

PAPER • OPEN ACCESS

Performance analysis of a hybrid agent for quantum-accessible reinforcement learning

To cite this article: Arne Hamann and Sabine Wölk 2022 *New J. Phys.* **24** 033044

View the [article online](#) for updates and enhancements.

You may also like

- [Parameterized reinforcement learning for optical system optimization](#)
Heribert Wankerl, Maike L Stern, Ali Mahdavi et al.
- [Variational quantum reinforcement learning via evolutionary optimization](#)
Samuel Yen-Chi Chen, Chih-Min Huang, Chia-Wei Hsing et al.
- [Reinforcement learning enhanced quantum-inspired algorithm for combinatorial optimization](#)
Dmitrii Beloborodov, A E Ulanov, Jakob N Foerster et al.



PAPER



Performance analysis of a hybrid agent for quantum-accessible reinforcement learning

OPEN ACCESS

RECEIVED
4 October 2021REVISED
26 January 2022ACCEPTED FOR PUBLICATION
7 March 2022PUBLISHED
30 March 2022

Original content from
this work may be used
under the terms of the
[Creative Commons
Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/).

Any further distribution
of this work must
maintain attribution to
the author(s) and the
title of the work, journal
citation and DOI.

Arne Hamann^{1,*} , and Sabine Wölk^{1,2} ¹ Institut für Theoretische Physik, Universität Innsbruck, Technikerstraße 21a, 6020 Innsbruck, Austria² Institute of Quantum Technologies, German Aerospace Center (DLR), D-89081 Ulm, Germany

* Author to whom any correspondence should be addressed.

E-mail: arne.hamann@uibk.ac.at**Keywords:** quantum reinforcement learning, reinforcement learning, amplitude amplification, hybrid quantum–classical algorithm, quantum search

Abstract

In the last decade quantum machine learning has provided fascinating and fundamental improvements to supervised, unsupervised and reinforcement learning (RL). In RL, a so-called agent is challenged to solve a task given by some environment. The agent learns to solve the task by exploring the environment and exploiting the rewards it gets from the environment. For some classical task environments, an analogue quantum environment can be constructed which allows to find rewards quadratically faster by applying quantum algorithms. In this paper, we analytically analyze the behavior of a hybrid agent which combines this quadratic speedup in exploration with the policy update of a classical agent. This leads to a faster learning of the hybrid agent compared to the classical agent. We demonstrate that if the classical agent needs on average $\langle J \rangle$ rewards and $\langle T \rangle_{\text{cl}}$ epochs to learn how to solve the task, the hybrid agent will take $\langle T \rangle_{\text{q}} \leq \alpha_s \alpha_o \sqrt{\langle T \rangle_{\text{cl}} \langle J \rangle}$ epochs on average. Here, α_s and α_o denote constants depending on details of the quantum search and are independent of the problem size. Additionally, we prove that if the environment allows for maximally $\alpha_o k_{\text{max}}$ sequential coherent interactions, e.g. due to noise effects, an improvement given by $\langle T \rangle_{\text{q}} \approx \alpha_o \langle T \rangle_{\text{cl}} / (4k_{\text{max}})$ is still possible.

1. Introduction

The application of quantum algorithms to machine learning provided promising results and evolved over the last years to the domain of quantum machine learning (QML) [1, 2]. The main types of machine learning are supervised, unsupervised and reinforcement learning (RL) [3] and each of them can be improved by quantum algorithms [1, 2, 4–8]. In RL an agent has to solve a task via interactions with an environment, perceiving a reward as a measure of its performance on the task. RL can be applied to solve problems from different areas such as robotics [9, 10], healthcare [11], or games such as Go [12].

The structure of RL allows for multiple quantum improvements. Various results show quantum advantages for quantum-enhanced agents interacting with a classical environment. In this way, improvements on the deliberation time of an agent [13–15] or a better performance via variational quantum circuits [16, 17] can be achieved.

Depending on the quantization of the environment, different methods can be applied [18–23] to gain quantum improvements in sample complexity, that is the number of interactions the agent has to perform with the environment to solve the task. In this work, we will focus on an approach with quantum-accessible environments introduced in [24]. This framework allows for a quadratic speedup in sample complexity during the exploration if access to some oraculized version of the environment is available (see [24] and section 3 for more details). The construction of such an oraculized environment is for example possible for deterministic strictly epochal environments, but also for stochastic environments [24] and epochal environments with variable epochal length [25]. There also exist specific environments for which super-polynomial and exponential improvements have been demonstrated [26].

The quadratic speedup in exploration will improve the agent's performance if the agent is luck-favoring³ on the environment. In this paper, we investigate how this speedup in exploration will lead to an improvement in sample complexity. For this purpose, we introduce a hybrid agent based on a feedback loop between quantum exploration and classical updates. We then analyze the resulting performance and compare it to a similar classical agent based on the same update rules. We determine possible speedups for two different situations: quantum-enhanced learning with (i) an ideal quantum computer and (ii) a quantum computer with only a limited number of possible coherent interactions.

An experimental implementation of such an agent with limited quantum resources based on a photonic quantum processor is described and demonstrated in [23]. In this paper, we concentrate on the theoretical background of such quantum-enhanced learning agents.

We will start with a brief review on RL in section 2. Then, we discuss quantum-enhanced exploration and introduce a hybrid learning agent in section 3. Consecutively, we analyze its behavior and compare the learning time of hybrid learning agents and their corresponding classical agents in section 4. We conclude by summarizing our results and discussing generalizations of the here discussed hybrid agent to more general scenarios in section 5.

2. Classical reinforcement learning

In reinforcement learning, an agent A is challenged to solve a task via interactions with an environment E . The interaction is usually modelled as a (partially observable) Markov decision process defined by the (finite) set of states S of the environment and a (finite) set of actions A performed by the agent.

The distribution of the initial state of the environment is described by a probability distribution $P(s)$. An action a performed by the agent leads to a state change of the environment from s to s' according to the transition probabilities $P(s'|a, s)$. The agent receives (partial) information, called observations or percepts $c(s)$, about the current state of the environment. The agent chooses its next action a based on the observed percept $c(s)$ according to its current policy $\Pi(a|c)$. Additionally, the agent receives a real-valued reward r rating its performance. The goal of the agent is to optimize the obtained reward in the long term by updating its policy Π based on its observations and thus to learn. Different classical algorithms have been developed such as SARSA, Q-learning, deep Q-learning or projective simulation [3, 27, 28] which provide good policies and update rules.

The performed actions of an agent together with the resulting observed percepts and rewards form its history

$$h_n = ((c_1, a_1, r_1), \dots, (c_{n-1}, a_{n-1}, r_{n-1}), (c_n, a_n, r_n)). \quad (1)$$

An agent interacting with an environment will update its policy $\Pi(a|c)$ according to the observed history. Also the evaluation of the performance of an agent is usually a function based on its history.

In general, the policy $\Pi(a|c)$ is probabilistic. Consider now a set of learning agents, with the same initial policy Π_{h_0} and update rules, which have performed n interactions. In general, the different agents observe different histories h_n leading to different consecutive policies Π_{h_n} . The average performance of these agents thus depends on the probability distribution $p_n(h_n)$ to observe different histories h_n .

In order to solve a given task, agents with different histories usually need a different number of interactions n . Therefore, it is necessary to extend the probability distribution $p_n(h_n)$ over histories with length n to the probability distribution $p(h_n)$ over the set of infinite histories. Here, $p(h_n)$ now determines the probability that an infinite long history h_∞ starts with h_n . A more explicit definition of $p(h_n)$ can be found in appendix A. In section 4, we will use the probability distribution of histories $p(h_n)$ to compare a set of classical and quantum-enhanced agents in order to determine possible quantum speedups.

3. A quantum-enhanced learning agent

To quantize RL we will use the approach of quantum accessible environments introduced in [24], where the agent and the environment are embedded in a communication scenario. The environment sends percepts and rewards to the agent, which responds with actions. This communication is quantized by encoding classical percepts c , actions a and rewards r into orthonormal quantum states $|c\rangle$, $|a\rangle$ and $|r\rangle$. In this scenario, the quantum-enhanced agent can choose to interact quantum by using superposition states as action states or classical by limiting itself to orthonormal basis states.

³ Meaning that lucky agents, which received more rewards in the past, are expected to outperform unlucky agents, which received less rewards.

In this paper, we consider strictly epochal environments. In strictly epochal environments, the interaction of an agent with its environment can be divided into epochs⁴. In each epoch, the environment is initialized in some initial state $|s_1\rangle$ and consecutively L percept-action pairs $(c_1, a_1), \dots, (c_L, a_L)$ are exchanged.

3.1. Quantum-enhanced exploration

For some environments, action sequences with a reward $r > 0$ can be found faster by using superpositions of action states and interference effects [24]. Similar to the wave-particle duality, interference effects between different sequences of actions $|\vec{a}\rangle$ are destroyed if information about the actions within one epoch are measured or memorized. As a consequence, all such information contained in the percepts c and states s of the environment need to be coherently deleted after an epoch in order to obtain interference effects. There exist different classes of environments where this is possible as discussed in [24], such as for example memoryless environments or deterministic strictly epochal environments.

In these cases, the environment can be used to create a quantum oracle O_E described by the unitary

$$O_E|\vec{a}\rangle_A|0\rangle_R = \begin{cases} |\vec{a}\rangle_A|0\rangle_R & \text{if } r(\vec{a}) = 0 \\ |\vec{a}\rangle_A|1\rangle_R & \text{if } r(\vec{a}) > 0. \end{cases} \quad (2)$$

by playing a single ($\alpha_o = 1$, e.g. memoryless environments) or several epochs ($\alpha_o = 2$, e.g. for deterministic strictly epochal environments). The constant α_o denotes how many epochs are required to effectively create one oracle O_E . The oracle O_E is equivalent to a controlled not-gate acting on the reward register and controlled by the action state $|\vec{a}\rangle_A$. We can create a quantum-enhanced learning agent, whenever such an oracle exist.

Quantum-enhanced agents use this oracle to find rewarded sequences of actions quadratically faster. That is, for a fixed policy, they need on average $\langle t \rangle_q = \alpha \sqrt{\langle t \rangle_{cl}}$ [29–32, 32, 33] epochs to find the next reward, whereas a classical agent would need $\langle t \rangle_{cl}$ epochs. Here, the constant $\alpha = \alpha_s \cdot \alpha_o$ is determined by the number of epochs α_o necessary to create the oracle O_E and a constant α_s depending on the applied quantum search algorithm [29–33]⁵. Given this quadratic speedup in exploration, it is possible to construct a basic quantum agent based on any classical agent. A basic quantum agent performs quantum searches for a certain amount of time and then trains an internal copy of the classical agent to reproduce the found rewarded sequences of actions. This basic quantum agent is on average luckier than the classical agent, as it will on average find rewarded sequences faster than the classical agent. Hence, if the agent-environment-setting is luck-favoring [1], it will outperform the classical agent [24].

However, more advanced quantum–classical hybrid agents can be constructed by alternating between quantum search and classical policy updating as discussed in this paper and experimentally demonstrated in [23]. Furthermore, we quantify the overall speedup in learning for these agents, which is in general not possible for the basic quantum agent described in [24].

3.2. Quantum-enhanced exploration and classical policy updates

A key feature of RL is the assumption that there exist not only correct and incorrect (sequences of) actions but rather a spectrum of better or worse (sequences of) actions. In addition, some resemblance between good (sequences of) action is usually implied. In these cases, finding good actions sequences which might be suboptimal (but with higher probability to find them) can help to find better (sequences of) actions, which are less probable. For example, there may exist many, not too long routes from city A to city B with different length. By slightly varying such a route, an even shorter and therefore better route might be found. As a consequence, many RL problems, like e.g. in deterministic strictly epochal environments, can in theory be solved by testing all possible action sequences and searching for the optimal one. However, the search space for such problems is usually too big and the optimum might be unknown such that solving the problem via straight forward search is not possible in practice and RL is used instead.

Quantum-enhanced exploration can speed-up the search for rewards quadratically. However, the search space might nevertheless be too large to find the optimal solution via direct search. We therefore use in these cases quantum-enhanced exploration to search for general rewarded actions and use classical methods for policy updates. The policy updates will change the underlying search space and help to find better solutions. In general, also these modified search spaces are still big such that quantum-enhanced exploration can be also advantageous after the first policy updates. Therefore, we introduce in the following

⁴ Epochs are often also called episodes. We chose epochs to be consistent with the notion of strictly epochal environments as used e.g. in [24, 25].

⁵ Typical values are $\alpha_s = \frac{\pi}{4}$ if the classical reward probability is known or $\alpha_s = \frac{9}{4}$ if it is unknown.

a hybrid quantum–classical learning agent. This agent uses quantum-enhanced exploration not only in the beginning, before any policy update took place. Thought, it alternates between quantum-enhanced exploration and classical policy updates. We establish a feed-back loop between quantum-enhanced exploration and classical policy updates such that both procedures can profit from each other.

3.3. Description of the hybrid learning agent

In the following, we consider classical agents with a policy which can be described by a probability distribution $\hat{\Pi}(\vec{a})$ for action sequences for a complete epoch and environments which allow the construction of an effective oracle O_E , equation (2). This is e.g. possible for agents in deterministic strictly epochal environments which use mapping as described in appendix B.

We can create a hybrid quantum classical learning agent based on this classical agent. The hybrid agent alternates between quantum epochs used for quantum exploration and classical epochs used for policy updates as shown in figure 1 and described in detail below:

- (a) Given the classical probability distribution $\hat{\Pi}(\vec{a})$ of action sequence, estimate a lower bound Q_{\min} on the winning probability

$$Q = \sum_{\{\vec{a}|r(\vec{a})>0\}} \hat{\Pi}(\vec{a}) \quad (3)$$

and prepare the action state

$$|\psi\rangle_A = \sum_{\{\vec{a}\}} \sqrt{\hat{\Pi}(\vec{a})} |\vec{a}\rangle_A \quad (4)$$

and the reward state $|-\rangle_R = (|0\rangle_R - |1\rangle_R)/\sqrt{2}$.

- (b) Use several epochs to perform amplitude amplification [29, 30, 33] until a reward is found. This consists e.g. of the following steps, if the winning probability Q is not known exactly:
- (1) Initialize $m = 1$ and $\lambda = 6/5$ and choose a random integer $k < m$.
 - (2) Use $\alpha_o k$ epochs to perform k Grover iterations

$$|\psi'\rangle_A |-\rangle_R = (FO_E)^k |\psi\rangle_A |-\rangle_R, \quad (5)$$

where $F = \mathbb{1} - 2|\psi\rangle\langle\psi|$ is the reflection around the initial action state.

- (3) Perform a measurement on the resulting state $|\psi'\rangle_A$ to determine a possible sequence of actions \vec{a} .
 - (4) Play one classical epoch with the measured sequence of actions \vec{a} and record the observed percepts and reward.
 - (5) Terminate if the sequence wins a reward; else set m to $\min(\lambda m, \sqrt{1/Q_{\min}})$ and restart.
- (c) Use the most recent classical information from step b(4) to update the classical policy.
- (d) Determine the new probability distribution $\hat{\Pi}(\vec{a})$ and Q_{\min} according to the new policy and repeat.

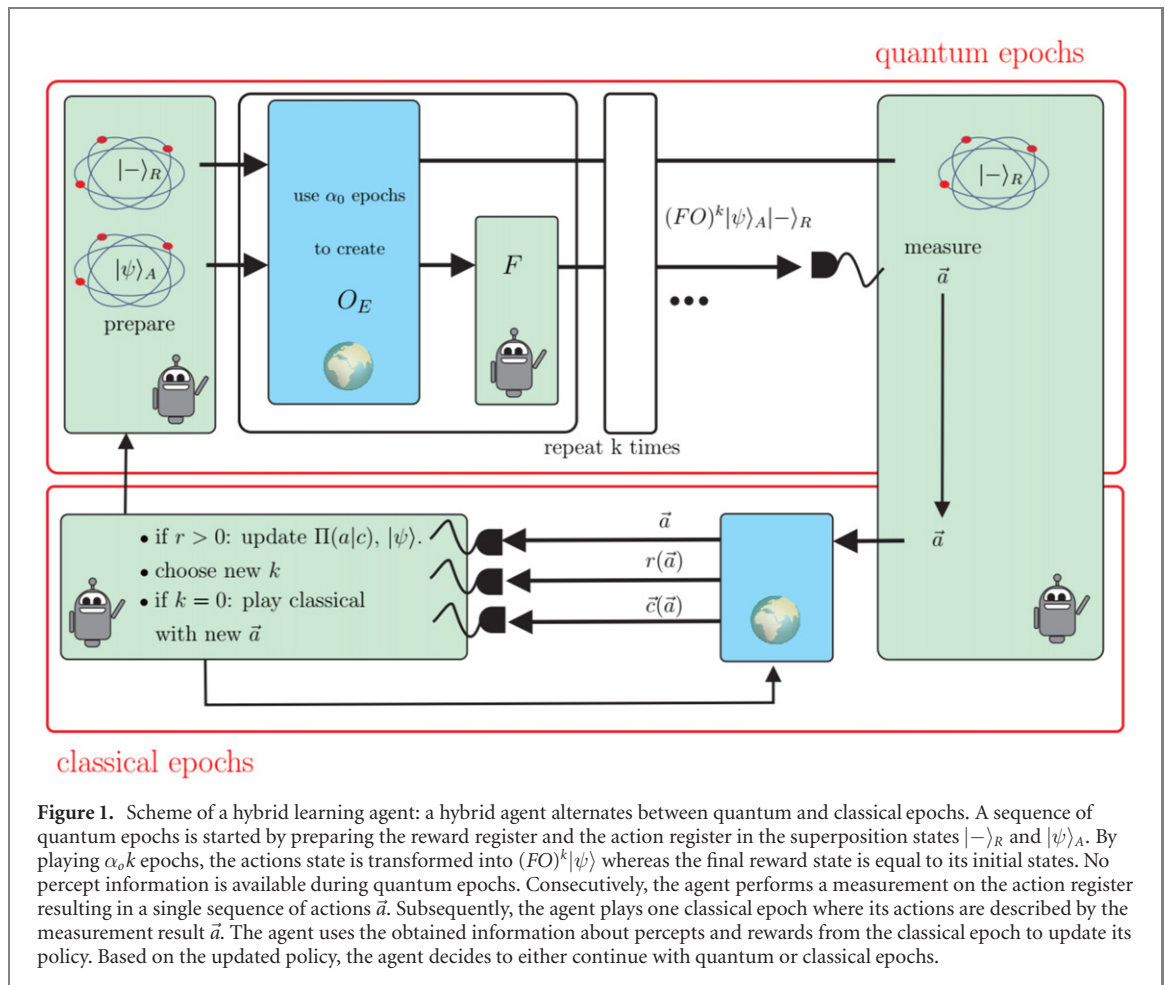
The probability to observe a reward in step b(4) after performing k Grover iterations and playing $\alpha_o k + 1$ epochs is given by (see appendix C)

$$G(Q, k) = \sin^2 \left[(2k + 1) \arcsin(\sqrt{Q}) \right]. \quad (6)$$

A classical agent sampling its actions directly from $\hat{\Pi}(\vec{a})$ would observe a reward with probability Q in each single epoch. As a consequence, performing Grover iterations leads to observing a reward more frequently only if $Q < G(Q, k)/(\alpha_o k + 1)$. This is only the case for winning probabilities Q below a certain threshold $Q \leq Q_{\max}$. This threshold is given by $Q_{\max} \approx 0.3964$ for simple learning problems with $\alpha_o = 1$ and the minimal number of Grover iterations $k = 1$.

A hybrid agent following the steps b(1)–(5) will automatically decrease the probability to perform Grover iterations because it always starts with $k = 0$ each time after it has found a reward. Thus performing Grover iterations becomes more and more unlikely the larger Q . Nevertheless, it might be advantageous to fix $k = 0$, and thus restrict the agent to classical behavior, at a certain point of learning. This behavior can be steered e.g. by the total number of found rewards or the observed frequency of rewards in the last few classically played epochs.

The above described hybrid agents finds rewards quadratically faster than a corresponding classical agent with the same policy $\hat{\Pi}(\vec{a})$. However, the effect of this quantum-enhanced exploration on the learning time depends on the used policy update rules and the environment. Two simple examples are (i) a very exploitative agent in a deterministic environment which chooses to play always the first found rewarded sequence of action, once it has found it or (ii) a very explorative agent which does not update its policy.



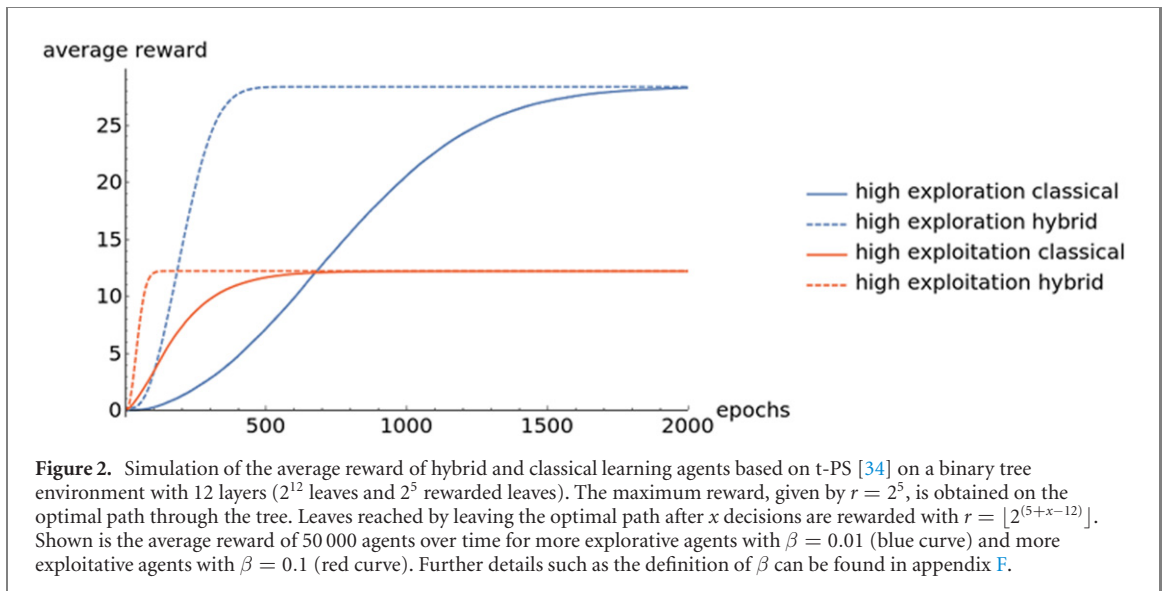
In the first case, the quantum-enhanced agent achieves a quadratic speed-up in the learning time. In the second case, the quantum-enhanced agent usually finds rewards more often than the classical agent. Yet, both agents never learn, such that no improvement in the learning time can be achieved. For settings where future rewards and policy updates depend solely on the found rewards, we prove in the next section a quasi-quadratic improvement of the learning time.

4. Analysis of the hybrid agent

In the following, we compare the behavior of the above described hybrid learning agents with corresponding classical agents. The behavior of a learning agent depends on the history it has observed. Hybrid agents only observe percepts and rewards during classical epochs (step b(4)). As a consequence, the general history of a hybrid agent, containing all actions, percepts and rewards for all epochs, is not well defined. However, we can define the rewarded history h_r , which is the history of an agent reduced to rewarded epochs. That is, epochs where a non-vanishing reward $r > 0$ was found. E.g. when a non-vanishing reward was obtained in every even epoch, the corresponding rewarded history would be (e_2, e_4, e_6, \dots) with e_j containing the sequence of actions, percepts and rewards observed in epoch j . We define the event of observing a rewarded history h_r as the set of all infinite length histories, which reduce to rewarded histories starting with h_r .

Therefore, we consider only environments and agents with a behavior which is completely defined by their rewarded history. In this case, the behavior of the agent can be described by $\hat{\Pi}(\vec{a})$ and it only updates its policy when receiving a reward. In addition, all agents (hybrid and classical) use the same update rules which are based solely on rewarded epochs. As a consequence, the behavior of all agents is solely determined by rewarded epochs.

The analysis of the hybrid agent will concentrate on three properties. First, we will show that the probabilities to observe a given rewarded history are identical for the classical and the corresponding hybrid agent. Based on these results, we then determine the scaling advantage in the learning time for a hybrid



agent with unlimited quantum resources. In the end, we will go one step into the direction of noisy intermediate-scale quantum (NISQ) computers and investigate achievable improvements based on a limited number of coherent interactions.

4.1. Distribution of rewarded histories h_r

The final quality of a trained agent depends on the initial policy and performed policy updates and thus in the here considered cases solely on the observed rewarded history. That is the history reduced to rewarded epochs. As a consequence, a classical agent and a hybrid agent which have observed the same rewarded history h_r obtain the same classical policy. In addition, the probability $p(h_r)$ to observe a given rewarded history h_r is equal for a hybrid agent and its corresponding classical agent as stated below and proven in appendix D:

Theorem 1 *Let E be a (time independent) epochal environment with corresponding oracle O_E and A_{cl} be a classical agent with a corresponding hybrid agent A_q as defined above. Then, the probability distribution of rewarded histories $p(h_r)$ for A_{cl} interacting with E and A_q interacting with O_E and E are equal.*

Theorem 1 leads to several direct consequences when comparing a group of trained hybrid agents with a group of corresponding classical agents. Consider for example a setup with different possible rewards r . Not all agents will learn to achieve the maximal possible reward in the long run due to their individual observed histories. Yet, a group of hybrid agents and a group of corresponding classical agents converge towards the same average reward as shown in figure 2 due to theorem 1. Or consider groups of agents, where each agent plays until it has found J rewards. Then, all hybrid agents switch to complete classical behavior and we compare the consecutive behavior of the group of hybrid agents with the group of classical agents. In this case, both groups behave exactly similar and the behavior of both groups are indistinguishable from each other.

In ML, there often exists a trade-off between exploration and exploitation. This manifests itself often in a trade-off between fast learning and optimal behavior in the long run [3, 35]. An example of such a trade-off is visualized in figure 2. Here, we considered classical policies depending on some parameter β influencing the ratio between exploration and exploitation (see appendix F). Exploitative agents ($\beta = 0.1$, red curve) learn faster but get stuck more likely in local maxima. As a consequence, the expected average reward will be lower in the long run. Whereas explorative agents ($\beta = 0.01$, blue curve) learn slower but reach in the long run a higher expected average reward. This is the case for classical agents as well as for hybrid agents. In contrast, going from a classical agent to its corresponding hybrid agent, both based on the same policy, leads to faster learning without sacrificing the expected average reward due to theorem 1.

4.2. Learning time

In order to quantify the speedup of our hybrid agent we define the learning time $T(h)$ of an agent with history h as the minimal number of epochs t this agent needs to reach a winning probability $Q = Q_t$, (3) above a predefined learning threshold that is $Q_t \geq Q_l$. We say the learning time is infinite if $Q_t < Q_l \forall t$. A learning threshold is achievable, if and only if the probability $p(\{h_\infty \in H_\infty | T(h_\infty) = \infty\}) = 0$ for histories of agents with infinite learning time is zero.

In general, a speedup in learning is only achieved while the hybrid agent performs amplitude amplification which is only the case for winning probabilities $Q < Q_{\max}$ (compare section 3). In these cases, our hybrid agent can achieve the following speedup compared to its corresponding classical agent:

Theorem 2 *Let E be a (time independent) epochal environment with corresponding oracle O_E and A_{cl} be a classical agent with a corresponding hybrid agent A_q as defined above. Then, for all achievable reward probabilities $Q_i < Q_{\max}$, the average learning time of a hybrid agent $\langle T \rangle_q$ and its corresponding classical agent $\langle T \rangle_{cl}$ are connected via⁶*

$$\langle T \rangle_q \leq \alpha \sqrt{\langle T \rangle_{cl} \langle J \rangle}, \quad (7)$$

where $\langle J \rangle$ denotes the average number of rewards agents need to observe in order to learn.

Hence the hybrid agent learns quadratically faster, while it converges towards the same average policy as its corresponding classical agent.

Proof. In order to determine the average $\langle T \rangle$, we split the learning time into intervals of length t_j of constant policy. Thus, the interval j starts after j non-vanishing rewards have been observed and ends with the observation of the next non-vanishing reward. In addition, we define $J(h)$ as the number of non-vanishing rewards an agent with history h has observed until it has learned. As a consequence, the learning time of an agent with history h is given by

$$T(h) = \sum_{j=0}^{J(h)-1} t_j(h). \quad (8)$$

The average learning time $\langle T \rangle$ is thus determined by averaging $T(h)$ over all possible histories h . We perform this averaging in two steps. First, we average over all histories h which can be reduced to the same rewarded history h_r . Consecutively, we average over all rewarded histories h_r .

The policy $\hat{\Pi}$, the winning probability Q and thus J depend not on the exact history h but solely on the rewarded history h_r for the learning agents and environments considered here. Thus, a classical agent which has found already j rewards with rewarded history h_r needs on average

$$\langle t_j(h_r) \rangle_{cl} = \frac{1}{Q_j(h_r)} \quad (9)$$

epochs to find the next reward (see appendix E for a more detailed discussion). A quantum-enhanced agent as described in section 3 finds rewards quadratically faster [29, 30]. As a consequence, the average interval time $\langle t_j(h_r) \rangle_q$ for such agents is given by

$$\langle t_j(h_r) \rangle_q \leq \frac{\alpha}{\sqrt{Q_j(h_r)}} = \alpha \sqrt{\langle t_j(h_r) \rangle_{cl}}. \quad (10)$$

The learning time $\langle T(h_r) \rangle$ averaged over all agents with the same rewarded history h_r is determined by

$$\langle T(h_r) \rangle = \sum_{j=0}^{J(h_r)-1} \langle t_j(h_r) \rangle. \quad (11)$$

As a consequence, quantum-enhanced learning agents with a rewarded history h_r learn on average in

$$\langle T(h_r) \rangle_q \leq \sum_{j=0}^{J(h_r)-1} \alpha \sqrt{\langle t_j(h_r) \rangle_{cl}} \quad (12)$$

$$\leq \alpha \sqrt{J(h_r)} \sqrt{\langle T(h_r) \rangle_{cl}} \quad (13)$$

epochs. Here, we have used the Cauchy–Schwarz inequality in the second step.

Averaging $\langle T(h_r) \rangle$ over all possible rewarded histories leads to (see appendix E)

$$\langle T \rangle = \sum_{\{h_r\}} p(h_r) \langle T(h_r) \rangle. \quad (14)$$

⁶ With constant $\alpha = \alpha_s \cdot \alpha_o$ where $\alpha_{s,o}$ are typically of the order $\alpha_o \in \{1, 2\}$ and $\alpha_s \approx \frac{9}{4}$.

Note, that the probability $p(h_r)$ to observe a given rewarded history is identical for quantum-enhanced learning agents and their corresponding classical agents due to theorem 1. As a consequence, using again the Cauchy–Schwarz inequality leads to

$$\langle T \rangle_q \leq \alpha \sum_{h_r} p(h_r) \sqrt{J(h_r) \langle T(h_r) \rangle_{cl}} \quad (15)$$

$$\leq \alpha \sqrt{\langle J \rangle \langle T \rangle_{cl}}, \quad (16)$$

where $\langle T \rangle_q$ and $\langle T \rangle_{cl}$ denote the average learning times of the quantum-enhanced agents and classical agents, respectively. $\langle J \rangle$ denotes the average number of rewards an agent with given policy update rules needs to find in order to learn. $\langle J \rangle$ is equal for classical and quantum enhanced agents. ■

4.3. Noisy quantum devices

Theorem 2 quantifies the achievable speedup of our hybrid agent assuming the existence of a perfect quantum computer. However, quantum computers which will be available soon will be noisy and thus possess limited coherence times. Such behavior can be approximated by assuming that noise can be neglected for up to $\alpha_o k_{\max}$ consecutive quantum epochs whereas the effect of noise starts to become crucial if more than $\alpha_o k_{\max}$ quantum epochs are performed consecutively. Therefore, the question arises which improvements can be achieved if only a limited number of epochs can be performed coherently in a row. That is, we investigate the achievable improvement assuming that the number of Grover iterations k in step b(2) of our hybrid agent is limited to $k \leq k_{\max}$. Obviously, the limitation of k plays only a role if the winning probability Q , (3), is small such that

$$Q \leq Q_{k_{\max}} = \sin^2 \left[\frac{\pi}{2(2k_{\max} + 1)} \right]. \quad (17)$$

Here, $Q_{k_{\max}}$ denotes the smallest reward probability which leads to $G(Q_{k_{\max}}, k_{\max}) = 1$, see (6), when performing k_{\max} Grover iterations. It is also possible to achieve an improvement in this regime albeit a linear one as stated in the following theorem:

Theorem 3 *Let E be a (time independent) epochal environment with corresponding oracle O_E and A_{cl} be a classical agent with a corresponding hybrid agent A_q as defined above. Then, a hybrid learning agent limited to maximal k_{\max} sequential Grover iterations interacting with O_E can reach achievable reward probabilities $Q_l < Q_{k_{\max}}$ in a learning time $\langle T \rangle_q$ with*

$$\langle T \rangle_q \leq \frac{\alpha_o \pi^2 \langle T \rangle_{cl}}{16 k_{\max}}, \quad (18)$$

where $\langle T \rangle_{cl}$ is the learning time of the corresponding classical agent interacting with E .

Note, this is a linear improvement in query complexity which can be crucial in certain settings independent of the running time of the algorithm. Figure 3 shows the performance of limited and unlimited agents on the binary tree environment described in appendix F. The separation between the hybrid and the limited agent increases for decreasing initial winning probabilities Q and thus typically with the size of the environment.

Proof. Again, we first consider a learning agent with a given rewarded history h_r before we average over all possible rewarded histories.

A classical winning probability $Q_j = Q_j(h_r)$ leads after k_{\max} Grover iterations due to (6) to an enhanced winning probability [29, 30]

$$G(Q_j, k_{\max}) = \sin^2 \left[(2k_{\max} + 1) \arcsin(\sqrt{Q_j}) \right]. \quad (19)$$

As a consequence, the expected interval time t_j (compare proof of theorem 2) of the hybrid agent is given by

$$\langle t_j(h_r) \rangle_q = \frac{\alpha_o k_{\max} + 1}{G(Q_j, k_{\max})}. \quad (20)$$

Here, we have taken into account that $\alpha_o k_{\max}$ epochs are necessary to create k_{\max} Grover iterations plus one epoch to determine the reward. The denominator can be approximated with the help of the inequality $\sin(x) \geq x \cdot \sin(x_0)/x_0$, which is valid in a range $0 \leq x \leq x_0 \leq \pi/2$. In our case, we consider the interval $0 \leq x \leq x_0 = \frac{\pi}{2}$.

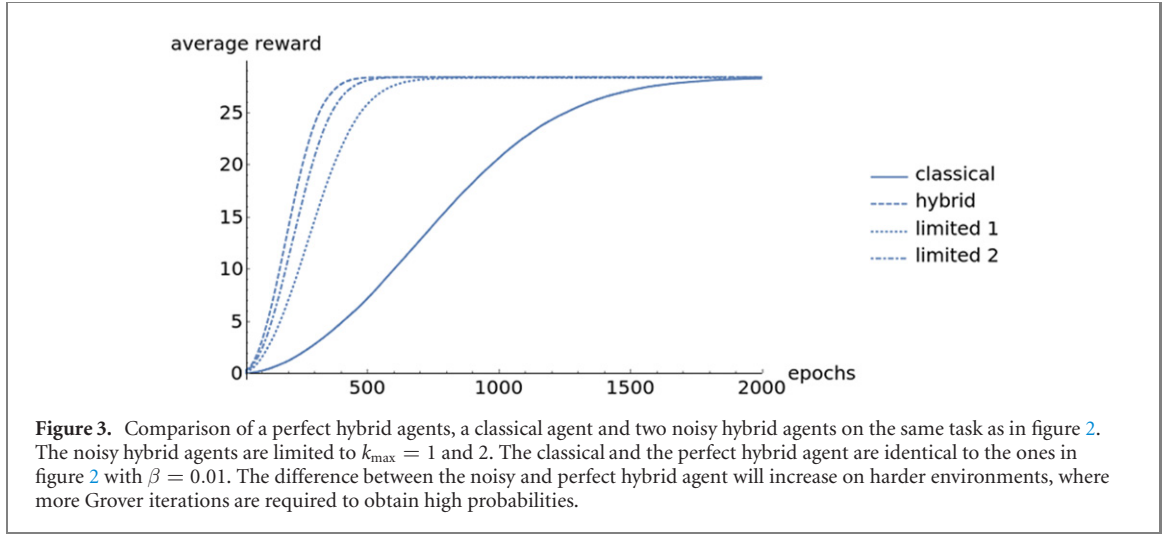


Figure 3. Comparison of a perfect hybrid agents, a classical agent and two noisy hybrid agents on the same task as in figure 2. The noisy hybrid agents are limited to $k_{\max} = 1$ and 2. The classical and the perfect hybrid agent are identical to the ones in figure 2 with $\beta = 0.01$. The difference between the noisy and perfect hybrid agent will increase on harder environments, where more Grover iterations are required to obtain high probabilities.

Identifying $x = (2k_{\max} + 1) \arcsin(\sqrt{Q_j})$ and using $\arcsin(\sqrt{Q_j}) > \sqrt{Q_j}$ leads to

$$G(Q_j, k_{\max}) \geq \frac{4}{\pi^2} (2k_{\max} + 1)^2 Q_j \quad (21)$$

$$\geq \frac{4}{\pi^2} 4k_{\max} (k_{\max} + 1) Q_j. \quad (22)$$

With the help of $\langle t_j(h_r) \rangle_{\text{cl}} = 1/Q_j$ and $1 \leq \alpha_o$, we find as a result

$$\langle t_j(h_r) \rangle_{\text{q}} \leq \frac{\alpha_o \pi^2}{16k_{\max}} \langle t_j \rangle_{\text{cl}}. \quad (23)$$

A summation over $0 \leq j < J(h_r)$ gives the average learning time $\langle T(h_r) \rangle$ and the average over all possible rewarded histories h_r leads to (20) due to theorem 1. ■

Notice that the inequality can be tightened by using smaller values for x_0 , allowing to prove higher improvements for smaller limits on Q_l . The extreme case with $x_0 \rightarrow 0$ leading to $\sin(x_0)/x_0 \rightarrow 1$ is approximately reachable for $Q_l \ll 1$ and $k_{\max} \ll \frac{1}{Q_l}$ leading to

$$\langle T \rangle_{\text{q}} \approx \alpha_o \frac{\langle T \rangle_{\text{cl}}}{4k_{\max}}. \quad (24)$$

In general, the total learning process of a quantum-enhanced learning agent with limited quantum resources can be split into three phases. The first phase is defined by winning probabilities $Q_j \leq Q_{k_{\max}}$. In this phase, a linear improvement proportional to k_{\max} is achievable according to theorem 3. The second phase is defined by $Q_{k_{\max}} < Q_j < Q_{\max}$. Here, a complete Grover search is possible and beneficial and a quadratic speedup is achieved according to theorem 2. The last phase $Q_j \geq Q_{\max}$ is the phase, where the hybrid agent reproduces the classical agent and therefore no speedup can be generated in this phase.

5. Conclusions and outlook

In this paper, we analyzed the quantum speedup which can be gained by combining classical RL agents with quantum exploration. We compared quantum-enhanced learning agents, which alternate between quantum exploration and classical policy updates, with classical learning agents based on the same policies and update rules.

For analytical reasons, we considered only agents which fulfilled the following two criteria: first, agents are able to determine the probabilities for all possible action sequences for one epoch beforehand. Second, policy updates are completely determined by the rewarded history, that is the history of an agent reduced to epochs with a non-vanishing reward. We also assumed that if the behavior of the environment changes from one epoch to another. Then, these changes are again completely determined by the rewarded history.

For this classes of agents and environments, we proved that the probability $p(h_r)$ to observe a given rewarded history for a quantum-enhanced agent is equal to the one for a corresponding classical agent. As a consequence, a hybrid agent and its corresponding classical agent behave similarly. That is, quantum and

classical agents with the same rewarded history h_r follow the same policy and quantum and classical agents tend towards the same average reward per epoch in the long run.

Based on this result, we proved a quadratic speedup in learning for quantum-enhanced agents compared to their classical counterparts without sacrificing the quality of the learned solution. Furthermore, we also analyzed speedups which can be obtained with limited quantum hardware. We demonstrated that a quantum improvement can also be obtained if only a limited number of consecutive epochs can be performed coherently.

A proof of principle experiment of a quantum-enhanced learning agent as discussed in this paper has been demonstrated experimentally with a nanophotonic quantum processor [23]. This concept of photonic quantum-enhanced learning agents can easily be expanded to more advanced architectures for RL. Indeed, it is possible to define the policy Π with the help of variational quantum circuits such as photonic networks [34].

In general, also classical learning agents and environments, which do not obey the criteria mentioned above, can be combined with quantum exploration. However, the behavior of quantum-enhanced learning agents in such scenarios might differ from the behavior of the corresponding classical agents. For example, the probability for an epoch with vanishing reward is different for quantum and classical agents. As a consequence, policy updates based on epochs with vanishing reward might lead in the long run to different policies for quantum-enhanced agents and classical agents. Therefore, no general comparison between quantum-enhanced agents and classical agents about the long term expected reward and learning time can be made in these cases.

Many learning setups are luck-favoring in the sense that agents which received more rewards in the beginning also receive more rewards on average in the future (compare [24]). In general, quantum-enhanced agents will learn faster in such setups. However, the obtainable speedup will depend on the exact setup. In very rare occasions, an agent which observed fewer rewards in the beginning might perform better in the long run. We do not expect any quantum speedups in learning in such setups.

In the future, it is necessary to further study realistic learning setups based on general classical agents and faulty quantum hardware for example with the help of simulations and further numerical analyzes. In this way, more specific predictions about possible quantum improvements might be gained.

Acknowledgments

AH and SW thanks Hans J Briegel, Vedran Dunjko, Davide Orsucci and Oliver Seifrin for fruitful discussions. AH acknowledges support from the Austrian Science Fund (FWF) through the project P 30937-N27. SW acknowledges support from the Austrian Science Fund (FWF) through SFB BeyondC F7102.

Data availability statement

The data supporting the findings of this study are openly available at the following URL/DOI: [10.5281/zenodo.5879296](https://doi.org/10.5281/zenodo.5879296).

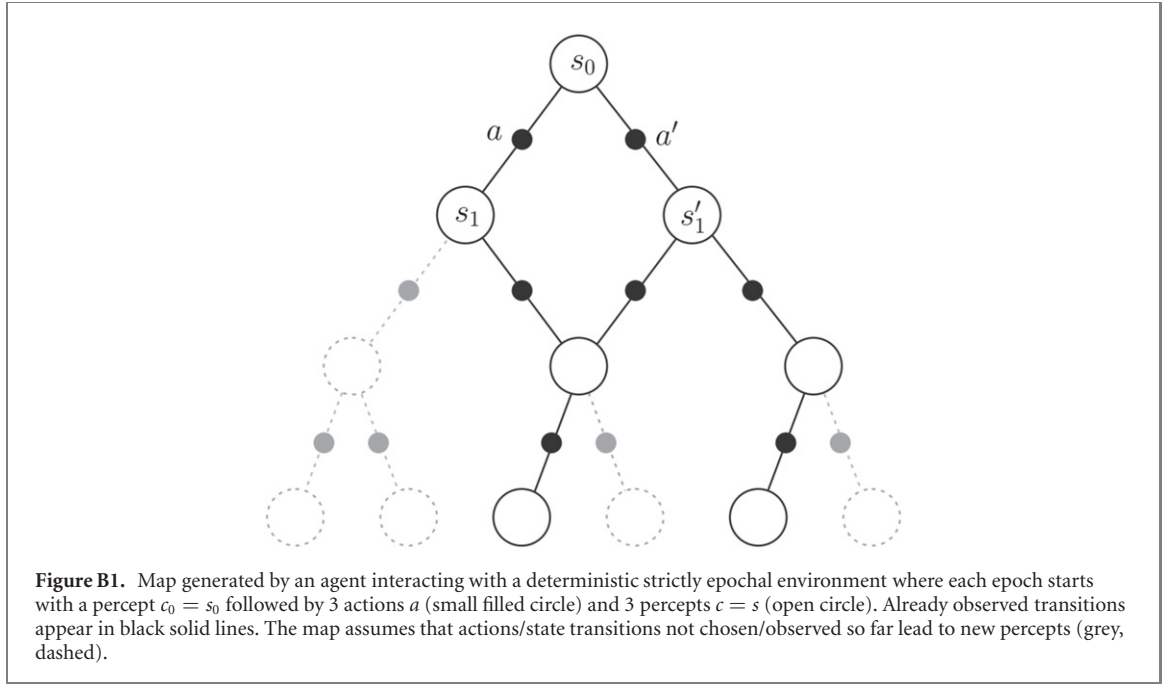
Appendix A. The history of an agent

We define by H_n the set of all possible histories h_n with length n which can be observed by a set of agents. The probability $p_n(h_n)$ to observe a history of length n can be determined recursively. Let $h_{n+1} = ((c_1, a_1, r_1), \dots, (c_n, a_n, r_n), (c_{n+1}, a_{n+1}, r_{n+1}))$ be an extension of the history $h_n = ((c_1, a_1, r_1), \dots, (c_n, a_n, r_n))$. Then, the probability p_{n+1} to observe the history h_{n+1} is given by

$$p_{n+1}(h_{n+1}) = PR(r|s_{n+1}, s_n) \cdot P(s_{n+1}|a_{n+1}, s_n) \cdot \prod_{h_n} (a_{n+1}|c_{n+1}) \cdot PC(c_{n+1}|s_n) \cdot p_n(h_n). \quad (\text{A.1})$$

Here, $PC(c_{n+1}|s_n)$ denotes the probability for a percept c_{n+1} if the environment is in the state s_n , \prod_{h_n} denotes the actual policy of an agent with observed history h_n , $P(s_{n+1}|a_{n+1}, s_n)$ the probability that the state of the environment changes from s_n to s_{n+1} due to action a_{n+1} , and $PR(r|s_{n+1}, s_n)$ the probability that a reward r is issued due to the state change from s_n to s_{n+1} . The recursion for an environment with initial state s_0 starts with $p_0(h_0) = 1$ and $PC(c_1|s_0)$.

In order to solve a given task, agents with different histories usually need a different number of interactions n . Therefore, it is necessary to extend the probability distribution $p_n(h_n)$ over the set H_n of



histories with length n to the probability distribution $p(h_n)$ over the set of infinite histories H^∞ [36]. Here, $p(h_n)$ now determines the probability that an infinite long history h_∞ starts with h_n via

$$p(h_n) := p(\{h_\infty | h_\infty \text{ starts with } h_n\}) = p_n(h_n). \quad (\text{A.2})$$

Appendix B. Mapping

In general, learning agents can use mapping or model building [3] in order to determine their next actions. A map of the environment can e.g. be generated by not only generating and storing a policy $\Pi(a|c)$ for every observed percept c , but also keeping track of observed transitions $(c_n, a_n) \rightarrow c_{n+1}$. Such a map can also include actions and percepts not taken/observed so far.

Figure B1 shows an example of a map for a deterministic strictly epochal environment where percepts c are equal to the state s of the environment and each epoch consists of 3 consecutive actions. In each step, the agent can choose between two different actions a and a' . Here, the agent already observed that starting an epoch with the sequence of actions $(a_1 = a, a_2 = a')$ leads to the same percept as starting with $(a_1 = a', a_2 = a)$. However, so far it has not observed e.g. the percept resulting from starting the epoch with the action sequence $(a_1 = a, a_2 = a)$. Therefore, the agent assumes that this sequence of actions will lead to a new, so far unobserved percept.

An agent can use such a map to plan its action for the next epoch by defining the policy

$$\hat{\Pi}(\vec{a}) = \prod_{j=1}^L \Pi(a_j | c(a_1, \dots, a_{j-1})) \quad (\text{B.1})$$

for complete action sequences of an epoch. Here, it uses for unknown percepts c the equal distribution

$$\Pi(a_j | c(a_1, \dots, a_{j-1})) = \frac{1}{\|A\|} \quad \text{if } c \text{ unknown}, \quad (\text{B.2})$$

where $\|A\|$ determines the number of possible actions.

The above described method is only one possible way how an agent can plan its actions for the next epoch. The important point is that it is possible to define $\hat{\Pi}(\vec{a})$ at the beginning of each epoch.

Appendix C. Turnover from quantum to classical search

A classical agent can observe a reward with probability Q , (3) in every epoch. As a result, the average reward obtained by classical agents following the policy $\hat{\Pi}$ is given by

$$\langle r \rangle_{\text{cl}} = \sum_{\{\vec{a}\}} r(\vec{a}) \hat{\Pi}(\vec{a}) = \bar{r}_{\text{cl}} Q. \quad (\text{C.1})$$

Here, we introduced the average reward of a winning sequence

$$\bar{r}_{\text{cl}} = \sum_{\{\vec{a}|r(\vec{a})>0\}} r(\vec{a}) \hat{\Pi}(\vec{a}) / Q \quad (\text{C.2})$$

for classical agents, which is equal to the average reward conditioned on observing a reward $r > 0$.

A quantum-enhanced agent uses $\alpha_o k$ epochs to perform amplitude amplification to determine a possible rewarded sequence of actions. The agent will receive in an additional consecutive epoch a reward with probability [30]

$$G(Q, k) = \sin^2((2k + 1)\Theta(Q)) \quad (\text{C.3})$$

$$\Theta(Q) = \arcsin \sqrt{Q}. \quad (\text{C.4})$$

Thus, quantum-enhanced agents receive on average a reward of

$$\langle r \rangle_{\text{q}} = \bar{r}_{\text{q}} \frac{G(Q, k)}{\alpha_o k + 1}, \quad (\text{C.5})$$

where \bar{r}_{q} is the average reward of a winning sequence of a quantum-enhanced agent. Due to theorem 1, we find $\bar{r}_{\text{q}} = \bar{r}_{\text{cl}} = \bar{r}$. As a consequence, a learning agent with a winning probability Q will receive on average more rewards when performing quantum exploration for k epochs if

$$Q < \frac{\sin^2[(2k + 1)\arcsin(\sqrt{Q})]}{\alpha_o k + 1}. \quad (\text{C.6})$$

The inequality can only be solved numerically. For $\alpha_o = 1$ and $k = 1$ we find that a quantum-enhanced agent finds on average more rewards if $Q < Q_{\text{max}} \approx 0.3964$.

Appendix D. Theorem 1

In the following, we give a detailed proof of theorem 1. Please keep in mind that we consider in this paper only agents and environment, where all changes in the general behaviors, such as e.g. policy updates, are completely determined by their rewarded history. Therefore, we introduce $\hat{\Pi}_j$ as the policy of an agent in the j th interval starting after the j th rewarded epoch and ending with the $j + 1$ th rewarded epoch. As a result, the probability to play the sequence of actions \vec{a} in an epoch in the interval j is given by $\hat{\Pi}_j(\vec{a})$.

The probability to get no reward in an epoch in the j th interval is given by

$$\sum_{\{\vec{a}|r(\vec{a})=0\}} \hat{\Pi}_j(\vec{a}) = 1 - Q_j \quad (\text{D.1})$$

with Q_j , (3), being the reward probability in this interval. The probability $p_j(t_j, \vec{a}_j)$ that the j th interval contains t_j epochs and ends with the action sequence \vec{a}_j with $r(\vec{a}_j) > 0$ is given by the probability to play $t_j - 1$ epochs without getting any reward and play then the action sequence \vec{a}_j leading to

$$p_j(t_j, \vec{a}_j) = (1 - Q_j)^{t_j - 1} \hat{\Pi}_j(\vec{a}_j). \quad (\text{D.2})$$

As a consequence, the probability $p_j(\vec{a}_j)$ that playing the action sequence \vec{a}_j will lead to the j th observed reward is given by

$$p_j(\vec{a}_j) = \sum_{t_j=1}^{\infty} p_j(t_j, \vec{a}_j) \quad (\text{D.3})$$

$$= \frac{1}{Q_j} \hat{\Pi}_j(\vec{a}_j), \quad (\text{D.4})$$

where we used the geometric series in the second step. The probability to observe a given rewarded history h_r can be expressed with the help of the conditional probability $p(h_r|\vec{a}_0)$, denoting the probability to observe h_r if \vec{a}_0 was observed as the first rewarded action sequence via

$$p(h_r) = p(h_r|\vec{a}_0)p_0(\vec{a}_0). \quad (\text{D.5})$$

The same considerations hold for all other time intervals, leading together with (D.4) to

$$p(h_r) = \prod_{j=0}^{J-1} \frac{\hat{\Pi}_j(\vec{a}_j)}{Q_j}, \quad (\text{D.6})$$

where we assumed that the rewarded history h_r contains J rewarded epochs.

Amplitude amplification [29, 30] enhances Q_j but preserves the ratio $\hat{\Pi}_j(\vec{a}_j)/Q_j$. The classical and the hybrid agent start with the same policy $\hat{\Pi}_0$. As a consequence, their probability to observe \vec{a}_0 as the first rewarded action sequence is equal leading to identical probabilities $p(h_r)$ for rewarded histories of length $J = 1$. If a quantum and a classical agent played the same \vec{a}_0 as first rewarded action sequences, they follow the same policy $\hat{\Pi}_1$ in the next interval leading together with (D.6) to identical distributions for rewarded histories.

Appendix E. Average learning times

In the following, we summarize more detailed discussions concerning the proof of theorem 2.

The expected learning time $\langle T(h_r) \rangle$ for an agent with a given rewarded history can be expressed by

$$\langle T(h_r) \rangle = \sum_{j=0}^{J(h_r)-1} \langle t_j(h_r) \rangle, \quad (\text{E.1})$$

because the behavior of the here considered agents depends solely on the rewards an agent has found so far.

The probability $p_j(t_j, \vec{a}_j)$ that the duration of interval j is given by t_j epochs and that interval j ends with the rewarded action sequence \vec{a}_j is given by (D.2). The probability that the duration of interval j is given by t_j conditioned on that the agent's rewarded history is given by h_r is therefore determined by

$$p_j(t_j|h_r) = \frac{p_j(t_j, \vec{a}_j)}{p_j(\vec{a}_j)} = Q_j(1 - Q_j)^{t_j-1}. \quad (\text{E.2})$$

As a consequence, the average interval time $\langle t_j(h_r) \rangle$ of an agent with rewarded history h_r is given by

$$\langle t_j(h_r) \rangle = \sum_{\tau=1}^{\infty} \tau p_j(\tau|h_r) = \frac{1}{Q_j} \quad (\text{E.3})$$

as used in (9).

In the main text, we then express the average learning time $\langle T \rangle$ as an average of the learning times $\langle T(h_r) \rangle$ for different given rewarded histories in (14). This equation follows from the following considerations.

Let $p(T, h_r)$ be the probability that an agent has learned after T epochs and has observed the rewarded history h_r and let H_r be the set of all possible rewarded histories. Then, the average learning time is determined by

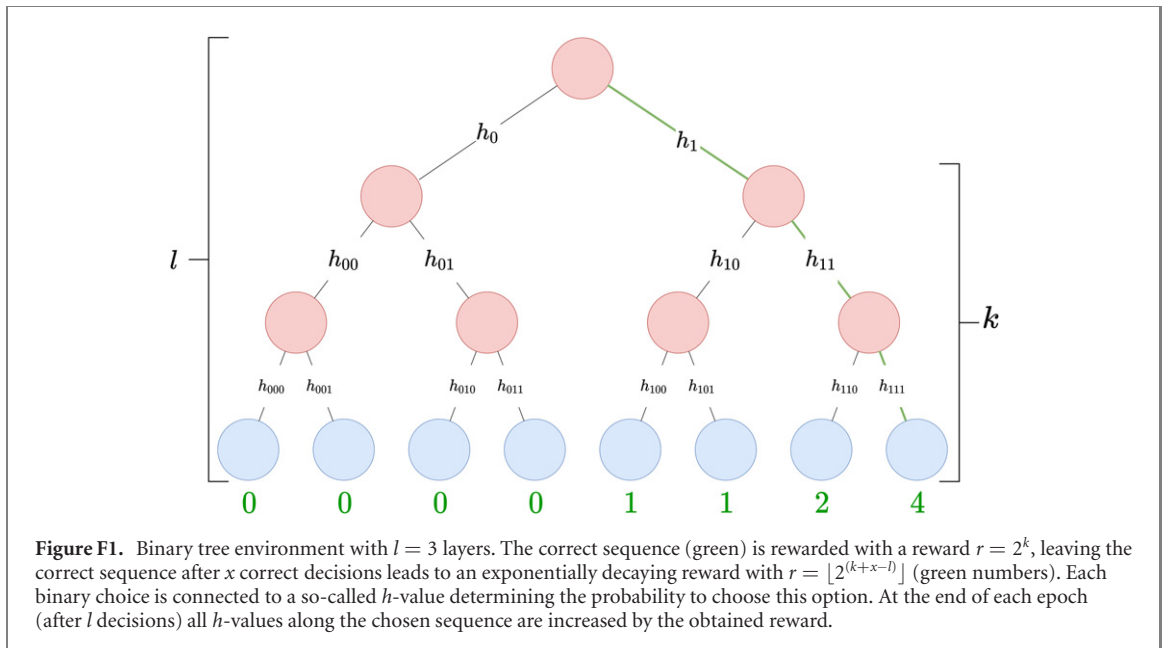
$$\langle T \rangle = \sum_{T=1}^{\infty} \sum_{h_r \in H_r} T p(T, h_r). \quad (\text{E.4})$$

The average learning time $\langle T(h_r) \rangle$ of an agent conditioned on observing h_r is given by

$$\langle T(h_r) \rangle = \sum_{T=1}^{\infty} T p(T|h_r) \quad (\text{E.5})$$

with $p(T|h_r) = p(T, h_r)/p(h_r)$. As a consequence, we find

$$\langle T \rangle = \sum_{T=1}^{\infty} \sum_{h_r \in H_r} T p(T|h_r) p(h_r) = \sum_{h_r \in H_r} \langle T(h_r) \rangle p(h_r). \quad (\text{E.6})$$



Appendix F. Binary tree environment

In this section, we describe the binary tree environment used as an example environment for figures 2 and 3. Additionally, we introduce a learning agent based on t-PS [34] simplified and reduced to the essentials for this binary tree environment. For a more detailed introduction of projective simulation or t-PS we recommend [28, 34]. The source for this simulation is available at [37].

In the binary tree environment, see figure F1, an agent has to perform l sequential binary decisions in an epoch. Then, a reward is issued and a new epoch starts. One sequence of action is marked as the correct sequence and rewarded with a reward $r = 2^k$. Leaving the correct sequence after x correct decisions leads to an exponentially decaying reward of $r = \lfloor 2^{(k+x-l)} \rfloor$. The number of rewarded action sequences ($r > 0$) relative to the complete number of possible action sequences reduces exponentially with $l - k$. On the other hand, the reward difference within the rewarded subspace grows exponentially with k .

This environment inhabits two important features making it an ideal example for comparing the here described hybrid agent with its classical counterpart: (i) it is hard to find a rewarded action sequence leading to a big possible speedup for the hybrid agent. (ii) Classical optimization within the rewarded subspace leads to higher rewards, allowing a comparison of the quality of the found solution between the hybrid agent and its classical counterpart.

To complete the example, we will need an agent interacting with this environment. As this paper does not focus on how to construct a good classical agent, we will use just a simple agent suited for this environment. We would like to emphasize that more advanced agents could be used, too. Our simple agent assigns an h value to each possible choice. Each of the l binary decisions is decided by a random decision governed by a softmax policy. The y th decision of an epoch is thus governed by the probabilities

$$p_{i_y} = 0.5 + 0.5 \tanh [\beta(h_{i_1, \dots, i_y} - h_{i_1, \dots, \neg i_y})], \quad (\text{E.1})$$

with $i_z \in \{0, 1\} \forall 1 \leq z \leq y$ and $\neg i$ denoting the alternative choice to i . The h values are initialized by the same value ($h = 1$). However, the exact initial value is irrelevant because the policy only depends on the difference between the h -values. If a rewarded was observed for the sequence $\vec{i} = (i_1, \dots, i_l)$, all h values of the sequence are increased by the obtained reward r via

$$h_{i_1, \dots, i_y} \rightarrow h_{i_1, \dots, i_y} + r \quad \forall 1 \leq y \leq l. \quad (\text{E.2})$$

ORCID iDs

Arne Hamann  <https://orcid.org/0000-0002-9016-3641>

Sabine Wölk  <https://orcid.org/0000-0001-9137-4814>

References

- [1] Dunjko V and Briegel H 2018 Machine learning & artificial intelligence in the quantum domain: a review of recent progress *Rep. Prog. Phys.* **81** 3
- [2] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195–202
- [3] Sutton R S and Barto A G 1998 *Reinforcement Learning* (Cambridge, MA: MIT Press)
- [4] Ciliberto C, Herbster M, Ialongo A D, Pontil M, Rocchetto A, Severini S and Wossnig L 2018 Quantum machine learning: a classical perspective *Proc. R. Soc. A* **474** 20170551
- [5] Havlicek V, Córcoles A D, Temme K, Harrow A W, Kandala A, Chow J M and Gambetta J M 2019 Supervised learning with quantum-enhanced feature spaces *Nature* **567** 209–12
- [6] Schuld M, Sinayskiy I and Petruccione F 2015 An introduction to quantum machine learning *Contemp. Phys.* **56** 172–85
- [7] Adcock J *et al* 2015 Advances in quantum machine learning (arXiv:1512.02900)
- [8] El-Mahalawy A M and El-Safty K H 2021 Classical and quantum regression analysis for the optoelectronic performance of NTCDA/p-Si UV photodiode *Optik* **246** 167793
- [9] Johannink T, Bahl S, Nair A, Luo J, Kumar A, Loskyll M, Ojea J A, Solowjow E and Levine S 2019 Residual reinforcement learning for robot control 2019 *Int. Conf. Robotics and Automation (ICRA)* (Montreal, QC, Canada) (Piscataway, NJ: IEEE) pp 6023–9
- [10] Tjandra A, Sakti S and Nakamura S 2018 Sequence-to-sequence ASR optimization via reinforcement learning 2018 *IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)* (Calgary, AB, Canada) (Piscataway, NJ: IEEE) pp 5829–33
- [11] Komorowski M, Celi L A, Badawi O, Gordon A C and Faisal A A 2018 The artificial intelligence clinician learns optimal treatment strategies for sepsis in intensive care *Nat. Med.* **24** 1716–20
- [12] Silver D *et al* 2016 Mastering the game of go with deep neural networks and tree search *Nature* **529** 484–9
- [13] Jerbi S, Trenkwalder L M, Nautrup H P, Briegel H J and Dunjko V 2021 Quantum enhancements for deep reinforcement learning in large spaces *PRX Quantum* **2** 010328
- [14] Paparo G D, Dunjko V, Makmal A, Martin-Delgado M A and Briegel H J 2014 Quantum speedup for active learning agents *Phys. Rev. X* **4** 031002
- [15] Sriarunothai T, Wölk S, Giri G S, Friis N, Dunjko V, Briegel H J and Wunderlich C 2018 Speeding-up the decision making of a learning agent using an ion trap quantum processor *Quantum Sci. Technol.* **4** 015014
- [16] Jerbi S, Gyurik C, Marshall S, Briegel H J and Dunjko V 2021 Variational quantum policies for reinforcement learning (arXiv:2103.05577)
- [17] Nagy D, Tabi Z, Haga P, Kallus Z and Zimboras Z 2021 Photonic quantum policy learning in OpenAI Gym 2021 *IEEE Int. Conf. Quantum Computing and Engineering (QCE)* (Broomfield, CO, USA) (Piscataway, NJ: IEEE) pp 123–9
- [18] Ronagh P 2019 Quantum algorithms for solving dynamic programming problems (arXiv:1906.02229)
- [19] Crawford D, Levit A, Ghadermarzy N, Oberoi J S and Ronagh P 2019 Reinforcement learning using quantum Boltzmann machines (arXiv:1612.05695)
- [20] Cornelissen A 2018 Quantum gradient estimation and its application to quantum reinforcement learning *Master Thesis* Delft University of Technology
- [21] Neukart F, Von Dollen D, Seidel C and Compostella G 2018 Quantum-enhanced reinforcement learning for finite-episode games with discrete state spaces *Front. Phys.* **5** 71
- [22] Casalé B, Di Molfetta G, Kadri H and Ralaivola L 2020 Quantum bandits *Quantum Mach. Intell.* **2** 11
- [23] Saggio V *et al* 2021 Experimental quantum speed-up in reinforcement learning agents *Nature* **591** 229–33
- [24] Dunjko V, Taylor J M and Briegel H J 2016 Quantum-enhanced machine learning *Phys. Rev. Lett.* **117** 130501
- [25] Hamann A, Dunjko V and Wölk S 2021 Quantum-accessible reinforcement learning beyond strictly epochal environments *Quantum Mach. Intell.* **3** 22
- [26] Dunjko V, Liu Y-K, Wu X and Taylor J M 2017 Exponential improvements for quantum-accessible reinforcement learning (arXiv:1710.11160)
- [27] Mnih V *et al* 2015 Human-level control through deep reinforcement learning *Nature* **518** 529–33
- [28] Briegel H J and De las Cuevas G 2012 Projective simulation for artificial intelligence *Sci. Rep.* **2** 400
- [29] Grover L K 1998 Quantum computers can search rapidly by using almost any transformation *Phys. Rev. Lett.* **80** 4329
- [30] Boyer M, Brassard G, Høyer P and Tapp A 1998 Tight bounds on quantum searching *Fortschr. Phys.* **46** 493–505
- [31] Yoder T J, Low G H and Chuang I L 2014 Fixed-point quantum search with an optimal number of queries *Phys. Rev. Lett.* **113** 210501
- [32] Saleh S Q and Younes A 2019 Different fixed-phases for quantum search operators *J. Phys. Soc. Japan* **88** 124002
- [33] Roy T, Jiang L and Schuster D I 2021 Deterministic Grover search with a restricted oracle (arXiv:2201.00091 [quant-ph])
- [34] Flamini F, Hamann A, Jerbi S, Trenkwalder L M, Nautrup H P and Briegel H J 2020 Photonic architecture for reinforcement learning *New J. Phys.* **22** 045002
- [35] Melnikov A A, Makmal A and Briegel H J 2018 Benchmarking projective simulation in navigation problems *IEEE Access* **6** 64639–48
- [36] Maschler M, Solan E and Zamir S 2020 *Game Theory* (Cambridge: Cambridge University Press)
- [37] Hamann A and Wölk S 2022 Performance analysis of a hybrid agent for quantum-accessible reinforcement learning <https://doi.org/10.5281/zenodo.5879295>