Twamley, J. Measurement-Based Feedback Quantum Control with Deep Reinforcement Learning for a Double-Well Nonlinear Potential. *Phys. Rev. Lett.* **2021**, *127*, 190403. [CrossRef]
36. Sivak, V.V.; Eickbusch, A.; Liu, H.; Royer, B.; Tsioutsios, I.; Devoret, M.H. Model-Free Quantum Control with Reinforcement Learning. *Phys. Rev. X* **2022**, *12*, 011059. [CrossRef]
37. Niu, M.Y.; Boixo, S.; Smelyanskiy, V.N.; Neven, H. Universal quantum control through deep reinforcement learning. *NPJ Quantum Inf.* **2019**, *5*, 33. [CrossRef]
38. He, R.-H.; Wang, R.; Nie, S.-S.; Wu, J.; Zhang, J.-H.; Wang, Z.-M. Deep reinforcement learning for universal quantum state preparation via dynamic pulse control. *EPJ Quantum Technol.* **2021**, *8*, 29. [CrossRef]
39. Bukov, M.; Day, A.G.R.; Sels, D.; Weinberg, P.; Polkovnikov, A.; Mehta, P. Reinforcement Learning in Different Phases of Quantum Control. *Phys. Rev. X* **2018**, *8*, 031086. [CrossRef]
40. Mackeprang, J.; Dasari, D.B.R.; Wrachtrup, J. A reinforcement learning approach for quantum state engineering. *Quantum Mach. Intell.* **2020**, *2*, 5. [CrossRef]
41. 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*, 1–7. [CrossRef]
42. Baum, Y.; Amico, M.; Howell, S.; Hush, M.; Liuzzi, M.; Mundada, P.; Merkh, T.; Carvalho, A.R.R.; Biercuk, M.J. Experimental Deep Reinforcement Learning for Error-Robust Gate-Set Design on a Superconducting Quantum Computer. *PRX Quantum* **2021**, *2*, 040324. [CrossRef]
43. Kuo, E.-J.; Fang, Y.-L.L.; Chen, S.Y.-C. Quantum Architecture Search via Deep Reinforcement Learning. *arXiv* **2021**, arXiv:2104.07715.
44. Pirhooshyaran, M.; Terlaky, T. Quantum circuit design search. *Quantum Mach. Intell.* **2021**, *3*, 25. [CrossRef]
45. Ostaszewski, M.; Trenkwalder, L.M.; Masarczyk, W.; Scerri, E.; Dunjko, V. Reinforcement learning for optimization of variational quantum circuit architectures. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 18182–18194.
46. August, M.; Hernández-Lobato, J.M. Taking Gradients Through Experiments: LSTMs and Memory Proximal Policy Optimization for Black-Box Quantum Control. In Proceedings of the High Performance Computing, Frankfurt, Germany, 24–28 June 2018; Yokota, R., Weiland, M., Shalf, J., Alam, S., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 591–613.
47. Hausknecht, M.; Stone, P. Deep Recurrent Q-Learning for Partially Observable MDPs. In Proceedings of the 2015 AAAI Fall Symposium Series, Arlington, VA, USA, 12–14 November 2015.
48. Lample, G.; Chaplot, D.S. Playing FPS Games with Deep Reinforcement Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31. [CrossRef]
49. Zhu, P.; Li, X.; Poupart, P.; Miao, G. On Improving Deep Reinforcement Learning for POMDPs. *arXiv* **2018**, arXiv:1704.07978.
50. Kimura, T.; Sakamoto, K.; Sogabe, T. Development of AlphaZero-based Reinforcment Learning Algorithm for Solving Partially Observable Markov Decision Process (POMDP) Problem. *Bull. Netw. Comput. Syst. Softw.* **2020**, *9*, 69–73.
51. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016; ISBN 978-0-262-03561-3.
52. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
53. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [CrossRef]
54. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hoo, NY, USA, 2019; Volume 32.
55. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362. [CrossRef] [PubMed]
56. Hunter, J.D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9*, 90–95. [CrossRef]

57. Treinish, M.; Gambetta, J.; Nation, P.; qiskit-bot; Kassebaum, P.; Rodríguez, D.M.; González, S.d.l.P.; Hu, S.; Krsulich, K.; Lishman, J.; et al. Qiskit/qiskit: Qiskit 0.37.1. 2022. Available online: https://elib.uni-stuttgart.de/handle/11682/12385 (accessed on 16 August 2022). [CrossRef]
58. Greenberger, D.M.; Horne, M.A.; Zeilinger, A. Going Beyond Bell's Theorem. In *Bell's Theorem, Quantum Theory and Conceptions of the Universe*; Fundamental Theories of Physics; Kafatos, M., Ed.; Springer: Dordrecht, The Netherlands, 1989; pp. 69–72, ISBN 978-94-017-0849-4.
59. Gasse, M.; Chételat, D.; Ferroni, N.; Charlin, L.; Lodi, A. Exact combinatorial optimization with graph convolutional neural networks. In Proceedings of the 33rd International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Curran Associates Inc.: Red Hook, NY, USA, 2019; pp. 15580–15592.
60. Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.-H.; Zhou, X.-Q.; Love, P.J.; Aspuru-Guzik, A.; O'Brien, J.L. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **2014**, *5*, 4213. [CrossRef] [PubMed]
61. McClean, J.R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **2016**, *18*, 023023. [CrossRef]
62. Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J.M.; Gambetta, J.M. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **2017**, *549*, 242–246. [CrossRef]