scientific reports

Check for updates

OPEN

Experimental semi-autonomous eigensolver using reinforcement learning

C.-Y. Pan[1], M. Hao[1], N. Barraza[1], E. Solano[1,2,3,4]✉ & F. Albarrán-Arriagada[1]✉

The characterization of observables, expressed via Hermitian operators, is a crucial task in quantum mechanics. For this reason, an eigensolver is a fundamental algorithm for any quantum technology. In this work, we implement a semi-autonomous algorithm to obtain an approximation of the eigenvectors of an arbitrary Hermitian operator using the IBM quantum computer. To this end, we only use single-shot measurements and pseudo-random changes handled by a feedback loop, reducing the number of measures in the system. Due to the classical feedback loop, this algorithm can be cast into the reinforcement learning paradigm. Using this algorithm, for a single-qubit observable, we obtain both eigenvectors with fidelities over 0.97 with around 200 single-shot measurements. For two-qubits observables, we get fidelities over 0.91 with around 1500 single-shot measurements for the four eigenvectors, which is a comparatively low resource demand, suitable for current devices. This work is useful to the development of quantum devices able to decide with partial information, which helps to implement future technologies in quantum artificial intelligence.

Increasing the computational capabilities of machines is an essential field in artificial intelligence. In this context, machine learning algorithms have emerged with great force in the last decades[1,2]. This class of algorithms can be divided into two families, learning from big data and learning from interactions. Learning from big data can be classified into two categories, supervised and unsupervised learning. In the supervised learning paradigm, we have a set of labeled data named training data, from which we want to infer some classification function to sort unlabeled new data. Unsupervised learning algorithms do not use training data. In this paradigm, the goal is to extract the statistical structure of an unsorted data set and divide it into different groups according to some criteria (clustering problem)[3–8].

In the category of learning from interactions we have the Reinforcement Learning (RL) algorithms[9–18]. The idea in this paradigm is that a known and manipulable system called *agent* (A) interacts with a non-manipulable system called *environment* (E). Here, the goal is to optimize a task $\mathscr{G}(A, E)$, which depends on the state of A and E. For this, we use feedback loops to change the state of A using the information extracted from the interaction with E. Some impressive and recent examples of RL are the AI players for different strategy games like Go[19], Chess[20], or StarCraft II[21].

On the other hand, it has been shown that quantum computing[22] can overcome some fundamental limits of classical computing, e.g., in searching problems[23], factorization algorithms[24], solving linear equation systems[25,26], and for linear differential equations[27]. Therefore, it was natural to merge machine learning techniques with the advantages of quantum computing in the topic known as Quantum Machine Learning (QML)[28–35].

With the development of Noisy Intermediate-Scale Quantum (NISQ) devices[36], the research on simple quantum information protocol (suitable for NISQ quantum computers) and in QML has grown in the last years. The IBM quantum computer is one of the most famous open NISQ devices, which can be programmed using Qiskit[37], an open-source python package, to create and run quantum programs using the IBM quantum cloud service[38].

One of the most useful algorithms for linear algebra, and hence for quantum mechanics, are the quantum eigensolvers. The hybrid quantum-classical algorithms like variational quantum eigensolver (VQE)[39–41] take advantage due to its easy implementation in NISQ devices. The main idea of this class of algorithm is to calculate some expectation value (like energy) with a quantum processor, and then use a classical optimizer (like variational one) to reach the solution[42]. Nevertheless, it has been recently proposed an algorithm that uses a quantum

[1]International Center in Quantum Artificial Intelligence for Science and Technology (QuArtist) and Physics Department, Shanghai University, Shanghai 200444, China. [2]Department of Physical Chemistry, University of the Basque Country UPV/EHU, Apartado 644, 48080 Bilbao, Spain. [3]IKERBASQUE, Basque Foundation for Science, Plaza Euskadi 5, 48009 Bilbao, Spain. [4]Kipu Quantum, Kurwenalstrasse 1, 80804 Munich, Germany. ✉email: enr.solano@gmail.com; pancho.albarran@gmail.com

optimizer[43]. Each iteration of the classical optimizer algorithm involves many single-shot measurements in the quantum system, which are required to calculate an expectation value. The development of an algorithm with more quantum features will involve the use of a more primitive classical subroutine.

In this paper, we implement the semi-autonomous eigensolver proposed in Ref.[44]. The protocol can obtain an approximation of all eigenvectors for an arbitrary observable using single-shot measurements instead of expectation values. Here, we use the most basic classical subroutine, which involves only pseudo-random changes handled by the outcome of the single-shot measurement and a feedback loop. Due to this feedback loop, this algorithm can be classified in the RL paradigm. Using our protocol, we can obtain a high fidelity approximation for all eigenvectors. In the single-qubit case, we get fidelities larger than 0.97 and larger than 0.91 for a two-qubit observable in around 200 and 5000 single-shot measurements, respectively. This work opens the door to explore alternative paradigms in hybrid classical-quantum algorithms, which is useful for developing semi-autonomous quantum devices that decide with incomplete information.

Methods

Basics on RL paradigm. We briefly describe the basic components of the RL paradigm. As mentioned above, in an RL algorithm, we define two systems: the agent A and the environment E. The interaction among these systems can be divided in three basic steps, the *policy*, the *reward function* (RF) and the *value function* (VF). The policy refers to the general rules of the algorithm and can be subdivided into three stages: first, the interaction, where we specify how A and E interact; second, the action, which refers to how A changes its perception of E modifying some internal parameters; and third, the information extraction, that defines the process used by A to infer information from E. The information extraction can be done directly by A or using an auxiliary system, named *register*, if A cannot read the response of the environment.

The RF is the criterion to reward or punish A in each iteration using the information collected from E. This step is the most important in any RL algorithm because the right choice of the RF ensures the optimization of the desired task $\mathscr{G}(A, E)$. Finally, the VF evaluates a figure of merit related to the task $\mathscr{G}(A, E)$, which provides us the utility of the algorithm. The main difference between RF and VF is that the first evaluates each iteration to increase the performance locally in time without considering the history of the algorithm. At the same time, VF depends on the history of the algorithm, which takes into consideration a large number of iterations given the global performance of the algorithm.

RL protocol. We define the basic parts of our protocol as an RL algorithm. The state of the agent is denoted by

$$|\mathscr{A}_k^{(j)}\rangle = \hat{D}_k|j\rangle, \tag{1}$$

where $\hat{D}_k$ is a unitary transformation to prepare the desired agent state, the state $|j\rangle$ is the initial state provided by the quantum processor in the computational basis, and the subindex k denotes the iteration of the algorithm. The environment is expressed as an unknown Hermitian operator $\hat{\mathscr{O}}$ written as

$$\hat{\mathscr{O}} = \sum_j \alpha^{(j)} |\mathscr{E}^{(j)}\rangle\langle\mathscr{E}^{(j)}|, \tag{2}$$

with $\alpha^{(j)}$ and $|\mathscr{E}^{(j)}\rangle$ the jth eigenvalue and eigenvector of $\hat{\mathscr{O}}$, respectively. The task $\mathscr{G}$ is set to maximize the fidelity between the state of the agent, $|\mathscr{A}_N^{(j)}\rangle$, after N iterations, and the eigenvectors $|\mathscr{E}^{(j)}\rangle$, or in other words, we want to find the matrix $\hat{D}_k$ that diagonalizes the observable $\mathscr{O}$.

Now, the policy is as follows:

Interaction: The observable $\hat{\mathscr{O}}$ generates an evolution given by the unitary transformation

$$\hat{E} = e^{-i\hat{\mathscr{O}}\tau} = \sum_j e^{-i\alpha^{(j)}\tau} |\mathscr{E}^{(j)}\rangle\langle\mathscr{E}^{(j)}|, \tag{3}$$

where τ is a constant related with the elapsed time of the interaction. The agent state after this evolution is

$$\hat{E}|\mathscr{A}_k^{(j)}\rangle = |\bar{\mathscr{A}}_k^{(j)}\rangle = \sum_\ell c^{(\ell)} |\mathscr{A}_k^{(\ell)}\rangle. \tag{4}$$

Information extraction: We measure the state $|\bar{\mathscr{A}}_k^{(j)}\rangle$ in the basis $\{|\mathscr{A}_k^{(\ell)}\rangle\}$. For this purpose we apply the transformation $\hat{D}_k^\dagger$ obtaining

$$\hat{D}_k^\dagger |\bar{\mathscr{A}}_k^{(j)}\rangle = \sum_\ell c^{(\ell)} |\ell\rangle, \tag{5}$$

followed by a single-shot measurement in the computational basis $\{|\ell\rangle\}$ obtaining the outcome value m with probability $|c^{(m)}|^2$. This outcome refers to the resulting state $|\mathscr{A}_k^{(m)}\rangle$ after the measuring process.

Action: According to Eq. (3) if $|\mathscr{A}_k^{(j)}\rangle$ is equal to some eigenvector of $\hat{\mathscr{O}}$, we obtain $c^{(j)} = 1$ in Eq. (4). Using this condition we define the next rule for the action. If the outcome is $m \neq j \Rightarrow c^{(j)} \neq 1$, then $|\mathscr{A}_k^{(j)}\rangle$ is not an eigenvector of $\hat{\mathscr{O}}$. In this case ($m \neq j$), we modify the agent for the next iteration defining operator $\hat{D}_{k+1}$ as

$$\hat{D}_{k+1} = \hat{D}_k \hat{u}_{j,m}(\theta, \phi, \lambda), \tag{6}$$

with

$$\hat{u}_{j,m}(\theta,\phi,\lambda) = e^{-i\lambda\hat{S}^{(z)}_{j,m}} e^{-i\theta\hat{S}^{(y)}_{j,m}} e^{-i\phi\hat{S}^{(z)}_{j,m}}, \tag{7}$$

where,

$$\begin{aligned} S^{(z)}_{j,m} &= \frac{1}{2}\big(|j\rangle\langle j| - |m\rangle\langle m|\big), \\ S^{(y)}_{j,m} &= -\frac{i}{2}\big(|j\rangle\langle m| - |m\rangle\langle j|\big). \end{aligned} \tag{8}$$

Then,

$$\begin{aligned} \hat{u}_{j,m}(\theta,\phi,\lambda) = \cos\left(\frac{\theta}{2}\right)\Big(|j\rangle\langle j| + e^{i(\lambda+\phi)}|m\rangle\langle m|\Big) \\ + \sin\left(\frac{\theta}{2}\right)\Big(-e^{i\phi}|j\rangle\langle m| + e^{i\lambda}|m\rangle\langle j|\Big) \end{aligned} \tag{9}$$

up to a global phase. Therefore, $\hat{u}(\theta,\phi,\lambda)$ is a general rotation in the $\{|j\rangle, |m\rangle\}$ subspace. The angles are random numbers given by

$$\{\theta,\lambda,\phi\} \in w_k \cdot [-\pi,\pi], \tag{10}$$

where the range amplitude w_k will be updated in each iteration according to the RF, which will be specified later. Now, for the case $m = j$, the state $|\mathscr{A}_k^{(j)}\rangle$ could be an eigenvector of $\hat{\mathscr{O}}$, then we define

$$\hat{D}_{k+1} = \hat{D}_k. \tag{11}$$

We can summarize Eqs. (6) and (11) as

$$\hat{D}_{k+1} = \hat{D}_k\left[\sum_{l\neq j}\hat{u}_{l,m}(\theta,\phi,\lambda)\cdot\delta_{l,m} + \mathbb{I}\cdot\delta_{j,m}\right]. \tag{12}$$

Now, we define the reward function as

$$w_{k+1} = w_k\left[p\cdot\sum_{l\neq j}\delta_{l,m} + r\cdot\delta_{j,m}\right] \tag{13}$$

where $p > 1$ is the punishment ratio, and $0 < r < 1$ is the reward ratio. This means that each time we obtain the outcome $m \neq j$, we increase the amplitude range w_{k+1}, because $m \neq j$ means that we are further away from an eigenvector and greater corrections are required. In the other case, when $m = j$ means that we are closer to an eigenvector, then, we reduce the value of w_{k+1} obtaining smaller changes for future iterations.

Finally, the value function will be the last value of the range amplitude w_N after N iterations. If $w_N \to 0$ signifies that we have measured $m = j$ several times, then $c^{(j)} \approx 1$, which implies that we obtain a good approximation of an eigenvector.

Results

Single-qubit case. We implement the algorithm described above in the IBM quantum computer. We start with the simplest case, which is to find the eigenvectors of a single-qubit observable. Since there are only two eigenvectors, we only need to obtain one of them, because the orthogonality property can determine the second one. Figure 1 shows the circuit diagram for this case. As we can see in Fig. 1 the agent in each iteration is given by

$$|\mathscr{A}_k^{(0)}\rangle = \hat{D}_k|0\rangle. \tag{14}$$

In this case, we have only one the rotation ($\hat{u}_{1,0}$) of the form of Eq. (7), then, for simplicity, we redefine the operator $\hat{D}_k = \hat{D}(\theta_k,\phi_k,\lambda_k)$ as

$$\hat{D}(\theta_k,\phi_k,\lambda_k) = e^{-i\frac{\lambda_k}{2}\hat{\sigma}^{(z)}} e^{-i\frac{\theta_k}{2}\hat{\sigma}^{(y)}} e^{-i\frac{\phi_k}{2}\hat{\sigma}^{(z)}}, \tag{15}$$

where $\hat{\sigma}^{(a)}$ is the a-Pauli matrix and

$$\begin{aligned} \theta_{k+1} &= \theta_k + \Delta_\theta\cdot\delta_{1,m}, \\ \phi_{k+1} &= \phi_k + \Delta_\phi\cdot\delta_{1,m}, \\ \lambda_{k+1} &= \lambda_k + \Delta_\lambda\cdot\delta_{1,m}, \end{aligned} \tag{16}$$

with $\{\Delta_\theta,\Delta_\phi,\Delta_\lambda\} \in w_k[-\pi,\pi]$ and w_k given by Eq. (13), considering only two outcomes ($m \in \{0,1\}$) and $j = 0$ for the whole algorithm. The gate in Eq. (15) has the form of the general qubit-rotation provided by qiskit, therefore, it can be efficiently implemented in the IBM quantum computer. We denote by, $\mathscr{F}$, the maximum

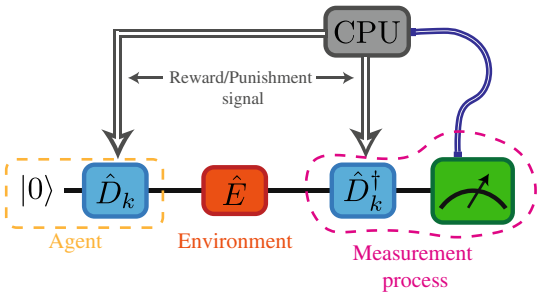

Figure 1. Diagram of the single-qubit protocol. The subindex k refers to the kth iteration. Blue lines represent the classical communication to the central processing unit. The gray arrows show feedback loops, where $\hat{D}_k$ and $\hat{D}_k^\dagger$ are updated according to the measurement outcome.

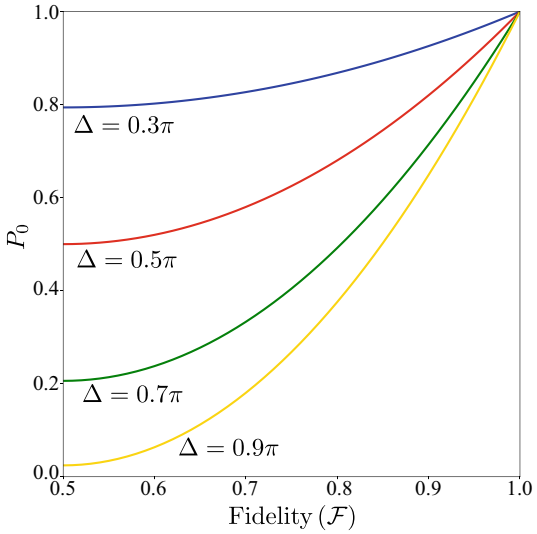

Figure 2. P_0 as a function of $\mathscr{F}$ for different values of Δ.

fidelity between the agent state, $|\mathscr{A}_N^{(0)}\rangle$, and one of the eigenvectors at the end of the algorithm. We find that $\mathscr{F}$ is related to the probability of obtaining the outcome $m = 0$ (P_0) by (see appendix A)

$$\begin{aligned} P_0 &= \frac{1-\cos(\Delta)}{2}\left[(2\mathscr{F}-1)^2-1\right]+1 \\ \Rightarrow \mathscr{F} &= \frac{1}{2}\left(1+\sqrt{\frac{2(P_0-1)}{1-\cos\Delta}+1}\right), \end{aligned} \tag{17}$$

where $\Delta = \tau|\alpha^{(0)} - \alpha^{(1)}|$ is the gap between the eigenvalues of $\tau\hat{\mathscr{O}}$ [see Eqs. (2) and (3)]. Figure 2 shows P_0 as a function of the fidelity $\mathscr{F}$ for different values of Δ.

For the implementation we use the initial values $\theta_1 = \phi_1 = \lambda_1 = 0$, $w_1 = 1$ and the quantum processor "ibmqx2". The algorithm is run until $w_N < 0.1$. Since the algorithm converges stochastically to the eigenvectors, we perform 40 experiments in order to characterize the performance of the algorithm by the central values of the data set. Also, we compare the performances of our algorithms with the *VQE* algorithm for the same environments using the same quantum processor. To test the algorithm, we use three different environment Hermitian operators:

1. $\tau\hat{\mathscr{O}} = \frac{\pi}{2}\sigma_x \Rightarrow \Delta = \pi \Rightarrow \mathscr{F} = \frac{1}{2}(1+\sqrt{P_0})$.

Here, we choose the reward ratio $r = 0.9$ and the punishment ratio $p = 1/r$. The results of the 40 experiments are collected in the Apendix Table 1 (Supplemental material) and summarized in the histograms of Fig. 3. From Fig. 3a, we can see that the probability P_0 is bigger than 0.85 in 36 cases, which implies, as is shown in Fig. 3b, that most cases give fidelities larger than 0.94. Also, we have 36 experiments with $\mathscr{F} > 0.96$, the average fidelity is $\bar{\mathscr{F}} = 0.98$ and the standard deviation is $\sigma = 0.019$ which represent the 2% of the average fidelity $\bar{\mathscr{F}}$. Also, the average number of iterations of the algorithm in the 40 experiments is $\bar{N} = 103$, the minimum number of

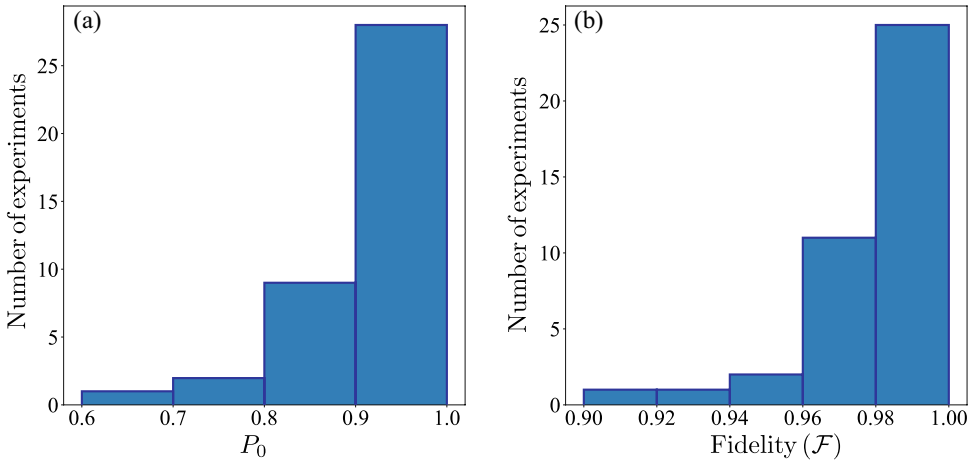

Figure 3. Histograms for the results of 40 independent experiments. with $\tau\hat{\mathcal{O}} = \frac{\pi}{2}\sigma_x, r = 0.9$ and $p = 1/r$. (**a**) Histogram for the probability to obtain $m = 0$. (**b**) Histogram for the fidelity between the agent and the nearest eigenvector using Eq. (17).

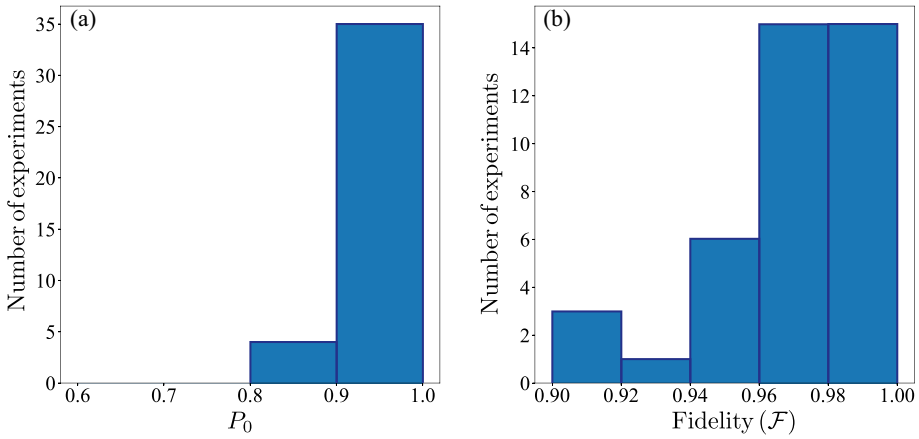

Figure 4. Histograms for the results of 40 independent experiments. with $\tau\hat{\mathcal{O}} = \frac{\pi}{4}\sigma_x, r = 0.9$ and $p = 1.5/r$. (**a**) Histogram for the probability to obtain $m = 0$. (**b**) Histogram for the fidelity between the agent and the nearest eigenvector using Eq. (17).

iterations $N_{min} = 25$, and the maximum number of iterations $N_{max} = 528$. This number may look large, but we remark that we using only one single-shot measurement per iteration. In comparison, if we want to calculate a given expectation value, we require at least 1000 single-shot measurements for a single qubit. Then for this case, our algorithm requires less resources than any other classical-quantum algorithm that utilizes expectation values. For the VQE algorithm, first we choose 500 single-shot measurements per step and COBYLA as the classical optimization method. VQE needs 33 COBYLA iterations to converge, which means 16500 single-shot measurements in total, i.e.100 times the resources needed in our algorithm, and get a fidelity of 0.997. If we change the number of single-shot measurements to 8192 per step (it is the maximum shots allowed by IBM), we need 35 COBYLA iterations to converge, which means 286720 single-shot measurements, 1000 times more resources than our algorithms, nevertheless, the fidelity is 0.999.

2. $\tau\hat{\mathcal{O}} = \frac{\pi}{4}\sigma_x \Rightarrow \Delta = \frac{\pi}{2} \Rightarrow \mathscr{F} = \frac{1}{2}(1 + \sqrt{2P_0 - 1})$.

Now, we choose the reward ratio $r = 0.9$ and the punishment ratio $p = 1.5/r$. The results of the 40 experiments are collected in the Appendix Table 2 (see supplemental material) and summarized in the histograms of Fig. 4. From Fig. 4a we can see that the probability P_0 is bigger than 0.9 in 35 cases, which implies, as is shown in Fig. 4b, that most cases give fidelities larger than 0.94. Also, we have 30 experiments with $\mathscr{F} > 0.96$, the average fidelity is $\bar{\mathscr{F}} = 0.97$ and the standard deviation is $\sigma = 0.022$ which represent the 2.3% of the average fidelity $\bar{\mathscr{F}}$. Also, the average number of iterations of the algorithm in the 40 experiments is $\bar{N} = 116$, the minimum number of iterations $N_{min} = 25$ and the maximum number of iterations $N_{max} = 572$, again for this case our algorithm

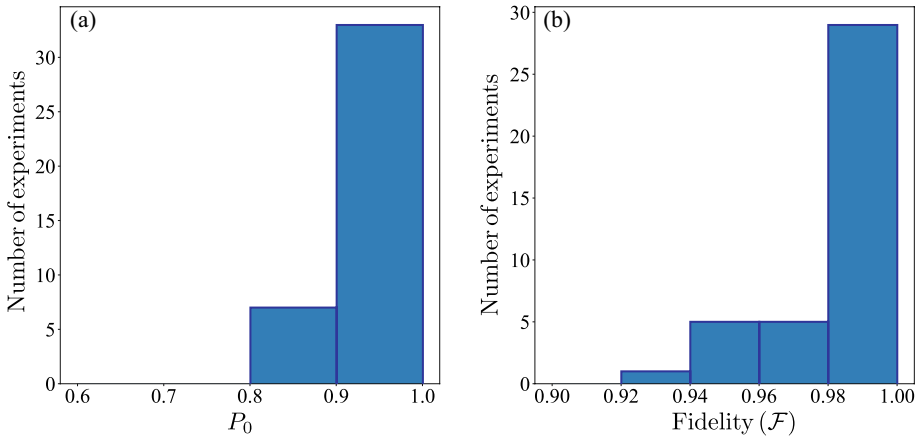

Figure 5. Histograms for the results of 40 independent experiments. with $\tau\hat{\mathcal{O}} = \cos\frac{1}{10}\sigma_x + \sin\frac{1}{10}\sigma_y, r = 0.9$ and $p = 1.5/r$. (**a**) Histogram for the probability to obtain $m = 0$. (**b**) Histogram for the fidelity between the agent and the nearest eigenvector using Eq. (17).

uses less resources than the algorithm that use expectation values. As in the previous case, we compare the results with the VQE algorithm. For 500 shots per step, we get a fidelity of 0.883 with 23 COBYLA iterations, which means 11500 single-shot measurements, i.e.100 times more resources than our algorithm. For 8192 shots per step, the fidelity is 0.891 and we need 23 COBYLA iterations, the total single-shot measurements are 188416, i.e.1000 times more resources than in our algorithm.

3.
$$\tau\hat{\mathcal{O}} = \cos\frac{1}{10}\sigma_x + \sin\frac{1}{10}\sigma_y \Rightarrow \Delta = 2$$
$$\Rightarrow \mathscr{F} = \frac{1}{2}\left(1 + \sqrt{1 + \frac{2(P_0 - 1)}{1 - \cos 2}}\right)$$

We choose the reward ratio $r = 0.9$ and the punishment ratio $p = 1.5/r$ as in the previous case. The results of the 40 experiments are collected in the Appendix Table 3 (see supplemental material) and summarized in the histograms of Fig. 5. From Fig. 5a we can see that the probability P_0 is bigger than 0.85 in 39 cases, which implies, as is shown in Fig. 5b, that most cases give fidelities larger than 0.94. Also, we have 30 experiments with $\mathscr{F} > 0.98$, the average fidelity is $\bar{\mathscr{F}} = 0.98$ and the standard deviation of $\sigma = 0.015$ which represent the 1.6% of the average fidelity $\bar{\mathscr{F}}$. Also, the average number of iterations of the algorithm in the 40 experiments was $\bar{N} = 227$, the minimum number of iterations $N_{min} = 26$ and the maximum number of iterations $N_{max} = 782$. In this case, as N_{max} is around 800, we compare the VQE algorithm, at first with 800 shots per step, obtaining a fidelity of 0.911 using 14 COBYLA iterations, which means, a total number of single-shot measurements of 11200, i.e.50 times more resources than our algorithms. When we use 8192 per step, the fidelity is 0.999 and we need 14 COBYLA iterations, obtaining a total number of single-shot measurements of 114688, i.e.500 times more resources than our algorithm.

Even if VQE allows us to reach fidelities larger than 0.98 (the mean fidelity of our algorithm), it needs several resources, more than 100 times the resources using by our algorithm, which implies a great advantage of our proposal.

Two-qubit case. In this case, we have three different agent states given by

$$\begin{aligned} |\mathscr{A}_k^{(0)}\rangle &= \hat{D}_k|00\rangle, \\ |\mathscr{A}_k^{(1)}\rangle &= \hat{D}_k|01\rangle, \\ |\mathscr{A}_k^{(2)}\rangle &= \hat{D}_k|10\rangle. \end{aligned} \tag{18}$$

We update the matrix $\hat{D}_k$ according to Eq. (12). To decompose the matrix $\hat{D}_k$ in a set of one- and two-qubit gates, we use the method already implemented in qiskit[45]. To find all the eigenvectors we divide the protocol in three stages. In the first stage, we consider the agent state $|\mathscr{A}_k^{(0)}\rangle = \hat{D}_k|00\rangle$, with $\hat{D}_1 = \mathbb{I}$ and $w_1 = 1$. The outcome of the measure have four possibilities $m \in \{00,\ 01,\ 10,\ 11\}$ and we run the algorithm until $w_{n_1} < 0.1$ (n_1 iterations). After this, we have that $|A_{n_1}^{(0)}\rangle = \hat{D}_{n_1}|00\rangle$ is the approximation of one of the eigenvectors of $\hat{\mathcal{O}}$.

In the second stage, we consider the agent state $|\mathscr{A}_k^{(1)}\rangle = \hat{D}_k|01\rangle$, with $\hat{D}_{n_1+1} = \hat{D}_{n_1}$ and $w_{n_1+1} = 1$. Now, we take into account only three outcome $m \in \{01,\ 10,\ 11\}$, since we suppose that $|\mathscr{A}_{N_1}^{(0)}\rangle$ is a good enough approximation. If we obtain $m = 00$, we consider it as an error, and we define $\hat{D}_{k+1} = \hat{D}_k$ and $w_{k+1} = w_k$, it means that we do nothing, and not apply the updating rule for $\hat{D}_{k+1}$ and w_{k+1}, we denote this error as c_{00}. We run this stage n_2 iterations until $w_{n_1+n_2} < 0.1$. As we do not do rotations in the subspace spanned by $\{|00\rangle,\ |01\rangle\}$ during this

stage, we have $|\mathscr{A}^{(0)}_{n_1+n_2}\rangle = |\mathscr{A}^{(0)}_{n_1}\rangle$. Now, we obtain the approximation of two eigenvectors $|A^{(1)}_{n_1+n_2}\rangle = \hat{D}_{n_1+n_2}|01\rangle$ and $|A^{(0)}_{n_1+n_2}\rangle = \hat{D}_{n_1+n_2}|00\rangle$.

Finally, in the third stage, we consider the agent state $|\mathscr{A}^{(2)}_k\rangle = \hat{D}_k|10\rangle$, with $\hat{D}_{n_1+n_2+1} = \hat{D}_{n_1+n_2}$ and $w_{n_1+n_2+1} = 1$. Now, we have only two possibilities for the outcome measurement $m \in \{10, 11\}$. Here, we also suppose that $\hat{D}_{n_1+n_2}|00\rangle$ and $\hat{D}_{n_1+n_2}|01\rangle$ are good enough approximations. If we obtain $m = 00$ or $m = 01$, we consider them again as an error and we do not apply the update rule, denoting these errors as $c_{00}^{'}$ and c_{01}, like in the previous stage. We run this case n_3 iterations until $w_{n_1+n_2+n_3} < 0.1$. In this stage, we only modify the subspace expanded by $\{|10\rangle, |11\rangle\}$, then, we have that $|\mathscr{A}^{(0)}_{n_1+n_2+n_3}\rangle = |\mathscr{A}^{(0)}_{n_1+n_2}\rangle = |\mathscr{A}^{(0)}_{n_1}\rangle$ and $|\mathscr{A}^{(1)}_{n_1+n_2+n_3}\rangle = |\mathscr{A}^{(1)}_{n_1+n_2}\rangle$. After this procedure we obtained the approximation of all the eigenvectors $\{|A^{(0)}_{n_T}\rangle = \hat{D}_{n_T}|00\rangle, |A^{(1)}_{n_T}\rangle = \hat{D}_{n_T}|01\rangle, |A^{(2)}_{n_T}\rangle = \hat{D}_{n_T}|10\rangle, |A^{(3)}_{n_T}\rangle = \hat{D}_{n_T}|11\rangle\}$, with $n_T = n_1 + n_2 + n_3$.

To test the algorithm, we choose three cases. First we consider the bi-local operator given by

$$1. \qquad \tau\hat{\mathscr{O}} = \sigma_x\sigma_x = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}. \tag{19}$$

In this case, the eigenstates and the eigenvalues are

$$\begin{aligned} |\mathscr{E}^{(0)}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle), \quad \alpha^{(0)} = -1, \\ |\mathscr{E}^{(1)}\rangle &= \frac{1}{\sqrt{2}}(-|01\rangle + |10\rangle), \quad \alpha^{(1)} = -1, \\ |\mathscr{E}^{(2)}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle), \quad \alpha^{(2)} = 1, \\ |\mathscr{E}^{(3)}\rangle &= \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle), \quad \alpha^{(3)} = 1. \end{aligned} \tag{20}$$

We note that the ground state is degenerate, then any linear state of the form $|\phi\rangle = a|\mathscr{E}^{(0)}\rangle + b|\mathscr{E}^{(1)}\rangle$ will be also ground state of the operator and the same for the other states. In this case we define the fidelity of our algorithm by the probability to measure the initial state $|j\rangle$

$$\mathscr{F}_j = P_j = |\langle j|\hat{D}^{\dagger}_{n_T}\hat{E}\hat{D}_{n_T}|j\rangle|^2. \tag{21}$$

We run this case using IBM backend "ibmq_vigo" and the results are shown in Appendix Table 4 (see supplemental material). In this case, we run the algorithm ten times and the mean fidelities are: $\mathscr{F}_{00} = 0.931$, $\mathscr{F}_{01} = 0.933$, $\mathscr{F}_{10} = 0.932$, and $\mathscr{F}_{11} = 0.919$. The mean number of iterations is $\bar{N} = 272$. In this case, the mean errors are: $\bar{c}_{00} = 10$, $\bar{c}^{'}_{00} = 8$ and $\bar{c}_{01} = 5$. Therefore, the fidelity of our algorithm was higher than 0.91 for each eigenstate in less than 300 single-shot measurements. The same as the single-qubit case, we will compare with the *VQE* algorithm. At first, we choose 300 shots per step, and 56 COBYLA iterations, which means 16800 single-shot measurements, obtaining a fidelity of 0.976 for the ground state. Using 8192 shots per step, VQE needs 54 COBYLA iterations to converge, which means 442368 single-shot measurements, obtaining a fidelity of 0.997 for the ground state. In this case, VQE get a significantly more accurate result, but it is only for the ground state and uses 1000 times more resources than our algorithm which obtain all the eigenvectors.

The second example is the molecular hydrogen Hamiltonian with a bound length of 0.2 [Å][46]:

$$H = g_0\mathbb{I} + g_1Z_0 + g_2Z_1 + g_3Z_0Z_1 + g_4Y_0Y_1 + g_5X_0X_1, \tag{22}$$

with $g_0 = 2.8489, g_1 = 0.5678, g_2 = -1.4508, g_3 = 0.6799, g_4 = 0.0791, g_5 = 0.0791$. In this case the environment is given by

$$2. \qquad \tau\hat{\mathscr{O}} = \begin{pmatrix} g_0+g_1+g_2+g_3 & 0 & 0 & g_5-g_4 \\ 0 & g_0+g_1-g_2-g_3 & g_4+g_5 & 0 \\ 0 & g_4+g_5 & g_0-g_1+g_2-g_3 & 0 \\ g_5-g_4 & 0 & 0 & g_0-g_1-g_2+g_3 \end{pmatrix}, \tag{23}$$

with the next eigenvectors and eigenvalues

$$\begin{aligned} |\mathscr{E}^{(0)}\rangle &= -0.03909568|01\rangle + 0.99923547|10\rangle, \quad \alpha^{(0)} = 0.14421033, \\ |\mathscr{E}^{(1)}\rangle &= |00\rangle, \quad \alpha^{(1)} = 2.6458, \\ |\mathscr{E}^{(2)}\rangle &= 0.99923547|01\rangle + 0.03909568|10\rangle, \quad \alpha^{(2)} = 4.19378967, \\ |\mathscr{E}^{(3)}\rangle &= |11\rangle, \quad \alpha^{(3)} = 4.4118 \end{aligned} \tag{24}$$

In this case, we choose the same method as the previous case to calculate the $\mathscr{F}$, we choose IBM backend "ibmq_valencia" and the results are shown in Appendix Table 5 (see supplemental material). In this case, we run the algorithm ten times and the mean fidelities are: $\mathscr{F}_{00} = 0.989$, $\mathscr{F}_{01} = 0.973$, $\mathscr{F}_{10} = 0.976$ and $\mathscr{F}_{11} = 0.979$. The mean errors are: $\bar{c}_{00} = 7$, $\bar{c}^{'}_{00} = 4$ and $\bar{c}_{01} = 3$ and the mean number of iterations is $\bar{N} = 111$. In this case, we need less than 150 single-shot measurements to obtain the fidelity over 0.97. For the VQE algorithm, at first we choose 120 shots per step and we need to use 59 COBYLA iterations, which means 7080 single-shot measurements,

obtaining a fidelity of 0.994 for the ground state. When we use 8192 shots per step and VQE needs 64 COBYLA iterations to converge, it means 507904 single-shot measurements, obtaining a fidelity of 0.999 for the ground state. In this case, VQE can get better fidelities (larger than 0.99) but use again much more resources than our proposal, around 1000 times more to get only one of the eigenvectors.

The third case that we consider to test the algorithm is the non-degenerate two-qubit operator

$$\tau\hat{\mathcal{O}} = \begin{pmatrix} \pi & -\frac{\pi}{2} & -\frac{\pi}{4} & -\frac{\pi}{4} \\ -\frac{\pi}{2} & \pi & -\frac{\pi}{4} & -\frac{\pi}{4} \\ -\frac{\pi}{4} & -\frac{\pi}{4} & \frac{\pi}{2} & 0 \\ -\frac{\pi}{4} & -\frac{\pi}{4} & 0 & \frac{\pi}{2} \end{pmatrix}, \tag{25}$$

3. with eigenvectors and eigenvalues given by

$$\begin{aligned} |\mathscr{E}^{(0)}\rangle &= \frac{1}{2}(|00\rangle + |01\rangle + |10\rangle + |11\rangle), \quad \alpha^{(0)} = 0, \\ |\mathscr{E}^{(1)}\rangle &= \frac{1}{\sqrt{2}}(|10\rangle - |11\rangle), \quad \alpha^{(1)} = \frac{\pi}{2}, \\ |\mathscr{E}^{(2)}\rangle &= \frac{1}{2}(|00\rangle + |01\rangle - |10\rangle - |11\rangle), \quad \alpha^{(2)} = \pi, \\ |\mathscr{E}^{(3)}\rangle &= \frac{1}{\sqrt{2}}(|00\rangle - |01\rangle), \quad \alpha^{(3)} = \frac{3\pi}{2}. \end{aligned} \tag{26}$$

We run the algorithm in the IBM quantum computer "ibmq_vigo". In order to reduce the total number of iterations, we run the three stages of the algorithm four times as follows:

1. We choose $r = 0.6$, $p = 1/r$, $\hat{D}_1 = \mathbb{I}$, $w_1 = 1$. Suppose that the total number of iteration after the three stages is $N_1 = \eta_1$.
2. We choose $r = 0.7$, $p = 1/r$, $\hat{D}_{\eta_1+1} = \hat{D}_{\eta_1}$, $w_{\eta_1+1} = 1$. Suppose that the total number of iteration after the three stages is $N_2 = \eta_1 + \eta_2$.
3. We choose $r = 0.8$, $p = 1/r$, $\hat{D}_{N_2+1} = \hat{D}_{N_2}$, $w_{N_2+1} = 1$. Suppose that the total number of iteration after the three stages is $N_3 = \eta_1 + \eta_2 + \eta_3$.
4. We choose $r = 0.9$, $p = 1/r$, $\hat{D}_{N_3+1} = \hat{D}_{N_3}$, $w_{N_3+1} = 1$, and suppose that the total number of iteration after the three stages is $N = \eta_1 + \eta_2 + \eta_3 + \eta_4$.

We define the fidelity of each approximation as

$$\mathscr{F}_{\ell m} = \max_{k=\{0,1,2,3\}} |\langle \mathscr{E}^{(k)} | \hat{D}_N | \ell m \rangle|^2. \tag{27}$$

To obtain a data set to evaluate the performance of our protocol, we perform ten independent experiments. These data are collected in Appendix Table 6 (see supplemental material). The average fidelities that we obtain are $\bar{\mathscr{F}}_{00} = 0.941$, $\bar{\mathscr{F}}_{01} = 0.933$, $\bar{\mathscr{F}}_{10} = 0.929$, $\bar{\mathscr{F}}_{11} = 0.935$, the average number of iterations is $\bar{N} = 1396$ and the mean errors are: $\bar{c}_{00} = 29$, $\bar{c}'_{00} = 19$ and $\bar{c}_{01} = 18$. Therefore, in this case we obtain the four eigenvectors with fidelities larger than 0.92 in less than 1500 single-shot measurements, which at least corresponds to 6 measurements of mean values, being not enough for a classical-quantum algorithm that uses the optimization of mean values. For the VQE algorithm, we choose 2000 shots per step using 77 COBYLA iterations, which means 157000 single-shot measurements obtaining a fidelity of 0.918 for the ground state. For 8192 shots per step, VQE needs 88 COBYLA iterations to converge, it means 720896 single-shot measurements obtaining a fidelity of 0.944. In this case, VQE cannot surpass the performance of our algorithm, and use more than 100 times resources than our proposal only for the ground state.

For n−qubit observable ($n > 2$), we can use the same protocol but considering more measurement outputs, which implies more stages in the algorithm.

Conclusions

In this work, we implement satisfactorily the approximate eigensolver[44] using the IBM quantum computer. For the single-qubit case, we obtain fidelities larger than 0.97 for both eigenvectors using around 200 single-shot measurements. For the two-qubit case, we use around 1500 single-shot measurements to obtain the approximation of the four eigenvectors with fidelity over 0.9. Due to the stochastic nature of this protocol, we cannot ensure that the approximation converges asymptotically with the number of iteration to the eigenvectors. Nevertheless, it is useful to obtain a fast approximation to use as a guess into another eigensolver that can reach maximal fidelity, like in the eigensolver of Ref.[43]. Also, we compare the performance of our proposal with the VQE algorithm, where VQE, in general, get better fidelities in the single-qubit case but use more than 100 times the number of resources than our algorithm. For two-qubit, the advantage in the maximal fidelity of VQE is a little better in comparison with our algorithm, but again, VQE needs several resources, i.e.more than 1000 times the resources used by our algorithm for all the eigenvectors. Also, the performance of the VQE algorithm depends on the variational ansatz used, which is not the case with our algorithm. This dependence of the VQE algorithms allows enhancing its performance using a better ansatz. The main goal of our algorithm is to get a high fidelity

approximation for all the eigenvectors with few resources. This goal is completely satisfied in comparison with the resources needed for VQE. On the other hand, by manipulating the convergence criteria of our algorithm, we can reach better fidelities. Finally, this work also paves the way for the development of future suitable quantum devices to work with limited resources.

Data availability

The qiskit codes of the one-qubit case and the two-qubit case are available in https://github.com/Panchiyue/Qiskit-Code/tree/main.

Received: 25 November 2020; Accepted: 22 March 2021
Published online: 10 June 2021

References

1. Russell, S. & Norvig, P. *Artificial Intelligence: A Modern Approach* (Prentice Hall, New Jersey, 1995).
2. Metha, P. *et al.* A high-bias, low-variance introduction to machine learning for physicists. *Phys. Rep.* **810**, 1–124 (2019).
3. Ghahramani, Z. *Advanced Lectures on Machine Learning* (Springer, Berlin, 2004).
4. Kotsiantis, S. B. Supervised machine learning: A review of classification techniques. *Informatica* **31**, 249–268 (2007).
5. Wiebe, N., Braun, D. & Lloyd, S. Quantum algorithm for data fitting. *Phys. Rev. Lett.* **109**, 050505 (2012).
6. Lloyd, S., Mohseni, M. & Rebentrost, P. Quantum algorithms for supervised and unsupervised machine learning. arXiv:1307.0411 [quant-ph] (2013).
7. Rebentrost, P., Mohseni, M. & Lloyd, S. Quantum support vector machine for big data classification. *Phys. Rev. Lett.* **113**, 130503 (2014).
8. Li, Z., Liu, X., Xu, N. & Du, J. Experimental realization of a quantum support vector machine. *Phys. Rev. Lett.* **114**, 140504 (2015).
9. Sutton, R. S. & Barto, A. G. *Reinforcement Learning: An Introduction* (MIT Press, Cambridge, 2018).
10. Jaderberg, M. *et al.* Human-level performance in 3D multiplayer games with population-based reinforcement learning. *Science* **364**, 859–865 (2019).
11. Lamata, L. Basic protocols in quantum reinforcement learning with superconducting circuits. *Sci. Rep.* **7**, 1609 (2017).
12. Kaelbling, L. P., Littman, M. L. & Moore, A. W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **4**, 237–285 (1996).
13. Dong, D., Chen, C., Li, H. & Tarn, T.-J. Quantum reinforcement learning. *IEEE Trans. Syst. Man Cybern. B Cybern.* **38**, 1207–1220 (2008).
14. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
15. Riedmiller, M., Gabel, T., Hafner, R. & Lange, S. Reinforcement learning for robot soccer. *Auton. Robot.* **27**, 55–73 (2009).
16. Yu, S. *et al.* Reconstruction of a photonic qubit state with reinforcement learning. *Adv. Quantum Technol.* **2**, 1800074 (2019).
17. Albarrán-Arriagada, F., Retamal, J. C., Solano, E. & Lamata, L. Measurement-based adaptation protocol with quantum reinforcement learning. *Phys. Rev. A* **98**, 042315 (2018).
18. Littman, M. L. Reinforcement learning improves behaviour from evaluative feedback. *Nature* **521**, 445–451 (2015).
19. Silver, D. *et al.* Mastering the game of Go without human knowledge. *Nature* **550**, 354–359 (2017).
20. Silver, D. *et al.* A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **362**, 1140–1144 (2018).
21. Vinyals, O. *et al.* Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575**, 350–354 (2019).
22. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, New York, 2010).
23. Grover, L. K. A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* 212–219 (1996).
24. Shor, P. W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.* **26**, 1484–1509 (1997).
25. Harrow, A. W., Hassidim, A. & Lloyd, S. Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* **103**, 150502 (2009).
26. Cai, X.-D. *et al.* Experimental quantum computing to solve systems of linear equations. *Phys. Rev. Lett.* **110**, 230501 (2013).
27. Xin, T. *et al.* Quantum algorithm for solving linear differential equations: Theory and experiment. *Phys. Rev. A* **101**, 032307 (2020).
28. Biamonte, J. *et al.* Quantum machine learning. *Nature* **549**, 195–202 (2017).
29. Schuld, M., Sinayskiy, I. & Petruccione, F. An introduction to quantum machine learning. *Contemp. Phys.* **56**, 172–185 (2015).
30. Dunjko, V., Taylor, J. M. & Briegel, H. J. Quantum-enhanced machine learning. *Phys. Rev. Lett.* **117**, 130501 (2016).
31. Gao, J. *et al.* Experimental machine learning of quantum states. *Phys. Rev. Lett.* **120**, 240501 (2018).
32. Schuld, M. & Killoran, N. Quantum machine learning in feature Hilbert spaces. *Phys. Rev. Lett.* **122**, 040504 (2019).
33. Lau, H.-K., Pooser, R., Siopsis, G. & Weedbrook, C. Quantum machine learning over infinite dimensions. *Phys. Rev. Lett.* **118**, 080501 (2017).
34. Wittek, P. *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, New York, 2014).
35. Lamata, L. Quantum machine learning and quantum biomimetics: A perspective. *Mach. Learn. Sci. Technol.* **1**, 033002 (2020).
36. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
37. Aleksandrowicz, G. *et al.* Qiskit: An open-source framework for quantum computing (2019).
38. IBM-Q Experience (2019).
39. McClean, J. R., Romero, J., Babbush, R. & Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **18**, 023023 (2016).
40. Kandala, A. *et al.* Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **549**, 242–246 (2017).
41. Peruzzo, A. *et al.* A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**, 4213 (2014).
42. Lavrijsen, W. *et al.* Classical optimizers for Noisy Intermediate-Scale Quantum devices, in *IEEE International Conference on Quantum Computing & Engineering (QCE20)* (2020).
43. Wei, S., Li, H. & Long, G.-L. A full quantum eigensolver for quantum chemistry simulations. *Research* **2020**, 1486935 (2020).
44. Albarrán-Arriagada, F., Retamal, J. C., Solano, E. & Lamata, L. Reinforcement learning for semi-autonomous approximate quantum eigensolver. *Mach. Learn. Sci. Technol.* **1**, 015002 (2020).
45. Qiskit command *operator*.
46. O'Malley, P. J. J. *et al.* Scalable quantum simulation of molecular energies. *Phys. Rev. X* **6**, 301007 (2016).

Acknowledgements
We acknowledge financial support from Spanish MCIU/AEI/FEDER (PGC2018-095113-B-I00), Basque Government IT986-16, projects QMiCS (820505) and OpenSuperQ (820363) of EU Flagship on Quantum Technologies, EU FET Open Grant Quromorphic, EPIQUS, and Shanghai STCSM (Grant No. 2019SHZDZX01-ZX04).

Author contributions
E.S. and F.A.-A. supervised and contributed to the theoretical analysis. N.B. carried out all calculations and prepared the figures. C.-Y.P. and M.H write the qiskit program to run in IBM quantum experience. All the authors wrote the manuscript. All authors contributed to the results discussion and revised the manuscript.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary Information The online version contains supplementary material available at https://doi.org/10.1038/s41598-021-90534-7.

Correspondence and requests for materials should be addressed to E.S. or F.A.-A.

Reprints and permissions information is available at www.nature.com/reprints.

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