Quantum Machine Intelligence (2025) 7:52
https://doi.org/10.1007/s42484-025-00269-1

RESEARCH ARTICLE

A hybrid learning agent for episodic learning tasks with unknown target distance

Oliver Sefrin[1] · **Sabine Wölk**[1,2]

Received: 17 December 2024 / Accepted: 8 March 2025

Abstract
The "hybrid agent for quantum-accessible reinforcement learning," as defined in (Hamann and Wölk New J Phys 24:033044 2022), provides a proven quasi-quadratic speedup and is experimentally tested. However, the standard version can only be applied to episodic learning tasks with fixed episode length. In many real-world applications, the information about the necessary number of steps within an episode to reach a defined target is not available in advance and especially before reaching the target for the first time. Furthermore, in such scenarios, classical agents have the advantage of observing at which step they reach the target. How to best deal with an unknown target distance in classical and quantum reinforcement learning and whether the hybrid agent can provide an advantage in such learning scenarios is unknown so far. In this work, we introduce a hybrid agent with a stochastic episode length selection strategy to alleviate the need for knowledge about the necessary episode length. Through simulations, we test the adapted hybrid agent's performance versus classical counterparts with and without similar episode selection strategies. Our simulations demonstrate a speedup in certain scenarios due to our developed episode length selection strategy for classical learning agents as well as an additional speedup for our resulting hybrid learning agent.

Keywords Quantum reinforcement learning · Amplitude amplification · Hybrid algorithm · Navigation problem

1 Introduction

Reinforcement learning (RL), the machine learning framework related to learning through interaction and experience, has shown tremendous success in recent years, surpassing human capability in Atari games (Mnih et al. 2015) or the board game Go (Silver et al. 2017), among many others. One of the main reasons for its success is the ever-growing computational power of classical hardware, which allows the implementation of deep learning techniques in RL such as deep Q-learning (DQN) (Mnih et al. 2015).

However, many problems and problem classes still prove to be difficult even to modern supercomputers due to their unfavorable scaling with the problem size. Here, quantum computation (Nielsen and Chuang 2010) with the prospect of polynomial or even exponential advantages in problem complexity has excited researchers across many disciplines and even created new research fields. Among the latter, quantum machine learning (QML) (Biamonte et al. 2017) has emerged with the idea of combining the computational benefits of quantum computation with the proven effectiveness of machine learning approaches. Given the current state of quantum computing hardware in the so-called noisy intermediate-scale quantum (NISQ) era (Preskill 2018), research in QML has focused on algorithms of low circuit depth and width, such as variational algorithms (Cerezo et al. 2021), or even so-called quantum-inspired methods such as tensor networks (Biamonte and Bergholm 2017; Orús 2014; Bridgeman and Chubb 2017; Huggins et al. 2019). Whereas these methods have the benefit of being applicable on current quantum devices (or even on classical machines in the case of quantum-inspired ansätze), their actual advantage compared to classical methods remains unclear (Schuld and Killoran 2022; Cerezo et al. 2024).

An example for a QML algorithm with a provable speedup, which is, however, not NISQ-ready, is the so-called

✉ Oliver Sefrin
oliver.sefrin@dlr.de

[1] Institute of Quantum Technologies, German Aerospace Center (DLR), Wilhelm-Runge-Straße 10, Ulm 89081, Germany

[2] Institute for Complex Quantum Systems, Ulm University, Albert-Einstein-Allee 11, Ulm 89081, Germany

Published online: 11 April 2025

"hybrid agent for quantum-accessible reinforcement learning" (Hamann and Wölk 2022). Here, a learning agent learns to solve a given problem by interacting via a set of actions with a problem environment. Based on amplitude amplification (Grover 1997; Brassard et al. 2002), a quasi-quadratic speedup in terms of the sample complexity compared to corresponding classical agents was proven for a class of RL environments called *deterministic and strictly episodic (DSE)* environments. These environments are reset to an initial state after a fixed number of interaction steps defining the *episode length*. Further, choosing an action in the current state yields one subsequent state with certainty. The *Gridworld* or maze scenario with a fixed episode length is one example of a DSE environment. The Gridworld with and without fixed episode length will serve as a toy model in this work.

While many learning scenarios are indeed episodic, a large subset of these is not strictly episodic. That is, episodes may differ in length because, e.g., their end is coupled to reaching some rewarded target state such that the episode length depends on the brevity of the solution found. The Gridworld environment without a fixed episode length is a standard example for such a learning scenario and a learning model with many real world applications (see, e.g., Hämmerle et al. (2024)). Classical reinforcement learning agents usually solve the problem of an unknown target distance by extending an episode until the target is found. Practically, an absolute maximal length for an episode based, e.g., on prior knowledge about the problem size or the available computational time is set to avoid a never-ending algorithm. How to best deal with unknown target distances is still under current investigation (see, e.g., Mandal et al. (2023)).

For the hybrid agent (Hamann and Wölk 2022), a practical application strategy in learning scenarios with an unknown target distance has not been formulated yet. In general, the hybrid agent can easily be applied to a deterministic and non-strictly episodic environment by choosing a fixed episode length L after which the episode ends, whether the target was reached or not. Yet, the chosen episode length L has crucial influence on the hardness of the learning problem and the expected total number of interaction steps to learn to solve the problem, both in the classical and the hybrid case. Figure 1 illustrates this influence emphasizing that a good episode length selection strategy can accelerate the learning both for classical and quantum learning agents. Here, we show the expected number of interaction steps, summed over all episodes, until the defined goal was reached for the first time for a classical and a hybrid agent depending on the fixed episode length L for a given maze example. On the one hand, choosing L small renders the problem hard or even unsolvable for an untrained agent. On the other hand, choosing a large L can lead to a high probability $p_{\text{init}}(L)$ to achieve the defined target within one episode even for an untrained agent. This reduces the number of necessary episodes but instead requires a larger number of interaction steps per episode. For small episode lengths, corresponding to small success probabilities $p_{\text{init}}(L)$, the hybrid agent offers a quasi-quadratic speedup compared to the classical agent. For intermediate episode lengths, the hybrid agent's quantum overhead results in a slightly worse performance, before it finally converges to the classical agent's behavior and thus to its performance in the limit of large episode lengths and $p_{\text{init}}(L) \to 1$. (For a detailed discussion, see Appendix C.) The choice of a suitable episode length is further complicated by the fact that upon not being rewarded, one does not gather any information about whether the chosen episode length is sufficient or not.

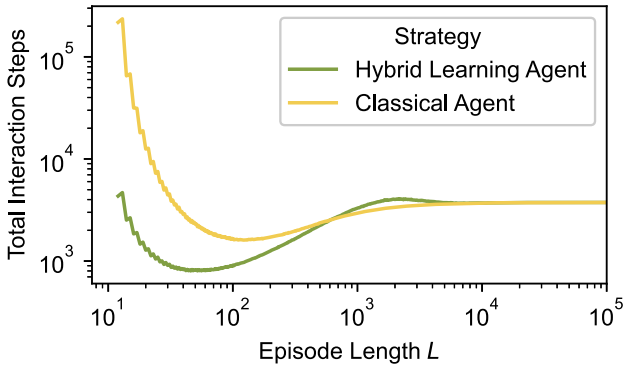


Fig. 1 Comparison of the expected number of total interaction steps performed before finding a defined target for the first time depending on the episode length L. The results stem from a simulation with a Gridworld of base size 7×7 and an outer wall distance of 16 (see Section 3.2 for information about the Gridworld layouts)

In summary, the choice of L can have a larger influence on the necessary effort to reach the goal than whether we use a hybrid or a purely classical agent. If enough information about the learning problem were available, an optimal episode length could be inferred. However, this is usually not the case. Furthermore, a classical learning agent can observe when it reaches the target and then end the episode. In quantum mechanics, observation usually suppresses the effect of interference, which is necessary to gain a quantum advantage. Thus, a predefined episode length is necessary for the hybrid agent. Whether the hybrid agent can outperform classical agents which use the advantage of a flexible episode length and if so, in which scenarios, has not been investigated so far.

In this work, we introduce a hybrid agent with a flexible episode length selection strategy for the case of a deterministic and episodic environment with an unknown target distance. More precisely, we introduce a probabilistic condition which doubles the current episode length when triggered until the target is found for the first time. Reaching the target for the first time is especially challenging while extremely important. Indeed, after having reached the target, the length of the found sequence of actions is obviously sufficient and serves as an upper bound for the optimal episode length. The necessity of reaching the target for the first time efficiently is further underlined by the fact that, at the start of learning, the RL algorithm's success probabilities are usually the lowest. This indicates that the search for reaching the target for the first time in such a sparse reward environment takes up a significant amount of the total learning time.

By interweaving our probabilistic episode length adaptation with the underlying amplitude amplification algorithm, we manage to solve learning problems without fixed episode length with no additional hyperparameters introduced. We test the extended hybrid agent versus a corresponding classical agent with the same episode length selection strategy and a classical agent with flexible episode length. This latter agent has no fixed episode length and ends an episode only when finding the target. Our research objective is to find a good episode length selection strategy by comparing the performance of different hybrid and classical agents and to investigate whether the resulting hybrid learning agent provides any benefits compared to the considered classical agents.

The article is structured as follows: in Section 2, the hybrid agent is introduced in more detail as well as placed in the broader context of quantum reinforcement learning (QRL). Next, we present the adapted algorithm and explain the simulation methodology in Section 3. In Section 4, we define a new figure of merit tailored to the new problem setting and motivate it before presenting the results. Finally, we provide a conclusion and an outlook in Section 5.

2 Background

2.1 Reinforcement learning

In RL (Sutton and Barto 2018), an agent interacts with an environment in a sequence of discrete time steps $t \in \mathbb{N}_0$. The interaction starts with an initial percept, or state, $s_0 \in \mathcal{S}$ from the set of possible percepts $\mathcal{S}$, which is given to the agent by the environment. At each time step $t \geq 1$, the agent selects an action a_t from the set of possible actions $\mathcal{A}$ based on the previous percept s_{t-1}. This action selection is governed by the agent's current *policy* $\pi(a|s)$, which is a probability distribution over the set of actions conditional to the current percept. Subsequently, the environment responds with the next percept s_t as well as a real-valued reward r_t. Generally, the response of the environment is probabilistic, with probability distribution

$$\tau(s_t, r_t|s_{t-1}, a_t). \tag{1}$$

Since the probability function only depends on the previous interaction step, it fulfills the Markov property. Hence, this type of RL interaction is a so-called finite *Markov decision process* (MDP).

In *deterministic* environments such as Gridworld, Eq. 1 is a trivial probability distribution in the sense that it returns unity for one specific combination of percept and reward value (s_t, r_t) and zero for any other combination. To simplify the notation in Algorithm 1, we can thus introduce functions $S : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ and $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ which return the next, deterministic percept s_t and reward r_t, respectively:

$$r_t = R(s_{t-1}, a_t) \tag{2}$$

$$s_t = S(s_{t-1}, a_t) \tag{3}$$

The agent's task is to adapt its policy such as to maximize the expectation value of the *cumulative reward*

$$\mathbb{E}_\pi \left[\sum_{t=1}^{\infty} \gamma^t r_t \right], \tag{4}$$

where the subscript π indicates the expectation value upon following policy π. The coefficient $\gamma \in (0, 1]$ is the so-called discount value, which adjusts a trade-off between current and future rewards.

2.2 Hybrid learning agent

Our approach builds on the QRL algorithm coined "hybrid agent for quantum-accessible reinforcement learning" (Hamann and Wölk 2022). This algorithm is embedded in a quantum communication scenario where the RL agent and the DSE environment interact by exchanging quantum states. That is, each action from the set of allowed actions $a \in \mathcal{A}$ is mapped to a quantum state $|a\rangle_A$ and the set of states $\{|a\rangle \mid a \in \mathcal{A}\}$ forms an orthonormal basis. Similarly, the set of percepts $\{s \mid s \in \mathcal{S}\}$ is mapped to orthonormal states $\{|s\rangle_S\}$, and the set of rewards $\{r \mid r \in \mathcal{R}\}$ to orthonormal states $\{|r\rangle_R\}$. Here, the indices A, S, and R indicate the Hilbert spaces for the quantized actions $\mathcal{H}_A$, percepts $\mathcal{H}_S$,

and rewards $\mathcal{H}_R$, respectively. We map the initial percept s_0 to

$$|\vec{s}_{\text{init}}\rangle_S = |s_0, \varnothing, \ldots, \varnothing\rangle_S \in \mathcal{H}_S^{\otimes(L+1)},$$

with a single-percept default state $|\varnothing\rangle_S \in \mathcal{H}_S$, as the initial quantum state for the sequence of percepts and use $|\vec{0}\rangle_R \in \mathcal{H}_R^{\otimes L}$ as the initial quantum state for the sequence of rewards. Then, in the quantum communication scenario, the response of the environment to the sequence of actions $\vec{a} = (a_1, \ldots, a_L)$ within one episode of L interaction steps can be modeled as a unitary U_{env} acting on the multipartite state $|\psi\rangle = |\vec{a}\rangle_A|\vec{s}_{\text{init}}\rangle_S|\vec{0}\rangle_R$ such that

$$U_{\text{env}}|\psi\rangle = |\vec{a}\rangle_A|\vec{s}\rangle_S|\vec{r}\rangle_R.$$

In our RL scenario, we assume a single binary reward r is given at the end of each episode to simplify the learning scenario, such that the reward Hilbert space is two-dimensional.

With α_k[1] queries of the environment unitary U_{env}, it was shown that an effective phase-kickback oracle

$$O_E|\vec{a}\rangle_A|\vec{s}_{\text{init}}\rangle_S|-\rangle_R = (-1)^{r(\vec{a})}|\vec{a}\rangle_A|\vec{s}_{\text{init}}\rangle_S|-\rangle_R \tag{5}$$

can be realized (Dunjko et al. 2016) when initializing the reward register in the state $|-\rangle_R = \frac{1}{\sqrt{2}}(|0\rangle_R - |1\rangle_R)$. Based on this oracle, a hybrid quantum classical learning agent has been defined in Saggio et al. 2021 and Hamann and Wölk 2022.

This hybrid learning agent consists of two parts: a quantum part and a classical part. In the quantum part of the hybrid learning agent (Hamann and Wölk 2022), the agent prepares instead of a single sequence of actions $|\vec{a}\rangle$ a superposition of possible action sequences $\sum_{\vec{a}} c_{\vec{a}}|\vec{a}\rangle$ where $|c_{\vec{a}}|^2$ is equal to the probability that the agent chooses the action sequence $\vec{a}$ according to its current policy π. By interacting with the environment, the agent applies a Grover operator on $\sum_{\vec{a}} c_{\vec{a}}|\vec{a}\rangle$ based on the oracle O_E. With this Grover operator, the amplitudes $\{c_{\vec{a}} \mid r(\vec{a}) > 0\}$ of rewarded action sequences can be amplified which enables a quadratic speedup in query complexity (Grover 1997; Brassard et al. 2002). Following the quantum part of the algorithm, the classical part of the agent starts by measuring the action register. Consecutively, one classical episode with the measured action sequence $\vec{a}$ is performed to determine the corresponding sequence of percepts $\vec{s}\,(\vec{a})$ and the corresponding reward $r\,(\vec{a})$ for the measured sequence of actions. This is necessary to infer the actual sequence of percepts encountered by the agent as well as the reward information, since this information was uncomputed in the amplitude amplification procedure to allow for interference. Finally, the policy π of the agent can be updated according to some chosen update rules such as Q-learning (Watkins and Dayan 1992) or projective simulation (Briegel and De las Cuevas 2012). Then, the agent proceeds by starting the next quantum part until a predefined ending condition (Hamann and Wölk 2022) is met.

The speedup of the hybrid learning agent has been verified in a proof-of-principle experiment using a nanophotonic processor (Saggio et al. 2021). Likewise, a speedup in decision-making using a quantum walk search approach has been formulated (Paparo et al. 2014) and experimentally verified (Sriarunothai et al. 2018) for a variant of the projective simulation algorithm called "Reflecting Projective Simulation" (Briegel and De las Cuevas 2012). A first extension of the standard RL scenario for the hybrid learning agent concerning changing oracles was investigated in Hamann et al. (2021).

2.3 Related work

QRL (Dong et al. 2008), just as quantum machine learning, can be interpreted in many different ways, according to which we aim to structure this overview of related work.

In the broader sense, QRL can be understood as the application of classical RL (Sutton and Barto 2018) to problems related to quantum computing or quantum technologies. This comprises the usage of RL for quantum circuit optimization (Ostaszewski et al. 2021; Lockwood 2022; Fösel et al. 2021; Ruiz et al. 2024; Rapp et al. 2025), quantum control (Sivak et al. 2022; Fösel et al. 2018; Bukov et al. 2018; Guatto et al. 2024; Yu et al. 2025), or quantum error correction (Nautrup et al. 2019; Sivak et al. 2023).

Another well-established category is QRL with parametrized quantum circuits (PQCs) (Jerbi et al. 2021). Here, PQCs encode the RL agent's current policy (in policy gradient algorithms such as PPO (Schulman et al. 2017) or DPG/DDPG (Silver et al. 2014; Lillicrap et al. 2019)) or encode an approximate action-value function such as in DQN (Mnih et al. 2015). In a hybrid approach, the PQC's parameters are updated using a classical feedback loop. Using simulated or real quantum hardware, these methods have been applied to standard Gymnasium (Kwiatkowski et al. 2024) (previously OpenAI Gym (Brockman et al. 2016)) environments such as Cart Pole, Frozen Lake, or Atari

[1] $\alpha_k = 1$ for basic environments with action-independent percepts s and $\alpha_k = 2$ for environments in which intermediate percepts s are action-dependent (Hamann and Wölk 2022).

Games (Jerbi et al. 2021; Skolik et al. 2022; Chen et al. 2020; Lockwood and Si 2020, 2021) as well as maze problems (Hohenfeld et al. 2024; Chen et al. 2024).

More recent approaches aim at speeding up the learning process using quantum sub-routines (Ganguly et al. 2023; Zhong et al. 2024), finding better policies with a combined approach of using quantum phase estimation (Kitaev 1995) and Grover's search algorithm (Grover 1997; Wiedemann et al. 2023), or combining quantum computing with the policy iteration algorithm (Cherrat et al. 2023). In Li et al. (2020), amplitude amplification is used to increase the chance of rewarded actions. Here, the kickback phases of the oracle and diffusion operator are not fixed to π, but instead depend on a so-called utility function.

First investigations of possible implementations of QRL on superconducting devices are presented in Lamata (2017). A different approach to the maze or Gridworld problem is shown in Dalla Pozza et al. (2022). Here, a classical RL agent is trained to modify the maze's walls such as to maximize the escape probability with a quantum random walk in a given time interval.

3 Methods

3.1 Algorithm

Our hybrid learning agent uses, similar to Hamann and Wölk (2022), the variation of Grover's algorithm where the number of solutions and thus the optimal number k_{opt} of amplitude amplification (AA) iterations is unknown (Boyer et al. 1998). This variation of the Grover algorithm appears also slightly varied as `QSearch` in Brassard et al. (2002) and is necessary due to the fact that in a RL problem, the *success probability* of being rewarded is typically unknown.

The main ingredient of what we call from here on *Boyer's algorithm* is a flexible interval $[0, m)$ with $m \in \mathbb{R}^+$, from which the integer number of AA iterations k is uniformly sampled. Starting from $m = 1$, the interval upper bound is multiplied by a constant factor $\lambda \in (1, 2)$ each time that the measurement yields no success. This parameter λ, which is set to $\frac{5}{4}$ in all simulations in this work, is the only hyperparameter of our algorithm. Once the parameter m reaches $\sqrt{1/p_{\text{min}}}$, with p_{min} being a lower bound for the current success probability, it is not increased further. At this point, Boyer's algorithm reaches its so-called *critical stage*, at which the success probability in each AA round is known to be at least $1/4$, supposed that a rewarded item exists (Boyer et al. 1998).

Require: initial percept s_0, set of actions $\mathcal{A}$, policy π, percept response function S, reward function R, Grover operator G

$L \leftarrow 1$
$m \leftarrow 1$
$\lambda \leftarrow \frac{5}{4}$
$N_{\text{act}} = 0$ ▷ *step counter*
while `true` **do**
  ▷ *probabilistic doubling of L* ◁
  $q \leftarrow$ random number in $[0, 1]$
  if $q < \frac{2\log(m)}{L\log(|\mathcal{A}|)}$ **then**
    $L \leftarrow 2 \cdot L$
    $m \leftarrow 1$
  ▷ *amplitude amplification* ◁
  $k \leftarrow$ random integer in $[0, \mathrm{m})$
  $|\psi\rangle \leftarrow \sum_{\vec{a} \in \mathcal{A}^{\otimes L}} \sqrt{\pi(\vec{a})}\, |\vec{a}\rangle_A\, |\vec{s}_{\text{init}}\rangle_S\, |-\rangle_R$
  $|\psi'\rangle \leftarrow G^k\, |\psi\rangle$
  $\vec{a}' \leftarrow$ **measure** $|\psi'\rangle$
  $N_{\text{act}} \leftarrow N_{\text{act}} + 2 \cdot k \cdot L$
  ▷ *classical verification episode* ◁
  for $i = 1$ **to** L **do**
    $s_i \leftarrow S_i(s_{i-1}, a'_i)$
    $r_i \leftarrow R_i(s_{i-1}, a'_i)$
    $N_{\text{act}} \leftarrow N_{\text{act}} + 1$
    if $r_i = 1$ **then**
      return $\vec{a}'$, N_{act}, $(s_0, \ldots, s_i)$, r_i
  ▷ *no reward* ◁
  $m \leftarrow \min\left(\lambda \cdot m, \sqrt{|\mathcal{A}|^L}\right)$

Our learning algorithm (see Algorithm 1) is strongly intertwined with this notion of two stages in Boyer's algorithm. The algorithm starts with a minimal episode length, which is $L = 1$ in the most uninformed case. We set the lower bound estimate $p_{\text{min}} = |\mathcal{A}|^{-L}$, assuming a uniform initial policy and at least one rewarded action sequence at the current episode length. In each round of the main loop, the episode length has a chance to be doubled, with probability

$$\varphi_L(m) \equiv \frac{\log(m)}{\log\left(\sqrt{|\mathcal{A}|^L}\right)} = \frac{2\log(m)}{L\log(|\mathcal{A}|)}. \tag{6}$$

This probability is chosen such that it reaches unity exactly when Boyer's algorithm reaches its critical stage. If the episode length doubling is triggered, m is reset to one.

This probabilistic episode length selection strategy serves several purposes. First, starting from low episode length values is resource-friendly, since the Hilbert space of action

sequences, $\mathcal{H}_A^{\otimes L}$, scales exponentially with the episode length L. It is also more efficient with regard to the total number of actions played before reaching the target, which will be our main metric (see Section 4.1). Second, the exponential increase of L enables the hybrid algorithm to reach large episode lengths reasonably quickly, which might be necessary in scenarios with distant targets. Finally, coupling the doubling probability to the parameter m ensures that the algorithm does not spend too many tries with an episode length which has no, or vanishing, success chance.

As mentioned in Section 2.2, the hybrid agent can be paired with the policy updating mechanism of many classical RL algorithms. Since in this work, we focus solely on finding the first reward, the subsequent updating of the policy is not of relevance. Therefore, we do not include the formulation of a specific policy update rule but instead just require a given (initial) policy π in Algorithm 1.

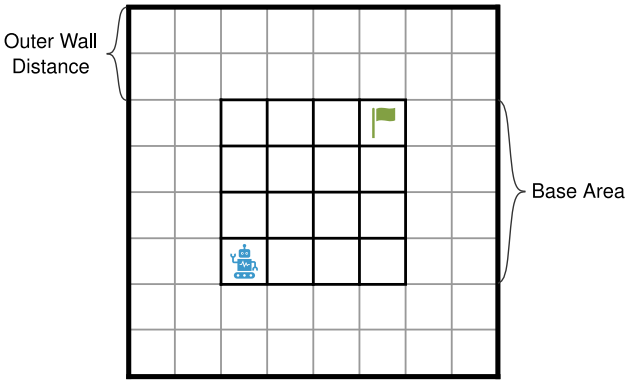


Fig. 2 Example of a Gridworld layout used in the simulations. The blue robot and the green flag symbols indicate the start and target position, respectively. The inner square of thick lines is the so-called *base area*, here with a size of 4×4 cells. The Gridworld has outer walls which prohibit the RL agent from moving away further. The example here shows an *outer wall distance* of 2

3.2 Simulation

To systematically test our hybrid method, we investigate the performance of our hybrid learning agent in a two-dimensional Gridworld scenario, which is a standard problem for classical RL. The standard Gridworld (Sutton and Barto 2018) consists of a grid of cells, and the goal is to find the shortest path from a start cell to a given target cell. In each cell, a RL agent may choose one action of the set $\mathcal{A} \in \{\text{up, down, left, right}\}$ changing the position/cell of the RL agent accordingly. Usually, the grid is surrounded by walls, and also walls between arbitrary cells are possible. Standing next to a wall and choosing an action towards it yields no change in the cell state. The first action to reach the target cell in each episode yields a reward of one; every other action is not rewarded.

The Gridworld layouts in our simulations have a quadratic *base area* with one start and one target cell placed in diagonally opposing corners. Further, the quadratic base area is surrounded by outer walls. We decide to have no inner walls, since their existence is not crucial for our investigations. To generate a variety of shapes for the function $p_{\text{init}}(L)$ on which the performance of our strategies depends, we vary this basic Gridworld layout in two ways, as illustrated in Fig. 2:

1. We vary the side lengths of the quadratic base area, which we denote b. This obviously has an effect on the *minimal episode length* $L_{\min}$ necessary to reach the target, $L_{\min} = 2(b-1)$. Additionally, it influences the initial success probability for the minimal episode length (assuming a uniform policy initialization): $p_{\text{init}}(L_{\min}) = \binom{2(b-1)}{b-1} \cdot 4^{-2(b-1)}$, with $\binom{2(b-1)}{b-1}$ being the number of distinct paths of length $2(b-1) = L_{\min}$ that reach the target.
2. We vary the distance of the outer walls around the Gridworld's base area. An *outer wall distance*, or d_{wall}, of zero indicates that the walls are directly surrounding the base area. Having $d_{\text{wall}} = n$ would result in a ring of cells of thickness n between the base area and the outer walls. This has no effect on $p_{\text{init}}(L_{\min})$. However, more distanced outer walls increase the general state space and, in particular, add cells to the state space which are further from the target cell than any cell in the Gridworld's base area. Therefore, it effectively decreases how quickly $p_{\text{init}}(L)$ rises with increasing episode length L.

In our simulations, we vary the Gridworld's base area from size 5×5 (with $p_{\text{init}}(L_{\min}) = 1.1 \times 10^{-3}$) to 9×9 ($p_{\text{init}}(L_{\min}) = 2.9 \times 10^{-6}$). For the outer wall distance, we test the values 0, 4, 8, 16, 32, and 64. The scenario of no outer walls, which is equal to the limit of an infinite outer wall distance, is not computationally feasible to realize within our simulation framework (for an explanation and more implementation details, see Appendix B). The dependency of $p_{\text{init}}(L)$ for a selection of the different Gridworld layouts is shown in Fig. 3.

In this paper, we concentrate on finding the first reward, since the main speedup our hybrid agent achieves compared to classical agents happens during this stage. In addition, from this point on, the learning process and the achievable quantum speedup depend crucially on the chosen policy update mechanism and chosen learning parameters, making

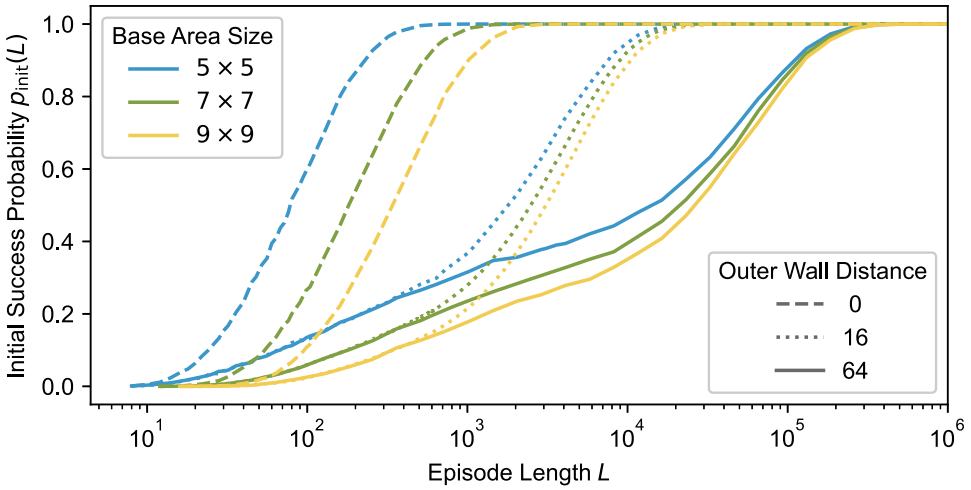


Fig. 3 Dependency of the agent's initial success probability on its episode length L for different Gridworld layouts (varied by their base area size and outer wall distance). The probability values are generated using a Monte Carlo simulation, see Appendix B for more details

general statements on the comparison of hybrid and classical learning agents difficult. Since we focus on the scenario of finding the first reward, we assume an untrained agent with an initial uniform policy $\pi(a) = \frac{1}{|\mathcal{A}|} \quad \forall a \in \mathcal{A}$. Given that in the Gridworld scenario, intermediate percepts of an episode depend on the actions played within that episode, the hybrid algorithm requires $\alpha_k = 2$ queries of the environment unitary U_{env} per iteration of the Grover operator G (cf. footnote 1 on 4). Every strategy is tested on each Gridworld layout for $N = 10000$ runs.

3.3 Classical strategies

We compare the extended hybrid algorithm to two classical strategies, which we present and motivate in the following.

A direct classical equivalent to the extended hybrid algorithm can be devised by employing the same probabilistic episode length doubling strategy. Here, the parameter m of Algorithm 1 loses its twofold function and only defines the respective probability to double the episode length L. This episode length then defines the number of steps the agent may take until it is reset to its starting position. Again, we set the hyperparameter $\lambda = \frac{5}{4}$ for this algorithm. From here on, we denote this the *probabilistic classical* strategy.

The second classical strategy arises from the idea that only in the hybrid algorithm an episode length needs to be given. Classical algorithms, however, are not restricted in such a way. Practically, not setting an episode length implies an uninterrupted random walk governed by the agent's current policy until the reward state is reached. We denote this the *unrestricted classical* strategy.

4 Results

Before presenting the results of our simulation, we discuss the novel figure of merit and its implications. This figure of merit is different from the original hybrid learning agent introduced in Hamann and Wölk (2022).

4.1 Figure of merit

The quadratic speedup that was proven theoretically and experimentally in the initial works on the hybrid learning agent (Hamann and Wölk 2022; Saggio et al. 2021) is based on the number of queries of the environment unitary U_{env}. This is equivalent to the number of episodes played in a classical context.

This figure of merit is misleading in this scenario, as we now argue. Unlike in Hamann and Wölk (2022), we no longer operate with a fixed episode length in our RL scenario. Given that the episode length has a crucial impact on the agent's initial success probability (cf. Fig. 3), omitting an episode's length from the figure of merit is unreasonable in this scenario. Additionally, due to the monotonously increasing success probability, in the limit of an infinite episode length, one singular episode is always sufficient to find the target.

Hence, we define the total number of actions taken, N_{act}, instead of the number of RL episodes as the new figure of merit in our scenario. With a current episode length of L, this metric is counted as follows:

- In the quantum part of the hybrid learning agent, k iterations of amplitude amplification add $2 \cdot k \cdot L$ steps to the count. Here, the factor two stems from the fact that we require $\alpha_k = 2$ applications of the environment unitary U_{env} per iteration of amplitude amplification.
- In the classical part of the hybrid learning agent and for a purely classical agent, an unsuccessful episode adds L steps to the count. If the agent reaches the target, only the actual number of steps $i \leq L$ necessary to reach the target is counted.

We derive theoretical expressions for the expected performance of the probabilistic hybrid and the unrestricted classical strategy in Appendix D.

4.2 Simulation results

A full overview of the results for each combination of strategy and Gridworld configuration is given in Table 1 of Appendix A. Figure 4 visualizes the results on a subset of Gridworld configurations.

According to the two ways by which we varied the basic Gridworld layout, *base area size* b and *outer wall distance* d_{wall}, two effects can be observed in the data:

(i) The necessary number of actions to reach the target increases with increasing base area size b as expected. This observation holds for all strategies and outer wall distance values.

(ii) The influence of the outer wall distance parameter, d_{wall}, differs for the different strategies. Thus, the question whether the hybrid agent or one of the classical agents is preferable depends on the outer wall distance.

At $d_{\text{wall}} = 0$ and $d_{\text{wall}} = 4$, the unrestricted classical agent requires on average the least number of actions to reach the target across all base area sizes. Both probabilistic strategies require approximately 1.2 to 5 times the number of actions, with the hybrid version requiring the most steps at larger base area sizes. For larger values of d_{wall}, both probabilistic strategies appear to stabilize in terms of N_{act}, which can be seen from the flat curves in Fig. 4 for outer wall distances of 8 and higher. Between the two, the hybrid strategy consistently has lower N_{act}, with the ratio of fewer actions ranging from 27 to 42% (for base area 9×9 with $d_{\text{wall}} = 8$ and base area 5×5 and $d_{\text{wall}} = 64$, respectively). For the unrestricted classical strategy, however, the number of actions increases continuously with increasing d_{wall}. At $d_{\text{wall}} = 8$, it is still more efficient than the hybrid agent for the two largest base area sizes, 8×8 and 9×9. Already for $d_{\text{wall}} = 16$, however, it requires more actions than either probabilistic strategy for any base area size. For the largest tested Gridworld configuration (base area size 9×9 and $d_{\text{wall}} = 64$), the unrestricted classical strategy trails the probabilistic hybrid one by more than an order of magnitude.

Another interesting quantity is the so-called *terminal episode length*. In the case of the unrestricted classical strategy which stops as soon as the target is reached, this is

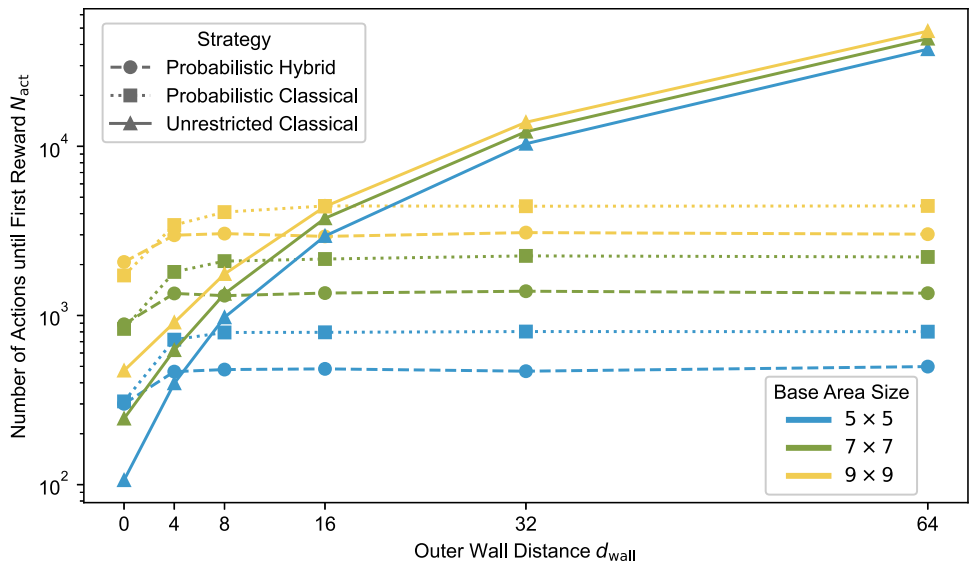


Fig. 4 Results of the tested hybrid and classical strategies for the first reward search problem. Gridworld layouts are varied by their base area size as well as their outer wall distance. Datapoints show the mean over $N = 10000$ runs per configuration, see Table 1 for the respective standard errors

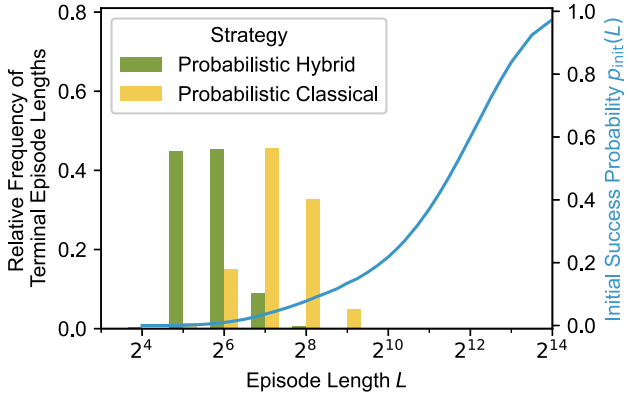


Fig. 5 Relative frequency of terminal episode lengths for the probabilistic hybrid and classical algorithms. The blue curve represents the initial success probability for the given Gridworld layout depending on the episode length. The results shown stem from the Gridworld configuration with a base area size of 9×9 and outer wall distance of 16

equivalent to the length of the solution. In the case of the probabilistic strategies, this quantity refers to the episode length currently used by the algorithm when the target was first reached. Due to the episode length doubling nature of the probabilistic strategies, this is always a power of two. In this case, the terminal episode length provides an upper bound to the length of the solution. A visual comparison of the relative frequencies with which either probabilistic strategy terminates at a specific episode length is given in Fig. 5.

The probabilistic hybrid strategy terminates on average at lower episode lengths than the classical probabilistic strategy. For the Gridworld configuration used for Fig. 5 (base area 9×9 and $d_{\text{wall}} = 16$), the most frequent terminal episode

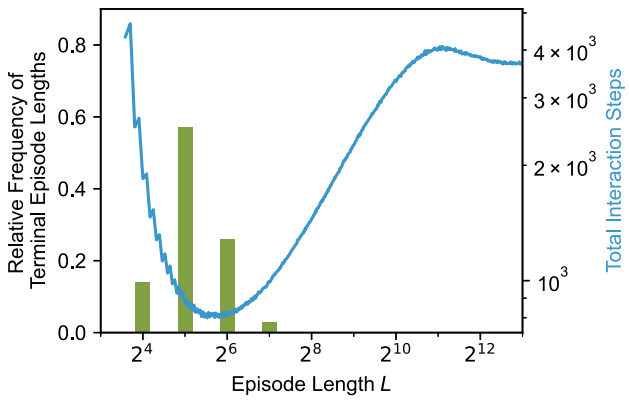


Fig. 6 Relative frequency of terminal episode lengths for the probabilistic hybrid algorithm and expected number of total interaction steps for the fixed-length hybrid algorithm (cf. Fig. 1). The Gridworld configuration used here has a base area size of 7×7 and an outer wall distance of 16

lengths for the hybrid strategy are 32 and 64, whereas the probabilistic classical strategy terminates most frequently at 128 and 256. In Fig. 6, we compare the relative frequency of terminal episode lengths of the probabilistic hybrid algorithm with the total number of interaction steps that the original hybrid learning agent of Hamann and Wölk (2022) would require with a fixed episode length (cf. Fig. 1).

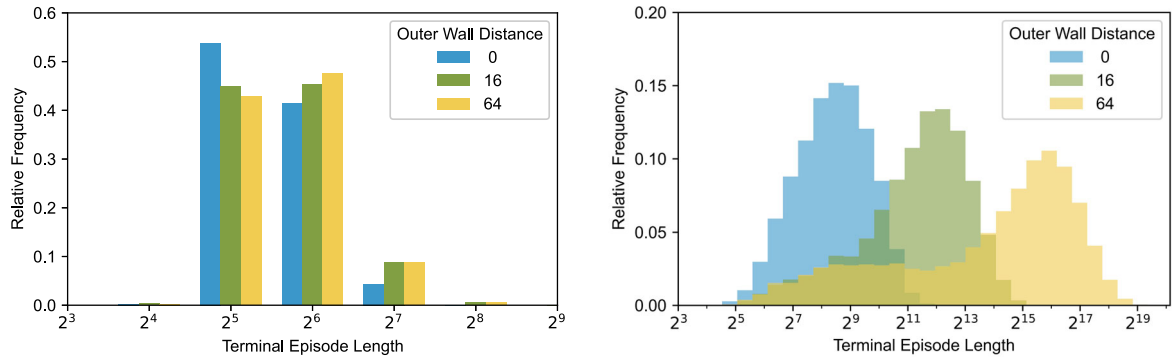
For the given Gridworld configuration, using the hybrid learning agent with fixed episode length $L = 34$ would be optimal in terms of the total interaction steps. As the distribution of terminal episode lengths in Fig. 6 shows, our probabilistic hybrid algorithm terminates most frequently at the nearest ($L = 2^5 = 32$) and second most frequently at the second nearest ($L = 2^6 = 64$) episode length. This suggests that the in-built episode length doubling mechanism is tuned well enough that the probabilistic hybrid algorithm reaches efficient episode lengths and at the same time does not massively overshoot to unnecessary large episode lengths.

Figure 7 shows the influence of the outer wall distance parameter on the terminal episode length. For the probabilistic hybrid strategy (Fig. 7a), increasing d_{wall} results in a slight shift of relative frequencies towards larger terminal episode lengths. Namely, the most frequent terminal episode length shifts from 32 for $d_{\text{wall}} = 0$ to 64 for $d_{\text{wall}} = 16$ and $d_{\text{wall}} = 64$. For the unrestricted classical strategy (Fig. 7b), the episode length shifts by several orders of magnitude for larger outer wall distances. As the figure shows, for $d_{\text{wall}} = 0$, it terminates most frequently at episode lengths between 2^8 and 2^9, whereas for $d_{\text{wall}} = 16$, this interval ranges from 2^{12} to 2^{13} and for $d_{\text{wall}} = 64$ from 2^{15} to 2^{16}. The on average much shorter terminal episode lengths of the probabilistic hybrid strategy compared to the unrestricted classical strategy directly imply much shorter solutions for the first rewarded action sequence.

5 Conclusion

In this work, we have introduced a hybrid agent for quantum-accessible reinforcement learning with a flexible episode length selection strategy. As we have argued, this extension is crucial for finding the first reward in RL scenarios which are equivalent to a Gridworld with no information about the length of the shortest rewarded path. Achieving this goal efficiently enables the swift continuation of the RL process with the newfound knowledge of a sufficient episode length.

The simulation results for different Gridworld configurations suggest that our proposed hybrid agent can (i) find the first reward faster and (ii) can find shorter solutions than the considered classical agents in many configurations. Namely,



(a) Probabilistic hybrid strategy. (b) Unrestricted classical strategy.

Fig. 7 Comparison of the terminal episode lengths between the probabilistic hybrid and the unrestricted classical strategy. The results shown are for the configuration with a base area size of 9×9

pairing the hybrid agent with the probabilistic episode length selection strategy appears beneficial compared to both the unrestricted and the probabilistic classical strategy in configurations with large state spaces (i.e., larger values of d_{wall} in our simulations) and with low initial success probabilities (i.e., larger base area sizes b in our simulations). Thus, even though the quadratic scaling advantage does not apply any more for this metric and scenario, the hybrid strategy outperforms its classical counterparts especially at harder problems.

In general, imposing an episode length results in better performance for large state spaces also in the comparison between the classical agents. Our intuition behind this is that in large state spaces, a random walk might on average increase the distance to the target compared to the starting position such that a reset to the start is beneficial in many situations. With regard to how quickly our probabilistic strategy moves to larger episode lengths, our analysis reveals that the doubling of the episode length happens slow enough such that (i) the maximal episode length stays in a reasonable regime and that (ii) noticeable speedups through amplitude amplification can be achieved (cf. Fig. 5).

Finally, we address a few design questions on the chosen Gridworld layout, especially regarding a few omissions of further Gridworld variations. First, one could conceive of a scenario where moving into some cells, or even all walls, stops the episode immediately without a reward. If we have a move sequence $\vec{a}$ which moves into such a terminal, but non-rewarded cell, concatenating any additional move sequence $\vec{a}'$ will not turn the full sequence into a rewarded one. Thus, with a uniform initial policy, the initial success probability does not converge to 1 in the limit of infinite episode lengths. Here, it can be assumed that the probabilistic hybrid strategy is beneficial as this strategy works well with a slowly increasing success probability and low success probabilities in general. Second, one could omit the outer walls altogether, yielding an infinite state space. As mentioned in Section 3.2, simulating this scenario is not feasible. However, we can extrapolate the trend for increasing outer wall distances, since having no outer walls is equal to the limit $d_{\text{wall}} \to \infty$. Here, we can see that the probabilistic strategies prove to be advantageous, with the hybrid version still requiring less steps than the classical one. Third and last, one could investigate higher-dimensional Gridworld layouts than the two-dimensional scenario shown here. For random walks in hypergrids of dimension $D \geq 3$, however, the probability to reach any point with a random walk in the limit of infinite steps does not converge to unity (Pólya 1921). Therefore, an infinite random walk in the fashion of the unrestricted classical strategy is certainly a bad choice. Given that, besides the asymptotic limit, the scenario is not fundamentally different, the hybrid probabilistic strategy can again be assumed to be the most efficient of the three, supposed that the initial success probability is low. With these generalizations of our basic Gridworld toy model, we expect our hybrid agent to be applicable in an even wider range of real-world problems.

There are two research questions that should be investigated further: (i) Is the coincidence of the most probable terminal episode length with the optimal episode length pure luck or a reliable property of our algorithm? (ii) In which ways does the on average shorter terminal episode length of our probabilistic hybrid agent compared to the probabilistic classical agent influence the further learning? In addition, the successful extension of the hybrid learning agent to episodic learning tasks with unknown target distance now enables the application to many more realistic learning tasks, which should be investigated in the future.

Appendix A. Table of full results

Table 1 Results of the tested hybrid and classical strategies for the first reward search problem

Outer wall distance	Strategy	Base area size				
		5×5	6×6	7×7	8×8	9×9
	Probabilistic hybrid	300 (2)	515 (3)	887 (5)	1400 (9)	2071 (14)
0	Probabilistic classical	310 (2)	534 (3)	833 (5)	1208 (7)	1724 (9)
	Unrestricted classical	**106 (1)**	**170 (2)**	**246 (2)**	**347 (3)**	**472 (4)**
	Probabilistic hybrid	466 (3)	806 (4)	1350 (7)	2087 (11)	2981 (17)
4	Probabilistic classical	720 (6)	1206 (9)	1809 (12)	2541 (16)	3426 (21)
	Unrestricted classical	**397 (4)**	**503 (5)**	**624 (6)**	**749 (7)**	**909 (8)**
	Probabilistic hybrid	**479 (3)**	**797 (5)**	1309 (8)	2106 (14)	3048 (18)
8	Probabilistic classical	793 (7)	1333 (11)	2096 (16)	3083 (23)	4085 (29)
	Unrestricted classical	975 (10)	1165 (11)	1343 (13)	**1539 (14)**	**1758 (16)**
	Probabilistic hybrid	**483 (3)**	**826 (4)**	**1357 (7)**	**1986 (11)**	**2934 (17)**
16	Probabilistic classical	795 (7)	1374 (12)	2157 (18)	3115 (24)	4442 (34)
	Unrestricted classical	2950 (34)	3375 (37)	3754 (41)	4095 (43)	4406 (45)
	Probabilistic hybrid	**468 (3)**	**816 (5)**	**1391 (8)**	**2047 (12)**	**3090 (18)**
32	Probabilistic classical	803 (7)	1364 (12)	2250 (18)	3291 (26)	4426 (34)
	Unrestricted classical	10,357 (135)	11,219 (139)	12,208 (151)	13,152 (154)	13,872 (157)
	Probabilistic hybrid	**499 (3)**	**847 (5)**	**1355 (7)**	**2173 (12)**	**3025 (18)**
64	Probabilistic classical	802 (7)	1385 (12)	2218 (18)	3147 (25)	4442 (34)
	Unrestricted classical	37,514 (543)	40,668 (555)	43,315 (578)	46,448 (606)	47,963 (615)

Numbers represent average N_{act} (with the respective standard error in parentheses) for the different Gridworld layouts, based on $N = 10000$ runs per configuration. For each configuration, the lowest value of N_{act} is printed in boldface

Appendix B. Simulation details

Given that amplitude amplification (AA) is not a NISQ-compatible algorithm, we have to fall back to simulating its effect instead of doing a full quantum circuit execution on simulated or real hardware.

To do so, we infer the initial success probability $p_{\text{init}}(L)$ for any episode length L which might occur in the hybrid strategy beforehand. Having knowledge of this probability, we can subsequently compute the amplified success probability using the well-known AA equation (Brassard et al. 2002)

$$p_{AA}(L,k) = \sin^2\left([2k+1]\arcsin\left[p_{\text{init}}(L)^{-1/2}\right]\right)$$

for k iterations of our Grover operator G. This probability can in turn be used to correctly sample a rewarded or non-rewarded action sequence.

Given that the initial policy that generates $p_{\text{init}}(L)$ is a uniform probability distribution over the space of actions, the agent's movement initially equals an unweighted random walk. Therefore, we can estimate $p_{\text{init}}(L)$ with a Monte Carlo simulation of random walks of length L.

For each combination of Gridworld configuration and episode length L to be tested, we perform at least $N_{\text{shots}}(L) = 2^{14}$ runs and count the number of random walks which terminated successfully, $N_{\text{success}}(L)$ (i.e., which have the target cell in their path). To improve numerical stability, we keep incrementing $N_{\text{shots}}(L)$ in batches of 2^{14} until $N_{\text{success}}(L)$ has reached at least 16. Doing so, we can finally estimate the initial success probability simply as the ratio of successes to shots:

$$p_{\text{init}}(L) \approx \frac{N_{\text{success}}(L)}{N_{\text{shots}}(L)}.$$

Due to the hybrid agent's doubling strategy, $p_{\text{init}}(L)$ only needs to be pre-computed for powers of 2 and until convergence of $p_{\text{init}}(L)$. For the plot in Fig. 3, however, we also computed $p_{\text{init}}(L)$ for intermediate values to create a smoother curve using linear interpolation.

Finally, in this section, we address why omitting outer walls at all is not feasible within this framework. As proven in Pólya (1921), on an infinite two-dimensional grid, the random walker's probability to pass by any given point $\mathbf{x} \in \mathbb{Z}^2$ converges to 1 in the limit of infinite steps. Therefore, even for the scenario of no walls, which is equivalent to an infinite two-dimensional grid, $p_{\text{init}}(L)$ should converge towards one in the limit of infinite steps,

$$\lim_{L\to\infty} p_{\text{init}}(L) = 1.$$

The issue for our simulation is, however, the slow rate of convergence. Even for the smallest base area size of 5×5, $p_{\text{init}}(L)$ has just reached approximately 60% for $L = 2^{22} = 4194304$ in the "no-walls" scenario, whereas for $d_{\text{wall}} = 64$, the probability already converges near $L = 2^{19} = 1048576$. By counting just the steps of unsuccessful random walks, we thus arrive at

$$\begin{aligned} &L \cdot (1 - p_{\text{init}}(L)) \cdot N_{\text{shots}} \\ \approx &2^{22} \cdot (1 - 0.6) \cdot 2^{14} \\ \approx &2.7 \times 10^{10} \end{aligned}$$

steps computed just for this episode length, which becomes soon fully intractable for even larger episode lengths due to the slow increase in $p_{\text{init}}(L)$.

Additionally, the run time scaling of the unrestricted classical strategy in Fig. 4 with increasing outer wall distance shows the intractability of simulating this strategy in a "no-walls" scenario.

Appendix C. Total interaction steps for fixed episode length

In this section, we give some background on the performance comparison shown in Fig. 1, which motivates the flexible episode length selection strategy for the hybrid agent.

The example stems from an RL setting of a Gridworld with a base area size of 7×7 and an outer wall distance of 16 (see Section 3.2 for our Gridworld layout definitions). We choose fixed episode lengths L in the interval ranging from $L_{\text{min}} = 12$ up to $2^{14} = 16834$. Further, we assume untrained agents initialized with uniform action selection probabilities such that the classical success probability $p_{\text{init}}(L)$ is the one of a random walk of length L through the maze. For both the classical and hybrid agent, we count the total number of interaction steps, i.e., the total number of actions performed until the first reward is reached.

The results presented in Fig. 1 are generated with a Monte Carlo simulation of 100,000 repetitions each on a logarithmically spaced grid of 868 different episode length values. For the classical agent, we perform a random walk that is periodically reset after L steps until the first reward is found, aggregating the total number of steps. For the hybrid agent, we rely on the simulation of amplitude amplification presented in Appendix B, using the precomputed initial success probabilities for each episode length. If the simulated amplitude amplification returns a rewarded episode, we sample the length of a rewarded action sequence by performing random walks of at most length L until there is a rewarded one.

The "zig-zag" behavior in Fig. 1 for small episode lengths can be explained as follows. For the chosen Gridworld layout,

the minimal episode length to reach the target is twelve. As we increase the episode length, non-optimal paths may now also reach the target, resulting in an increase of the success probability. However, this increase only occurs in episode length intervals of two as one cannot land on the target cell with an odd number of steps. Thus, for small episode lengths, the success probability only increases from an even number to the next but stays constant for the next larger odd value. Only when the episode length is large enough that the agent may run into a wall and thus "waste" a step, the success probability increases for every incrementally larger episode length. This piecewise constant success probability leads to the spikes for odd episode length values in Fig. 1. Indeed, there are on average as many unsuccessful episodes as with the next lower even episode length, but these episodes are more "costly" due to the additional step.

Appendix D. Theoretical performance investigations

In this section, we derive expressions for the expected performance in terms of the total number of actions taken to find a reward, N_{act}, for a given success probability function $p(L)$. Here, we assume that the probability to find the target converges to unity in the limit of infinitely many steps, $\lim_{L\to\infty} p(L) = 1$. We focus on the comparison between the probabilistic hybrid and the unrestricted classical strategy.

D.1 Unrestricted classical strategy

For the unrestricted classical strategy, the expected number of actions to find a reward is

$$\begin{aligned}&\mathbb{E}_{\text{unres. class.}}[N_{\text{act}}]\\ &= \sum_{L=1}^{\infty} L \cdot \big[p(L) - p(L-1)\big] \qquad \text{(D1)}\\ &\approx \int_0^{\infty} \big[1 - p(L)\big] dL. \qquad \text{(D2)}\end{aligned}$$

This follows from the idea that $\big[p(L) - p(L-1)\big]$ is the probability to have success *exactly* after L steps, which is indeed a valid probability distribution due to the fact that $\lim_{L\to\infty} p(L) = 1$. The approximation as an integral follows from the geometric idea that Eq. D1 describes the area which is bounded by $p(L)$ from below and by unity from above in the $\big(L,\ p(L)\big)$-graph.

D.2 Probabilistic hybrid strategy

Computing the expected number of actions taken until a reward is found for the probabilistic hybrid strategy is slightly more involved than in the unrestricted classical case.

Let $L = 2^n$ with $n \in \mathbb{N}_0$ denote the current episode length and i be the count of completed iterations of the main loop of the probabilistic algorithm (Algorithm 1) with that L. The possible events and their probabilities within one execution of the main loop of the probabilistic hybrid algorithm are presented schematically in Fig. 8.

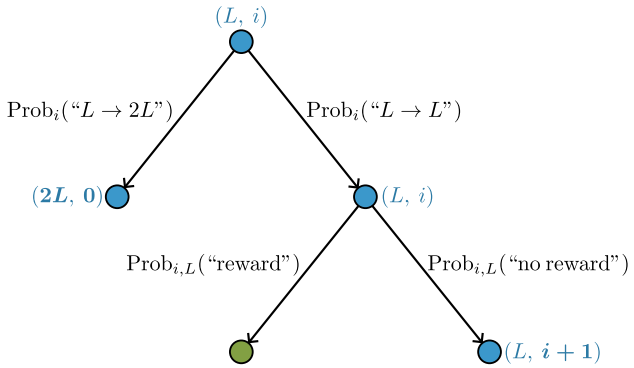


Fig. 8 Schematic of the decisions within one iteration of the probabilistic hybrid algorithm (Algorithm 1) after i completed iterations at episode length L. Starting from the root node at the top, a stochastic decision is first taken whether the episode length is doubled. If it is not doubled, an actual sampling step is performed. If the green leaf node is reached, a reward is found and the algorithm terminates. The corresponding probabilities are printed in black, boldface entries in the blue $(L,\ i)$-tuples highlight a change in either the episode length L, the count of completed iterations i, or both

Before presenting the individual probabilities occurring in the probabilistic hybrid algorithm, we recall the connection between the count of completed loop iterations i and the parameter m, which controls both the sampling interval for the number of Grover iterations, which is $[0, m)$, and the probability to double the episode length, denoted $\varphi_L(m)$ in the main text. In the first iteration (i.e., $i = 0$) at the current episode length, m is set to 1. At the end of each iteration without success, it is multiplied by λ until a maximal value of $m_{\max} = \sqrt{|\mathcal{A}|^L}$ is reached. Before m reaches $m_{\max}$ after

$$i_{\max}(L) = \left\lceil \frac{L \log(|\mathcal{A}|)}{2 \log(\lambda)} \right\rceil \qquad \text{(D3)}$$

loop iterations, we can therefore identify $m = \lambda^i \quad \forall\, i < i_{\max}$.

We can now rewrite the probability to double the episode length, $\varphi_L(m)$, in terms of i. For clarity and consistency with the subsequently introduced probabilities, we denote

this probability $\text{Prob}_i(\text{“}L \to 2L\text{”})$:

$$\begin{aligned}\text{Prob}_i(\text{“}L \to 2L\text{”}) &= \frac{2\log(m)}{L\log(|\mathcal{A}|)} \\ &= \begin{cases} \frac{i}{L}\cdot\frac{2\log(\lambda)}{\log(|\mathcal{A}|)}, & 0 \le i < i_{\max}, \\ 1, & i = i_{\max}. \end{cases}\end{aligned} \tag{D4}$$

We denote the corresponding complementary probability to continue with the current episode length as

$$\text{Prob}_i(\text{“}L \to L\text{”}) = 1 - \text{Prob}_i(\text{“}L \to 2L\text{”}).$$

Further, since the probabilistic doubling of the episode length happens first in an iteration before any RL interaction takes place, we can always assume in the following that $i < i_{\max}$ and, therefore, express m as λ^i.

The probability $\text{Prob}_{i,L,k}(\text{“reward”})$ to sample a rewarded action sequence at the current episode length L after i completed loop iterations (and thus complete the search for the first reward) depends on the classical success probability $p(L)$ and the randomly sampled number of Grover iterations k and is given by

$$\text{Prob}_{i,L,k}(\text{“reward”}) = \sin^2\left([2k+1]\theta_L\right) \tag{D5}$$

with $\theta_L = \arcsin[\sqrt{p(L)}]$. To find the expected success probability $\text{Prob}_{i,L}(\text{“reward”})$ for some L and i, we compute the expectation value over the uniform interval $[0, \lambda^i)$:

$$\begin{aligned}&\text{Prob}_{i,L}(\text{“reward”}) \\ &= \mathbb{E}_{k\sim\text{unif}\{0,\lambda^i\}}[\text{Prob}_{i,L,k}(\text{“reward”})] \\ &= \frac{1}{\lceil\lambda^i\rceil}\sum_{k=0}^{\lceil\lambda^i\rceil-1}\sin^2\left([2k+1]\theta_L\right) \\ &= \begin{cases} \frac{1}{2} - \frac{\sin(4\lceil\lambda^i\rceil\theta_L)}{4\lceil\lambda^i\rceil\sin(2\theta_L)}, & 0 < \theta_L < \pi/2, \\ 0, & \theta_L = 0, \end{cases}\end{aligned} \tag{D6}$$

where the last equality stems from Lemma 2 in Boyer et al. (1998). The corresponding complementary probability is

$$\text{Prob}_{i,L}(\text{“no reward”}) = 1 - \text{Prob}_{i,L}(\text{“reward”}).$$

Using these basic probabilities, we can express the probability that at episode length L, a reward is found before the doubling of the episode length is triggered:

$$\begin{aligned}&\text{Prob}(\text{“reward at length } L\text{”}) \\ &= \sum_{i=1}^{i_{\max}}\text{Prob}(\text{“reward exactly after } i \text{ iter. at } L\text{”}) \\ &= \sum_{i=1}^{i_{\max}}\Bigg(\text{Prob}_{i-1}(\text{“}L\to L\text{”})\cdot\text{Prob}_{i-1,L}(\text{“reward”}) \\ &\quad\cdot\prod_{j=0}^{i-2}\Big[\text{Prob}_j(\text{“}L\to L\text{”})\cdot\text{Prob}_{j,L}(\text{“no reward”})\Big]\Bigg)\end{aligned} \tag{D7}$$

The term $\text{Prob}(\text{“reward exactly after } i \text{ iter. at } L\text{”})$ defines the probability that a reward is found exactly after i iterations at the current episode length L. This probability can be derived from passing the decision tree in Fig. 8, starting with some L and $i = 0, i-1$ times following the rightmost path, followed by a consecutive pass towards the rewarded (green) leaf node.

The probability terms stated above now allow the computation of the probability that the probabilistic algorithm reaches a certain combination of episode length $L = 2^n$ and number of passed iterations i and, furthermore, actually performs a sampling step with amplitude amplification. This probability is denoted $\text{Prob}(\text{“play at } (L = 2^n,\ i)\text{”})$. A combination of L and i may not be reached due to either the strategy terminating at a smaller episode length or due to doubling the episode length before or within the $(i+1)$-th iteration at the current episode length. The probability can be decomposed as follows:

$$\begin{aligned}&\text{Prob}(\text{“play at } (L = 2^n,\ i)\text{”}) \\ &= \left[\prod_{m=0}^{n-1}\Big(1 - \text{Prob}(\text{“reward at } L = 2^m\text{”})\Big)\right] \\ &\quad\cdot\left[\prod_{j=0}^{i-1}\text{Prob}_j(\text{“}2^n\to 2^n\text{”})\Big(1-\text{Prob}_{j,2^n}(\text{“reward”})\Big)\right] \\ &\quad\cdot\text{Prob}_i(\text{“}2^n\to 2^n\text{”})\end{aligned} \tag{D8}$$

Finally, before stating the full expected value of N_{act} for the probabilistic hybrid algorithm, we must compute the expected value of N_{act} for a certain combination of i and L, named $\mathbb{E}^{(i,L)}[N_{\text{act}}]$. This is again an expectation over the number of Grover iterations k, which is uniformly sampled from the interval $[0, \lambda^i)$. Performing k Grover iterations contributes $2kL$ actions to N_{act} in the quantum part of the algorithm and up to L actions in the classical verification of the sampled sequence. Hence, the total of $(2k+1)L$ actions

provides an upper bound:

$$\begin{aligned}\mathbb{E}^{(i,L)}[N_{\text{act}}] &\leq \frac{1}{\lceil\lambda^i\rceil}\sum_{k=0}^{\lceil\lambda^i\rceil-1}(2k+1)L \\ &= \frac{L}{\lceil\lambda^i\rceil}\left(2\frac{(\lceil\lambda^i\rceil-1)\lceil\lambda^i\rceil}{2}+\lceil\lambda^i\rceil\right) \\ &= L\cdot\lceil\lambda^i\rceil\end{aligned} \tag{D9}$$

Now, we can write the expected value of N_{act} for the probabilistic hybrid algorithm as the sum of the expected number of actions performed for a certain combination of i and L, $\mathbb{E}^{(i,L)}[N_{\text{act}}]$, weighted by the probability Prob("play at $(L=2^n,\ i)$") for all possible combinations of L and i:

$$\begin{aligned}&\mathbb{E}_{\text{hybrid}}[N_{\text{act}}] \\ &=\sum_{n=0}^{\infty}\left(\sum_{i=0}^{i_{\max}(2^n)}\left[\mathbb{E}^{(i,L=2^n)}[N_{\text{act}}]\right.\right. \\ &\left.\left.\cdot\,\text{Prob}(\text{"play at }(L=2^n,\ i)\text{"})\right]\right).\end{aligned} \tag{D10}$$

Acknowledgements The authors thank Alessio Belenchia and Benjamin Desef for helpful comments.

Author Contributions S.W. and O.S. conceived the study. O.S. carried out the simulation and the analysis of the results. S.W. supervised the project. Both authors contributed to writing, editing, and reviewing the manuscript.

Funding Open Access funding enabled and organized by Projekt DEAL.

Data Availability No datasets were generated or analyzed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

Biamonte J, Bergholm V (2017) Tensor networks in a nutshell. arXiv:1708.00006

Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S (2017) Quantum machine learning. Nature 549:195–202. https://doi.org/10.1038/nature23474

Boyer M, Brassard G, Høyer P, Tapp A (1998) Tight bounds on quantum searching. Fortschr Phys 46:493–505. https://doi.org/10.1002/(SICI)1521-3978(199806)46:4/5<493::AID-PROP493>3.0.CO;2-P

Brassard G, Høyer P, Mosca M, Tapp A (2002) Quantum amplitude amplification and estimation. Contemp Math 305:53–74. https://doi.org/10.1090/conm/305/05215

Bridgeman JC, Chubb CT (2017) Hand-waving and interpretive dance: an introductory course on tensor networks. J Phys A 50:223001. https://doi.org/10.1088/1751-8121/aa6dc3

Briegel HJ, Cuevas G (2012) Projective simulation for artificial intelligence. Sci Rep 2:400. https://doi.org/10.1038/srep00400

Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W (2016) OpenAI Gym. arXiv:1606.01540

Bukov M, Day AGR, Sels D, Weinberg P, Polkovnikov A, Mehta P (2018) Reinforcement learning in different phases of quantum control. Phys Rev X 8:031086. https://doi.org/10.1103/PhysRevX.8.031086

Cerezo M, Arrasmith A, Babbush R, Benjamin SC, Endo S, Fujii K, McClean JR, Mitarai K, Yuan X, Cincio L, Coles PJ (2021) Variational quantum algorithms. Nat Rev Phys 3:625–644. https://doi.org/10.1038/s42254-021-00348-9

Cerezo M, Larocca M, García-Martín D, Diaz NL, Braccia P, Fontana E, Rudolph MS, Bermejo P, Ijaz A, Thanasilp S, Anschuetz ER, Holmes Z (2024) Does provable absence of barren plateaus imply classical simulability? Or, why we need to rethink variational quantum computing. arXiv:2312.09121

Chen SY-C, Yang C-HH, Qi J, Chen P-Y, Ma X, Goan H-S (2020) Variational quantum circuits for deep reinforcement learning. IEEE Access 8:141007–141024. https://doi.org/10.1109/ACCESS.2020.3010470

Chen H-Y, Chang Y-J, Liao S-W, Chang C-R (2024) Deep Q-learning with hybrid quantum neural network on solving maze problems. Quant Mach Intell 6:2. https://doi.org/10.1007/s42484-023-00137-w

Cherrat EA, Kerenidis I, Prakash A (2023) Quantum reinforcement learning via policy iteration. Quant Mach Intell 5:30. https://doi.org/10.1007/s42484-023-00116-1

Dalla Pozza N, Buffoni L, Martina S, Caruso F (2022) Quantum reinforcement learning: the maze problem. Quant Mach Intell 4:11. https://doi.org/10.1007/s42484-022-00068-y

Dong D, Chen C, Li H, Tarn T-J (2008) Quantum reinforcement learning. IEEE Trans Syst Man Cybern B Cybern 38:1207–1220. https://doi.org/10.1109/TSMCB.2008.925743

Dunjko V, Taylor JM, Briegel HJ (2016) Quantum-enhanced machine learning. Phys Rev Lett 117:130501. https://doi.org/10.1103/PhysRevLett.117.130501

Fösel T, Tighineanu P, Weiss T, Marquardt F (2018) Reinforcement learning with neural networks for quantum feedback. Phys Rev X 8:031084. https://doi.org/10.1103/PhysRevX.8.031084

Fösel T, Niu MY, Marquardt F, Li L (2021) Quantum circuit optimization with deep reinforcement learning. arXiv:2103.07585

Ganguly B, Wu Y, Wang D, Aggarwal V (2023) Quantum computing provides exponential regret improvement in episodic reinforcement learning. arXiv:2302.08617

Grover LK (1997) Quantum mechanics helps in searching for a needle in a haystack. Phys Rev Lett 79:325–328. https://doi.org/10.1103/PhysRevLett.79.325

Guatto M, Susto GA, Ticozzi F (2024) Improving robustness of quantum feedback control with reinforcement learning. Phys Rev A 110:012605. https://doi.org/10.1103/PhysRevA.110.012605

Hamann A, Dunjko V, Wölk S (2021) Quantum-accessible reinforcement learning beyond strictly epochal environments. Quant Mach Intell 3:22. https://doi.org/10.1007/s42484-021-00049-7

Hamann A, Wölk S (2022) Performance analysis of a hybrid agent for quantum-accessible reinforcement learning. New J Phys 24:033044. https://doi.org/10.1088/1367-2630/ac5b56

Hämmerle A, Heindl C, Stübl G, Thapa J, Lamon E, Pichler A (2024) Applying grid world based reinforcement learning to real world collaborative transport. Procedia Comput Sci 232:388–396. https://doi.org/10.1016/j.procs.2024.01.038

Hohenfeld H, Heimann D, Wiebe F, Kirchner F (2024) Quantum deep reinforcement learning for robot navigation tasks. IEEE Access 12:87217–87236. https://doi.org/10.1109/ACCESS.2024.3417808

Huggins W, Patil P, Mitchell B, Whaley KB, Stoudenmire EM (2019) Towards quantum machine learning with tensor networks. Quant Sci Technol 4:024001. https://doi.org/10.1088/2058-9565/aaea94

Jerbi S, Gyurik C, Marshall SC, Briegel HJ, Dunjko V (2021) Parametrized quantum policies for reinforcement learning. In: Advances in neural information processing systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, Virtual event, pp 28362–28375. https://proceedings.neurips.cc/paper/2021/hash/eec96a7f788e88184c0e713456026f3f-Abstract.html

Kitaev AY (1995) Quantum measurements and the Abelian stabilizer problem. https://arxiv.org/abs/quant-ph/9511026

Kwiatkowski A, Towers M, Terry J, Balis JU, De Cola G, Deleu T, Goulão M, Kallinteris A, Krimmel M, KG A, Perez-Vicente R, Pierré A, Schulhoff S, Tai JJ, Tan H, Younis OG (2024) Gymnasium: a standard interface for reinforcement learning environments. arXiv:2407.17032

Lamata L (2017) Basic protocols in quantum reinforcement learning with superconducting circuits. Sci Rep 7:1609. https://doi.org/10.1038/s41598-017-01711-6

Li J-A, Dong D, Wei Z, Liu Y, Pan Y, Nori F, Zhang X (2020) Quantum reinforcement learning during human decision-making. Nat Hum Behav 4:294–307. https://doi.org/10.1038/s41562-019-0804-2

Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D (2019) Continuous control with deep reinforcement learning. arXiv:1509.02971

Lockwood O (2022) Optimizing quantum variational circuits with deep reinforcement learning. arXiv:2109.03188

Lockwood O, Si M (2020) Reinforcement learning with quantum variational circuits. In: Proceedings of the AAAI conference on artificial intelligence and interactive digital entertainment, Lexington, KY. https://doi.org/10.1609/aiide.v16i1.7437

Lockwood O, Si M (2021) Playing atari with hybrid quantum-classical reinforcement learning. In: NeurIPS 2020 Workshop on Pre-registration in Machine Learning, Virtual event. http://proceedings.mlr.press/v148/lockwood21a.html

Mandal D, Radanovic G, Gan J, Singla A, Majumdar R (2023) Online reinforcement learning with uncertain episode lengths. arXiv:2302.03608

Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D (2015) Human-level control through deep reinforcement learning. Nature 518:529–533. https://doi.org/10.1038/nature14236

Nautrup HP, Delfosse N, Dunjko V, Briegel HJ, Friis N (2019) Optimizing quantum error correction codes with reinforcement learning. Quantum 3:215. https://doi.org/10.22331/q-2019-12-16-215

Nielsen MA, Chuang IL (2010) Quantum computation and quantum information: 10th, Anniversary. Univ. Press, Cambridge, Camb. https://doi.org/10.1017/CBO9780511976667

Orús R (2014) A practical introduction to tensor networks: matrix product states and projected entangled pair states. Ann Phys 349:117–158. https://doi.org/10.1016/j.aop.2014.06.013

Ostaszewski M, Trenkwalder LM, Masarczyk W, Scerri E, Dunjko V (2021) Reinforcement learning for optimization of variational quantum circuit architectures. In: Advances in neural information processing systems 34: Annual conference on neural information processing systems 2021, NeurIPS 2021, Virtual event, pp 18182–18194. https://proceedings.neurips.cc/paper/2021/hash/9724412729185d53a2e3e7f889d9f057-Abstract.html

Paparo GD, Dunjko V, Makmal A, Martin-Delgado MA, Briegel HJ (2014) Quantum speedup for active learning agents. Phys Rev X 4:031002. https://doi.org/10.1103/PhysRevX.4.031002

Pólya G (1921) Über eine Aufgabe der Wahrscheinlichkeitsrechnung betreffend die Irrfahrt im Straßennetz. Math Ann 84:149–160. https://doi.org/10.1007/BF01458701

Preskill J (2018) Quantum computing in the NISQ era and beyond. Quantum 2:79. https://doi.org/10.22331/q-2018-08-06-79

Rapp F, Kreplin DA, Huber MF, Roth M (2025) Reinforcement learning-based architecture search for quantum machine learning. Mach Learn: Sci Technol 6:015041. https://doi.org/10.1088/2632-2153/adaf75

Ruiz FJR, Laakkonen T, Bausch J, Balog M, Barekatain M, Heras FJH, Novikov A, Fitzpatrick N, Romera-Paredes B, Wetering J, Fawzi A, Meichanetzidis K, Kohli P (2024) Quantum circuit optimization with AlphaTensor. arXiv:2402.14396

Saggio V, Asenbeck BE, Hamann A, Strömberg T, Schiansky P, Dunjko V, Friis N, Harris NC, Hochberg M, Englund D, Wölk S, Briegel HJ, Walther P (2021) Experimental quantum speed-up in reinforcement learning agents. Nature 591:229–233. https://doi.org/10.1038/s41586-021-03242-7

Schuld M, Killoran N (2022) Is quantum advantage the right goal for quantum machine learning? PRX Quantum 3:030101. https://doi.org/10.1103/PRXQuantum.3.030101

Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O (2017) Proximal policy optimization algorithms. arXiv:1707.06347

Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M (2014) Deterministic policy gradient algorithms. In: Proceedings of the 31st International Conference on Machine Learning, Beijing, China, pp 387–395. https://proceedings.mlr.press/v32/silver14.html

Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, Chen Y, Lillicrap T, Hui F, Sifre L, Driessche G, Graepel T, Hassabis D (2017) Mastering the game of go without human knowledge. Nature 550:354–359. https://doi.org/10.1038/nature24270

Sivak VV, Eickbusch A, Liu H, Royer B, Tsioutsios I, Devoret MH (2022) Model-free quantum control with reinforcement learning. Phys Rev X 12:011059. https://doi.org/10.1103/PhysRevX.12.011059

Sivak VV, Eickbusch A, Royer B, Singh S, Tsioutsios I, Ganjam S, Miano A, Brock BL, Ding AZ, Frunzio L, Girvin SM, Schoelkopf RJ, Devoret MH (2023) Real-time quantum error correction beyond break-even. Nature 616:50–55. https://doi.org/10.1038/s41586-023-05782-6

Skolik A, Jerbi S, Dunjko V (2022) Quantum agents in the gym: a variational quantum algorithm for deep Q-learning. Quantum 6:720. https://doi.org/10.22331/q-2022-05-24-720

Sriarunothai T, Wölk S, Giri GS, Friis N, Dunjko V, Briegel HJ, Wunderlich C (2018) Speeding-up the decision making of a learning agent using an ion trap quantum processor. Quant Sci Technol 4:015014. https://doi.org/10.1088/2058-9565/aaef5e

Sutton RS, Barto AG (2018) Reinforcement learning: an introduction. 2nd Edition. A Bradford Book, Cambridge. http://incompleteideas.net/book/the-book-2nd.html

Watkins CJCH, Dayan P (1992) Q-learning. Mach Learn 8:279–292. https://doi.org/10.1007/BF00992698

Wiedemann S, Hein D, Udluft S, Mendl C (2023) Quantum policy iteration via amplitude estimation and Grover search – towards quantum advantage for reinforcement learning. arXiv:2206.04741

Yu H, Zhao X, Chen C (2025) Quantum-inspired reinforcement learning for quantum control. IEEE Trans Control Syst Technol 33:61–76. https://doi.org/10.1109/TCST.2024.3437142

Zhong H, Hu J, Xue Y, Li T, Wang L (2024) Provably efficient exploration in quantum reinforcement learning with logarithmic worst-case regret. arXiv:2302.10796