Published in final edited form as:
Nature. 2021 March 01; 591(7849): 229–233. doi:10.1038/s41586-021-03242-7.

Experimental quantum speed-up in reinforcement learning agents

V. Saggio[1], **B. E. Asenbeck**[1], **A. Hamann**[2], **T. Strömberg**[1], **P. Schiansky**[1], **V. Dunjko**[3], **N. Friis**[4], **N. C. Harris**[5], **M. Hochberg**[6], **D. Englund**[5], **S. Wölk**[2,7], **H. J. Briegel**[2,8], **P. Walther**[1,9]

[1]University of Vienna, Faculty of Physics, Vienna Center for Quantum Science and Technology (VCQ), Boltzmanngasse 5, A-1090 Vienna, Austria

[2]Institut für Theoretische Physik, Universität Innsbruck, Technikerstraße 21a, 6020 Innsbruck, Austria

[3]LIACS, Leiden University, Niels Bohrweg 1, 2333 CA Leiden, Netherlands

[4]Institute for Quantum Optics and Quantum Information - IQOQI Vienna, Austrian Academy of Sciences, Boltzmanngasse 3, A-1090 Vienna, Austria

[5]Research Laboratory of Electronics, Massachusetts Institute of Technology, Cambridge, MA 02139, USA

[6]Nokia of America Corporation, USA

[7]Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Institut für Quantentechnologien, Söflingerstr. 100, 89077 Ulm, Germany

[8]Fachbereich Philosophie, Universität Konstanz, Fach 17, 78457 Konstanz, Germany

[9]Christian Doppler Laboratory for Photonic Quantum Computer, Faculty of Physics, University of Vienna, A-1090 Vienna, Austria

Abstract

As the field of artificial intelligence advances, the demand for algorithms that can learn quickly and efficiently increases. An important paradigm within artificial intelligence is reinforcement learning [1], where decision-making entities called agents interact with environments and learn by updating their behaviour based on obtained feedback. The crucial question for practical applications is how fast agents learn [2]. While various works have made use of quantum mechanics to speed up the agent's decision-making process [3, 4], a reduction in learning time has not been demonstrated yet. Here, we present a reinforcement learning experiment where the

Correspondence to: V. Saggio; P. Walther.

Correspondence should be addressed to philip.walther@univie.ac.at or valeria.saggio@univie.ac.at.

Author Statement

V.S. and B.E.A. implemented the experiment and performed data analysis. A.H., V.D., N.F., S.W., and H. J.B. developed the theoretical idea. T.S. and P.S. provided help with the experimental implementation. N.C.H., M.H., and D.E. designed the nanophotonic processor. V.S., S.W., and P.W. supervised the project. All the authors contributed to writing the paper.

The authors declare no competing interests.

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

learning process of an agent is sped up by utilizing a quantum communication channel with the environment. We further show that combining this scenario with classical communication enables the evaluation of such an improvement, and additionally allows for optimal control of the learning progress. We implement this learning protocol on a compact and fully tuneable integrated nanophotonic processor. The device interfaces with telecom-wavelength photons and features a fast active feedback mechanism, allowing us to demonstrate the agent's systematic quantum ad-vantage in a setup that could be readily integrated within future large-scale quantum communication networks.

Rapid advances in the field of machine learning (ML) and in general artificial intelligence (AI) are paving the way towards intelligent algorithms and automation. An important paradigm within AI is reinforcement learning (RL), where decision-making entities called 'agents' interact with an environment, 'learning' to achieve a goal via feedback [1]. Whenever the agent performs well (i.e., makes the right decision), the environment rewards its behaviour, and the agent uses this information to progressively increase the likelihood of accomplishing its task. In this sense, an agent 'learns' by 'reinforcement'. RL has applications in many sectors, from robotics [5, 6] to the healthcare domain [7], to brain-like computing simulation [8] and neural network implementations [6, 9]. Also the celebrated AlphaGo algorithm [10], able to beat even the most skilled human players at the game of Go, employs RL.

At the same time, quantum technologies have experienced remarkable progress [11]. At the heart of quantum mechanics lies the superposition principle, dictating that even the simplest, two-dimensional quantum system is described by a continuum of infinitely many possible choices via a state vector $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with $|\alpha|^2 + |\beta|^2 = 1$, while only two possible states, $|0\rangle$ and $|1\rangle$, exist classically. Advantageous RL algorithms [12, 13] inspired by quantum mechanics have been successful in aiding problems in quantum information processing, e.g., decoding of errors [14–16], quantum feedback [17], adaptive code-design [18], quantum state reconstruction [19], and even the design of quantum experiments [20, 21]. Conversely, quantum technologies have enabled quadratically faster decision-making processes for RL agents via the quantization of their internal hardware [3, 4, 22, 23].

In all of these applications, agent and environment interact entirely classically. Here, we consider a novel RL setting where they can also interact quantumly, formally via a quantum channel [2]. We therefore introduce a quantum-enhanced hybrid agent capable of quantum as well as classical information transfer. This makes it possible to achieve and quantify a quantum speed-up in the agent's learning time with respect to RL based solely on classical interaction.

We realize this protocol using a fully programmable nanophotonic processor interfaced with photons at telecom wavelengths. The setup enables the implementation of active feedback mechanisms, thus proving suitable for demonstrations of RL algorithms. Moreover, such photonic platforms hold the potential of integrating RL quantum speed-ups in future quantum networks thanks to the photons' telecom wavelengths. A long-standing goal in the development of quantum communication lies in establishing a form of 'quantum internet' [24, 25], a highly interconnected network able to distribute and manipulate quantum states

via optical links. We therefore envisage AI and RL to play important roles in future quantum networks, including a potential quantum internet, much in the same way that AI forms integral part of the internet today.

Quantum-Enhanced Reinforcement Learning

The conceptual idea of RL is shown in Fig. 1(a). A decision-making entity called agent interacts with an environment by receiving perceptual input ('percepts') s_i, and outputting specific 'actions' a_i accordingly. Different 'rewards' r issued by the environment for correct combinations of percepts and actions incentivize agents to improve their decision-making, and thus to learn [1].

Although RL has already been shown amenable to quantum enhancements, the interaction has so far been restricted exclusively to classical communication, meaning that signals can only be composed from a fixed, discrete alphabet. For signals carried by quantum systems (e.g., single photons considered here) this corresponds to a fixed preferred basis, e.g., 'vertical' or 'horizontal' photon polarization, as shown in Fig. 1(b).

In general, it has been shown that granting agents access to quantum hardware (while still considering classical communication) does not reduce the learning time, although it allows to output actions quadratically faster [3, 4]. To achieve reductions in learning times, quantum communication becomes necessary.

We therefore consider an environment and a quantum-enhanced hybrid agent with access to internal quantum (as well as classical) hardware interacting by exchanging quantum states $|a_i\rangle$, $|s_i\rangle$, and $|r\rangle$, representing actions a_i, percepts s_i, and rewards r, respectively. Such agents may behave 'classically', i.e., use a classical channel, or 'quantumly', meaning that communication is no longer limited to a fixed preferred basis, but allows for exchanges of arbitrary superpositions via a quantum channel, as shown in Fig. 1(c). In general, agents react to (sequences of) percepts $|s_{i-1}\rangle$ with (sequences of) actions $|a_i\rangle$ according to a policy $\pi(a_i|s_{i-1})$ that is updated during the learning process via classical control.

Within this framework, we focus on so-called [2] deterministic strictly epochal (DSE) learning scenarios, also called episodic instead of epochal [1]. Here, 'epochs' consist of strings of percepts $\vec{s} = (s_0, \ldots, s_{L-1})$ with fixed s_0, actions $\vec{a} = (a_1, \ldots, a_L)$ of fixed length L, and a final reward r, and both $\vec{s} = \vec{s}(\vec{a})$ and $r = r(\vec{a})$ are completely determined by $\vec{a}$. Therefore, no explicit representation of the percepts is required in our experiment (see Methods). A non-trivial feature of the DSE scenario is that the effective behaviour of the environment can be modelled via a unitary U_E [2] on the action and reward registers A and R as

$$U_E|\vec{a}\rangle_A|0\rangle_R = \begin{cases} |\vec{a}\rangle_A|1\rangle_R \text{ if } r(\vec{a}) > 0 \\ |\vec{a}\rangle_A|0\rangle_R \text{if } r(\vec{a}) = 0 \end{cases}. \quad (1)$$

U_E is similar to a generalized controlled-NOT gate such that in case of rewarded action sequences ($r(\vec{a}) > 0$), the reward state is flipped. U_E can therefore be used to perform a quantum search for such sequences.

A hybrid agent can choose between quantum and classical behaviour in each epoch. In classical epochs, the agent prepares the state $|\vec{a}\rangle_A|0\rangle_R$, where $\vec{a}$ is determined by sampling from a classical probability distribution $p(\vec{a})$ determined by its policy π. With a winning probability

$$\varepsilon = \sin^2(\xi) = \sum_{\{\vec{a}\,|\,r(\vec{a})>0\}} p(\vec{a}), \tag{2}$$

with $\xi \in [0, 2\pi]$, the agent receives a reward and updates its policy according to a rule, presented in Eq. (4), based on projective simulation [26] (see also Methods). In quantum epochs, the following steps are performed:

1. The agent prepares the state $|\psi\rangle_A|-\rangle_R$, with $|\psi\rangle_A = \sum_{\vec{a}} \sqrt{p(\vec{a})}\,|\vec{a}\rangle_A = \cos(\xi)|\ell\rangle_A + \sin(\xi)|w\rangle_A$, and sends the state to the environment. $|w\rangle_A$ and $|\ell\rangle_A$ are superpositions of all winning (rewarded) and losing (non-rewarded) action sequences, respectively, and $|-\rangle_R = (|0\rangle_R - |1\rangle_R)/\sqrt{2}$.
2. The environment applies U_E from Eq. (1) to $|\psi\rangle_A|-\rangle_R$, flipping the sign of the winning state:

$$U_E|\psi\rangle_A|-\rangle_R = [\cos(\xi)|\ell\rangle_A - \sin(\xi)|w\rangle_A]|-\rangle_R, \tag{3}$$

   and returns the resulting state to the agent.
3. The agent performs a reflection $U_R = 2|\psi\rangle\langle\psi|_A - \mathbb{1}_A$ over the initial state $|\psi\rangle_A$.

The last step leads to amplitude amplification similar to Grover's algorithm [27] and thus to an increased probability $\sin^2(3\xi)$ [28] to find rewarded action sequences (see Methods). In our experiment, the hybrid agent performs a single query (and thus a single step of amplitude amplification) during a quantum epoch. However, the general framework allows for multiple steps of amplitude amplification in consecutive quantum epochs.

While quantum epochs lead to an increased winning probability, they do not reveal the reward (or corresponding percept sequence, in general). The reward can be determined only via classical test epochs, where the obtained action sequence is used as input. Thus, the hybrid agent alternates between quantum and classical test epochs, updating its policy every time a reward is obtained after a test epoch. Such agents accomplish their task of finding winning action sequences faster, and hence learn faster than entirely classical agents. This approach allows us to quantify the speed-up in learning time, which is not possible in the general setting discussed in [2]. The learning speed-up manifests in a reduced average learning time $\langle T\rangle_Q$, i.e., the average number of epochs necessary to achieve a certain winning probability P_L. In general, a quadratic improvement can be achieved if the maximal

number of coherent interactions between agent and environment scales with the problem size (see Methods).

Experimental Implementation

Quantized RL protocols can be compactly realized using state-of-the-art photonic technology [29]. Nowadays, integrated photonic platforms hold the advantage of providing scalable architectures where many elementary components can be accommodated on small devices [30]. Here, we use a programmable nanophotonic processor comprising 26 waveguides fabricated to form 88 Mach-Zehnder interferometers (MZIs). An MZI is equipped with two configurable phase shifters as shown in Figs. 2(a), (b), and acts as a tuneable beam splitter. Information is spatially encoded onto two orthogonal modes, $|0\rangle = (1, 0)^T$ and $|1\rangle = (0, 1)^T$, which constitute the computational basis.

As illustrated in Fig. 2(c), pairs of single photons are generated (at telecom wavelengths) from a single-photon source. One photon is coupled into a waveguide and then detected by single-photon detectors D1, D2 or D3, while the other one is sent to D0 for heralding (i.e., clicks in detectors D1, D2 or D3 are registered in coincidence with clicks in D0). The detectors are superconducting nanowires with efficiences of up to ~90% (see Methods for experimental details). The processor is divided into three regions, where the first and last are assigned to the agent, and the middle region to the environment, in order to carry out, in quantum epochs, steps 1-3 listed above. The agent is further equipped with a classical control mechanism (a feedback loop) that updates its learning policy.

In our experiment, we represent the winning and losing action states $|w\rangle_A$ and $|\ell\rangle_A$ by a single qubit via $|0\rangle_A = |\ell\rangle_A$ and $|1\rangle_A = |w\rangle_A$, and use another qubit to encode the reward ($|0\rangle_R$, $|1\rangle_R$). This results in a four-level system, where each level is a waveguide in our processor, as shown in Fig. 3. The winning probability for the agent is initially set to $\varepsilon = \sin^2(\xi) = 0.01$, representing a single rewarded action sequence out of 100. After a single photon is coupled into the mode $|0_A 0_R\rangle$, the agent creates the state $|\psi\rangle_A = (\cos(\xi)|0\rangle_A + \sin(\xi)|1\rangle_A)|0\rangle_R$ by applying a unitary U_P. Next, it can decide to play classically or quantum-mechanically.

Classical strategy. The environment flips the reward qubit only if the action qubit is in the winning state via U_E, see Fig. 3(a). Next, the photon is coupled out and detected in either D1 or D2 with probability $\cos^2(\xi)$ and $\sin^2(\xi)$, respectively. If D2 is triggered (i.e., the agent has been rewarded), a feedback mechanism updates the policy π by updating the winning probability ε_j after having obtained j rewards as

$$\varepsilon_j = \frac{1+2j}{100+2j}. \tag{4}$$

π is related to $p(\vec{a})$, and therefore to ε, via Eq. (A.1) (see Methods).

Quantum strategy. The hybrid agent uses this strategy to speed up its learning process. After the reward qubit is rotated to $|-\rangle_R$ via U_{H0} and U_{H1}, the environment acts as an oracle via U_E as in Eq. (3). Consecutively, the agent reverses the effect of U_{H0} and U_{H1}, and performs

the reflection U_R, see Fig. 3(b). Measuring in the computational basis of the action register then leads to the detection of a rewarded action sequence with increased probability $\sin^2(3\xi)$ in D3. For practical reasons, the classical test epoch is implemented only in software (see Methods). The update rule remains the same as in the classical case.

In general, any Grover-like algorithm faces a drop in amplitude amplification after the optimal point is reached. Since different agents will reach this optimal point in different epochs, one can identify the probability $\varepsilon = 0.396$ until which it is beneficial for all agents to use a quantum strategy, as they will observe more rewards than in the classical strategy on average (see Methods). When this probability is surpassed, it is advantageous to switch to an entirely classical strategy. This *combined strategy* thus avoids the typical amplitude amplification drop without introducing additional overheads in terms of experimental resources.

Results

At the end of each classical epoch, we record outcomes 1 and 0 for the rewarded and non-rewarded behaviour, respectively, obtaining a binary sequence whose length equals the number of played epochs in the classical learning strategy and half of the number of played epochs in the quantum strategy (as here two epochs, quantum and classical test, are needed to obtain the reward). For a fair comparison between these scenarios, in the quantum strategy the reward is distributed (i.e., averaged) over the quantum and classical test epochs. The reward is then averaged over different independent agents. Fig. 4 shows this average reward η for the different learning strategies.

The theoretical data is simulated for $n = 10{,}000$ agents and the experimental data obtained from $n = 165$. Fig. 4(a) visualizes the quantum improvement originating from the use of amplitude amplification in comparison with a purely classical strategy. For completeness, the comparison between not distributing and distributing the reward over two epochs in the quantum strategy is shown in Figs. 4(b), (c).

When $\varepsilon = 0.396$, η for the quantum strategy starts decreasing, as visible in Fig. 4(a). Our setup allows the agents to choose the favorable strategy by switching from quantum to classical when the latter becomes advantageous. This combined strategy outperforms the purely classical scenario, as shown in Fig. 4(d). As previously discussed, a certain winning probability P_L has to be defined to quantify the learning time. Choosing $P_L = 0.37$ (note however, that any probability below $\varepsilon = 0.396$ can be chosen), the learning time $\langle T \rangle$ for P_L decreases from $\langle T \rangle_C = 270$ in the classical strategy to $\langle T \rangle_Q = 100$ in the combined strategy. This implies a reduction of 63%, which fits well to the theoretical values $T_C^{\text{theory}} = 293$ and $T_Q^{\text{theory}} = 97$, accounting for small experimental imperfections.

In general, hybrid agents can experience a quadratic speed-up in their learning time if arbitrary numbers of coherent Grover iterations can be performed [31], even if the number of rewarded actions is unknown [32].

Methods

I Quantum enhancement in reinforcement learning agents

Here, we present an explicit method for combining a classical agent with quantum amplitude amplification. Introducing a feedback loop between classical policy update and quantum amplitude amplification, we are able to determine achievable improvements in sample complexity, and thus in learning time. Additionally, the final policy of our agent has properties similar to those of the underlying classical agent, leading to a comparable behaviour as discussed in more detail in [31] (a paper dedicated to discuss the theoretical background of the hybrid agent more specifically).

In the following, we focus on simple deterministic strictly epochal (DSE) environments, where the interaction between the agent and the environment is structured into epochs. Each epoch starts with the same percept s_0, and at each time step i an action-percept pair (a_i, s_i) is exchanged. Many interesting environments are epochal, e.g., in applications of RL to quantum physics [21, 36–38] or popular problems such as playing Go [10]. At the end of each epoch, after L action-percept pairs are communicated, the agent receives a reward $r \in \{0, 1\}$. The rules of the game are deterministic and time independent, such that performing a specific action a_i after receiving a percept s_{i-1} always leads to the same following percept s_i.

The behaviour of an agent is determined by its policy described by the probability $\pi(a_i|s_{i-1})$ to perform the action a_i given the percept s_{i-1}. In deterministic settings, the percept s_i is completely determined by all previously performed actions $a_1, \ldots, a_i$ such that $\pi(a_i|s_{i-1}) = \pi(a_i|a_1, \ldots, a_{i-1})$. Thus, the behaviour of the agent within one epoch is described by action sequences $\vec{a} = (a_1, \ldots, a_L)$ and their corresponding probabilities

$$p(\vec{a}) = \prod_{i=1}^{L} \pi(a_i \mid a_1, \ldots, a_{i-1}). \tag{A.1}$$

Our learning agent uses a policy based on projective simulation [26], where each action sequence $\vec{a}$ is associated with a weight factor $h(\vec{a})$ initialized to $h = 1$. Its policy is defined via the probability distribution

$$p(\vec{a}) = \frac{h(\vec{a})}{\sum_{\vec{a}'} h(\vec{a}')}. \tag{A.2}$$

In our experiment, the initial winning probability ε, i. e., the probability to choose rewarded action sequences ($r(\vec{a}) > 0$), is given by

$$\varepsilon = \sum_{\{\vec{a} \mid r(\vec{a}) > 0\}} p(\vec{a}) = \frac{\sum_{\{\vec{a} \mid r(\vec{a}) > 0\}} h(\vec{a})}{\sum_{\{\vec{a}\}} h(\vec{a})}, \tag{A.3}$$

and is set to 1/100. If the agent has chosen the sequence $\vec{a}$, it updates the corresponding weight factor via

$$h\left(\vec{a}\right) \rightarrow h\left(\vec{a}\right) + \lambda r\left(\vec{a}\right), \tag{A.4}$$

where $\lambda = 2$ in our experiment and $r(\vec{a}) = 1$ (0) if $\vec{a}$ is rewarded (non-rewarded). Thus, the winning probability after the agent has found j rewards is given by Eq. (4). In general, the update method for quantum-enhanced agents is not limited to projective simulation and can be used to enhance any classical learning scenario, provided that $p(\vec{a})$ exists and that the update rule is solely based on the observed rewards. We generalize the given learning problem to the quantum domain by encoding different action sequences $\vec{a}$ into orthogonal quantum states $|\vec{a}\rangle$ defining our computational basis. Additionally, we create a fair unitary oracular variant of the environment [2], whose effective behaviour on the action register can be described by $\tilde{U}_{\mathrm{E}}$ as

$$\tilde{U}_{\mathrm{E}}|\vec{a}\rangle = \begin{cases} -|\vec{a}\rangle & \text{if } r\left(\vec{a}\right) > 0 \\ |\vec{a}\rangle & \text{if } r\left(\vec{a}\right) = 0 \end{cases}. \tag{A.5}$$

The unitary oracle $\tilde{U}_{\mathrm{E}}$ can be used to perform, for instance, a Grover search or amplitude amplification for rewarded action sequences by performing Grover iterations

$$U_{\mathrm{G}} = (2|\psi\rangle\langle\psi| - \mathbb{1})\tilde{U}_{\mathrm{E}} \tag{A.6}$$

on an initial state $|\psi\rangle$. A quantum-enhanced agent with access to $\tilde{U}_{\mathrm{E}}$ can thus find rewarded action sequences faster than a corresponding classical agent defined by the same initial policy $\pi(a_i|s_{i-1})$ and update rules.

In general, the optimal number k of Grover iterations $U_{\mathrm{G}}^{k}|\psi\rangle$ depends on the winning probability ε via $k \sim 1/\sqrt{\varepsilon}$ [27]. In the following, we assume that ε is known at least to a good approximation. This is for instance possible if the number of rewarded action sequences is known. However, a similar agent can also be developed if ε is unknown by adapting methods from [32] as described in [31].

I.1 Description of the agent—A quantum-enhanced hybrid agent is constructed via the following steps:

1. Given the classical probability distribution $p(\vec{a})$, determine the winning probability $\varepsilon = \sin^2(\xi)$ based on the current policy and prepare the quantum state in the action register:

$$|\psi\rangle = \sum_{\{\vec{a}\}} \sqrt{p\left(\vec{a}\right)}|\vec{a}\rangle \tag{A.7}$$

$$= \cos(\xi)|\ell\rangle + \sin(\xi)|w\rangle\,. \tag{A.8}$$

Here, the quantum states

$$|\ell\rangle \sim \sum_{\{\vec{a}\,|\,r(\vec{a})=0\}} \sqrt{p(\vec{a})}|\vec{a}\rangle, \tag{A.9}$$

$$|w\rangle \sim \sum_{\{\vec{a}\,|\,r(\vec{a})>0\}} \sqrt{p(\vec{a})}|\vec{a}\rangle, \tag{A.10}$$

contain all losing and winning components, respectively. In our experiment, we identify $|\ell\rangle \triangleq |0\rangle$ and $|w\rangle \triangleq |1\rangle$. The task assigned to the agent is to (learn to) perform the winning sequences $|w\rangle$ via policy update. This translates to a maximization of the obtained reward.

2. Apply the optimal number $k(\sqrt{\varepsilon})$ of Grover iterations leading to

$$|\psi'\rangle = U_{\mathrm{G}}^{k}|\psi\rangle, \tag{A.11}$$

and perform a measurement in the computational basis on $|\psi'\rangle$ to determine a test action sequence $\vec{a}$.

3. Play one classical epoch by using the test sequence $\vec{a}$ determined in step 2 and obtain the corresponding percept sequence $\vec{s}(\vec{a})$ and the reward $r(\vec{a})$.

4. Update ε, and thus the classical policy π, using the rule in Eq. (4).

There exists a limit P on ε determining whether it is more advantageous for the agent to perform k Grover iterations with $k(\sqrt{\varepsilon}) \geq 1$ or sample directly from $p(\vec{a})$ (therefore $k(\sqrt{\varepsilon}) = 0$) to determine $\vec{a}$. In the latter case, the agent would interact only classically (as in step 3) with the environment.

After each epoch, a classical agent receives a reward with probability $\sin^2(\xi)$. A quantum-enhanced agent can instead use one epoch to either perform one Grover iteration (step 2) or to determine the reward of a given test sequence $\vec{a}$ (step 3). After k Grover iterations, the winning probability is $\sin^2[(2k+1)\xi]$ [28] (see next section). Thus, for $k = 1$, the agent receives a reward after every second epoch with probability $\sin^2(3\xi)$. Therefore, we define the expected average reward of an agent playing a classical strategy as $\eta_{\mathrm{C}} = \sin^2(\xi)$ and of an agent playing a quantum strategy with $k = 1$ as $\eta_{\mathrm{Q}} = \sin^2(3\xi)/2$. For $\varepsilon < P$, $\eta_{\mathrm{Q}} > \eta_{\mathrm{C}}$, meaning that the quantum strategy proves advantageous over the classical case. However, as soon as $\eta_{\mathrm{Q}} = \eta_{\mathrm{C}}$ (at $P = 0.396$), a classical agent starts outperforming a quantum-enhanced agent that still performs Grover iterations.

Determining the winning probability ε exactly as in the example presented here is not always possible. In general, additional information like the number of possible solutions and

Europe PMC Funders Author Manuscripts

model building helps to perform this task. Note that a P smaller than 0.396 should be chosen if ε can only be estimated up to some range. To circumvent this problem, methods like Grover search with unknown reward probability [32], or fixed-point search [39], can be used to determine if and how many steps of amplitude amplification should be performed [31].

I.2 Enhancement of the winning probability—After a quantum epoch, the amplitude $\sin(\xi)$ of the winning state $|w\rangle$ increase to $\sin(3\xi)$. Here, we derive this result. The projections onto the winning and losing subspaces are given by $P_w = \sum\{\vec{a} \mid r(\vec{a}) > 0\} |\vec{a}\rangle\langle\vec{a}|$ and $P_\ell = \sum\{\vec{a} \mid r(\vec{a}) = 0\} |\vec{a}\rangle\langle\vec{a}|$, respectively, which are orthogonal and sum to Identity. Therefore, the initial state (A.8) can be decomposed into a normalized winning $|w\rangle \propto P_w |\psi\rangle$ and losing $|\ell\rangle \propto P_\ell |\psi\rangle$ component, and the unitary (A.6) implementing one Grover iteration can be written as

$$U_{\mathrm{G}} = (2|\psi\rangle\langle\psi| - \mathbb{1})(P_\ell - P_w). \tag{A.12}$$

Now, let us investigate the effect of U_{G} on an arbitrary real superposition

$$|\alpha\rangle = \sin(\alpha)|w\rangle + \cos(\alpha)|\ell\rangle \tag{A.13}$$

in the plane spanned by $|w\rangle$ and $|\ell\rangle$. Using the trigonometric addition theorems, the application of one Grover iteration to $|a\rangle$

$$U_{\mathrm{G}}|\alpha\rangle = \sin(\alpha + 2\xi)|w\rangle + \cos(\alpha + 2\xi)|\ell\rangle \tag{A.14}$$

can be identified as a rotation of 2ξ in the plane. Assuming $\alpha = \xi$, we therefore find the amplitude of $|w\rangle$ to be $\sin(3\xi)$, and thus obtain a winning probability $\sin^2(3\xi)$. Implementing k Grover iterations leads to $\sin^2[(2k + 1)\xi]$.

I.3 Learning time—We define the learning time T as the number of epochs an agent needs on average to reach a certain winning probability P_{L}. The hybrid agent can reach P with fewer epochs on average than its classical counterpart. However, once both reach P, they need on average the same number of epochs to reach $P + \Delta P$ with $0 \leq \Delta P < 1 - P$. Therefore, we choose $P_{\mathrm{L}} \leq P$ in order to quantify the achievable improvement of a hybrid agent compared to its classical counterpart. In our experiment, we choose $P_{\mathrm{L}} = 0.37$ to define the learning time.

Let $l_J = \{\vec{a}_1, \ldots, \vec{a}_J\}$ be a time-ordered list of all the rewarded action sequences an agent has found until it reaches P_{L}. Note that the actual policy π_j, and thus p_j, of our agents depend only on the list l_j of observed rewarded sequences of actions, and this is independent of whether they have found them via classical sampling or quantum amplitude amplification. As a result, a classical agent and its quantum-enhanced hybrid version are described by the same policy $\pi(l_j)$ and behave similarly if they have found the same rewarded action sequences. However, the hybrid agent finds them faster.

In general, the actual policy and overall winning probability might depend on the rewarded action sequences that have been found. Thus, the number J of observed rewarded action

sequences necessary to learn might vary. However, this is not the case for the experiment reported here. In our case, the learning time can be determined via

$$T(J) = \sum_{j=0}^{J-1} t_j, \tag{A.15}$$

where t_j determines the number of epochs necessary to find the next rewarded sequence $\vec{a}_{j+1}$ after it has observed j rewards. For a purely classical agent, the average time is given by

$$\langle t_j \rangle_{\mathrm{C}} = \frac{1}{\varepsilon_j}. \tag{A.16}$$

This time is quadratically reduced to

$$\langle t_j \rangle_{\mathrm{Q}} = \frac{\alpha}{\sqrt{\varepsilon_j}} \tag{A.17}$$

for the hybrid agent. Here, α is a parameter depending only on the number of epochs needed to create one oracle query $\tilde{U}_{\mathrm{E}}$ [2] and on whether ε_j is known. In the case considered here, we find $\alpha = \pi/4$. As a consequence, the average learning time for the hybrid agent is given by

$$\langle T(J) \rangle_{\mathrm{Q}} = \sum_{j=0}^{J-1} \frac{\alpha}{\sqrt{\varepsilon_j}} \leq \alpha \sqrt{J} \sqrt{\langle T(J) \rangle_{\mathrm{C}}}, \tag{A.18}$$

where we used the Cauchy-Schwarz inequality and equations A.15 and A.16. The classical learning time typically scales with $\langle T \rangle_{\mathrm{C}} \sim A^K$ for a learning problem with episode length K and the choice between A different actions in each step. The number J of observed rewarded sequences in order to learn depends on the specific policy update and sometimes also on the list l_J of observed rewarded action sequences. For an agent sticking with the first rewarded action sequence, we would find $J = 1$. However, typical learning agents are more explorative, and common scalings are $J \sim K$ such that we find

$$\langle T(J) \rangle_{\mathrm{Q}} \sim \sqrt{\log(\langle T(J) \rangle_{\mathrm{C}})} \sqrt{\langle T(J) \rangle_{\mathrm{C}}} \tag{A.19}$$

for these cases. This is equivalent to a quasi-quadratic speed-up in the learning time if arbitrary numbers of Grover iterations can be performed.

In more general settings, there exist several possible l_J with different length J such that the learning time $\langle T(J) \rangle$ needs to be averaged over all possible l_J, which again leads to a quadratic speed-up in learning time [31].

I.4 Limited coherent evolutions—In general, all near-term quantum devices allow for coherent evolution only for a limited time and are thus limited to a maximal number of Grover iterations. For winning probabilities $\varepsilon = \sin^2(\xi)$ with $(2k+1)\xi \leq \pi/2$, performing k Grover iterations leads to the highest probability of finding a rewarded action.

Again, we assume that the actual policy of an agent only depends on the number of observed rewards an agent has found. As a consequence, the average time a hybrid agent limited to k Grover iterations needs to achieve the winning probability $P < \sin^2[\pi/(4k+2)]$ is given by [31]

$$\langle T(J,k)\rangle_{\mathrm{Q}} = \sum_{j=0}^{J-1} \frac{\alpha_0 k + 1}{\sin^2\left[(2k+1)\xi_j\right]}, \tag{A.20}$$

with α_0 determining the number of epochs necessary to create one oracle query $\tilde{U}_{\mathrm{E}}$. For $\alpha_0 = 1$, $k \gg 1$ and $(2k+1)\xi_J \ll \pi/2$ we can approximate the learning time for the hybrid agent via

$$\langle T(J,k)\rangle_{\mathrm{Q}} \approx \sum_{j=0}^{J-1} \frac{1}{4k\epsilon_j} = \frac{\langle T(J)\rangle_{\mathrm{C}}}{4k}, \tag{A.21}$$

where we used $\sin x \approx x$ for $x \ll 1$. In general, it can be shown [31] that the winning probability $P_k = \sin^2[\pi/(4k+2)]$ can be reached by a hybrid agent limited to k Grover iterations in a time

$$\langle T(k)\rangle_{\mathrm{Q}} \leq \gamma \frac{\langle T\rangle_{\mathrm{C}}}{k}, \tag{A.22}$$

where γ is a factor depending on the specific setting.

In our case, Eq. (A.20) can be used to compute the lower bound for the average quantum learning time, with $\alpha_0 = k = 1$. For the classical strategy, Eq. (A.16), together with Eq. (A.15), is used. Thus, given $P_{\mathrm{L}} = 0.37$, the predictions for the learning time in our experiment are $T_{\mathrm{Q}}^{\mathrm{theory}} = 97$ and $T_{\mathrm{C}}^{\mathrm{theory}} = 293$.

II Experimental details

A continuous wave laser (Coherent Mira HP) is used to pump a single-photon source producing photon pairs in the telecom wavelength band. The laser light has a central wavelength of 789.5 nm and pumps the single photon source at a power of approximately 100 mW. The source is a periodically poled $KTiOPO_4$ non-linear crystal placed in a Sagnac interferometer [40, 41], where the emission of single photons occurs via a type-II spontaneous parametric down-conversion (SPDC) process. The crystal (produced by Raicol) is 30 mm long, set to a temperature of 25 °C, has a poling period of 46.15 μm and is quasi-phase matched for degenerate emission of photons at 1570 nm when pumping with coherent laser light at 785 nm. As the processor is calibrated for a wavelength of 1580 nm, we shift the wavelength of the laser light to 789.75 nm in order to produce one photon at 1580 nm (that is then coupled into the processor) and another one at 1579 nm (the heralding photon).

The processor is a silicon-on-insulator (SOI) type, designed by the Quantum Photonics Laboratory at MIT (Massachusetts Institute of Technology) [30]. Each programmable unit on the device acts as a tuneable beam splitter implementing the unitary

$$U_{\theta,\phi} = \begin{pmatrix} e^{i\phi}\sin\frac{\theta}{2} & e^{i\phi}\cos\frac{\theta}{2} \\ \cos\frac{\theta}{2} & -\sin\frac{\theta}{2} \end{pmatrix}, \quad \text{(A.23)}$$

where θ and ϕ are the internal and external phases shown in Figs. 2(a), (b), set via thermo-optical phase shifters controlled by a voltage supply. The achievable precision for phase settings is higher than 250 μrad. The bandwidth of the phase shifters is around 130 kHz. The waveguides, spatially separated from one another by 25.4 μm, are designed to admit one linear polarization only. The high contrast in refractive index between the silicon and silica (the insulator) allows for waveguides with very small bend radius (less than 15 μm), thus enabling a high component density (in our case 88 MZIs) on small areas (in our case 4.9 x 2.4 mm). Given the small dimensions, the in-(out-)coupling is realized with the help of $Si_3N_4 - SiO_2$ waveguide arrays (produced by Lionix International), that shrink (and enlarge) the 10 μm optical fibers' mode to match the 2 μm mode size of the waveguides in the processor. The total input-output loss is around 7 dB. The processor is stabilized to a temperature of 28 °C and calibrated at 1580 nm for optimal performance. To reduce the black-body radiation emission due to the heating of the phase shifters when voltage is applied, wavelength division multiplexers with a transmission peak centered at 1571 nm and bandwidth of 13 nm are used before the photons are sent to the detectors. In our processor, two external phase shifters in the implemented circuits were not responding to the supplied voltage. These defects were accounted for by employing an optimization procedure.

The single-photon detectors are multi-element superconducting nanowires (produced by Photon Spot) with efficiencies up to 90% in the telecom wavelength band. They have a dark count rate of ~ 100 c.p.s, low timing jitter (hundreds of ps) and a reset time < 100 ns [42]. Coincidence events are those detection events registered in D0 and at the output of the processor that fall into a temporal window of 1.3 ns (the coincidence window), and are found using a time tagging module (TTM). In more detail, in order to record coincidences and then update the agent's policy accordingly, the following steps are performed in the classical/quantum strategy (after initially setting $h_w = \sum_{\{\vec{a} \mid r(\vec{a}) > 0\}} h(\vec{a}) = 1$ and $h_\ell = \sum_{\{\vec{a} \mid r(\vec{a}) = 0\}} h(\vec{a}) = 99$ such that $\varepsilon = 0.01$):

1. The TTM records the time tags for photons in D0, D1 and D2/D3.
2. A Python script converts the time tags into arrival times, and it iterates through until it finds a coincidence event between either D0 and D1, or D0 and D2/D3.
3. If a coincidence event between D0 and D1 (or D0 and D2/D3) is first found, a 0 (1) is sent to another Python script controlling the MZIs' phase shifters operating on a different computer. If a 0 is sent, the ratio $\varepsilon = h_w/(h_\ell + h_w)$ remains unchanged. If a 1 is sent, h_w is updated as $h_w + 2$, which follows the update in Eq. (4). In the quantum strategy, this step also includes the implementation of a classical test epoch.

Implementing classical test epochs on hardware would require 'testing' the measured action state, i.e., using the measured action sequence as input and making the environment act via

Europe PMC Funders Author Manuscripts

Europe PMC Funders Author Manuscripts

U_E, thus leading to detection of a reward $|0\rangle_R$ or $|1\rangle_R$. However, since this simple circuit works in very close agreement with theoretical predictions (its visibility exceeds 0.99), this part has been implemented in software only.

The update rate is ~ 1 Hz for both the classical and quantum epochs, and can be reduced up to the phase shifters' bandwidth.

Conclusions

We have demonstrated a novel RL protocol where an agent can boost its performance by allowing quantum communication with the environment. This enables a quantum speed-up in its learning time and optimal control of the learning process. Emerging photonic technology provides the advantages of compactness, tunability and low-loss communication, thus proving suitable for RL algorithms where active feedback mechanisms, even over long distances, need to be implemented. Future scaled-up implementations of our protocol rely on a linearly increasing number of waveguides with the action space size when considering action sequences of length $L = 1$, and the use of just a single photon. In this case, a learning task with N different actions requires a processor with $2N$ modes, while $3N + 1$ or maximally $\frac{\pi}{4}\sqrt{N}(3N + 1)$ gates are needed for a single or multiple rounds of amplitude amplification, respectively. In general, multiple photons will be required to deal with a combinatorially big space of action sequences of arbitrary length L. We envision our protocol to aid specifically in problems where frequent search is needed, e.g., network routing problems, where, for instance, tens of qubits, waveguides and detectors would be employed to represent search spaces of 10^4 elements. In general, the development of superconducting detectors, on-demand single-photon sources [33], or the large-scale integration of artificial atoms within photonic circuits [34] suggest significant steps towards scalable multi-photon applications. Although photonic architectures are particularly suitable for such learning algorithms, our theoretical background is applicable to different platforms, e.g., trapped ions or superconducting qubits. Here, future implementations can feature the implementation of agent and environment as spatially separated systems, and a light-matter quantum interface for coherent exchange between them [24, 35].

Acknowledgments

The authors thank L. A. Rozema, I. Alonso Calafell, and P. Jenke for help with the detectors. A.H. acknowledges support from the Austrian Science Fund (FWF) through the project P 30937-N27. V.D. acknowledges support from the Dutch Research Council (NWO/OCW), as part of the Quantum Software Consortium programme (project number 024.003.037). N.F. acknowledges support from the Austrian Science Fund (FWF) through the project P 31339-N27. H.J.B. acknowledges support from the Austrian Science Fund (FWF) through SFB BeyondC F71. P.W. acknowledges support from the research platform TURIS, the European Commission through ErBeStA (no. 800942), HiPhoP (no. 731473), UNIQORN (no. 820474), and AppQInfo (no. 956071), from the Austrian Science Fund (FWF) through CoQuS (W1210-N25), BeyondC (F 7113-N38) and NaMuG (P30067-N36), AFOSR via QAT4SECOMP (FA2386-17-1-4011), and Red Bull GmbH. The MIT portion of the work was supported in part by AFOSR award FA9550-16-1-0391.

Data Availability

All the datasets used in the current work are available under the DOI 10.5281/zenodo.4327211.

Europe PMC Funders Author Manuscripts

References

[1]. Sutton, RS, Barto, AG. Reinforcement Learning: An Introduction. MIT press; Cambridge: 1998.

[2]. Dunjko V, Taylor JM, Briegel HJ. Quantum-Enhanced Machine Learning. Phys Rev Lett. 2016; 117 doi: 10.1103/PhysRevLett.117.130501

[3]. Paparo GD, Dunjiko V, Makmal A, Martin-Delgrado MA, Briegel HJ. Quantum Speedup for Active Learning Agents. Phys Rev X. 2014; 4 doi: 10.1103/PhysRevX.4.031002

[4]. Sriarunothai T, et al. Speeding-up the decision making of a learning agent using an ion trap quantum processor. Quantum Sci Technol. 2019; 4 doi: 10.1088/2058-9565/aaef5e

[5]. Johannink, T; , et al. Residual Reinforcement Learning for Robot Control. In 2019 International Conference on Robotics and Automation (ICRA); Montreal, QC, Canada. IEEE; 2019. 6023–6029.

[6]. Tjandra, A; Sakti, S; Nakamura, S. Sequence-to-Sequence ASR Optimization via Reinforcement Learning. In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); Calgary, AB, Canada. IEEE; 2018. 5829–5833.

[7]. Komorowski M, Celi LA, Badawi O, Gordon AC, Faisal AA. The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care. Nat Med. 2018; 24 :1716–1720. DOI: 10.1038/s41591-018-0213-5 [PubMed: 30349085]

[8]. Thakur CS, et al. Large-scale neuromorphic spiking array processors: A quest to mimic the brain. Frontiers in neuroscience. 2018; 12 :891. doi: 10.3389/fnins.2018.00891 [PubMed: 30559644]

[9]. Steinbrecher GR, Olson JP, Englund D, Carolan J. Quantum optical neural networks. npj Quantum Information. 2019; 5 :1–9. DOI: 10.1038/s41534-019-0174-7

[10]. Silver D, et al. Mastering the game of Go without human knowledge. Nature. 550 :354–359. 2017; doi: 10.1038/nature24270 [PubMed: 29052630]

[11]. Arute F, et al. Quantum supremacy using a programmable superconducting processor. Nature. 2019; 574 doi: 10.1038/s41586-019-1666-5

[12]. Dong D, Chen C, Li H, Tarn T-J. Quantum Reinforcement Learning. IEEE T Syst, Man Cy B. 2008; 38 :1207–1220. DOI: 10.1109/TSMCB.2008.925743

[13]. Dunjko V, Briegel HJ. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Rep Progr Phys. 2018; 81 doi: 10.1088/1361-6633/aab406

[14]. Baireuther P, O'Brien TE, Tarasinski B, Beenakker CWJ. Machine-learning-assisted correction of correlated qubit errors in a topological code. Quantum. 2018; 2 :48. doi: 10.22331/q-2018-01-29-48

[15]. Breuckmann NP, Ni X. Scalable Neural Network De-coders for Higher Dimensional Quantum Codes. Quantum. 2018; 2 :68–92. DOI: 10.22331/q-2018-05-24-68

[16]. Chamberland C, Ronagh P. Deep neural decoders for near term fault-tolerant experiments. Quant Sci Techn. 2018; 3 doi: 10.1088/2058-9565/aad1f7

[17]. Fösel T, Tighineanu P, Weiss T, Marquardt F. Reinforcement Learning with Neural Networks for Quantum Feedback. Phys Rev X. 2018; 8 doi: 10.1103/PhysRevX.8.031084

[18]. Poulsen Nautrup H, Delfosse N, Dunjko V, Briegel HJ, Friis N. Optimizing quantum error correction codes with reinforcement learning. Quantum. 2019; 3 :215. doi: 10.22331/q-2019-12-16-215

[19]. Yu S, et al. Reconstruction of a photonic qubit state with reinforcement learning. Adv Quantum Technol. 2019; 2 doi: 10.1002/qute.201800074

[20]. Krenn M, Malik M, Fickler R, Lapkiewicz R, Zeilinger A. Automated Search for new Quantum Experiments. Phys Rev Lett. 2016; 116 doi: 10.1103/PhysRevLett.116.090405

[21]. Melnikov AA, et al. Active learning machine learns to create new quantum experiments. Proc Natl Acad Sci USA. 2018; 115 :1221–1226. DOI: 10.1073/pnas.1714936115 [PubMed: 29348200]

[22]. Dunjko V, Friis N, Briegel HJ. Quantum-enhanced deliberation of learning agents using trapped ions. New J Phys. 2015; 17 doi: 10.1088/1367-2630/17/2/023006

[23]. Jerbi, S; Poulsen Nautrup, H; Trenkwalder, LM; Briegel, HJ; Dunjko, V. A framework for deep energy-based reinforcement learning with quantum speed-up. 2019. Preprint at https://arxiv.org/abs/1910.12760

[24]. Kimble HJ. The quantum internet. Nature. 2008; 453 doi: 10.1038/nature07127

[25]. Cacciapuoti AS, et al. Quantum Internet: Networking Challenges in Distributed Quantum Computing. IEEE Network. 2020; 34 :137–143. DOI: 10.1109/MNET.001.1900092

[26]. Briegel HJ, De las Cuevas G. Projective simulation for artificial intelligence. Sci Rep. 2012; 2 :1–16. DOI: 10.1038/srep00400

[27]. Grover LK. Quantum mechanics helps in searching for a needle in a haystack. Phys Rev Lett. 1997; 79 :325. doi: 10.1103/PhysRevLett.79.325

[28]. Nielsen, MA, Chuang, IL. Quantum Computation and Quantum Information. Cambridge University Press; Cambridge: 2000.

[29]. Flamini F, et al. Photonic architecture for reinforcement learning. New J Phys. 2020; 22 doi: 10.1088/1367-2630/ab783c

[30]. Harris NC, et al. Quantum transport simulations in a programmable nanophotonic processor. Nat Photon. 2017; 11 :447–452. DOI: 10.1038/nphoton.2017.95

[31]. Hamann A, et al. A hybrid agent for quantum-accessible reinforcement learning. In preparation.

[32]. Boyer M, Brassard G, Hoyer P, Tappa A. Tight bounds on quantum searching. Fortschr Phys. 1998; 46 :493. doi: 10.1002/3527603093.ch10

[33]. Senellart P, Solomon G, White A. High-performance semiconductor quantum-dot single-photon sources. Nat Nanotech. 2017; 12 :1026–1039. DOI: 10.1038/nnano.2017.218

[34]. Wan NH, et al. Large-scale integration of artificial atoms in hybrid photonic circuits. Nature. 2020; 583 :226–231. DOI: 10.1038/s41586-020-2441-3 [PubMed: 32641812]

[35]. Northup TE, Blatt R. Quantum information transfer using photons. Nat Photon. 2014; 8 :356–363. DOI: 10.1038/nphoton.2014.53

[36]. Denil, M; , et al. Learning to Perform Physics Experiments via Deep Reinforcement Learning. 2016. Preprint at https://arxiv.org/abs/1611.01843

[37]. Bukov M, et al. Reinforcement Learning in Different Phases of Quantum Control. Phys Rev X. 2018; 8 doi: 10.1103/PhysRevX.8.031086

[38]. Poulsen Nautrup, H; , et al. Operationally meaningful representations of physical systems in neural networks. 2020. Preprint at https://arxiv.org/abs/2001.00593

[39]. Yoder TJ, Low GH, Chuang IL. Fixed-Point Quantum Search with an Optimal Number of Queries. Phys Rev Lett. 2014; 113 doi: 10.1103/PhysRevLett.113.210501

[40]. Kim T, Fiorentino M, Wong FNC. Phase-stable source of polarization-entangled photons using a polarization Sagnac interferometer. Phys Rev A. 2006; 73 doi: 10.1103/PhysRevA.73.012316

[41]. Saggio V, et al. Experimental few-copy multipartite entanglement detection. Nat Phys. 2019; 15 :935–940. DOI: 10.1038/s41567-019-0550-4 [PubMed: 31485254]

[42]. Marsili F, et al. Detecting single infrared photons with 93% system efficiency. Nat photon. 2013; 7 :210–214. DOI: 10.1038/nphotonc.2013.13

Europe PMC Funders Author Manuscripts

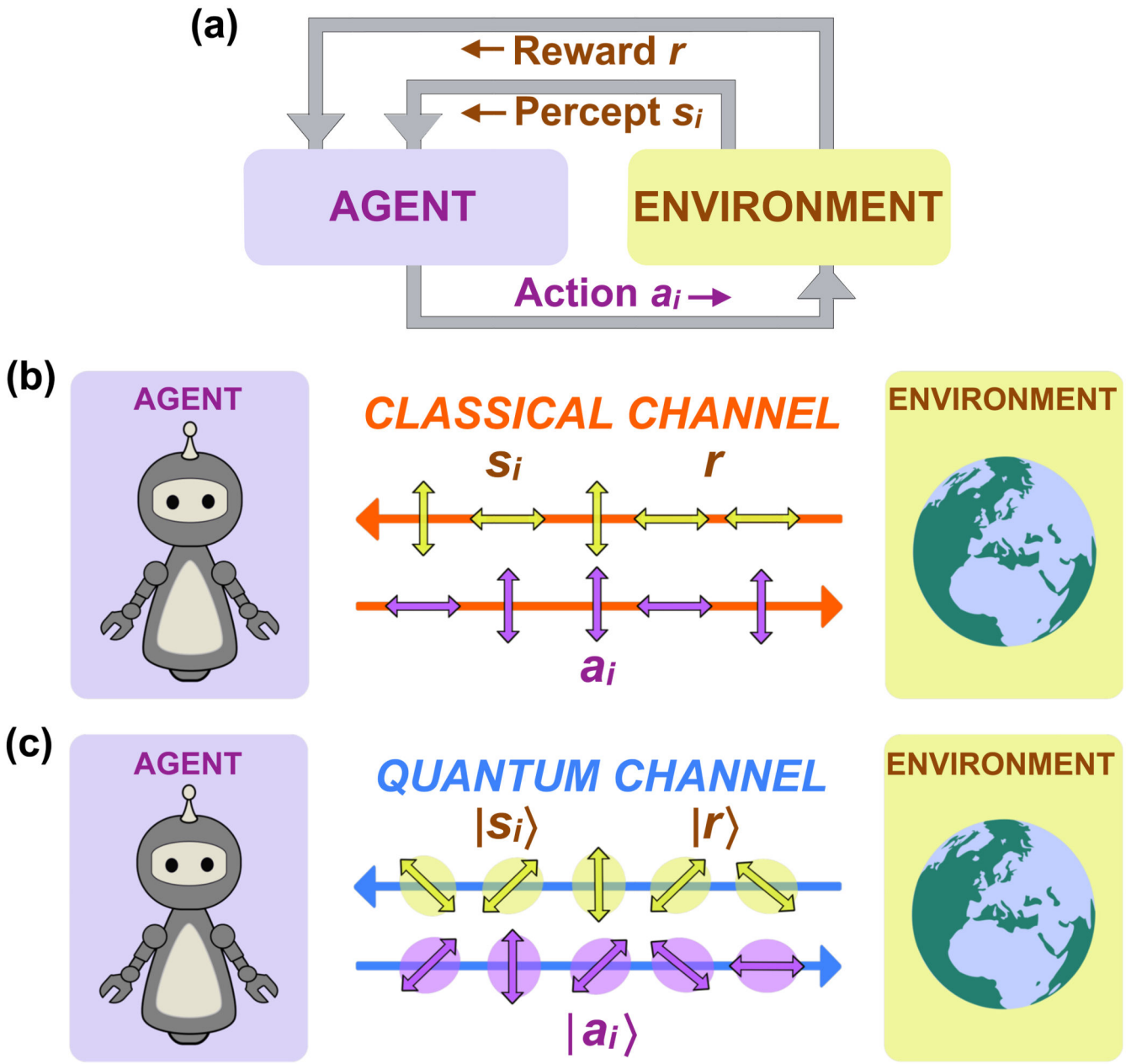

Fig. 1.

Schematic of a learning agent. (a) An agent interacts with an environment by receiving perceptual input s_i and outputting actions a_i. When the correct a_i is chosen, the environment issues a reward r that the agent uses to enhance its performance in the next round of interaction. (**b**) Agent and environment interacting classically, i.e., using a classical channel, where communication is only possible via a fixed preferred basis (e.g., vertical or horizontal photon polarization). (**c**) Agent and environment interacting via a quantum channel, where arbitrary superposition states are exchanged.

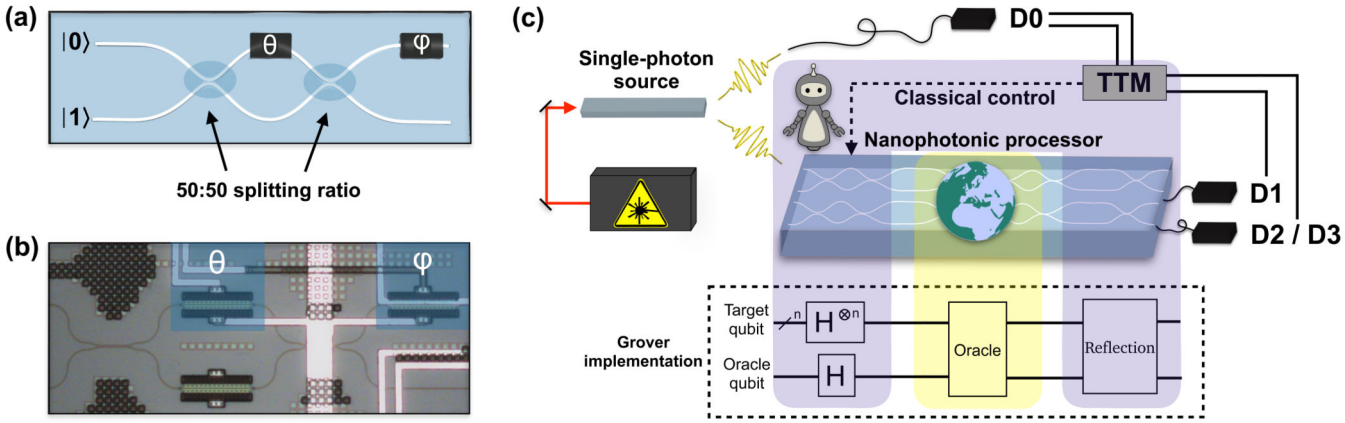

Fig. 2.

Experimental setup. (**a**) Single programmable unit consisting of a Mach-Zehnder interferometer (MZI) equipped with two fully tuneable phase shifters, one internal allowing for a scan of the output distribution over $\theta \in [0, 2\pi]$, and one external dictating the relative phase $\phi \in [0, 2\pi]$ between the two output modes. This makes the MZI act as a fully tuneable beam splitter and allows for coherent implementation of sequences of quantum gates. (**b**) Image of a single MZI in the processor. The third phase shifter in the bottom arm of the interferometer is not used. (**c**) Overview of the setup. A single-photon source generates single-photon pairs at telecom wavelength. One photon is sent to a single-photon detector D0, while the other one is coupled into the processor and undergoes the desired computation. It is then detected, in coincidence with the photon in D0, either in detector D1 or in D2/D3 after the agent plays the classical/quantum strategy (see Fig. 3 for more details). The coincidence events are recorded with a custom-made time tagging module (TTM). Different areas of the processor are assigned to either the agent or the environment, that can perform a Grover-like quantum search to look for rewarded action sequences in quantum epochs. The agent has access to a classical control that updates its policy.

Europe PMC Funders Author Manuscripts

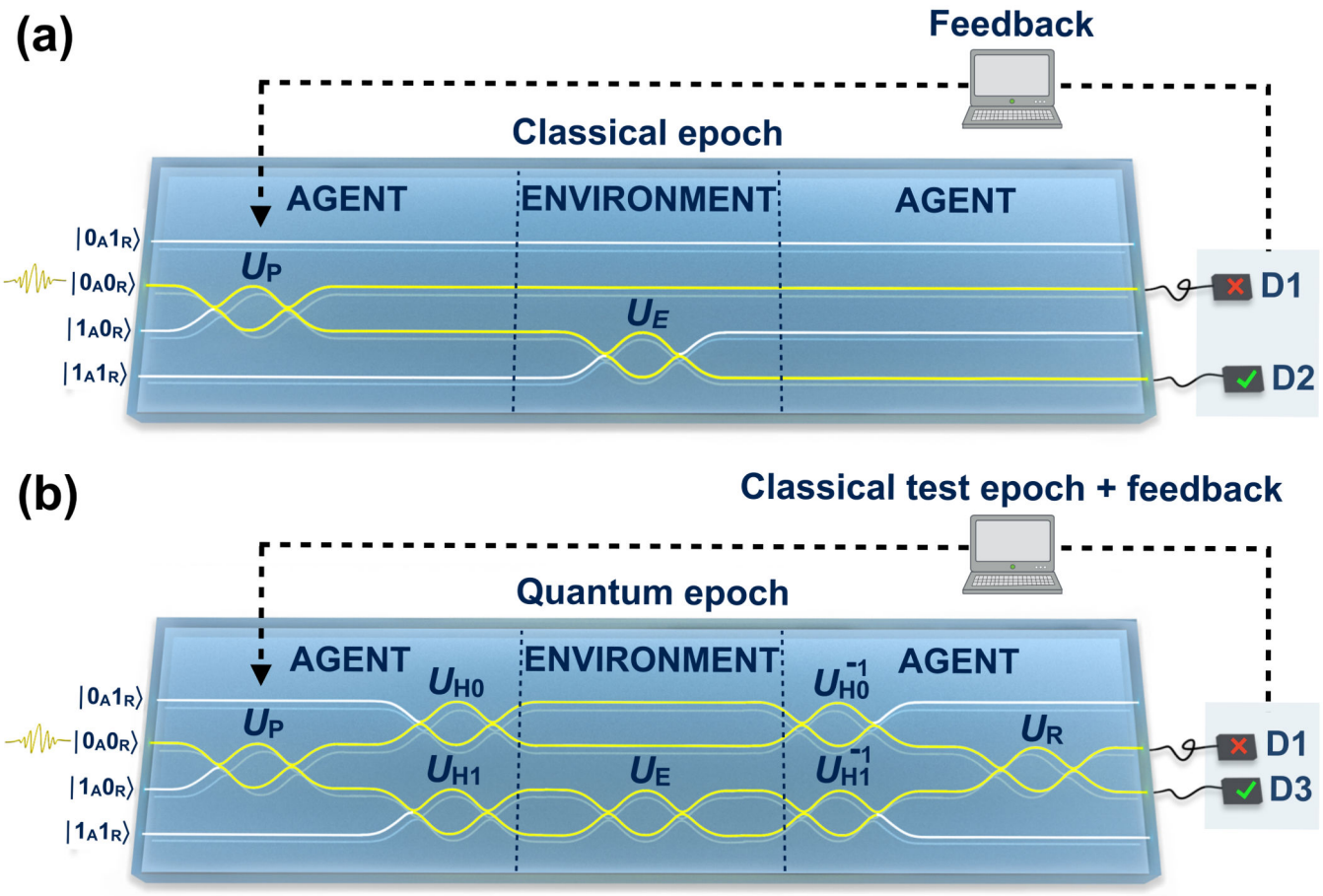

Fig. 3.

Circuit implementation. One photon is coupled into the $|0_A0_R\rangle$ waveguide and undergoes different unitaries depending on whether a classical (**a**) or a quantum (**b**) epoch is implemented. The waveguides highlighted in yellow show the photon's possible paths. Identity gates are represented by straight waveguides. Only the part of the processor needed for the computation is represented.

Europe PMC Funders Author Manuscripts

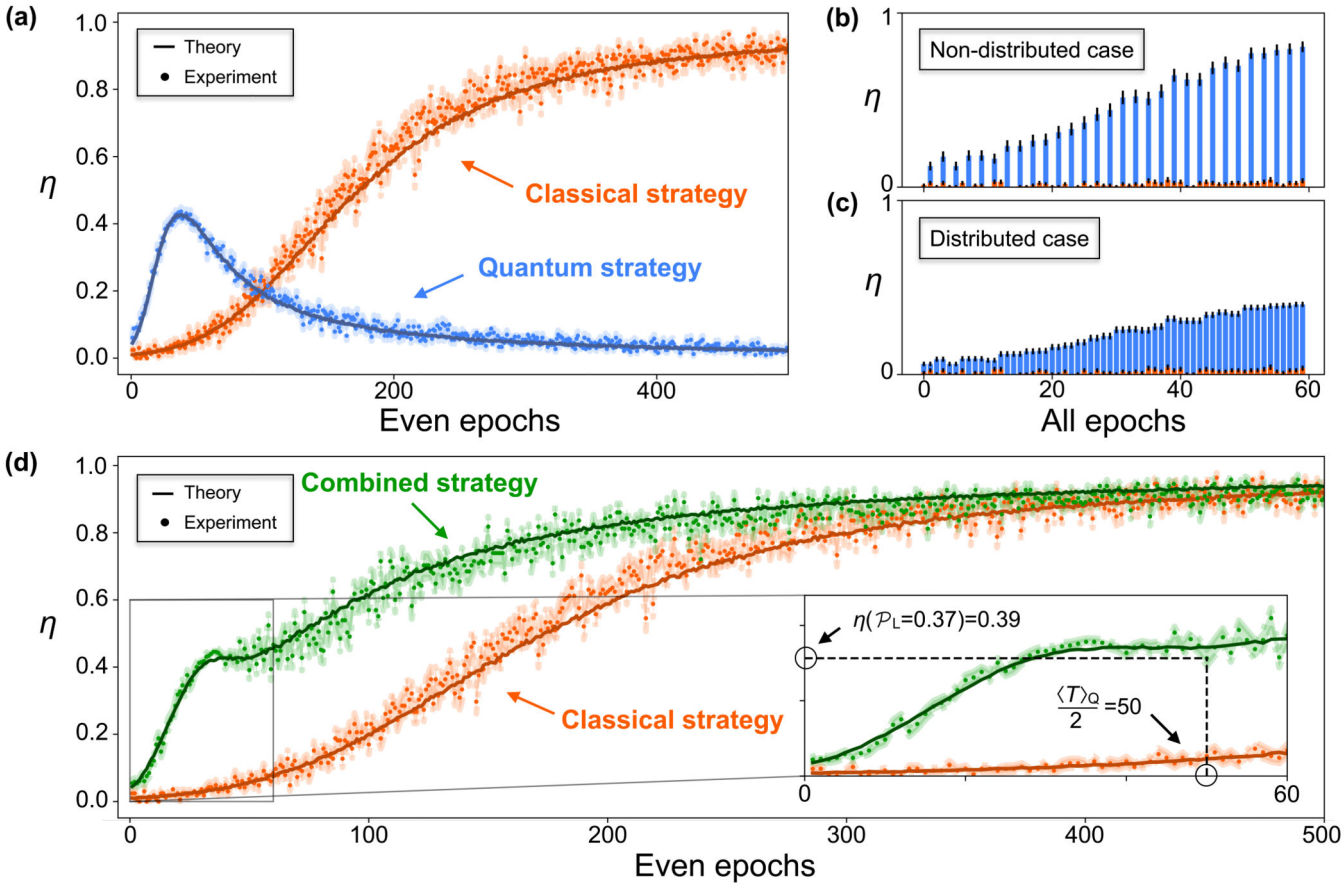

Fig. 4.

Behaviour of the average reward η for different learning strategies. The solid line represents the theoretical data simulated with $n = 10{,}000$ agents, while the dots represent the experimental data measured with $n = 165$ agents. The shaded regions indicate the errors associated to each single data point. (**a**) η of agents playing a quantum (blue) or classical (orange) strategy. (**b**) η accounting for rewards obtained only every second epoch in the quantum strategy, compared to (**c**) the case where the reward is distributed over the two epochs needed to acquire it. (**d**) Comparison between the classical (orange) and combined (green) strategy, where an advantage over the classical strategy is visible. Here, the agents stop the quantum strategy at their best performance (at $\varepsilon = 0.396$) and continue playing classically. The inset shows the point where agents playing the quantum strategy reach the winning probability $P_L = 0.37$, after $\langle T \rangle_Q = 100$ epochs.

Europe PMC Funders Author Manuscripts