Quantum Policy Iteration via Amplitude Estimation and Grover Search – Towards Quantum Advantage for Reinforcement Learning

Anonymous authors
Paper under double-blind review

Abstract

We present a full implementation and simulation of a novel quantum reinforcement learning (RL) method. Our approach shows in detail how to combine amplitude estimation and Grover search into a policy evaluation and improvement scheme. We first develop quantum policy evaluation (QPE) which is quadratically more efficient compared to an analogous classical Monte Carlo (MC) estimation and is based on a quantum mechanical realization of a finite Markov decision process (MDP). Building on QPE, we derive a quantum policy iteration that repeatedly improves an initial policy using Grover search until the optimum is reached. Finally, we present an implementation of our algorithm for a two-armed bandit MDP which we then simulate. Our work is a detailed and formal proof of concept for how quantum algorithms can be used to solve RL problems and shows that they can yield provable improvements over classical MC based methods in terms of sample complexity.

1 Introduction

Successful reinforcement learning (RL) algorithms for challenging tasks come at the cost of computationally expensive training (Silver et al., 2016). Since quantum computers can perform certain complex tasks much faster than classical machines (Grover, 1996; Shor, 1999) a hope is that quantum computers could also enable more efficient RL algorithms. Today, there is theoretical and practical evidence that quantum interaction between agent and environment can indeed reduce the time it takes for the agent to learn (Section 2).

Herein, we describe an optimization-based quantum algorithm for RL that makes use of a direct quantum mechanical realization of a finite MDP, in which agent and environment are modelled by unitary operators and exchange states, actions and rewards in superposition (Section 3.1). Using this model, we show in detail how to use amplitude estimation to estimate the value function of a policy in a finite MDP with finite horizon (Section 3.2). This quantum policy evaluation (QPE) algorithm can be compared to a classical Monte Carlo (MC) method, i.e., sampling from agent-environment interactions to estimate the value function. QPE uses *quantum samples* (qsamples) instead of classical ones and provides a quantum advantage over any possible classical MC approach; it needs quadratically fewer qsamples to estimate the value function of a policy up to a given precision. We embed QPE into an optimization scheme that uses repeated Grover-like searches of the set of policies to find a decision rule with ϵ-optimal performance (Section 3.3). The resulting routine shares similarities with the policy iteration method from *dynamic programming* (DP) (Sutton & Barto, 2018) and, thus, we call it *quantum (approximate) policy iteration.* A major contribution of our work is a detailed description of how to implement quantum policy iteration for a two-armed bandit MDP at the level of standard single and multi qubit gates on a digital quantum computer. Simulations of this implementation numerically confirm the mathematically proven quantum advantage of QPE over classical MC policy evaluation and illustrate the behavior of quantum policy iteration (Section 4).

2 Related Work

Most existing quantum RL approaches can be roughly divided into two categories: quantum enhanced agents that learn in classical environments, and scenarios where the agent and environment can interact quantum mechanically.

An example for the first category, is *quantum projective simulation* (Briegel & De las Cuevas, 2012), where an agent makes decisions using a quantum random walk on a learned graph that represents its internal memory. In this approach, the special properties of quantum walks allow for fast decision making and thus less active learning time than classical agents. Furthermore, various quantum-enhanced agents have been described that learn in classical environments by replacing components of classical learning algorithms with quantum counterparts. There are approaches for *deep Q-learning* and *policy gradient* methods, where classical artificial neural networks are replaced by analogous *variational quantum circuits* (VQCs) with trainable parameters (Chen et al., 2020; Lockwood & Si, 2020; Moll & Kunczik, 2021; Skolik et al., 2021; Jerbi et al., 2021).

Theoretical work by Dunjko et al. (2015) indicates that the possibility of quantum interaction between agent and environment can be leveraged to design quantum-enhanced agents that provably outperform the best possible classical learner in certain environments (Dunjko et al., 2015; 2016; 2017a). One such approach is related to our ideas and uses Grover searches in deterministic environments to find sequences of actions that lead to high rewards (Dunjko et al., 2016). The possibility of using amplitude estimation similar to QPE to be applied to stochastic environments are also discussed. Further extensions were proposed by Dunjko et al. (2017b) and Hamann et al. (2020).

More recently, quantum RL methods based on classical DP were proposed that require quantum access to a classical environment similar to our environment operator **E** from Section 3.1. Wang et al. (2021) used similar tools as in this work (i.e., amplitude estimation and quantum optimization) to obtain quantum speedups for subroutines of otherwise classical algorithms. Cherrat et al. (2022) developed a quantum policy iteration algorithm in which the Bellman equations for policy evaluation are solved using a quantum method for solving systems of linear equations.

Contrary to the methods by Wang et al. (2021) and Cherrat et al. (2022), our novel quantum (approximate) policy iteration is not a quantum-enhanced version of an existing classical DP method and uses no classical subroutines. It also differs from the ideas of Dunjko et al. (2016) as it directly searches for the optimal policy instead of looking for rewarded actions that are then used to improve the decision rule.

3 Quantum (Approximate) Policy Iteration

3.1 Quantum Mechanical Realization of a Finite Markov Decision Process

Consider an MDP with finitely many states $\mathtt{S}$, actions $\mathtt{A}$ and rewards $\mathtt{R}$. We can identify these sets with finite-dimensional complex Hilbert spaces $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$, which we call *state, action* and *reward spaces.* Formally, we use maps from the sets to the corresponding spaces, i.e.,

$$s \mapsto |s\rangle\,, \qquad a \mapsto |a\rangle\,, \qquad r \mapsto |r\rangle\,. \tag{1}$$

In order to be able to distinguish quantum states, actions and rewards we require that

$$\langle s|s'\rangle = \delta_{s,s'}, \quad \langle a|a'\rangle = \delta_{a,a'}, \quad \langle r|r'\rangle = \delta_{r,r'}, \tag{2}$$

holds for all pairs $s, s' \in \mathtt{S}$, $a, a' \in \mathtt{A}$ and $r, r' \in \mathtt{R}$. On a digital quantum computer, we can for example arbitrarily enumerate all states and actions and use strings of qubits as quantum representatives, i.e., $s_i \mapsto |\mathrm{bin}(i)\rangle$ and $a_j \mapsto |\mathrm{bin}(j)\rangle$. The rewards can be approximated by fixed point binaries which can also be represented as strings of qubits.

Agent and environment are modelled by unitary operators that act on $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$. For any policy π, we define the *policy operator* as a unitary operator $\mathbf{\Pi} \in U\big(\mathcal{S} \otimes \mathcal{A}\big)$ which satisfies

$$|s\rangle\,|0\rangle_{\mathcal{A}} \overset{\mathbf{\Pi}}{\longmapsto} |\pi_s\rangle = \sum_{a \in \mathtt{A}} c_{a|s}\,|s\rangle\,|a\rangle\,, \tag{3}$$

for all states $s \in \mathtt{S}$. The state $|0\rangle_{\mathcal{A}}$ is an arbitrary reference state in $\mathcal{A}$ and each amplitude $c_{a|s} \in \mathbb{C}$ has to satisfy $|c_{a|s}|^2 = \pi(a|s)$. The policy operator $\mathbf{\Pi}$ can be chosen as any unitary that satisfies these conditions. To see why such a unitary operator indeed exists, note that (3) describes a bijection between two orthonormal bases (ONBs) of two subspaces of $\mathcal{S} \otimes \mathcal{A}$. We can extend both ONBs to ONBs of the full space and define an arbitrary bijection between the newly added basis states while maintaining (3) on the original ones. The linear extension of this assignment is then unitary by construction. Using an analogous argument, we define the *environment operator* $\mathbf{E} \in U\big(\mathcal{S} \otimes \mathcal{A} \otimes \mathcal{R} \otimes \mathcal{S}\big)$ as any unitary operator that satisfies

$$|s\rangle\,|a\rangle\,|0\rangle_{\mathcal{R}}\,|0\rangle_{\mathcal{S}} \overset{\mathbf{E}}{\longmapsto} \sum_{r,s'} c_{r,s'|s,a}\,|s\rangle\,|a\rangle\,|r\rangle\,|s'\rangle\,, \tag{4}$$

with amplitudes that satisfy $|c_{r,s'|s,a}|^2 = p(r,s'|s,a)$ for all state-action pairs $s \in \mathtt{S}$, $a \in \mathtt{A}$. A single interaction between agent and environment is modelled by the *step operator* $\mathbf{S} := \mathbf{E} \circ \mathbf{\Pi}$. For any state $s \in \mathtt{S}$, it holds that

$$|s\rangle\,|0\rangle_{\mathcal{A}}\,|0\rangle_{\mathcal{R}}\,|0\rangle_{\mathcal{S}} \overset{\mathbf{S}}{\longmapsto} \sum_{a,r,s'} c_{a,r,s'|s}\,|s\rangle\,|a\rangle\,|r\rangle\,|s'\rangle\,, \tag{5}$$

where $c_{a,r,s'|s} = c_{r,s'|s,a}c_{a|s}$. We use $\mathbf{S}$ to construct a unitary operator that prepares a quantum state which represents the distribution of all trajectories with a fixed finite horizon H. We call it *MDP operator* and denote it as $\mathbf{M}$. While the step operator acts on $\mathcal{S} \otimes \mathcal{A} \otimes \mathcal{R} \otimes \mathcal{S}$, the MDP operator is a unitary on the *trajectory space* $\mathcal{T}^H := \mathcal{S} \otimes \big(\mathcal{A} \otimes \mathcal{R} \otimes \mathcal{S}\big)^{\otimes H}$ which is large enough to store H quantum agent-environment interactions. The states in this space represent trajectories of length H and are of type

$$|t^H\rangle = |s_0^{t^H}\rangle\,|a_0^{t^H}\rangle\,|r_1^{t^H}\rangle\,|s_1^{t^H}\rangle \cdots |r_H^{t^H}\rangle\,|s_H^{t^H}\rangle\,. \tag{6}$$

We define the MDP operator as

$$\mathbf{M} := \prod_{h=1}^{H} \mathbf{S}_h \in U\big(\mathcal{T}^H\big), \tag{7}$$

where $\mathbf{S}_h$ denotes a local version of $\mathbf{S}$ that acts on the h-th $\mathcal{S} \otimes \mathcal{A} \otimes \mathcal{R} \otimes \mathcal{S}$ subsystem of $\mathcal{T}^H$. It holds that

$$|s\rangle \otimes \big(\,|0\rangle_{\mathcal{A}}\,|0\rangle_{\mathcal{R}}\,|0\rangle_{\mathcal{S}}\,\big)^{\otimes H} \overset{\mathbf{M}}{\longmapsto} \sum_{t^H} c_{t^H}\,|t^H\rangle\,, \tag{8}$$

where $|c_{t^H}|^2 = p(t^H)$ which is the classical probability of trajectory t^H. This can be seen by inductive application of (5). If we measure the state from (8), we observe a trajectory t^H (or more precisely its quantum analogue $|t^H\rangle$) with probability $p(t^H)$. Therefore, this state is a quantum version of the distribution of all trajectories and we refer to it as qsample of the trajectories.

For QPE, we need the distribution of the returns in order to approximate the value function. With this in mind, we use the qsample of the trajectories and calculate for each trajectory state $|t^H\rangle$ the associated return defined as $G(t^H) := \sum_{h=1}^{H} \gamma^{h-1} r_h$, where $\gamma \in [0,1]$ is a fixed discount factor. As the reward states of $|t^H\rangle$ are qubit binary encodings of (real) numbers, we can use quantum arithmetic to calculate their discounted sum. Formally, we use a unitary *return operator* $\mathbf{G}$ that maps

$$|r_1\rangle \cdots |r_H\rangle\,|0\rangle_{\mathcal{G}} \overset{\mathbf{G}}{\longmapsto} |r_1\rangle \cdots |r_H\rangle\,\left|\sum_{h=1}^{H} \gamma^{h-1} r_h\right\rangle, \tag{9}$$

where $|0\rangle_{\mathcal{G}}$ is a reference state in the *return space* $\mathcal{G}$, which represents a system of sufficiently many qubits to encode all returns. Ruiz-Perez & Garcia-Escartin (2017) show an explicit construction of such an operator which performs the weighted addition in the Fourier domain. By letting $\mathbf{G}$ act on all reward subsystems and the return component of $\mathcal{T}^H \otimes \mathcal{G}$, we can define $\mathbf{G} \circ \mathbf{M}$ which satisfies

$$|0\rangle \overset{\mathbf{G}\circ\mathbf{M}}{\longmapsto} |\psi\rangle := \sum_{t^H} c_{t^H}\,|t^H\rangle\,|G(t^H)\rangle\,. \tag{10}$$

As all $|t^H\rangle\,|G(t^H)\rangle$ states are (by construction) orthonormal, it follows that if we measure the second subsystem of $|\psi\rangle$, we receive an individual return with the classical probability determined by the MDP and the agent's policy. $|\psi\rangle$ can be thought of as qsample of the return.

3.2 Quantum (Approximate) Policy Evaluation

Consider the problem of (approximately) evaluating the value function v_π^H of a policy π for some finite horizon $H \in \mathbb{N}$. The value function is defined pointwise via

$$v_\pi^H(s) = \mathbb{E}_{t^H}[G(t^H)|S_0 = s], \tag{11}$$

where $s \in \mathtt{S}$ is the initial state, and the expectation is taken with respect to the distribution of all trajectories of length H. Even if we assume perfect knowledge of the MDP dynamics p, evaluating $v_\pi^H(s)$ exactly is usually infeasible: the complexity of directly calculating the expectation grows exponentially in the horizon H. Techniques from DP such as *iterative policy evaluation* reduce the computational complexity but come at the cost of high memory consumption (Sutton & Barto, 2018). An alternative approach is to use an MC method to estimate $v_\pi^H(s)$. The arguably most straightforward MC algorithm for policy evaluation is to collect a dataset of trajectories and to average the returns. This requires the possibility to sample from the distribution of the trajectories.

Montanaro (2015) developed a quantum algorithm to estimate the expectation of a random variable using qsamples. The method is quadratically more sample efficient than the best possible classical MC routine and it forms the foundation of QPE. To approximate the expected return, we use qsamples of the returns to which we have access via the operator $\mathbf{A}_{\mathrm{QPE}} := \mathbf{G} \circ \mathbf{M}$ from (10). The first step towards QPE is to note that we can encode $v_\pi^H(s)$ as amplitude of a basis vector in a quantum superposition. To this end, assume that we know a lower bound $\underline{g}$ and an upper bound $\bar{g} \neq \underline{g}$ on all returns. Using an affine function ϕ defined by $\phi(x) = (x - \underline{g})/)(\bar{g} - \underline{g})$, we construct a unitary operator $\mathbf{\Phi}$ that maps

$$|x\rangle\,|0\rangle \overset{\mathbf{\Phi}}{\longmapsto} |x\rangle \Big(\sqrt{1-\phi(x)}\,|0\rangle + \sqrt{\phi(x)}\,|1\rangle\Big), \tag{12}$$

where $|x\rangle$ is a binary (qubit) representation of a real number $x \in \mathbb{R}$ and the second subsystem is a single qubit. If we now apply $\mathbf{\Phi}$ to the return component of $|\psi\rangle$ from (10) and an additional qubit, we get

$$|\psi\rangle \overset{\mathbf{\Phi}}{\longmapsto} |\psi^\phi\rangle := c_0\,|\psi_0^\phi\rangle + c_1\,|\psi_1^\phi\rangle\,, \tag{13}$$

where

$$|\psi_1^\phi\rangle \propto \sum_{t^H} c_{t^H} \sqrt{\phi\big(G(t^H)\big)}\,|t^H\rangle\,|G(t^H)\rangle\,|1\rangle\,, \tag{14}$$

The state $|\psi_0^\phi\rangle$ contains all the states in which the last qubit is in state $|0\rangle$ and is therefore orthonormal to $|\psi_1^\phi\rangle$. The amplitude $c_1 \in \mathbb{C}$, which is the norm of the (non-normalized) state on the right-hand side of (14), satisfies

$$|c_1|^2 = \sum_{t^H} p(t^H)\phi\big(G(t^H)\big) = \mathbb{E}_{t^H}\Big[\phi\big(G(t^H)\big)\Big]. \tag{15}$$

By affinity of ϕ, it holds that

$$\phi^{-1}\big(|c_1|^2\big) = \mathbb{E}_{t^H}\big[G(t^H)\big] = v_\pi^H(s). \tag{16}$$

Therefore, approximately evaluating the value function can be done by estimating $|c_1|^2$ for which we can use an *amplitude estimation* algorithm.

One way to estimate $|c_1|^2$ (or any amplitude in general) is to encode it via the phases of two eigenvalues of a unitary operator and to estimate them using *phase estimation*. Following the original construction of Brassard et al. (2002), we define this unitary as

$$\mathbf{Q}_{\mathrm{QPE}} := -\mathbf{A}_{\mathrm{QPE}}^\phi \circ \mathbf{S}_0 \circ \big(\mathbf{A}_{\mathrm{QPE}}^\phi\big)^\dagger \circ \big(\mathbf{id}_{\mathcal{T}^H \otimes \mathcal{G}} \circ \mathbf{Z}\big), \tag{17}$$

where $\mathbf{A}_{\mathrm{QPE}}^\phi := \big(\mathbf{id}_{\mathcal{T}^H} \otimes \mathbf{\Phi}\big) \circ \mathbf{A}_{\mathrm{QPE}}$. In this expression, $\mathbf{\Phi}$ acts on the return subsystem and the ancillary qubit. The operator $\mathbf{S}_0$ is a *phase oracle* that, considering the basis states, flips the phase of a state precisely when all components (in this case the trajectory and the return component as well as the additional qubit) are in the corresponding $|0\rangle$ ground states. The $\mathbf{Z}$ operator corresponds to a Pauli-Z gate and therefore the

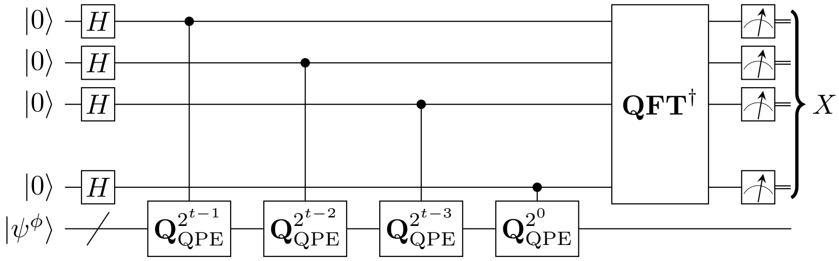
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|0\rangle$
$|\psi^\phi\rangle$
H
$\mathbf{Q}_{\mathrm{QPE}}^{2^{t-1}}$
$\mathbf{Q}_{\mathrm{QPE}}^{2^{t-2}}$
$\mathbf{Q}_{\mathrm{QPE}}^{2^{t-3}}$
$\mathbf{Q}_{\mathrm{QPE}}^{2^{0}}$
$\mathbf{QFT}^\dagger$
X

Figure 1: Phase estimation circuit for amplitude estimation in QPE.

last operator is also a phase oracle which flips the phase of a state precisely when the ancillary qubit is in state $|1\rangle$. Brassard et al. (2002) showed that $|\psi^\phi\rangle$ can be decomposed as $|\psi^\phi\rangle = c_+ |\psi^\phi_+\rangle + c_- |\psi^\phi_-\rangle$ where $|\psi^\phi_+\rangle$ and $|\psi^\phi_-\rangle$ are orthonormal, $|c_+|^2 = |c_-|^2 = 1/2$ and $\mathbf{Q}_{\mathrm{QPE}} |\psi^\phi_\pm\rangle = e^{\pm 2\phi i\theta} |\psi^\phi_\pm\rangle$, which means that $|\psi^\phi_\pm\rangle$ are eigenstates of $\mathbf{Q}_{\mathrm{QPE}}$. The phase θ is the unique value in $[0, 1)$ that satisfies $\sin^2(\pi\theta) = \sin^2(-\pi\theta) = |c_1|^2$.

Therefore, we can use a *phase estimation* algorithm to estimate θ and $-\theta$. The classical phase estimation routine shown in Figure 1 uses a system of

$$t := n + \left\lceil \log_2\left(\frac{1}{2\delta} + \frac{1}{2}\right)\right\rceil, \tag{18}$$

qubits where $n \in \mathbb{N}$ and $\delta \in (0, 1]$ are arbitrary. The basis states of these qubits are interpreted as binary numbers. Phase estimation exploits the fact that $|\psi^\phi_\pm\rangle$ are eigenstates of $\mathbf{Q}_{\mathrm{QPE}}$ to bring the t qubits into a joint state X which, when measured, satisfies

$$P\Big(|X/2^t - \theta| \le 1/2^{n+1}\Big) \ge 1/2 - \delta/2, \tag{19}$$

$$P\Big(|X/2^t + \theta| \le 1/2^{n+1}\Big) \ge 1/2 - \delta/2, \tag{20}$$

i.e., the distribution of $X/2^t$ is concentrated at θ and $-\theta$. In case $t = n$, we get the same result with total probability at least $8/\pi^2$. These bounds follow from a detailed analysis of the phase estimation circuit as was done for example by Cleve et al. (1998).

As $|c_1|^2 = \sin^2(\pi\theta)$ and $v^H_\pi(s) = \phi^{-1}(|c_1|^2)$, we define the QPE approximation $\tilde{v}^H_\pi(s)$ of $v^H_\pi(s)$ for any realization x of X as

$$\tilde{v}^H_\pi(s) := \phi^{-1}\Big(\sin^2\left(\pi x/2^t\right)\Big). \tag{21}$$

The approximation error of θ given in (19) can be translated to an approximation error of the value function

$$P\big(|\tilde{v}^H_\pi(s) - v^H_\pi(s)| \le \epsilon\big) \ge 1 - \delta, \tag{22}$$

where the error ϵ is (as shown in the Appendix) given by

$$\epsilon = (\bar{g} - \underline{g})\big(\pi/2^{n+1} + \pi^2/2^{2n+2}\big) \in \mathcal{O}(1/2^n). \tag{23}$$

Phase estimation uses $\mathcal{O}(2^t) = \mathcal{O}(1/\epsilon \cdot 1/\delta)$ applications of $\mathbf{A}^\phi_{\mathrm{QPE}}$ and $\big(\mathbf{A}^\phi_{\mathrm{QPE}}\big)^\dagger$ (via $\mathbf{Q}_{\mathrm{QPE}}$) to achieve this error bound (c.f. Figure 1). Each such application corresponds to the collection of one qsample. According to the optimality results on MC methods due to Dagum et al. (2000), the sample efficiency of the best possible classical MC type algorithm to estimate $v^H_\pi(s)$ via averaging has sample complexity in $\Omega(1/\epsilon^2 \cdot \log_2(1/\delta))$. Therefore, QPE yields a quadratic reduction of the sample complexity w.r.t. to the approximation error ϵ. This quantum advantage holds over the best possible classical MC approach for policy evaluation. An advantage over other approaches is not guaranteed. Comparing the sample complexity of QPE to those of other classical methods is an interesting direction for future research.

3.3 Quantum Policy Improvement

The task of improving a given policy μ is to find another policy μ' such that $v^H_{\mu'}(s) \geq v^H_\mu(s)$ holds for all $s \in \mathtt{S}$ and that strict inequality holds for at least one $s' \in \mathtt{S}$. Once we know the value function v^H_μ of μ, such a μ' can be explicitly constructed via *greedification.* In this section, we derive *quantum policy improvement* (QPI) which instead uses a Grover search over a set of policies to find an improved decision rule. For simplicity, we assume from now on that the MDP always starts in some fixed initial state $s_0 \in \mathtt{S}$. In this case, instead of using the whole value function, we can restrict ourselves to the *value* of policy π which, under abuse of notation, we define as $v^H_\pi := v^H_\pi(s_0)$.

QPI uses Grover search on QPE results to find an improved decision rule. For a fixed policy π, QPE can be summarized as one unitary **QPE** (consisting of the phase estimation circuit from Figure 1 without the measurements) that maps $|0\rangle \xmapsto{\mathbf{QPE}} |\psi^\pi_{\mathrm{QPE}}\rangle$. Now consider the finite set $\mathtt{P}$ that contains all deterministic policies. By mapping each $\pi \in \mathtt{P}$ onto a member $|\pi\rangle$ of an ONB of some suitable Hilbert space $\mathcal{P}$, we can construct a unitary, π-controlled version $\mathbf{QPE}_\pi$ of **QPE** that satisfies

$$|\pi'\rangle |0\rangle \xmapsto{\mathbf{QPE}_\pi} \begin{cases} |\pi\rangle |\psi^\pi_{\mathrm{QPE}}\rangle & \text{if } \pi' = \pi \\ |\pi'\rangle |0\rangle & \text{else} \end{cases}, \tag{24}$$

for all policies $\pi' \in \mathtt{P}$. Using these operators, we define another unitary that prepares the search state for QPI Grover search. Consider

$$\mathbf{A}_{\mathrm{QPI}} := \left(\prod_{\pi \in \mathtt{P}} \mathbf{QPE}_\pi \right) \circ \big(\mathbf{H}_\mathcal{P} \otimes \mathbf{id}\big), \tag{25}$$

where $\mathbf{H}_\mathcal{P}$ is a Hadamard transform on $\mathcal{P}$. By construction, this operator maps

$$|0\rangle_\mathcal{P} |0\rangle \xmapsto{\mathbf{A}_{\mathrm{QPE}}} \frac{1}{\sqrt{|\mathtt{P}|}} \sum_{\pi \in \mathtt{P}} |\pi\rangle |\psi^\pi_{\mathrm{QPE}}\rangle . \tag{26}$$

As the $|\psi^\pi_{\mathrm{QPE}}\rangle$ states encode the value of policy π, we can define a suitable oracle and use it to amplify the amplitudes of those policies that are likely to yield higher returns. For a policy μ and an approximation $\tilde{v}_\mu$ of its value, consider a phase oracle $\mathbf{O}_{>\tilde{v}_\mu}$ that satisfies

$$|x\rangle \xmapsto{\mathbf{O}_{>\tilde{v}_\mu}} \begin{cases} -|x\rangle & \text{if } \phi^{-1}\big(\sin^2(\pi x/2^t)\big) > \tilde{v}_\mu \\ |x\rangle & \text{else} \end{cases}. \tag{27}$$

Using this oracle, we define the QPI Grover operator as

$$\mathbf{Q}^{\tilde{v}_\mu}_{\mathrm{QPI}} := -\mathbf{A}_{\mathrm{QPI}} \circ \mathbf{S}_0 \circ \mathbf{A}^\dagger_{\mathrm{QPI}} \circ \big(\mathbf{id}_\mathcal{P} \otimes \mathbf{O}_{>\tilde{v}_\mu}\big). \tag{28}$$

According to the principle of *amplitude amplification*, which is used in Grover search and was described in detail by Brassard et al. (2002), if we apply this operator a certain number of times to the search state prepared by $\mathbf{A}_{\mathrm{QPI}}$, this amplifies the amplitude of all $|\pi\rangle |x\rangle$ states whose $|x\rangle$ component satisfies the oracle condition $\phi^{-1}\big(\sin^2(\pi x/2^t)\big) > \tilde{v}_\mu$. As a result, the probability of measuring a policy that has better performance than $\tilde{v}_\mu$ (up to an error of ϵ) is increased. Due to the QPE failure probability of δ, the procedure may also amplify amplitudes of policies that do not yield an improvement (within ϵ) as QPE may overestimate their performance. However, the probability of such errors can be limited by choosing small values for δ.

For the amplitude amplification to work, we need to specify the number of *Grover rotations*, i.e., the number of times we apply $\mathbf{Q}^{\tilde{v}_\pi}_{\mathrm{QPE}}$, which determines how the amplitudes of the desired states are scaled. Although there is a theoretically optimal number of Grover rotations, this value is inaccessible in QPI as it would require knowledge of the probability of obtaining a desired state when measuring the search state (Brassard et al., 2002). To circumvent this problem, we use the *exponential Grover search* strategy introduced by Boyer et al. (1998): We initialize a parameter $m = 1$. In each iteration, we uniformly sample the Grover rotations

Algorithm 1 Quantum Policy Iteration

```
input: A policy $\pi_0 \in \texttt{P}$, an estimate $\tilde{v}_{\pi_0}$ of its value, parameters for QPE and QPI
output: Guesses $\tilde{\pi}^* \in \texttt{P}$ and $\tilde{v}^*$ for $\pi^*$ and $v^*$
$c \leftarrow 0$
for k=1,2,3,... while $c \leq C$ do
    Run QPI on $\pi_{k-1}$, $\tilde{v}_{\pi_{k-1}}$ to obtain $\tilde{\pi}_k$, $\tilde{v}_k$
    if $\tilde{v}_{\pi_k} > \tilde{v}_{\pi_{k-1}}$ then
        $\pi_k \leftarrow \tilde{\pi}_k$, $v_k \leftarrow \tilde{v}_k$, $c \leftarrow 0$
    else
        $\pi_k \leftarrow \pi_{k-1}$, $v_k \leftarrow v_{k-1}$, $c \leftarrow c + 1$
    end if
end for
return $\tilde{\pi}^* = \pi_T$, $\tilde{v}^* = v_T$ from last iteration $T$
```

from $\{0, ..., \lceil m-1 \rceil\}$ and measure the resulting state. If the result corresponds to an improvement, we reset $m \leftarrow 1$. Otherwise, we overwrite $m \leftarrow \lambda m$ for some $\lambda > 1$ which stays the same over all iterations. For a theoretical justification of this technique, please refer to the original work of Boyer et al. (1998)

Quantum (approximate) policy iteration as described in Algorithm 1 starts with an initial policy π_0 and repeatedly applies QPI to generate a sequence of policies with increasing values. Note that the procedure is similar to a quantum minimization algorithm introduced by Dürr & Høyer (1996). In each iteration k, quantum policy iteration runs QPI which is a Grover search and can therefore also fail to produce a policy whose estimated value exceeds the current best v_k. In this case, the policy and its value are not updated. As we typically do not know the value of the optimal policy, the algorithm uses a patience criterion to interrupt the iteration if there was no improvement in the last C steps.

As quantum policy iteration relies on QPE, it can only find a policy with ϵ-optimal behavior and may also fail to do so. This is because QPE yields an ϵ-approximation of the value only with probability $1 - \delta$. We conjecture that the failure probability of quantum policy iteration can be made arbitrarily small by choosing a sufficiently small δ which, however, comes at the cost of increasing the qubit complexity and run-time of QPE. Moreover, note that if in some iteration k the QPE output v_k overestimates v_{π_k} but still $v_k < v_*$, QPI may nevertheless find a policy π_{k+1} with $v_{\pi_{k+1}} > v_k$. This mitigates QPE errors.

We propose that the complexity of quantum policy iteration is best measured in terms of the total number of Grover rotations performed in all QPI steps. This number is proportional to the runtime of the procedure and also measures the number of times we ran QPE which relates via the MDP model to the total number of quantum agent-environment interactions. Therefore, the number of Grover rotations also measures the qsample complexity. We conjecture that the complexity grows linear in $\sqrt{|\texttt{P}|}$. Our intuition behind this is that quantum policy iteration is essentially Grover search except that it uses changing, inaccurate oracles.

4 Experiments

Existing quantum hardware is not yet ready to run QPE, let alone quantum policy iteration. Both methods use far more gates than what is possible with existing, non-error-corrected digital quantum computers. Therefore, we have to resort to simulations. However, as quantum circuits are known to be hard to simulate classically, the complexity of the RL problems we can consider is limited. Simulating QPE alone requires processing a statevector whose size grows exponentially with the horizon, which is infeasible for large environments that require many interactions. Considering these facts, we resorted to a two-armed bandit problem. As we will see below, this setup is too simple to use it to fully validate our methods. Rather than that, the idea behind the simulation study in this section is to illustrate the steps needed to realize our algorithms on real quantum hardware in full detail and to demonstrate their behavior.

In this setup, the environment is a slot machine with two arms (levers). Upon pulling an arm, one receives a reward of either 0$ or 1$. This translates to an MDP with one state and two actions, i.e., $\texttt{A} = \{\leftarrow, \rightarrow\}$

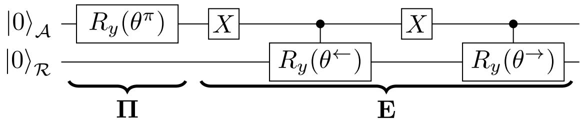
$|0\rangle_{\mathcal{A}}$
$|0\rangle_{\mathcal{R}}$
$R_y(\theta^\pi)$
X
$R_y(\theta^{\leftarrow})$
X
$R_y(\theta^{\rightarrow})$
$\mathbf{\Pi}$
$\mathbf{E}$

Figure 2: The step operator $\mathbf{S}$ for the two-armed bandit MDP in the form of a quantum circuit.

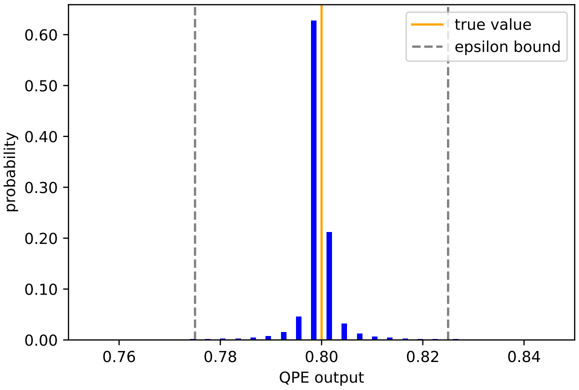

Figure 3: Distribution of the $\tilde{v}_\pi^2$ output of QPE for the concrete instance of the two-armed bandit MDP.

where "$\leftarrow$" means "pull the left arm" and "$\rightarrow$" means "pull the right arm". The reward set is $\mathtt{R} = \{0, 1\}$. The MDP dynamics are determined by the two probabilities $p(0|\leftarrow)$ and $p(0|\rightarrow)$ of losing, i.e., "winning" 0\$, when pulling the left or right arm. Each policy is determined by $\pi(\leftarrow)$ which denotes the probability of choosing the left arm. The learning problem is to identify the arm that yields the highest value v^*, given by $v^* = \max\big\{p(1|\leftarrow), p(1|\rightarrow)\big\}$.

We encode the actions via $|\leftarrow\rangle = |0\rangle_{\mathcal{A}}$ and $|\rightarrow\rangle = |1\rangle_{\mathcal{A}}$ which requires a single qubit. We use another qubit to encode the rewards as $|0\$\rangle = |0\rangle_{\mathcal{R}}$ and $|1\$\rangle = |1\rangle_{\mathcal{R}}$. In this encoding, we can represent the step operator $\mathbf{S}$ as a quantum circuit shown in Figure 2. The angles of the Y-axis rotations R_y are given by $\theta^\pi = 2\arccos\left(\sqrt{\pi(\leftarrow)}\right)$, $\theta^{\leftarrow} = 2\arccos\left(\sqrt{p(0|\leftarrow)}\right)$ and $\theta^{\rightarrow} = 2\arccos\left(\sqrt{p(0|\rightarrow)}\right)$. A direct calculation shows that this $\mathbf{S}$ indeed prepares a qsample of one agent-environment interaction.

Now consider the two-armed bandit for horizon $H = 2$, i.e., the agent gets to play two rounds. The MDP operator of this decision process is given by two step operators according to Figure 2 that act on two separate $\mathcal{A} \otimes \mathcal{R}$ subsystems represented by four qubits. For simplicity, we set the discount factor $\gamma = 1$. The return operator $\mathbf{G}$ adds the two reward qubits and stores the result using another two-qubit subsystem that represents the return space $\mathcal{G}$. It can be implemented using two CNOT gates and one Toffoli gate to realize the logical expressions that determine which qubit of the return register must be set to $|1\rangle$ or $|0\rangle$ depending on the rewards. $\mathbf{\Phi}$ can be realized using three doubly-controlled R_y gates that rotate another ancillary qubit depending on the two reward qubits.

4.1 Simulations of Quantum Policy Evaluation

Putting all of the above together, we receive a gate-decomposition of $\mathbf{A}^\phi_{\text{QPE}}$. To simulate QPE, we implemented the operator in IBM's `qiskit` (2021) framework for the programming language `Python`. We chose a concrete bandit with dynamics $p(0|\leftarrow) = 0.55$ and $p(0|\rightarrow) = 0.65$ and considered the policy π given by $\pi(\leftarrow) = 0.50$ which has the value $v_\pi^2 = 0.80$. All of these values were chosen arbitrarily. We used simulated QPE to obtain an estimate of this value with absolute error of at most $\epsilon = 0.025$. As maximum error probability, we chose $\delta = 0.05$. According to (18) and (23), we have to use $t = 7 + 4 = 11$ qubits in the first register of the phase estimation circuit to achieve the ϵ-approximation with probability at least $1 - \delta$. The distribution of the QPE outputs for these parameters is shown in Figure 3. `Qiskit` offers the possibility to calculate measurement probabilities analytically, which is what we used to generate the plot. The probability distribution of the QPE output $\tilde{v}_\pi^2$ is concentrated at the true value at $v_\pi^2 = 0.80$ and the mass that lies outside the ϵ region is less than 0.025 which is even better than our guaranteed error probability bound of $\delta = 0.05$.

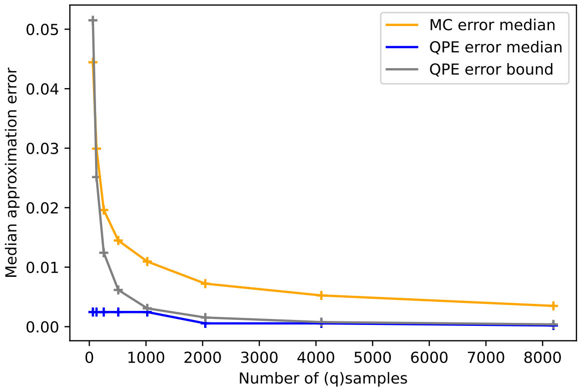

Figure 4: Median approximation errors of 1000 runs of both QPE and classical MC for one round of two-armed bandit.

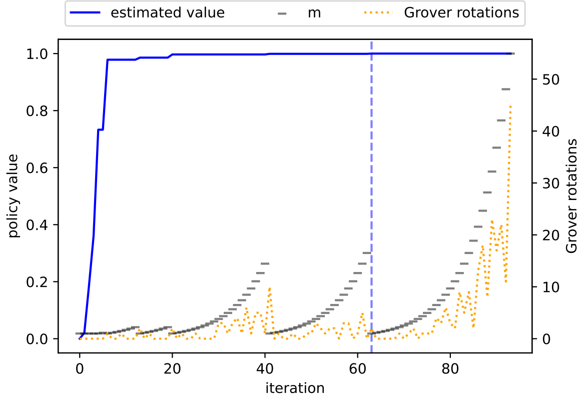

Figure 5: The estimated policy value v_k and the Grover rotations applied during one successful run of quantum policy iteration.

Next, we wanted to empirically confirm the quantum advantage of QPE over classical MC. To this end, we used both methods to evaluate a fixed policy for a fixed bandit using an increasing number of (q)samples. To reduce computational complexity, we chose horizon $H = 1$. According to (23), the approximation error $|v_\pi^1 - \tilde{v}_\pi^1|$ of QPE is in $\mathcal{O}(1/2^n)$, where n is from the definition of t in (18). For our experiment, we set $t = n$. From Figure 1, we see that QPE then uses $2^{n+1} - 1$ qsamples via applications of $\mathbf{A}_{\text{QPE}}$ and $\mathbf{A}_{\text{QPE}}^\dagger$. We chose a range of integer values for n and for each n executed QPE 1000 times and calculated the median approximation error. Recall that the failure probability for QPE with $t = n$ is $8/\pi^2$ so taking the median delivers a run in which the algorithm indeed returned an ϵ-approximation of the true value with high probability. The results of this experiment are shown in Figure 4. We also included the error bound ϵ according to (23) and see that the approximation errors satisfy the theoretical guarantee. For each n, we also collected $2^{n+1} - 1$ classical samples, averaged them, repeated this 1000 times and calculated the median. The approximation error of this classical MC approach is always higher than the one of QPE. This empirically confirms the quantum advantage of QPE over MC in terms of (q)sample complexity.

4.2 Simulations of Quantum Policy Iteration

We now turn to policy iteration. As a toy learning problem, we used a two-armed bandit where the agent always loses when it chooses the left arm and always wins when it chooses the right arm, i.e., $p(0|\leftarrow) = 1$ and $p(1|\rightarrow) = 0$. In DP, one typically only considers deterministic policies as, under mild assumptions, every finite MDP has an optimal deterministic policy (Sutton & Barto, 2018). For the two-armed bandit, there are only two such policies, which results in an uninteresting problem. To make the search for an optimal decision rule more challenging, we instead used a set of $N \in \mathbb{N}$ stochastic policies given by

$$\mathtt{P}_N = \left\{ \pi^n : \pi^n(\leftarrow) = \frac{n-1}{N-1}; n = 1, ..., N \right\}. \tag{29}$$

We let the agent start with the worst policy π^1 which chooses the left arm with unit probability and want to find the optimal decision rule $\pi^* = \pi^N$.

Figure 5 documents one successful run of quantum policy iteration for the toy learning problem with $N = 1000$ policies and parameters $\epsilon = 0.0125$, $\delta = 0.07$, $C = 30$ and $\lambda = 8/7$. We set $H = 1$ so the agent plays one round and the value of the optimal policy is $v^* = 1$. In the beginning, the policy value rapidly increases even when no or only few Grover rotations are applied. This is because we start with the worst possible policy, and better policies are easily found by chance. The steep increase stops after a few iterations, as the policy then is close to optimal. It takes increasingly more rotations and some minor improvements until the procedure makes the final jump (dotted vertical line) after which the values stagnate at the maximum of 1

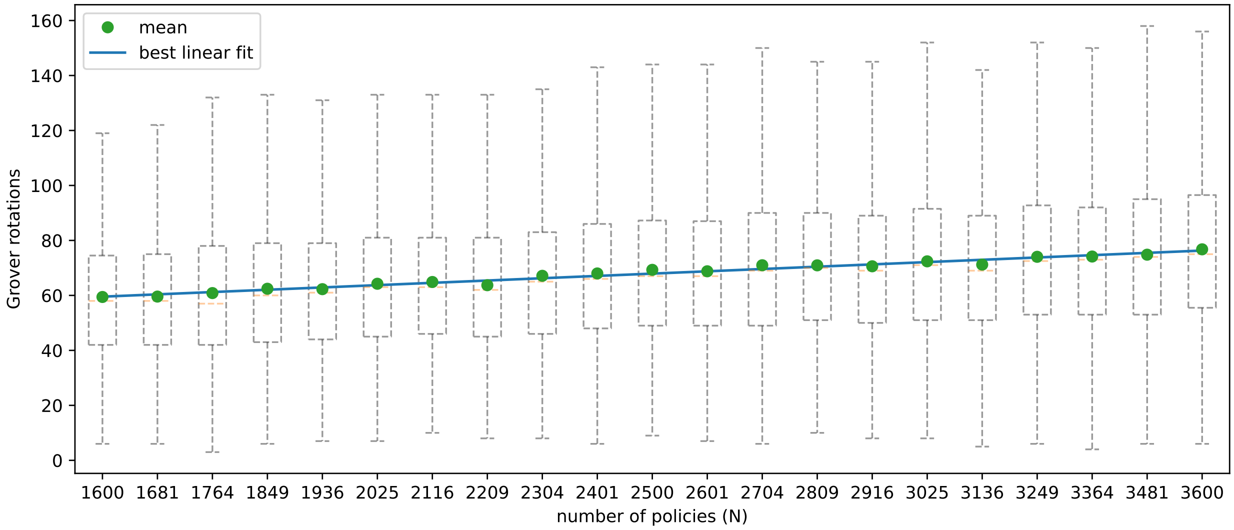

Figure 6: The total numbers of Grover rotations until quantum policy iteration finds an ϵ-optimal policy for the bandit problem.

as from then on, no further improvement is possible. In the end, the algorithm tries more and more Grover rotations as m (that determines their maximum number) grows exponentially.

Finally, we investigated our conjecture that the runtime of quantum policy iteration is proportional to $\sqrt{|\mathtt{P}|}$. To empirically test this, we ran quantum policy iteration for the same bandit and using the same parameters as in the previous experiment. We quadratically increased the size N of the policy set $\mathtt{P}_N$ from $N = 40^2 = 1600$ to $N = 60^2 = 3600$. For each N, we ran quantum policy iteration 1000 times and calculated statistics of the resulting distribution of the number of Grover rotations. As the number of rotations depends on the patience C, we omitted the ones that happened in the last C iterations. We only considered "successful" runs where quantum policy iteration returned an ϵ-optimal policy. Due to the low QPE failure probability, at least 99% of all runs were successful for any N. The results of the complexity experiment are shown in Figure 6. We see that the average number of rotations indeed seems to grow linear in $\sqrt{N}$ The line through the means was found by linear regression and fits the data well; the mean squared error is 1.50.

5 Conclusion

In this work, we used amplitude estimation and Grover search to derive a quantum version of classical policy evaluation and improvement based on a novel quantum realization of an MDP. Our work is a concrete proof-of-concept for how to solve reinforcement learning problems exclusively on quantum computers and how this can reduce sample complexity over classical MC by leveraging the superposition principle.

In their current form and given the likely limitations of near-term quantum hardware, our methods are not (yet) applicable to real problems: The qubit and gate complexity of QPE scales linear with the horizon and quantum policy iteration requires enough qubits to enumerate all possible policies of a finite MDP. Moreover, we assumed an ideal setup in which policy and environment operators are readily available and error-free and the improvement in sample complexity of QPE over classical MC only holds if this is the case. Finally, how to efficiently implement general environment operators for arbitrary MDP dynamics is an open question (a promising idea for future work in this direction is to parameterize the environment operator via a VQC and to learn the gate parameters from data).

We consider the nature of our ideas and algorithms more conceptual and we believe that the in-depth and formal discussion in this work provides useful insight on how to design quantum-based methods that yield other provable speedups in RL. However, QPE and quantum policy iteration are also likely to benefit from future advances in general quantum algorithms (especially in the areas of optimization and amplitude estimation). Hence, we believe that our or conceptually similar methods could one day indeed be applied to solve real RL problems on real quantum hardware.

References

Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505, 1998.

Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.

Hans J Briegel and Gemma De las Cuevas. Projective simulation for artificial intelligence. *Scientific reports*, 2(1):1–16, 2012.

Samuel Yen-Chi Chen, Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Hsi-Sheng Goan. Variational quantum circuits for deep reinforcement learning. *IEEE Access*, 8:141007–141024, 2020.

El Amine Cherrat, Iordanis Kerenidis, and Anupam Prakash. Quantum reinforcement learning via policy iteration. *arXiv preprint arXiv:2203.01889*, 2022.

Richard Cleve, Artur Ekert, Chiara Macchiavello, and Michele Mosca. Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 454 (1969):339–354, 1998.

Paul Dagum, Richard Karp, Michael Luby, and Sheldon Ross. An optimal algorithm for Monte Carlo estimation. *SIAM Journal on computing*, 29(5):1484–1496, 2000.

Vedran Dunjko, Jacob M Taylor, and Hans J Briegel. Framework for learning agents in quantum environments. *arXiv preprint arXiv:1507.08482*, 2015.

Vedran Dunjko, Jacob M Taylor, and Hans J Briegel. Quantum-enhanced machine learning. *Physical review letters*, 117(13):130501, 2016.

Vedran Dunjko, Yi-Kai Liu, Xingyao Wu, and Jacob M Taylor. Exponential improvements for quantum-accessible reinforcement learning. *arXiv preprint arXiv:1710.11160*, 2017a.

Vedran Dunjko, Jacob M Taylor, and Hans J Briegel. Advances in quantum reinforcement learning. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 282–287. IEEE, 2017b.

Christoph Dürr and Peter Høyer. A quantum algorithm for finding the minimum. *arXiv preprint quant-ph/9607014*, 1996.

Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pp. 212–219, 1996.

A Hamann, V Dunjko, and S Wölk. Quantum-accessible reinforcement learning beyond strictly epochal environments. *arXiv preprint arXiv:2008.01481*, 2020.

Sofiene Jerbi, Casper Gyurik, Simon Marshall, Hans J Briegel, and Vedran Dunjko. Variational quantum policies for reinforcement learning. *arXiv preprint arXiv:2103.05577*, 2021.

Owen Lockwood and Mei Si. Reinforcement learning with quantum variational circuits. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, pp. 245–251, 2020.

Maximilian Moll and Leonhard Kunczik. Comparing quantum hybrid reinforcement learning to classical methods. *Human-Intelligent Systems Integration*, 3(1):15–23, 2021.

A. Montanaro. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 471(2181):20150301, 2015.

Lidia Ruiz-Perez and Juan Carlos Garcia-Escartin. Quantum arithmetic with the quantum Fourier transform. *Quantum Information Processing*, 16(6):152, 2017.

Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

Andrea Skolik, Sofiene Jerbi, and Vedran Dunjko. Quantum agents in the gym: a variational quantum algorithm for deep Q-learning. *arXiv preprint arXiv:2103.15084*, 2021.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction.* MIT press, 2018. ISBN 9780262039246.

A tA v, MD SAJID ANIS, Abby-Mitchell, Héctor Abraham, AduOffei, Rochisha Agarwal, Gabriele Agliardi, Merav Aharoni, Vishnu Ajith, Ismail Yunus Akhalwaya, et al. Qiskit: An open-source framework for quantum computing, 2021.

Daochen Wang, Aarthi Sundaram, Robin Kothari, Ashish Kapoor, and Martin Roetteler. Quantum algorithms for reinforcement learning with a generative model. In *International Conference on Machine Learning*, pp. 10916–10926. PMLR, 2021.

A Proof of the error bound of QPE

In this appendix, we prove (22) and (23). The proof uses use the following technical result:

Lemma 1 ((Brassard et al., 2002), Lemma 7)**.** *Let* $\mu = \sin^2(\alpha)$ *and* $\tilde{\mu} = \sin^2(\tilde{\alpha})$ *with* $0 \leq \alpha, \tilde{\alpha} \leq 2\pi$. *Then it holds that*

$$|\tilde{\alpha} - \alpha| \leq a \Rightarrow |\tilde{\mu} - \mu| \leq 2a\sqrt{\mu(1-\mu)} + a^2. \tag{30}$$

To see that (22) is true, recall from (19) that the random variable X that corresponds to the final measurements of phase estimation in QPE satisfies

$$P\Big(\big\{|X/2^t - \theta| \leq 1/2^{n+1}\big\} \cup \big\{|X/2^t + \theta| \leq 1/2^{n+1}\big\}\Big) \geq 1 - \delta. \tag{31}$$

We set $\tilde{\mu} := \sin^2(\pi X/2^t) = \phi\big(\tilde{v}_\pi^H(s)\big)$ and $\mu := \sin^2(\pi\theta) = \phi\big(v_\pi^H(s)\big)$ and obtain

$$P\left(|\tilde{\mu} - \mu| \leq \frac{\pi}{2^{n+1}} + \frac{\pi^2}{2^{2n+2}}\right) \geq P\left(|\tilde{\mu} - \mu| \leq \pi\frac{\sqrt{\mu(1-\mu)}}{2^n} + \frac{\pi^2}{2^{2n+2}}\right) \geq 1 - \delta, \tag{32}$$

where the first inequality holds as $\sqrt{\mu(1-\mu)} \leq 1/2$ and the second inequality is due to Lemma 1. Now the form of ϵ as stated in (23) follows from

$$\big|\tilde{v}_\pi^H(s) - v_\pi^H(s)\big| = \big|\phi^{-1}(\tilde{\mu}) - \phi^{-1}(\mu)\big| = (\bar{g} - \underline{g})|\tilde{\mu} - \mu|, \tag{33}$$

where in the last step, we used the definition of ϕ.