Article

Measurement-Based Adaptation Protocol with Quantum Reinforcement Learning in a Rigetti Quantum Computer

Julio Olivares-Sánchez [1], Jorge Casanova [1,2] , Enrique Solano [1,2,3,4] and Lucas Lamata [1,5,*]

1   Department of Physical Chemistry, University of the Basque Country UPV/EHU, Apartado 644, 48080 Bilbao, Spain; juliolivaresanchez@gmail.com (J.O.-S.); jcasanovamar@gmail.com (J.C.); enr.solano@gmail.com (E.S.)
2   IKERBASQUE, Basque Foundation for Science, Maria Diaz de Haro 3, 48013 Bilbao, Spain
3   International Center of Quantum Artificial Intelligence for Science and Technology (QuArtist) and Physics Department, Shanghai University, Shanghai 200444, China
4   IQM, 80336 Munich, Germany
5   Departamento de Física Atómica, Molecular y Nuclear, Universidad de Sevilla, 41080 Sevilla, Spain
*   Correspondence: llamata@us.es

check for updates

Abstract: We present an experimental realisation of a measurement-based adaptation protocol with quantum reinforcement learning in a Rigetti cloud quantum computer. The experiment in this few-qubit superconducting chip faithfully reproduces the theoretical proposal, setting the first steps towards a semiautonomous quantum agent. This experiment paves the way towards quantum reinforcement learning with superconducting circuits.

1. Introduction

Quantum machine learning [1–29] is a field of research that has raised much attention in the past few years, especially for the expectation that it may enhance the machine learning calculations in current and future technology. The machine learning field, inside artificial intelligence, is divided into three main areas: supervised learning, unsupervised learning and reinforcement learning [30]. The first two rely on training the system via labelled or unlabelled data, respectively. Moreover, the third one considers an intelligent agent that interacts with its outer world, the environment, gathering information from it, as well as acting on it, being employed, e.g., in robotics. In each learning iteration, the agent decides a strategy, or policy, on the best action to take, depending on its past history and goal oriented. Reinforcement learning can be considered as the most similar way in which human beings learn, via interactions with their outer world.

Among the protocols being developed in quantum machine learning, the ones based on quantum reinforcement learning will enable the future deployment of semiautonomous quantum agents, which may produce significant advances in quantum computation and artificial intelligence [1,2,25–29]. Already some possible speedups when considering quantum agents have been suggested [25], while other interesting aspects of quantum reinforcement learning are related to quantum systems autonomously learning quantum data [1].

In this article, we implement a protocol for measurement-based adaptation with quantum reinforcement learning, proposed in [1], in an 8-qubit cloud quantum computer provided by Rigetti. The experimental results have a good agreement with the theoretical expectations, establishing a

plausible avenue for the future development of semiautonomous quantum agents. We point out that further efforts in experimental quantum machine learning, both in superconducting circuits [31] and in quantum photonics [32], are being pursued.

2. Results

2.1. Measurement-Based Adaptation Protocol with Quantum Reinforcement Learning

We review the protocol of [1], which we subsequently implement in the Rigetti cloud quantum computer. The aim of this algorithm is to adapt a quantum state to a reference unknown state via successive measurements. Many identical copies of the reference state are needed for this protocol and, after each measurement, which destroys the state, more information about it is obtained. The system that we consider is composed of the following parts:

1. The *environment* system (E) contains the reference state copies.
2. The *register* (R) interacts with E and obtains information from it.
3. The *agent* (A) is adapted by digital feedback depending on the outcome of the measurement of the register.

Let us assume that we know the state of a quantum system called agent and that many copies of an unknown quantum state called environment are provided. Let us also consider an auxiliary system called register that interacts with E. Thus, we extract information from E by measuring R and use the result as an input for a reward function (RF).

Subsequently, we perform a partially-random unitary transformation on A, which depends on the output of the RF.

Let us present the simplest case in which each subsystem is described by a qubit state: Agent $|0\rangle_A$, Register $|0\rangle_R$ and Environment $|\varepsilon\rangle_E = \left[\cos\left(\frac{\theta^{(1)}}{2}\right) |0\rangle_E + e^{i\phi^{(1)}} \sin\left(\frac{\theta^{(1)}}{2}\right) |1\rangle_E \right]$.

Therefore, the initial state is,

$$|\psi_0^{(1)}\rangle = |0\rangle_A |0\rangle_R \left[\cos\left(\frac{\theta^{(1)}}{2}\right) |0\rangle_E + e^{i\phi^{(1)}} \sin\left(\frac{\theta^{(1)}}{2}\right) |1\rangle_E \right]. \tag{1}$$

Then, we apply a CNOT gate with E as control and R as target (policy) to obtain information from E, namely

$$C_E \text{NOT}_R |\psi_0^{(1)}\rangle = |0\rangle_A \left[\cos\left(\frac{\theta^{(1)}}{2}\right) |0\rangle_R |0\rangle_E + e^{i\phi^{(1)}} \sin\left(\frac{\theta^{(1)}}{2}\right) |1\rangle_R |1\rangle_E \right]. \tag{2}$$

Secondly, we measure the register qubit in the basis $\{|0\rangle, |1\rangle\}$ with probability $P_0^{(1)} = \cos^2\left(\frac{\theta^{(1)}}{2}\right)$, or $P_1^{(1)} = \sin^2\left(\frac{\theta^{(1)}}{2}\right)$ of obtaining the state $|0\rangle$ or $|1\rangle$, respectively. Depending on the result of the measurement, we either do nothing if the result is $|0\rangle$, since it means that we collapse E into A, or perform a partially-random unitary operator on A, if the result is $|1\rangle$. This unitary transformation (action) is given by

$$U_A^{(1)}\left(\alpha^{(1)}, \beta^{(1)}\right) = e^{-iS_A^{Z(1)}\alpha^{(1)}} e^{-iS_A^{X(1)}\beta^{(1)}}, \tag{3}$$

where $S_A^{k(1)}$ is the kth spin component; $\alpha^{(1)} = \xi_\alpha \Delta^{(1)}$ and $\beta^{(1)} = \xi_\beta \Delta^{(1)}$ are random angles; and $\xi_\alpha, \xi_\beta \in \left[-\frac{1}{2}, \frac{1}{2}\right]$ are random numbers. The range of the random numbers is $\alpha^{(1)}, \beta^{(1)} \in \left[-\frac{\Delta^{(1)}}{2}, \frac{\Delta^{(1)}}{2}\right]$.

Now, we initialise the register qubit state and use a new copy of the environment obtaining the following initial state for the second iteration,

$$|\psi_0^{(2)}\rangle = \mathcal{U}_A^{(1)} |0\rangle_A |0\rangle_R |\varepsilon\rangle_E, \tag{4}$$

with $\mathcal{U}_A^{(1)} = \left[m^{(1)} U_A^{(1)} \left(\alpha^{(1)}, \beta^{(1)} \right) + \left(1 - m^{(1)} \right) \mathbb{I}_A \right]$ where $m^{(1)} = \{0, 1\}$ is the outcome of the measurement, and we call $\mathcal{U}_A^{(1)} |0\rangle_A \equiv |\bar{0}\rangle_A^{(2)}$.

Later, we define the RF as

$$\Delta^{(k)} = \left[(1 - m^{(k-1)}) \mathcal{R} + m^{(k-1)} \mathcal{P} \right] \Delta^{(k-1)}. \tag{5}$$

As we can see, the exploration range of the kth iteration, $\Delta^{(k)}$, is modified by $\mathcal{R}\Delta$ when the outcome of the $k - 1$ iteration was $m^{(k-1)} = 0$ and by $\mathcal{P}\Delta$, when it was $m^{(k-1)} = 1$. Here, we chose $\mathcal{R} = \epsilon < 1$, $\mathcal{P} = \frac{1}{\epsilon}$, with ϵ being a constant.

We define the value function (VF) in this protocol as the value of $\Delta^{(n)}$ after many iterations assuming that $\Delta \to 0$, i.e., that the agent converges to the environment state.

In the kth iteration of the protocol, we assume that the system starts in the following state,

$$|\psi\rangle^{(k)} = |\bar{0}\rangle_A^{(k)} |0\rangle_R |\bar{\varepsilon}\rangle_E, \tag{6}$$

where $|\bar{0}\rangle_A^{(k)} = \mathbb{U}^{(k)} |0\rangle_A$, $|\bar{\varepsilon}\rangle_E = \mathbb{U}^{(k)\dagger} |\varepsilon\rangle_E$, and the accumulated rotation operator is,

$$\mathbb{U}^{(k)} = \left[m^{(k-1)} U^{(k-1)} \left(\alpha^{(k-1)}, \beta^{(k-1)} \right) + \left(1 - m^{(k-1)} \right) \mathbb{I} \right] \mathbb{U}^{(k-1)}, \tag{7}$$

with $\mathbb{U}^{(1)} = \mathbb{I}$.

Then, we perform the gate $C_E NOT_R$,

$$|\phi\rangle^{(k)} = C_E NOT_R |\bar{0}\rangle_A^{(k)} |0\rangle_R |\bar{\varepsilon}\rangle_E = |\bar{0}\rangle_A^{(k)} \left[\cos\left(\frac{\theta^{(k)}}{2} \right) |0\rangle_R |0\rangle_E + e^{i\phi^{(k)}} \sin\left(\frac{\theta^{(k)}}{2} \right) |1\rangle_R |1\rangle_E \right], \tag{8}$$

and measure R, with probabilities $P_0^{(k)} = \cos^2\left(\frac{\theta^{(k)}}{2} \right)$ and $P_1^{(k)} = \sin^2\left(\frac{\theta^{(k)}}{2} \right)$, for the outcomes $m^{(k)} = 0$ and $m^{(k)} = 1$, respectively. Finally, we update the reward function, $\Delta^{(k)}$, for the next iteration.

2.2. Experimental Setup: Rigetti Forest Cloud Quantum Computer

Cloud quantum computers are dedicated quantum processors operated by users logged through the Internet (cloud). Although D-Wave Systems, Inc. [33] was among the first companies to commercialise quantum computers in 2011, it was not until the arrival of IBM Quantum Experience [34] in May 2016 that there was a quantum computer openly available in the cloud. Approximately one year later, in June 2017, Californian company Rigetti Computing [35] announced the availability of a cloud quantum computing platform. This last quantum platform is the one that we used in this work, because of convenience of their web interface to implement our protocol with feedback. In the past year, Rigetti built a brand new processor of 8 qubits called 8Q-Agave: this is the chip we employed. An advantage of this device is that one does not have to adapt the algorithm to the topology of the system. The compiler does it for us. In particular, it is the possibility of defining quantum gates in matrix form that makes the adaptation significantly simpler. Among other features, Rigetti Forest offers the possibility of simulating the experiments in their quantum virtual machine (QVM).

Python-Implemented Algorithm

In this section, we explain how the algorithm of Section 2.1 is adapted to be implemented in Rigetti simulator and quantum processor. Firstly, we must initialise some variables and constants to correctly perform the first iteration,

- Reward and punishment ratios: $\epsilon \in (0, 1) \Rightarrow \mathcal{R} = \epsilon$ and $\mathcal{P} = 1/\epsilon$.
- Exploration range: $\Delta = 4\pi$.

- The unitary transformation matrices: $\mathcal{U} = \mathbb{U} = \mathbb{U}^\dagger = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$.

- Partially-random unitary operator: $U(x, y) = \begin{pmatrix} e^{-i\frac{x}{2}} & 0 \\ 0 & e^{i\frac{x}{2}} \end{pmatrix} \begin{pmatrix} \cos\frac{y}{2} & -i\sin\frac{y}{2} \\ -i\sin\frac{y}{2} & \cos\frac{y}{2} \end{pmatrix}$.

- Initial values of the random angles: $\alpha = \beta = 0$. Makes $U(\alpha, \beta) = \mathbb{I}_2$ for the first iteration.
- Initial value of the iteration index: $k = 1$.
- Number of iterations: N.

The algorithm is composed of the following steps,

1. Step 1: While $k < N + 1 \Rightarrow$, go to Step 2.
2. Step 2: If $k \neq 1 \Rightarrow$

$$\xi_\alpha = random\ number \in \left[-\frac{1}{2}, \frac{1}{2}\right] \tag{9}$$

$$\xi_\beta = random\ number \in \left[-\frac{1}{2}, \frac{1}{2}\right],$$

$$\begin{matrix} \alpha = \xi_\alpha \Delta \\ \beta = \xi_\beta \Delta \end{matrix}' \tag{10}$$

$$\mathcal{U} = [mU(\alpha, \beta) + (1 - m)\mathbb{I}_2], \tag{11}$$

$$\mathbb{U} = \mathcal{U} \cdot \mathbb{U}, \tag{12}$$

$$\mathbb{U}^\dagger = (\mathbb{U}^*)^T. \tag{13}$$

3. Step 3: First quantum algorithm.
    First, we define the agent, environment and register qubits as,

$$|0\rangle_A |0\rangle_R |0\rangle_E, \tag{14}$$

and act upon the environment,

$$|\varepsilon\rangle_E = U^E |0\rangle_E. \tag{15}$$

Then, we have

$$\mathbb{U}^\dagger |\varepsilon\rangle_E = |\bar{\varepsilon}\rangle_E. \tag{16}$$

We apply the policy

$$C_E NOT_R |0\rangle_R |\bar{\varepsilon}\rangle_E, \tag{17}$$

and measure the register qubit storing the result in $m = \{0, 1\}$.

4. Step 4: Second quantum algorithm.
    Subsequently, we act with $\mathbb{U}$ on the agent qubit in order to approach it to the environment state, $|\varepsilon\rangle_E$:

$$\mathbb{U}|0\rangle_A, \tag{18}$$

Afterwards, we measure this qubit and store the result in a classical register array. We repeat Step 4 a total of 8192 times to determine the state created after applying $\mathbb{U}$.

5. In this last step, we apply the reward function,

$$\Delta = [(1 - m)\mathcal{R} + m\mathcal{P}]\Delta, \tag{19}$$

and increase the iteration index by one after it: $k = k + 1$. Go to Step 1.

2.3. Experimental Results of Quantum Reinforcement Learning with the Rigetti Cloud Quantum Computer

In this section, we describe the experimental results with the Rigetti cloud quantum computer of the measurement-based adaptation protocol of [1].

The algorithm has been proved for six different initial states of the environment. The states contain different weights of the quantum superpositions of $|0\rangle$ and $|1\rangle$ states, as well as different complex relative phases, to illustrate the performance of the protocol in a variety of situations. These states are the following,

$$|\varepsilon_1\rangle_E = R_Z\left(\frac{\pi}{3}\right) R_Y\left(\frac{4\pi}{9}\right) |0\rangle_E = \begin{pmatrix} e^{-i\frac{\pi}{6}} \cos\left(\frac{2\pi}{9}\right) \\ e^{i\frac{\pi}{6}} \sin\left(\frac{2\pi}{9}\right) \end{pmatrix} \approx \sqrt{0.6}|0\rangle + \sqrt{0.4}e^{i\frac{\pi}{3}}|1\rangle, \qquad (20)$$

$$|\varepsilon_2\rangle_E = R_Z\left(\frac{\pi}{4}\right) R_Y\left(\frac{5\pi}{9}\right) |0\rangle_E = \begin{pmatrix} e^{-i\frac{\pi}{8}} \cos\left(\frac{5\pi}{18}\right) \\ e^{i\frac{\pi}{8}} \sin\left(\frac{5\pi}{18}\right) \end{pmatrix} \approx \sqrt{0.4}|0\rangle + \sqrt{0.6}e^{i\frac{\pi}{4}}|1\rangle, \qquad (21)$$

$$|\varepsilon_3\rangle_E = R_Y\left(\frac{\pi}{3}\right) |0\rangle_E = \begin{pmatrix} \cos\left(\pi/6\right) \\ \sin\left(\pi/6\right) \end{pmatrix} = \sqrt{0.75}|0\rangle + \sqrt{0.25}|1\rangle, \qquad (22)$$

$$|\varepsilon_4\rangle_E = R_Y\left(\frac{2\pi}{3}\right) |0\rangle_E = \begin{pmatrix} \cos\left(\pi/3\right) \\ \sin\left(\pi/3\right) \end{pmatrix} = \sqrt{0.25}|0\rangle + \sqrt{0.75}|1\rangle, \qquad (23)$$

$$|\varepsilon_5\rangle_E = R_Z\left(\frac{\pi}{2}\right) H|0\rangle_E = \frac{1}{\sqrt{2}}\left(|0\rangle + i|1\rangle\right), \qquad (24)$$

$$|\varepsilon_6\rangle_E = H|0\rangle_E = |+\rangle_E = \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right). \qquad (25)$$

Hereunder, we plot the exploration range, Δ, and the fidelity, which we define in Equation (26), in terms of the total number of iterations. The results obtained using the 8Q-Agave chip of Rigetti are shown with the corresponding ideal simulation with the Rigetti quantum virtual machine, which includes no noise. We point out that each of the experimental plots in this article contains a single realisation of the experiment, given that, in the adaptation protocol, single instances are what are relevant instead of averages. This is due to the fact that the successive measurements influence the subsequent ones and are influenced by the previous ones, in each experiment, in order for the agent to adapt to the environment. This has as a consequence the fact that the theory curve and the experimental curve for each example match qualitatively but not totally quantitatively, as they are both probabilistic. On the other hand, when the exploration range converges to zero we always observe convergence to a large final fidelity. The blue solid line represents the real experiment result while the red dashed-dotted one corresponds to the ideal simulation. It is also worth mentioning that after doing many experiments with the ideal and real simulators for different values of the parameter ϵ, we fixed it to $\epsilon = 0.95$. We found it to yield a balanced exploration–exploitation ratio. The exploration–exploitation balance is a feature of reinforcement learning, where the optimal strategy must deal with exploring enough new possibilities while at the same time exploiting the ones that work best [30].

To begin with, let us have a look at Figure 1, in which the environment state is $|\varepsilon_1\rangle_E$. It shows the exploration ratio, Δ, and the fidelity as defined in Equation (26) in terms of the number of iterations. As we can see, 140 iterations are enough for Δ to take a value of almost zero. It is a good example of a balanced exploration versus exploitation ratio, that is, the exploration ratio decreases making continuous peaks. Each of them represents an exploration stage where Δ increases at the same time that the fidelity changes significantly. These changes might bring a positive result, such that the agent receives a reward, which means that Δ decreases, or a punishment and it keeps increasing. Thus, the fidelity is not constant and it does not attain a constant value of around 95% until it has done 100 iterations. In this case real and ideal experiments yield a similar result. It is true that the ideal Δ

decreases more smoothly and quickly than the real one. However, the values of the fidelity after 130 iterations are practically the same.
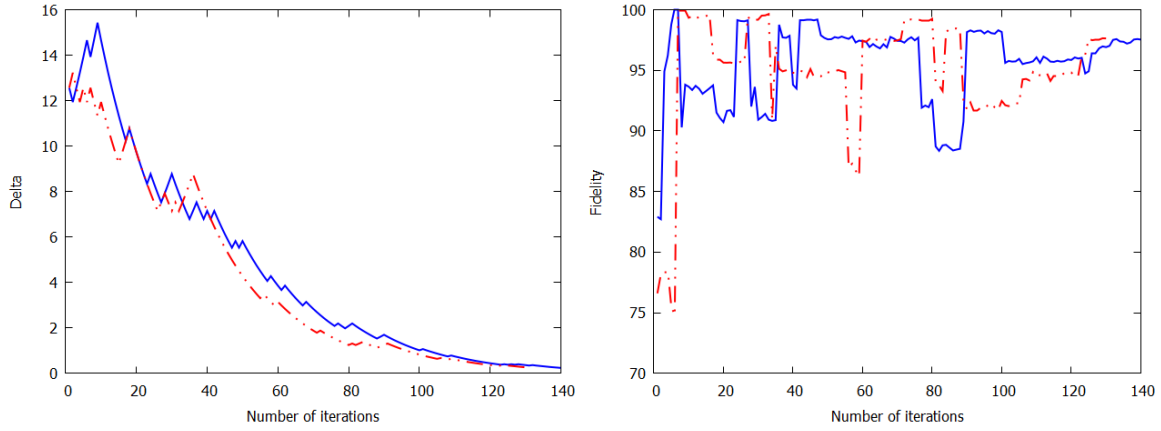

Figure 1. Measurement-based adaptation protocol for the environment state $|\varepsilon_1\rangle_E$. The blue solid line corresponds to the real experiment and the red dashed-dotted line represents the ideal simulation. The fidelity is given in percentage (%).

In our calculations, we employ a classical measure of the fidelity. The reason to use the classical fidelity, and not the quantum version, is the reduction of needed resources in the experiments. We cannot make full quantum state tomography as it is exponentially hard. Therefore, the definition used in the algorithm is,

$$\mathcal{F} = \sqrt{p_0 p_0^T} + \sqrt{p_1 p_1^T} \tag{26}$$

for one-qubit measurements. Here, p_0 and p_1 stand for the probability of obtaining $|0\rangle$ or $|1\rangle$ as an outcome when measuring the real qubit and p_0^T and p_1^T are the same probabilities for the corresponding theoretical qubit state that we expect. This fidelity coincides at lowest order with the fully quantum one, illustrating the convergence of the protocol for a large number of iterations.

Let us continue with the discussion focussing on Figure 2. In this experiment, the algorithm has to take the agent from the $|0\rangle$ state to the environment state $|\varepsilon_2\rangle_E$ which is closer to one than to zero (0.6 is the probability of getting $|1\rangle$ as an outcome when measuring the environment). Bearing this in mind, it seems reasonable that 70 iterations are not enough for Δ to reach a value below 2, in the real case. Apart from this, despite achieving a value above 99% of fidelity in fewer than 20 iterations, the exploration still continues. Consequently, the agent drops from its current state to one further from $|\varepsilon_2\rangle_E$.

In general, we notice a clear relationship between how smooth the Δ line is and how constant the fidelity remains. Indeed, the exploration ratio Δ decreases smoothly from fewer than 20 iterations to fewer than 40. In this range, the fidelity does not change because the agent is being rewarded. The price to pay for not exploring at so early stages of the learning is that the convergence of the delta is produced for a larger number of iterations than in other experiments. After 140 iterations, we see that the convergence of $\Delta \to 0$ is not guaranteed, namely, in the real experiment, it has a value above 1. Regarding the ideal simulation result, we draw the conclusion that fewer than 20 iterations could be enough to converge to the environment state with fidelity larger than 99.9% and, what is more, remaining on the same state until the exploration range has converged to zero (see inset in Figure 2).
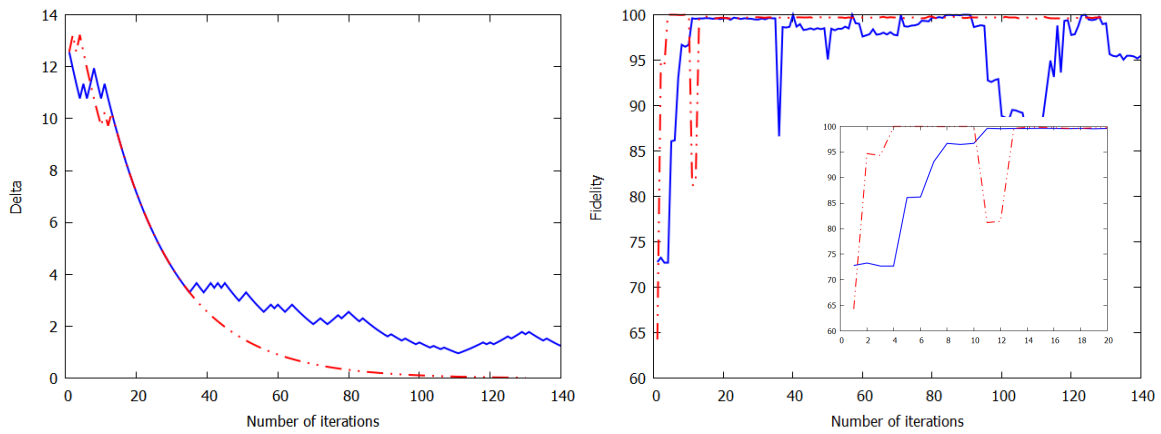
Figure 2. Measurement-based adaptation protocol for the environment state $|\varepsilon_2\rangle_E$. The blue solid line corresponds to the real experiment and the red dashed-dotted line represents the ideal simulation. The fidelity is given in percentage (%).

In third place, we have the environment state $|\varepsilon_3\rangle_E$. The results obtained using this initial state of the environment are presented in Figures 3 and 4. Unlike the previous examples, we do not compare the ideal theory and real experiments, which have similarly good agreement. Instead, we contrast two different real experiment outcomes. In this way, we can show how even for the same experiment, i.e., initial state of the environment, the algorithm can experience different behaviours. In both cases, Δ goes to zero and the fidelity reaches a constant value above 94% in the first case and 99% in the second one. However, this convergence is achieved in two different ways. On the one hand, we observe that exploitation predominates over exploration (see Figure 4), except for several spots where the algorithm keeps exploring new options. Then, as the initial fidelity is larger than 90%, the state of the agent converges to the environment with fewer than 70 iterations. On the other hand, when exploration is more important (as shown in Figure 3) the fidelity is erratic, changing from low to high values. Moreover, it takes longer for Δ to converge and for the fidelity to be stabilised—more than 80 iterations.

Let us focus just on the first stages of the learning process, for fewer than 40 iterations. Comparing both experiments, we see that, in the first case, it starts exploring from the very beginning; thus, with fewer than 20 iterations, the fidelity takes a value above 99.6%. In the other case, there is more exploitation at the beginning and around 25 iterations are required to reach a fidelity of 99%.

Among the six states that we have chosen, this is the one in which agent and environment are the closest. Nevertheless, 70 iterations are not enough to reach a value of Δ below 1. Thus, we can state that a smaller distance in the Bloch sphere between agent and environment does not imply in general a faster learning.
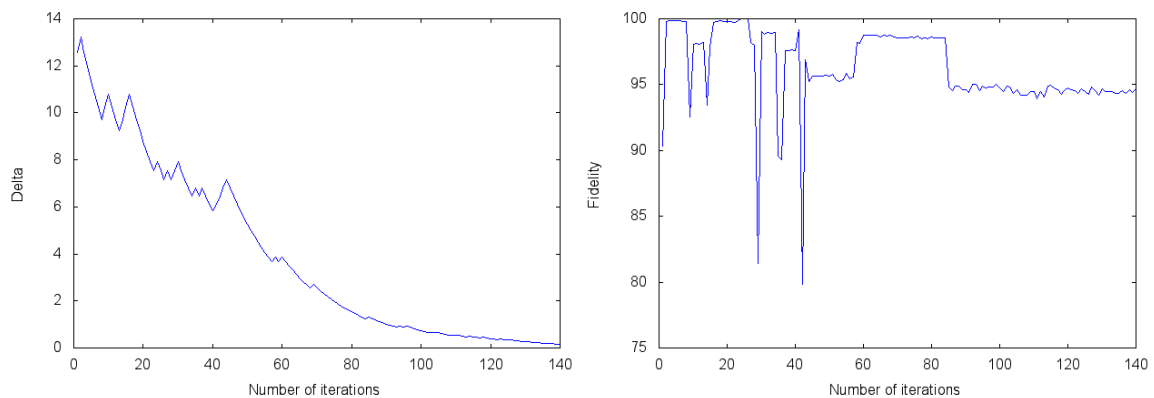

Figure 3. Measurement-based adaptation protocol for the environment state $|\varepsilon_3\rangle_E$. Experiment one. The blue solid line corresponds to the real experiment. The fidelity is given in percentage (%).
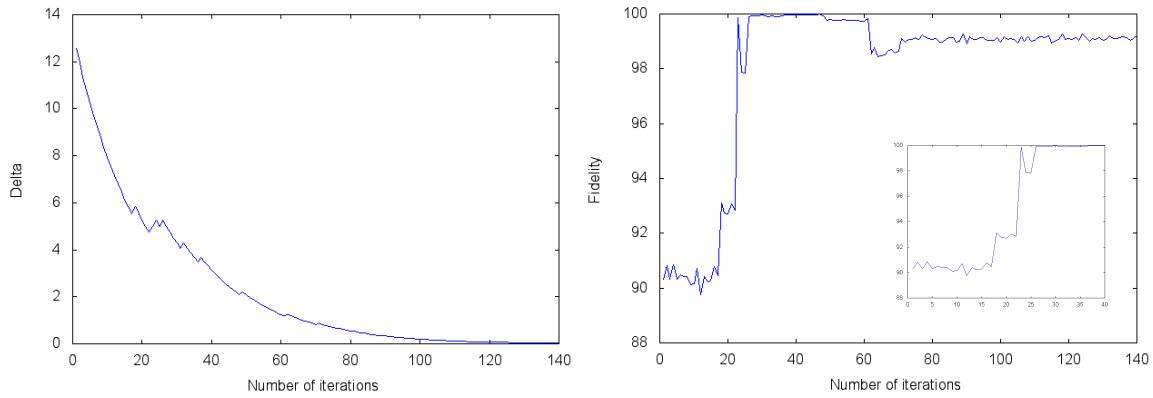
Figure 4. Measurement-based adaptation protocol for the environment state $|\varepsilon_3\rangle_E$. Experiment two. The blue solid line corresponds to the real experiment. The fidelity is given in percentage (%).

Let us analyse now Figure 5. This state is again the most asymmetric case along with the previous one. However, unlike the previous experiment, this one begins with the lowest value of fidelity. The environment state is the farthest one to the initial agent state $|0\rangle_A$, with just a probability of 0.25 of achieving this outcome (zero) when measuring the environment. Therefore, as might be expected, the algorithm is still exploring after 70 iterations rather than exploiting the knowledge it has already acquired from the environment. It is also proved that fewer than 100 iterations can be enough to reach a value of Δ below 1. Then, in this case, it is proved that the agent has already converged to a state with fidelity larger than 97%. It is also remarkable how, in this case, with fewer than 10 iterations, the fidelity has attained a value larger than 99%. However, as the delta had not converged yet, it goes out of this value later, exploring again. Once again, as a general rule, we can see that the algorithm is exploring for all the iterations. To explore is a synonym of changing fidelity, while, whenever the delta decreases smoothly, the fidelity remains constant. With this result, we wanted to show how sometimes the real experiment converged more quickly, e.g., with just nine iterations, to the environment state. On the top of that, the exploration range also went to zero more quickly than the ideal experiment. Nevertheless, the value of fidelity when Δ has converged is exactly the same in both cases.
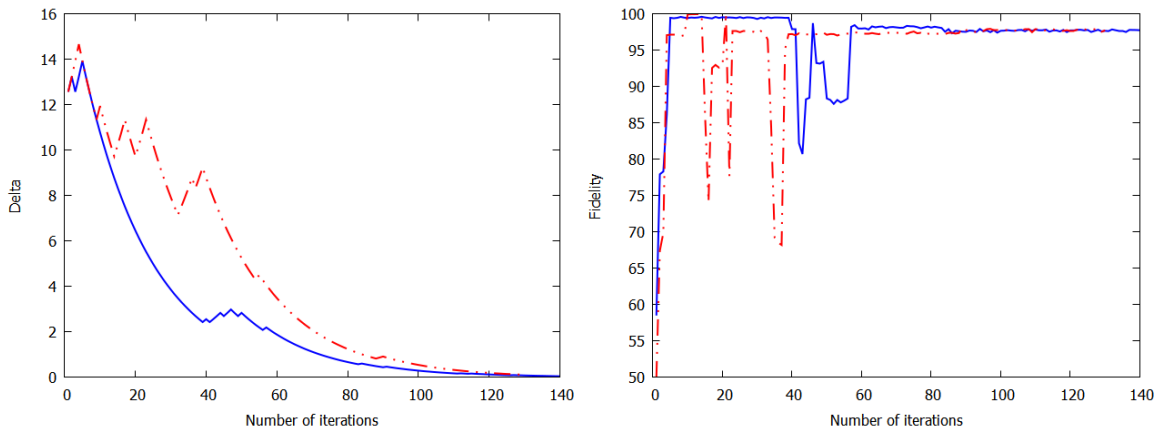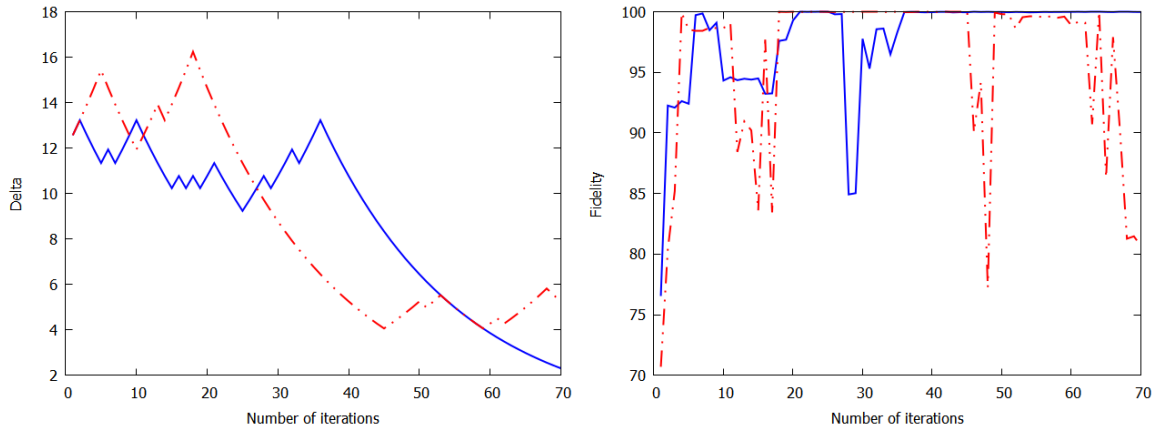

Figure 5. Measurement-based adaptation protocol for the environment state $|\varepsilon_4\rangle_E$. The blue solid line corresponds to the real experiment and the red dashed-dotted line represents the ideal simulation. The fidelity is given in percentage (%).

We analyse now the most symmetric cases, where the environment is prepared in a uniform superposition with a relative phase between both states. The experiment chosen to highlight here (see Figure 6) is the one in which the fidelity reaches a constant value above 99.9% in fewer than 40 iterations. The corresponding Δ evolves with a good balance between exploration and exploitation until reaching a point where it does not explore anymore and decreases very smoothly. Comparing it

to the ideal case, we notice two opposing behaviours: the ideal case makes a larger exploration at the beginning which yields a larger constant value of the fidelity with fewer than 20 iterations, whereas the real system needs almost 40 iterations to get to this point. On the other hand, in the real experiment, there is a larger learning stage from 0 to 40 iterations. Thus, in this particular case, the exploration ratio diminishes more quickly in the real experiment for a large number of iterations. It happens because of the larger value of Δ attained in the ideal experiment.
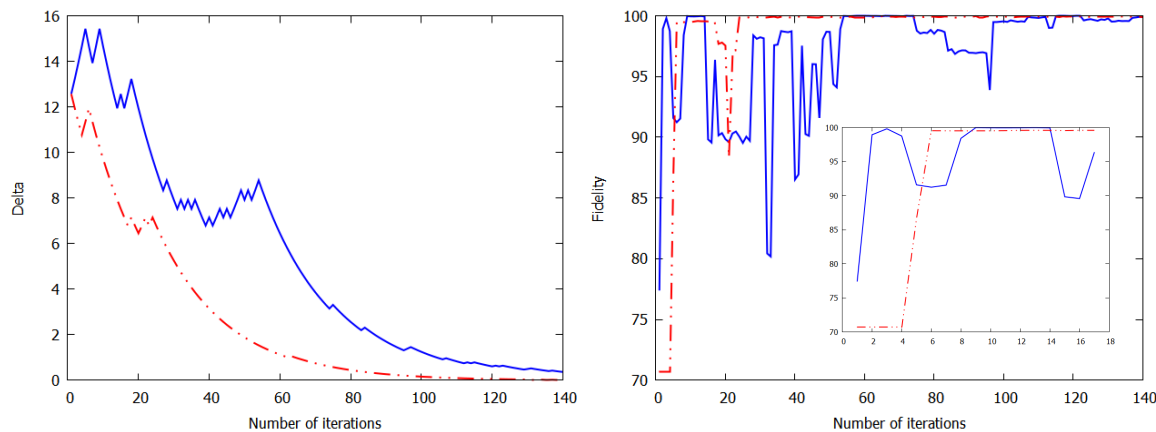

Figure 6. Measurement-based adaptation protocol for the environment state $|\varepsilon_5\rangle_E$. The blue solid line corresponds to the real experiment and the red dashed-dotted line represents the ideal simulation. The fidelity is given in percentage (%).

Finally, in this second symmetric case $|\varepsilon_6\rangle_E$, there are no relative phases between the states of the computational basis. This experiment is very rich in phenomena and exemplifies very well how the algorithm works. Figure 7 shows clearly the fast increase of the fidelity until reaching values above 99.9% with just nine iterations. Initially, there is an exploration stage that makes the fidelity grow up to 99.9% with just nine iterations. At the same time, the exploration range, Δ, grows, making two consecutive peaks and then it decreases smoothly, while the fidelity remains constant. Subsequently, there are a couple of exploration peaks that make the fidelity oscillate. Now, after a few iterations where the fidelity decreases smoothly, we come to a third and most important exploration phase where we observe how the fidelity has an increasing tendency. It suffices to check that the subsequent minima of the fidelity take larger and larger values. Such an amount of exploration has as a long-term reward a fidelity of 99.99% after fewer than 70 iterations. However, the exploration range is still large and leaves room for trying new states which make the fidelity drop again. Finally, the algorithm is able to find a good exploration-exploitation balance which makes the fidelity increase and remain constant with values above 99.5%. On the top of that, the exploration range goes progressively to zero. The ideal experiment is an excellent example of how quickly the algorithm could reach a high fidelity above 99.9% and also guarantee the convergence of $\Delta \to 0$. In this way, once the exploration range has become so small, it is assured that the agent does not go to another state. In other words, it is proved that the agent has definitely converged to the environment to a large fidelity.

In Table 1, we sort some results of the experiments run in Rigetti 8Q-Agave cloud quantum computer from larger to smaller value of the fidelity. As we can see, $\mathcal{F}$ is close to 100% in most of the experiments when the exploration ratio Δ is approaching zero. Thus, we are succeeding in adapting the agent to the environment state. From these data, we can also draw the conclusion that the convergence of the agent to the environment state is guaranteed whenever $\Delta \to 0$. We did not find any case where the exploration range is close to zero and the fidelity below 90%.

Table 1. Value of the fidelity for $\Delta \to 0$.

Δ	0.36	0.24	0.18	0.03	0.05	0.24	0.16
$\mathcal{F}$ (%)	99.89	99.72	99.53	99.20	97.72	97.53	94.72
Initial environment state	$\lvert \varepsilon_6 \rangle$	$\lvert \varepsilon_2 \rangle$	$\lvert \varepsilon_1 \rangle$	$\lvert \varepsilon_3 \rangle$	$\lvert \varepsilon_4 \rangle$	$\lvert \varepsilon_1 \rangle$	$\lvert \varepsilon_3 \rangle$

Figure 7. Measurement-based adaptation protocol for the environment state $\lvert \varepsilon_6 \rangle_E$. The blue solid line corresponds to the real experiment and the red dashed-dotted line represents the ideal simulation. The fidelity is given in percentage (%).

3. Discussion

In this study, we performed the implementation of the measurement-based adaptation protocol with quantum reinforcement learning of [1]. Consistently with it, we checked that indeed the fidelity increases reaching values over 90% with fewer than about 30 iterations in real experiments. We did not observe any case where $\Delta \to 0$ and $\mathcal{F} < 90$%. Thus, to a large extent, the protocol succeeds in making the agent converge to the environment. If we wanted to apply this algorithm to any subroutine, it would be possible to track the evolution of the exploration range and deduce from it the convergence of the agent to the environment. This is because the behaviour of Δ has proven to be closely related to the fidelity performance.

We can conclude that there is still a long way to be travelled until the second quantum revolution gives rise to well-established techniques of quantum machine learning in the lab. However, this work is encouraging since it sows the seeds for turning quantum reinforcement learning into a reality, and the future implementation of semiautonomous quantum agents.

We point out that another implementation of [1] in a different platform, namely quantum photonics, has been recently achieved, in parallel to this work [36]. The quantum photonics platform is more suited to quantum communication, while the current one, superconducting circuits, is more suited to quantum computation. On the other hand, there are interesting efforts for building quantum computers based on quantum photonics that exploit quantum machine learning techniques [32]. The importance of these two experiments of the proposal in [1] is the realisation that this proposal works well irrespective of the quantum platform in which it is implemented. Both experiments are complementary and it is valuable to show their good performance for different quantum technologies. The impact of quantum machine learning, and in particular, quantum reinforcement learning, in industry could be large in the short to mid term, and companies such as Xanadu [32], D-Wave [33], IBM [34] and Rigetti [35] are strongly investing on these efforts.

Conflicts of Interest: The authors declare no competing interests.

References

1. Albarrán-Arriagada, F.; Retamal, J.C.; Solano, E.; Lamata, L. Measurement-based adaptation protocol with quantum reinforcement learning. *Phys. Rev. A* **2018**, *98*, 042315. [CrossRef]

2. Albarrán-Arriagada, F.; Retamal, J.C.; Solano, E.; Lamata, L. Reinforcement learning for semi-autonomous approximate quantum eigensolver. *Mach. Learn. Sci. Technol.* **2020**, *1*, 015002. [CrossRef]

3. Harrow, A.W.; Hassidim, A.; Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **2009**, *103*, 150502. [CrossRef] [PubMed]

4. Wiebe, N.; Braun, D.; Lloyd, S. Quantum algorithm for data fitting. *Phys. Rev. Lett.* **2012**, *109*, 050505. [CrossRef]

5. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. *arXiv* **2013**, arXiv:1307.0411.

6. Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **2014**, *113*, 130503. [CrossRef]

7. Wiebe, N.; Kapoor, A.; Svore, K.M. Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *Quantum Inf. Comput.* **2015**, *15*, 316.

8. Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **2014**, *10*, 631. [CrossRef]

9. Lau, H.K.; Pooser, R.; Siopsis, G.; Weedbrook, C. Quantum machine learning over infinite dimensions. *Phys. Rev. Lett.* **2017**, *118*, 080501. [CrossRef]

10. Schuld, M.; Sinayskiy, I.; Petruccione, F. An introduction to quantum machine learning. *Contemp. Phys.* **2015**, *56*, 172. [CrossRef]

11. Adcock, J.; Allen, E.; Day, M.; Frick, S.; Hinchli, J.; Johnson, M.; Morley-Short, S.; Pallister, S.; Price, A.; Stanisic, S. Advances in quantum machine learning. *arXiv* **2015**, arXiv:1512.02900.

12. Dunjko, V.; Taylor, J.M.; Briegel, H.J. Quantum-enhanced machine learning. *Phys. Rev. Lett.* **2016**, *117*, 130501. [CrossRef] [PubMed]

13. Dunjko, V.; Briegel, H.J. Machine learning & artificial intelligence in the quantum domain: A review of recent progress. *Rep. Prog. Phys.* **2018**, *81*, 074001. [PubMed]

14. Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195. [CrossRef]

15. Lamata, L. Quantum machine learning and quantum biomimetics: A perspective. *arXiv* **2020**, arXiv:2004.12076.

16. Biswas, R.; Jiang, Z.; Kechezhi, K.; Knysh, S.; Mandrà, S.; O'Gorman, B.; Perdomo-Ortiz, A.; Petukhov, A.; Realpe-Gómez, J.; Rieffel, E.; et al. A NASA perspective on quantum computing: Opportunities and challenges. *Parallel Comput.* **2017**, *64*, 81. [CrossRef]

17. Perdomo-Ortiz, A.; Benedetti, M.; Realpe-Gómez, J.; Biswas, R. Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quantum Sci. Technol.* **2018**, *3*, 030502. [CrossRef]

18. Perdomo-Ortiz, A.; Feldman, A.; Ozaeta, A.; Isakov, S.V.; Zhu, Z.; O'Gorman, B.; Katzgraber, H.G.; Diedrich, A.; Neven, H.; de Kleer, J.; et al. Readiness of quantum optimization machines for industrial applications. *Phys. Rev. Appl.* **2019**, *12*, 014004. [CrossRef]

19. Sasaki, M.; Carlini, A. Quantum learning and universal quantum matching machine. *Phys. Rev. A* **2002**, *66*, 022303. [CrossRef]

20. Benedetti, M.; Realpe-Gómez, J.; Biswas, R.; Perdomo-Ortiz, A. Quantum-Assisted Learning of Hardware-Embedded Probabilistic Graphical Models. *Phys. Rev. X* **2017**, *7*, 041052. [CrossRef]

21. Benedetti, M.; Realpe-Gómez, J.; Perdomo-Ortiz, A. Quantum-assisted helmholtz machines: A quantum-classical deep learning framework for industrial datasets in near-term devices. *Quantum Sci. Technol.* **2018**, *3*, 034007. [CrossRef]

22. Aïmeur, E.; Brassard, G.; Gambs, S. Quantum speed-up for unsupervised learning. *Mach. Learn.* **2013**, *90*, 261. [CrossRef]

23. Melnikov, A.A.; Nautrup, H.P.; Krenn, M.; Dunjko, V.; Tiersch, M.; Zeilinger, A.; Briegel, H.J. Active learning machine learns to create new quantum experiments. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 1221. [CrossRef] [PubMed]

24. Alvarez-Rodriguez, U.; Lamata, L.; Escandell-Montero, P.; Martín-Guerrero, J.D.; Solano, E. Supervised Quantum Learning without Measurements. *Sci. Rep.* **2017**, *7*, 13645. [CrossRef]

25. Paparo, G.D.; Dunjko, V.; Makmal, A.; Martin-Delgado, M.A.; Briegel, H.J. Quantum Speedup for Active Learning Agents. *Phys. Rev. X* **2014**, *4*, 031002. [CrossRef]

26. Lamata, L. Basic protocols in quantum reinforcement learning with superconducting circuits. *Sci. Rep.* **2017**, *7*, 1609. [CrossRef]

27. Cárdenas-López, F.A.; Lamata, L.; Retamal, J.C.; Solano, E. Multiqubit and multilevel quantum reinforcement learning with quantum technologies. *PLoS ONE* **2018**, *13*, e0200455. [CrossRef]

28. Dong, D.; Chen, C.; Li, H.; Tarn, T.J. Quantum reinforcement learning. *IEEE Trans. Syst. Man Cybern. Part B (Cybern.)* **2008**, *38*, 1207–1220. [CrossRef]

29. Crawford, D.; Levit, A.; Ghadermarzy, N.; Oberoi, J.S.; Ronagh, P. Reinforcement learning using quantum Boltzmann machines. *Quant. Inf. Comput.* **2018**, *18*, 51.

30. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Prentice Hall: Upper Saddle River, NJ, USA, 1995.

31. Ristè, D.; Silva, M.P.D.; Ryan, C.A.; Cross, A.W.; Córcoles, A.D.; Smolin, J.A.; Gambetta, J.M.; Chow, J.M.; Johnson, B.R. Demonstration of quantum advantage in machine learning. *NPJ Quantum Inf.* **2017**, *3*, 16. [CrossRef]

32. Quantum Computing Powered by Light. Available online: https://www.xanadu.ai (accessed on 16 May 2020).

33. The Quantum Cloud Service Built for Business. Available online: https://www.dwavesys.com/home (accessed on 16 May 2020).

34. Quantum Computing. Available online: https://www.research.ibm.com/ibm-q/ (accessed on 16 May 2020).

35. Think Quantum. Available online: https://www.rigetti.com (accessed on 16 May 2020).

36. Yu, S.; Albarrán-Arriagada, F.; Retamal, J.C.; Wang, Y.-T.; Liu, W.; Ke, Z.-J.; Meng, Y.; Li, Z.-P.; Tang, J.-S.; Solano, E.; et al. Reconstruction of a Photonic Qubit State with Quantum Reinforcement Learning. *Adv. Quantum Technol.* **2019**, *2*, 1800074. [CrossRef]