Quantum speed-up for active learning agents

Giuseppe Davide Paparo,1,∗ Vedran Dunjko,2,3,4,∗ Adi Makmal,2,3 Miguel Angel Martin-Delgado,1 and Hans J. Briegel2,3

1Departamento de Física Teórica I, Universidad Complutense, 28040 Madrid, Spain
2Institut für Theoretische Physik, Universität Innsbruck, Technikerstraße 25, A-6020 Innsbruck, Austria
3Institut für Quantenoptik und Quanteninformation der Österreichischen Akademie der Wissenschaften, Innsbruck, Austria
4Division of Molecular Biology, Ruder Bošković Institute, Bijenička cesta 54, 10002 Zagreb, Croatia.

Can quantum mechanics help us in building intelligent robots and agents? One of the defining characteristics of intelligent behavior is the capacity to learn from experience. However, a major bottleneck for agents to learn in any real-life situation is the size and complexity of the corresponding task environment. Owing to, e.g., a large space of possible strategies, learning is typically slow. Even for a moderate task environment, it may simply take too long to rationally respond to a given situation. If the environment is impatient, allowing only a certain time for a response, an agent may then be unable to cope with the situation and to learn at all. Here we show that quantum physics can help and provide a significant speed-up for active learning as a genuine problem of artificial intelligence. We introduce a large class of quantum learning agents for which we show a quadratic boost in their active learning efficiency over their classical analogues. This result will be particularly relevant for applications involving complex task environments.

I. INTRODUCTION

The discovery that the laws of quantum physics can be employed for new and dramatically enhanced ways of information processing has revolutionized research in physics and theoretical computer science [1–6], and established quantum information science as a new interdisciplinary field of research.

However, advances employing quantum mechanics in the field of artificial intelligence (AI) have so far been modest. The theory of quantum information has only been employed, with varying degrees of success, to specific algorithmic AI tasks such as data clustering, pattern matching, binary classification, and similar [7–12], while the problem of general artificial intelligence, that is, the problem of designing autonomous intelligent agents, enhanced through the laws of quantum mechanics, has, up to now, not been addressed.

Partial reason for this is that the existing results in the field of quantum computing cannot be straightforwardly exploited in the latter tasks because a computer is not the same as an agent. The framework of embodied cognitive sciences [13–18] promotes a behavior-based approach to intelligence which puts a strong emphasis on the physical aspects of an agent. It regards an agent as a physically embodied entity with respect to both its internal constitution and its interactions with its environment, and studies the role and implications of this embodiment for learning and intelligence [15].

In this paper, we engage the embodied agent perspective and treat the observation that an agent is an embodied entity, that is ultimately to be described by the laws of physics, as fundamental. Physics does not only provide constraints to any agent architecture, it also may open entirely new avenues and tell us what is possible in principle, in a positive sense, in particular when we take the laws of quantum mechanics into account.

We introduce a class of quantum agents all of which are active learning agents: they operate in an unknown environment which they can actively explore, and they have access to a certain type of quantum memory that helps them to process previous experience. The quantum memory we are describing will involve quantum Markov chains over graphs that encode percept-action-reward correlations in their previous experience. These quantum models can be seen as quantum analogues of certain classical learning models based on classical Markov chains. A particular instance of the latter is the model of projective simulation (PS) introduced in [19]. While we discuss most of our work in the framework of the PS model, the results will be more general.

The notion of active learning is well-motivated both from a biological [20, 21] and a contemporary educational perspective [22], where animal or human agents actively explore their environment. In the case of static environments, active agents can, e.g., control the frequency with which they intervene with the environment and thus expose themselves to new perceptual input. Conversely, active agents should also be able to handle dynamic environments which change, and can impose variable interaction tempos.

Modern examples of active scenarios are the so-called real-time computer games, as opposed to the static

* The first two authors have contributed equally to this work.

1 We note that the notion of active learning is also used in some branches of machine learning, however, with a different meaning.
turn-based games. As a main result of our paper we show that in such an active learning context, the learning efficiency of a large class of agents, whose program is based on Markov chains, can be quadratically enhanced by our quantization procedure.

II. LEARNING AGENTS AND QUANTUM PHYSICS

Embodied agents are best thought of as robots or biological systems, including animals and humans, but they also include so-called internet robots [13, 14, 16, 17]. They are entities situated in an environment with which they communicate. Viewed externally, an agent can receive a sensory input (percept) from the environment and, based on this, produce an action, see Fig. 1.

However, intelligent behavior demands more: the agent must learn from its experiences. A learning agent also receives feedback from the environment in form of rewards (which can be seen as special types of percepts), given when the produced action, for a given percept, was right\(^2\). The agent commits its experiences to its memory and uses the acquired knowledge in the subsequent steps of interactions. This is formalized in the framework of reinforcement learning agents [23, 24].

A reinforcement learning agent is, at every instance in some internal state which comprises its memory and reflects the agent’s experience, that is, elapsed sequences of percepts, actions and rewards. Each sequence of percept-action-reward constitutes an external time-step (or cycle) of the activity of an agent. Formally, we can distinguish two characterizing elements of such agents: the decision function \(D\) which defines which action will be output given the current internal state and the received percept. This function defines the policy of the agent (as given in e.g. [24]) at each time-step. The learning strategy of the agent is characterized by an update function \(U\) which defines how the internal state changes, based on the percept-action pair, and the ensued reward. A successful agent will, over time, increase its reward frequency.

The agent’s learning process is reminiscent to computational oracle query models, in which an unknown oracle (environment), is queried (via an action) by the agent, in an iterative quest for the best responses. It is tantalizing to consider employing the powerful quantum searching machinery [3, 25, 26], which has been proven to outperform classical algorithms in computational settings, in an attempt to improve the agent. Here, one may think of simply applying quantum analogs of the internal decision and update functions, and sequentially perform quantum searches for the optimal action, and thereby achieve better than classical performance.

However, contrary to computer algorithms, an embodied agent, such as a robot, operates in a physical environment which is, for most existing applications, classical\(^3\). This prohibits querying in quantum superposition, a central ingredient to all quantum search algorithms. Thus, naïve approaches to quantizing learning agents are doomed to fail\(^4\).

 Nonetheless, while the physical nature of the agent and the environment prohibits speed-up through quantum queries of the environment, the physical processes within the agent, which lead to the performed actions, can be significantly improved by employing full quantum mechanics\(^5\). In particular, the time the agent requires for these processes (the internal time) can be sped up.

In active learning settings, this alone will constitute an overall improvement of performance, for instance when the environment imposes its own required interaction pace, or where the environment simply changes on

\[^2\] In this paper we restrict our analysis to a binary reward scheme for simplicity, but all our results can be extended to arbitrary reward systems.

\[^3\] Learning agents that operate in a quantum environment are conceivable and not without interest. They may have interesting applications in future quantum physics laboratories, but we will not consider such a scenario in this paper.

\[^4\] Even if we were to allow superpositions of actions, the amount of control the agent must have over the degrees of freedom of the environment, in order to apply quantum query algorithms, may be prohibitive. This constitutes one of the fundamental operative distinctions between quantum algorithms, where full control is assumed, and quantum agents, where such control is limited.

\[^5\] In embodied agents, these physical processes will realize some internal model of the environment, which the agent itself has to develop as it interacts with the environment. For example, in the context of artificial neural networks such internal models are known as self-organizing maps and, more specifically, sensorimotor maps [27, 28].
time-scales not overwhelmingly larger than the agent’s internal ‘thinking’ time. In the latter case the quantum agents may even develop behavioral patterns different from their their classical counterparts, which we will comment on later.

For formal definitions of reinforcement learning agents, we refer the reader to the Appendix, section VI A and proceed by further formalizing the behavioral characteristics of agents.

A. Behavioral equivalence and active time performance comparison

In the algorithmic tradition of machine learning, the learning pace is measured by external time (steps) alone, and the typical figure of merit is the percentage of awarded actions of the agent, as the function of external time. From an embodied agent perspective, this corresponds to a special passive setting where a static environment always waits for the responses of the agent. This constraint imposes a restriction on the universality of statements which can at all be made about the performance of an agent. In particular, in that setting it is well-known that no two agents can be meaningfully compared without reference to a specific (or a class of) learning tasks - a collection of results dubbed ‘no free lunch theorems’, and ‘almost no free lunch theorems’ [29, 30]. These results prove that when one agent outperforms another in a certain environment, there exists a different environment where the ranking according to performance is reversed. This, for instance, implies that every choice of environment settings, for which results of agent performance are presented, must be first well justified. More critically for our agenda, in which we wish to make no assumptions on the environment, passive settings would imply no comparative statements relating the performances of agents are possible.

Here, we consider active scenarios where internal time does matter, but nonetheless the passive setting plays a part. It provides a baseline for defining a passive behavioral equivalence of agents, which will be instrumental in our analysis of active scenarios.

Let us denote the elapsed sequence of triplets (percept, action, reward) which had occurred up to time-step \( k \) (the history of the agent) with \( H_k \), for two agents \( A \) and \( A' \) who can perceive and produce the same sets of percepts (\( S \)) and actions (\( A \)), respectively. Then we will say that \( A \) and \( A' \) are passively \((\epsilon-)\)equal if at every external time step \( k \) the probabilities \( P_A \) and \( P_{A'} \) of agents \( A \) and \( A' \), respectively, outputting some action \( a \in A \), given every percept \( s \in S \), and given all possible identical histories \( H_k \) are \((\epsilon-)\)equal, in terms of the variational distance on distributions:

\[
\frac{1}{2} \sum_a |P_A(a|H_k, s) - P_{A'}(a|H_k, s)| \leq \epsilon, \quad (1)
\]

which we abbreviate with

\[ A \approx_{\epsilon} A'. \]  

The relation above induces passive behavioral equivalence classes for fixed sets of possible percepts and actions. In passive settings, by definition, two passively equal agents perform equally well, and comparison of agents within a class is pointless.

However, in the active scenario, with the classes in place, we can meaningfully compare agents within the same class, with no assumptions on the environment\(^7\). Indeed, in an active learning setting, two passively equal agents \( A \) and \( A' \) may have vastly different success chances. To see this, suppose that the environment changes its policies on a timescale that is long compared to the internal timescale of agent \( A \), but short relative to the internally slower agent \( A' \). The best policy of both agents is to query the environment as frequently as possible, in order to learn the best possible actions. However, from the perspective of the slow agent, the environment will look fully inconsistent - once rewarded actions are no longer the right choice, as that agent simply did not have the time to learn. In active scenarios, internal speed of the agent is vital, which is a property we shall exploit further.

B. How fast is my embodied agent

As we have mentioned earlier, the behavior of an agent, once the space of percepts it can perceive, and actions it can perform is fixed, is characterized by its decision function \( D \) and the update function \( U \). It will be beneficial to see this in the light of behavioral equivalence classes we introduced above. In particular, it is straightforward to see that if two agents realize identical update functions, and similar decision functions, then they will be behaviorally similar as well\(^8\). In fact, this characterization will give us a simple means of establishing behavioral equivalence between classical agents, and their quantum counterparts. To establish

---

\( A \approx_{\epsilon} A' \)

We acknowledge that the interpretation of these results in the sense of their practical impact on the field are not without controversy. Nonetheless, the validity of the mathematical statements is not contested. See [31] for more details.

\(^7\) That is, no further assumptions beyond the trivial ones - that the percept and action sets are compatible with the environment and that the environment provides a rewarding scheme.

\(^8\) Converse does not hold however. In principle, significantly different agents, with differing sets of internal states, and differing update and decision functions can nonetheless be behaviorally passively equivalent.
that, given two agents in the same equivalence class, one is actively faster than another, one must show that the internal times of evaluating the decision and the update function favor one of the two. Given that two agents may be implemented in significantly different physical systems\(^9\), properly gauging speed may be difficult\(^{10}\). The standard approach in addressing this issue is by defining primitive processes, like ‘gates’ or ‘Turing machine steps’\(^{11}\), and as we describe next, in our case they will effectively be evolutions of Markov chains.

The AI model for the agents we consider in the following is the so-called Projective Simulation (PS) model [19]. Here, to percepts and actions, the agent assigns internal representations, i.e., memories, of percepts and actions, and it can also represent sequences of such memories. We will refer to these objects as clips[19], and we envision the implementations of clips as excitations of physical systems. The elementary internal processes of the agent, which implement the transitions from clip to clip internally, are discrete-time stochastic diffusion processes. In particular, the internal states of the agents are represented by weighted directed graphs (digraphs) over the clips and the decision making involves diffusion, as dictated by the digraphs, a sufficient number of times. Mathematically, these diffusion processes are described by Markov chains (MC), the transition probabilities by the transition matrix of the MC, and are standardly formulated in terms of random walks over weighted directed graphs. We will inherit the notion of quantizing these diffusion processes (and hence the agents themselves) from the theory of quantum random walks on directed graphs [25, 26, 34]. Consequently, in our setting, the primitive processes will be represented by the steps of classical and quantum random walks, for the classical and quantum agents, respectively. In the following section we expose the PS model for the readers benefit, and then give the main results of this paper. For more details on the theory of discrete-time classical and quantum walks see VI.B.

III. QUANTIZATION OF LEARNING AGENTS BASED ON PROJECTIVE SIMULATION

A. The PS agent model

The PS model is based on a specific memory system, which is called episodic and compositional memory (ECM). This memory provides the platform for simulating future action before real action is taken. The ECM can be described as a stochastic network of the previously mentioned clips, which constitute the elementary excitations of episodic memory. In the following, we formalize the PS model in terms which are slightly more general than given in [19]; this will simplify the construction of the desired quantum PS agents later. For the original construction, and the explicit description of the generalization, we refer the reader to the Appendix, section VI.C.

In formal terms the general PS model comprises the spaces of percepts \( \mathcal{S} = \{s_1, s_2, \ldots \} \) and actions \( \mathcal{A} = \{a_1, a_2, \ldots \} \), which constitute the space of environmental stimuli the agent can perceive and the space of actions it can effect, respectively. The basic actions and percepts, along with sequences thereof, are represented within an agent as clips \( c \), and the set of these comprises the clip space \( C = \{c = (c^{(1)}, c^{(2)}, \ldots) | c^{(k)} \in \mathcal{S} \cup \mathcal{A} \} \).

In this work, we focus on PS agents in which clips are unit length sequences, representing a memorized percept or an action, which we denote using the same symbols, so \( C = \mathcal{S} \cup \mathcal{A} \). Each agent is equipped with input and output couplers, which translate, through sensors and actuators (see Fig. 1), real percepts to the internal representations of percepts, and internal representations of actions to real actions, respectively. The internal states of the agent, i.e. the total memory, comprise weighted graphs \( G_s \) over a subset of the clip space, which are assigned to each percept \( s \). This graph dictates the hopping probabilities from one clip to another, and is a Markov process, i.e., a MC. The hopping probabilities themselves are encoded in the so-called \( h \)-matrix, defining un-normalized relative hopping probabilities. This matrix encodes \( G_s \) for every \( s \). Each agent can also perceive rewards \( \Lambda = \{0, 1\} \) and, based on the received percept \( s \), realized action \( a \), and the resulting reward, the \( h \)-matrix is updated via simple rules, which in turn, incurs a change in the transition probabilities. For a more detailed description of the PS model, and an example of previously studied update rules see the Appendix, section VI.C.

---

\(^9\) Which also implies the natural laws which govern the dynamics so such systems which the agent uses to realize the decision and update functions.

\(^{10}\) In fact, for some pairs of implementations, like the biological brain versus a classical computer running an algorithm, even showing behavioral equivalence may be difficult.

\(^{11}\) In general, the internal process of the agent may be inherently time-continuous. An example where similar problems were successfully mitigated is that of adiabatic quantum computation [32, 33].

\(^{12}\) In the PS framework one formally distinguishes between real percepts and actions \( s, a \), and their internal representations denoted \( \mu(s), \mu(a) \), which comprise clips. For our purposes, by abuse of notation, we will omit the mapping \( \mu \). For more details see the Appendix, section VI.C.
In the process of deliberation (which realizes the agent’s decision function $D$), the relevant Markov chain is diffused a particular number of times (depending on the particular PS model) until an action is output. The choice of the action - and thereby the policy of the agent - is dictated by the probability distribution over the actions, which is realized by the diffusion process. The latter depends on the agent’s experience manifest in the matrix $h$. This distribution represents the agent’s state of belief on what is the right action in the given situation. Conceptually, projective simulation thus consists of a random walk through episodic memory – a stochastic network of clips – which serves the agent to re-invoke past experience and to compose fictitious experience before real action is taken. Learning is then achieved by updating and reconfiguring episodic memory under environmental feedback.

Aside from the basic structural and diffusion rules, the PS model allows for additional structures, some of which we briefly address here. 1) *Emoticons* - a form of short-term memory, i.e., *flags* which notify the agent whether the currently found action, given a percept, was previously rewarded or not. For our purposes, we shall use only a rudimentary mode of flags, which designate that the particular action (given a particular percept) was not already unsuccessfully tried before. If it was, the agent can ‘reflect on its decision’ and re-evaluate its strategies, by re-starting the diffusion process. We make this more formal presently. 2) *Edge and clip glow* - mechanisms which allow for establishing additional *temporal correlations*. 3) *Clip composition* - the PS model based on episodic memory allows for the creation of new clips under certain variational and compositional principles. These enable the agent to develop *new behavioral patterns* under certain conditions, and allow for a dynamic reconfiguration of the agent itself. For more information on these additional structures, we refer the reader to [19, 35]. In this work we will focus on the bare PS model with the most rudimentary emoticon structure, but note that most of the additional structures are compatible with the quantization procedure we explain next.

Next, we specify a particular broad class of PS agents, called *reflecting agents*, for which we give a generic quantization procedure, which yields a quadratic speed-up.

### B. Reflecting PS agents

As we have previously mentioned, the diffusion process within the agent is the constitutive part of the projection (simulation of future action) before real action is taken. From an operational perspective, the same process realizes some distribution over the action space, from which the actual action is sampled. This distribution represents the agent’s state of belief about the optimal response, given its accumulated knowledge. *Reflecting* PS agents draw their name from the reflection process [19] in which the diffusion processes are repeated multiple times - in this model the agents approximate the complete mixing of their diffusion processes, simulating infinite deliberation times. While this may suggest that reflecting agents are comparatively slow (in terms of internal time), we will show in section IV that they are often on par with ‘more eager’ models. Concretely, the reflecting PS agent is a PS agent, where the internal states (the memory) are, for each percept, *irreducible, regular MC’s* over subsets of the clip space, such that *all actions* are included in the subsets. The additional structure we will consider consists of percept-specific *flags* (corresponding to rudimentary *emoticons* in [19]) which are just subsets of actions assigned to each percept, formally $F = \{f(s) | f(s) \subseteq A, s \in \mathcal{S}\}$. These constitute the agent’s short-term memory.

For example, one can consider a possible mechanism in which for each percept, all actions are initially flagged. If the agent outputs some action $a$, given a percept $s$, and this action is not rewarded, $a$ is removed from $f(s)$. The meaning of *flags* may, however, in principle be more general.

Recall, the policy of the agent is defined by the decision function $D$. For *ideal* reflecting agents, given a percept $s$, and the relevant part of the internal state, that is, the corresponding Markov chain $P_s$, $D$ outputs an action $a$ distributed according to $\pi_s$, given as follows: let $s_i$ be the stationary distribution of $P_s$, and let $f(s)$ be the subset of flagged actions, then

$$\tilde{\pi}_s(i) = \left\{ \begin{array}{ll} \pi_s(i) / \sum_{j \in f(s)} \pi_s(j), & \text{if } i \in f(s) \\ 0, & \text{otherwise.} \end{array} \right.$$ \hspace{1cm} (3)

that is the re-normalized stationary distribution $\pi_s$ modified to have support only over flagged actions. We will often refer to $\tilde{\pi}$ as the *tailed distribution*. The reflecting agents also have an update function, which defines the agent’s learning strategies by updating the internal states (including the flag assignments), but the reflecting agent model imposes no restrictions on the update functions. The abstractly given decision function above has a natural interpretation within the conceptual framework of PS: the process of projective simulation represents a stochastic replay of episodic memory, in particular of percept-to-action sequences, the transition probabilities of which are dictated by previous experiences. The (ideal) reflecting agent is defined as the limit in which the agent allows this process to run forever, until an equilibrium distribution over the

---

13 In section IV we will give a concrete example of a reflecting agent model, where particular update functions are given.
clips is found. This distribution may have support over the entire clip space and, since only actions can be output, it must be truncated to the action space alone. In our case, due to the additional structure of flags, we truncate it further to only actions which are not known to have been incorrect in the recent past. For a full formal specification of the ideal reflecting agent we refer the reader to the Appendix, section VI D.

Any embodied agent will, however, have to stop the deliberation in some finite time, which constitutes the effective internal time of the agent, vital in active scenarios. As we have explained, the elementary processes of the agents we consider are diffusion processes governed by the MC’s. The classical reflecting agent will then be defined by a process realizing the decision function which we give next. In the following we will be using standard results from the theory of classical and quantum random walks \cite{25, 26, 34}, and refer the reader using standard results from the theory of classical and quantum random walks to Appendix, section VI B, for more details. Let, diffusion processes above, output an action uniformly at random.

1. Initialize: Prepare

\[ \pi = P^t_s \pi_0 \]

for some fixed distribution \( \pi_0 \).

2. For \( t_2 \) time steps do:

(a) Check: Sample \( y \) from \( \pi \). If \( y \) is a flagged action, break and output \( y \).

(b) Diffuse: re-mix the Markov chain by

\[ \pi = P^t_s \pi \]

3. If no flagged action is encountered by the diffusion processes above, output an action uniformly at random.

As we can see, the internal time of the classical agent, i.e. the number of primitive processes (diffusion steps), is fully dictated by the quantities \( \delta_s \) and \( \epsilon_s \).

At this point, the theory of quantum walks provides us with analogs of discrete-time diffusion processes, using which we can describe the quantum analog of the agent above. The process we will describe here closely follows the quantum walk approaches to the algorithmic problems of element presence detection, and marked item search, as given in \cite{26} but to a fundamentally different end. Standardly, quantum search algorithms have been employed in computer science to reduce the number of oracular function evaluations with the task of efficiently outputting an item satisfying some logical predicate \cite{34, 36}. In contrast, the goal of the protocol which the agent follows is more general. The task here is to output a (fully known) flagged action, but according to a specific probability distribution: a good approximation of the aforementioned tailed distribution, see Eq. (3). We note that using quantum walks for the purpose of sampling from desired distributions has been studied previously, predominantly in the context of quantum counterparts of Markov chain Monte Carlo methods \cite{37–40}. There the quantum walks were mostly used for the important purpose of sampling from Boltzmann-Gibbs distributions of (classical and quantum) Hamiltonians. Aside from the construction we provide next, concrete applications of quantum walks specified by non-symmetric Markov chains have, to our knowledge, only been studied in \cite{41, 42}, by two of the authors and other collaborators.

In the following, we will use the standard quantum discrete time diffusion operators \( U_P \) and \( V_P \) which act on two quantum registers, sufficiently large to store the labels of the nodes of the Markov chain \( P_s \). The construction we present follows the one given in \cite{26}. The diffusion operators are defined as \( U_P \{ |0 \rangle = |i \rangle |p_i \rangle \) and \( V_P \{ |j \rangle = |p_j \rangle |j \rangle \) where \( |p_i \rangle = \sum_j \sqrt{|P_{ij}|} |j \rangle \), \( |p_j \rangle = \sum_i \sqrt{|P_{ij}|} |i \rangle \). Here, \( P_s \) is the time reversed MC defined by \( P_s \pi_j = \pi_j P_s \), where \( \pi = (\pi_i) \) is the stationary distribution of \( P_s \). Using four applications of the diffusion operators above, it has been shown that one can construct the standard quantum walk operator \( W(P_s) \), which is a composition of two reflections in the mentioned two-register state space. Let \( \Pi_1 \) be a projection operator on the space \( \text{Span}\{ |i \rangle |p_i \rangle \} \), and \( \Pi_2 \) be the projector on \( \text{Span}\{ |p_j \rangle |j \rangle \} \). Then \( W(P_s) = (2\Pi_2 - I)(2\Pi_1 - I) \).

Using the quantum walk operator and the well-known phase detection algorithm the agent can realize the \( R(P_s) (q, k) \) subroutine which approximates the reflection operator \( 2|\pi_s\rangle \langle \pi_s| - 1 \), where \( |\pi_s\rangle = \sum_i \sqrt{\pi_s(i)} |i \rangle \) is the coherent encoding of the stationary distribution \( \pi_s \). The parameters \( q \) and \( k \) control the fidelity (and the time requirement) of this process – how well the reflection is approximated as a function of the number of applications of the quantum walk operator.

\footnote{In the following we will use the label \( P_s \) of the transition matrix as a synonym for the MC itself.}

\footnote{In this paper we do not consider logarithmically contributing terms in the complexity analysis, thus we use the \( O \)-level analysis of the limiting behavior (instead of the standard ‘big \( O \)’ \( O \)). We note that some of the logarithmically contributing terms which appear, can be further avoided using more complicated constructions as has, for instance, been done in \cite{26}.}
The fidelity can be shown to approach unity exponentially quickly in $k$ if $q \in O(1/\sqrt{\delta_s})$ [26]. For more details on the basics of classical and quantum walks we refer the reader to the Appendix VI.B where we also present an explicit treatment of the approximate reflection operator.

Now, we can define the quantum reflecting agent’s deliberation process.

Let $t_1 \in \tilde{O}(1/\sqrt{\delta_s})$ and $t_2 \in [0, \tilde{O}(1/\sqrt{\epsilon_s})]$ where we adhere to the notation used for the classical reflecting agent, and $\epsilon_s$ denotes that the relevant choice is performed uniformly at random.

Then, the steps of the quantum agent are as follows:

1. **Initialize**: Prepare
   \[
   |\pi_{\text{init}}\rangle = \sum_{i \in X} \sqrt{\pi_s(i)} |i\rangle |p_i\rangle,
   \]
   with $|p_i\rangle = \sum_j \sqrt{P_{ji}} |j\rangle$, and $P_{ji}$ is the transition probability from $i$ to $j$ as dictated by the MC $P_s$.

2. For $t_2$ time-steps do
   
   (a) **Check**: Apply the $V_{f(s)}$ operator which flips the phase of all components of the current state of the first register which are not in $f(s)$.
   
   (b) **Diffuse**: Apply the approximate reflection operator $R(P_s)(t_1, O(1))$, as described previously.

3. Measure the first register, and if it is a flagged action, output it, else output a random action.

For the first step above we assume that the agent can prepare the initial state $|\pi_{\text{init}}\rangle$ which requires just one application of the diffusion operator $U_{P_s}$, provided the state $|\pi_s\rangle = \sum_i \sqrt{\pi_s(i)} |i\rangle$ is available. In this framework, like in the standard frameworks of algorithms based on quantum walks with non-symmetric Markov chains, we assume this is the case. Later we will provide an example of reflecting classical and quantum agents where this is easily achieved. We also assume that the agent can perform the phase flip operator $V_{f(s)}$ defined with $V_{f(s)} |i\rangle = |i\rangle$ if $i \in f(s)$, and $-|i\rangle$ otherwise, which is easy given that the agent has full access to $f(s)$. Finally, we mention that the quantum agent may repeat the entire procedure a logarithmic number of times, but this overhead will not significantly contribute to our analysis.

We can immediately see that the quantum agent is, by construction, quadratically faster than the classical counterpart, in terms of the figures $\delta_s$ and $\epsilon_s$, which alone determine the number of applications of the agents’ respective elementary processes. What constitutes our main result is the fact that both the presented classical and the quantum reflective agent (defined over the same percept and action sets) belong to the same passive behavioral class within distance $\alpha$, where the class is defined by the ideal reflecting agent (see paragraph before Eq. (3)). To state this more formally, if $A_{\text{ideal}}, A_c$ and $A_q$ denote the ideal, the classical and the quantum reflecting agents, respectively, with the same sets of percepts and actions, then

\[
A_c, A_q \in \{A| A \approx_{\alpha} A_{\text{ideal}}\},
\]

where $A$ is any agent with the same set of percepts and actions. Crucially, we show that $\alpha$ can be made, in both the classical and the quantum case, exponentially small within the same limiting behaviors given with $O(1/\delta_s)$ and $O(1/\epsilon_s)$, for the classical, and $O(1/\sqrt{\delta_s})$ and $O(1/\sqrt{\epsilon_s})$, for the quantum agent. That is, the two proposed agents are passively approximately equivalent, and the quantum agent is quadratically faster than the classical agent. We summarize this result in the following theorem:

**Theorem 1.** (main) The proposed classical and quantum agents are $\alpha-$passively equivalent. Moreover, they both approximate the passive behavior of the ideal reflecting agent within $\alpha$. The parameter $\alpha$ can be made arbitrarily small with at most a logarithmic overhead in the number of the required elementary processes for both agents. As a consequence of the given constructions of the agents, the quantum reflecting agents are generically quadratically faster than their classical counterparts.

For a proof of this theorem, we refer the reader to the Appendix, section VI.E, but note that the proof builds on the known results for speed-ups in quantum search algorithms proposed in [3, 25, 26]. Thus, what we have presented is a method for generic quantization of reflecting PS agents, which maintains the behavior of the agents, and provably yields a quadratic speed-up in the active setting.

In the following section we compare a well-studied class of PS agents (which we refer to as standard PS agents) with the proposed ‘quantizable’ reflecting PS agents. We show that the classical reflecting PS agent performs equally well as the standard model. This immediately implies that the quantum reflecting agent outperforms standard constructions quadratically as well, in terms of internal time.
IV. COMPARISON OF PS AGENT MODELS

While the definition of reflecting PS agents we introduced in the previous section fits into the broad framework of PS agents, it still differs from the standard PS agents, which have been previously studied \[19, 35\]. The main difference between the standard and the reflecting model is that the standard agents evolve their Markov chain until the first instance where an action clip has been hit, whereas the reflecting agents allow for the Markov chain to fully mix. In the standard PS model, the underlying Markov chain should never mix, as indeed in that case, the construction will guarantee that the actual action performed is independent from the received percept (if the Markov chain is irreducible). This implies that large clip networks of high connectivity should not be used in the standard PS, although, from the conceptual perspective, they could appear in agents which have undergone a rich complex set of episodic experiences (for instance any type of clip compositions we mentioned earlier). The reflecting agents could be expected to yield behavioral advantages in such scenarios. Conversely, since the reflecting agent must always fully mix its Markov chain, and the standard does not, it would seem that the standard model should typically outperform the reflecting model on simple clip networks (as the hitting time may be much smaller than the mixing time). This would suggest that the standard and reflecting PS models should not be compared on simple structures, as no quantum advantage can be demonstrated.

In contrast to this, here we show that even for the simplest (but non-trivial) standard PS agent construction \[19, 35\], there exists a natural reflecting agent analog, where the performance of the classical reflecting agent matches the standard PS construction, and consequently, the quantum reflecting agent yields a quadratic speed-up over both.

A. Simple standard PS agent with flags

Recall that, in the standard PS model, the internal state of the agent is defined by an $h-$matrix, which defines the transition probabilities of the directed Markov chain over the clip space. In the simplest case, the clip space comprises only percepts and actions, and the agent is initialized so that each percept is connected to each action with unit weight (implying equiprobable transitions), and no other connections exist. Furthermore, the update function we assume is the standard update function as defined in \[19\] and repeated in the Appendix VI C. In this particular model, the internal elementary time-step is one evolution of the Markov chain, so the agent always decides in one step, and there is nothing to improve. However, if we introduce short term memory, which significantly improves the performance of the model \[19\], then the fact that the agent may hit an un-flagged item implies it may have to walk again. If $\epsilon_s$ denotes the probability (for the clip $s$) that the agent hits a flagged item, then the expected number of elementary transitions, and checks, the agent will perform is $O(1/\epsilon_s)$ (see Fig. 2 for an illustration). For this model, we can introduce a direct reflecting agent analog.

B. Simple reflecting agent with flags

It is relatively straightforward to construct a reflective PS agent which is behaviorally equivalent to the simple standard agent with flags, given above. To each percept we assign a Markov chain over just the action space, such that the transition matrix is column-constant, and the column simply contains the transition probabilities for that particular percept in the standard model above, see Fig. 2 for an illustration. The stationary distribution of this Markov chain then matches the desired transition probabilities. The update function for this agent is then exactly the same as in the standard model (that is, internally, the reflecting agent can also keep and update the same $h-$matrix which induces the Markov chains), and the flags are also treated in exactly the same way. Note that the transition matrix of
such a Markov chain is rank-1, meaning it has only one non-zero eigenvalue, and since the trace of this matrix is unit (and trace is invariant under diagonalization/basis change), this implies the largest eigenvalue is 1 and all the others are zero. Thus we have that the spectral gap is always 1. This immediately implies that the Markov chain is fully mixed after just a single transition (as \( \delta = 1 \)). Thus the classical reflecting agent will also perform \( O(1/\epsilon) \) transitions and checks until an action is performed, given the percept \( s \).

However, the quantum agent, will only require \( O(1/\sqrt{\epsilon}) \) calls to the quantum walk operator. Thus, even in the very simple setting we see that the reflecting agents can be compared to the standard PS agent model, and maintain a generic quadratic speed-up when quantized.

While the scenario considered in this section is somewhat restricted, it is not without importance as for the standard PS agents we already have a body of results in the classical case \([19, 35]\). We emphasize however that the quadratic speed-up proven in Theorem 1 is not restricted to this scenario.

V. DISCUSSION

We have presented a class of quantum learning agents that use quantum memory for their internal processing of previous experience. These agents are situated in a classical task environment that rewards a certain behavior but is otherwise unknown to the agent, which corresponds to the situation of conventional learning agents.

The agent’s internal ‘program’ is realized by physical processes that correspond to quantum walks. These are derived from classical random walks over directed weighted graphs, which represent the structure of its episodic memory. We have shown how, using quantum coherence and known results from the study of quantum walks, the agent can explore its episodic memory in superposition in a way which guarantees a provable quadratic speedup in its active learning time over its classical analogue.

Regarding potential realizations for such quantum learning agents, modern quantum physics laboratories offer a variety of systems. Quantum walks can naturally be implemented by linear optics setups with single photons propagating in arrays of polarizing beam splitters (PBS) \([44]\). Feedback from the environment can, for example, change the polarization of the photons in each path of such an interferometer which in turn updates the probability amplitudes for taking certain subsequent paths. Highly versatile setups can also be realized using internal states of trapped ions \([45]\) or atoms as the quantum memory, and lasers to control the transitions between different internal states. Feedback from the environment may here change the laser intensity and/or phase. These are just two out of several systems that are studied in the field of quantum computing and simulation \([46]\), all of which allow, in principle, a level of coherent control that seems to be necessary to implement quantum walks as described in this paper. An entirely different route towards realizing the proposed quantum (and classical) learning agents might employ condensed matter systems in which the proposed Markov chains could e.g. be realized through cooling or relaxation processes towards target distributions that then encode the state of belief of the agent. Here we envision rather non-trivial cooling/relaxation schemes in complex many-body systems, the study of which is a prominent topic in the field of quantum simulation.

In conclusion, it seems to us that the embodied approach to artificial intelligence acquires a further fundamental perspective by combining it with concepts from the field of quantum physics. The implications of embodiment are, in the first place, described by the laws of physics, which tell us not only about the constraints but also the ultimate possibilities of physical agents. In this paper we have given one example how the laws of quantum physics can be fruitfully employed in the design of future intelligent agents that will outperform their classical relatives in complex task environments.

ACKNOWLEDGEMENTS

MAMD acknowledges support by the Spanish MICINN grant FIS2009-10061, FIS2012-33152, the CAM research consortium QUITEMAD S2009-ESP-1594, the European Commission PICC: FP7 2007-2013, Grant No. 249958, and the UCM-BS grant GICC-910758. HJB acknowledges support by the Austrian Science Fund (FWF) through the SFB FoQuS: F 4012.

VI. APPENDIX

A. Formal definitions of reinforcement learning agents

Here we formally define the model of reinforcement learning agents as employed in this work.

**Definition 1.** (Reinforcement learning agent) A reinforcement learning agent is an ordered sextuplet \((\mathcal{S}, \mathcal{A}, \Lambda, \mathcal{C}, \mathcal{D}, \mathcal{U})\) where:

- \(\mathcal{S} = \{s_1, \ldots, s_m\}\), \(\mathcal{A} = \{a_1, \ldots, a_n\}\) are the sets of percepts and actions, respectively.
- \(\Lambda = \{0, 1\}\) is the set of rewards, offered by the environment.
- \(\mathcal{C} = \{C_1, \ldots, C_p\}\) is the set of possible internal states of the agent.
• \(D : S \times C \rightarrow A\) is the decision function, which outputs some action given a percept and the internal state.

• \(U : S \times A \times \Lambda \times C \rightarrow C\) is the update function, which updates the internal state based on the success or failure of the last percept-action sequence.

A few comments are in order. In this work, the sets of percepts, actions and internal states are defined to be finite, but, in general, this need not be the case. The set of rewards is binary, and this can again be generalized. In our settings, however, we assume a binary rewarding system. The update function may take additional information into account, based on additional outputs of the decision function, which are only processed internally, but this does not occur in the models we consider.

The decision function is not necessarily deterministic. In the non-deterministic case it can be formally defined as

\[ D : S \times C \rightarrow \text{Distr}(A) \] (5)

that is, a function which takes values in the set of distributions over \(A\). In this case we also assume that this distribution is sampled before actual output to the environment is produced, and the realized sampled action is the input to the update function.

B. Classical and quantum walk basics

A random walk on a graph is described by a MC, specified by a transition matrix \(P\) which has entries \(P_{ij} = \text{Prob}(j|i)\). For an irreducible MC there exists a stationary distribution \(\pi\) such that \(P\pi = \pi\). For an irreducible, regular MC this distribution can be approximated by \(P^t\pi_0\) i.e. by applying, to any initial distribution \(\pi_0\), the MC \(t\) number of times where \(t \geq \tau^{\text{mix}}\). This time is known as mixing time and is defined as follows.

Definition 2. (Mixing Time).
The mixing time is:

\[ \tau^{\text{mix}} = \min \{ t | \forall s \geq t, \pi_0 : \|P^t\pi_0 - \pi\| \leq \epsilon' \} \] (6)

The latter can be related to the spectral properties of the MC \(P\) via the following theorem [17]:

Theorem 2. The mixing time satisfies the following inequalities:

\[ \frac{\lambda_2}{\delta \log 2\epsilon'} \leq \tau^{\text{mix}} \leq \frac{1}{\delta} \left( \max \lambda_1^{-1} + \log(\epsilon')^{-1} \right) \] (7)

In the above we use \(\epsilon'\) instead of the standard \(\epsilon\) for consistency with the rest of the Appendix, as \(\epsilon\) has a reserved meaning.

For the purpose of clarity let us introduce some definitions and theorems, originally provided in [25, 26] that will be useful to introduce the notation and to prove the main results for the speedup of quantum agents.

The quantum analog of the applying the MC \(P\) is given by:

Definition 3. (Quantum Diffusion Operators).
The quantum diffusion operators, the analogs of the classical diffusion operators are given by the following transformations:

\[ U_P : |i\rangle |0\rangle \mapsto |i\rangle |p_i\rangle \] (8)
\[ V_P : |0\rangle |j\rangle \mapsto |p_j^*\rangle |j\rangle , \] (9)

where \(p_i = \sum_j \sqrt{P_{ij}} |j\rangle\), \(p_j^* = \sum_i \sqrt{P_{ij}}^* |i\rangle\) and \(P^*\) is the time reversed MC defined by \(p_i^* = \pi_j P_{ji}\).

We will consider the application of the MC \(P\) (for the classical agent) and the quantum diffusion operators (for the quantum agent) as the (equally time consuming) primitive processes, as is done in the theory of quantum random walks [26]. Next, we can define the quantum walk operator \(W(P)\), for the Markov chain \(P\).

Definition 4. (Walk Operator or Quantum Markov Chain).
The walk operator or Quantum Markov Chain is given by

\[ W(P) = (2\Pi_2 - I)(2\Pi_1 - I), \] (10)

where \(\Pi_1\) is the projection operator onto the space \(\text{Span}\{ |i\rangle |p_i\rangle \}\), and \(\Pi_2\) is the projection operator onto \(\text{Span}\{ |p_j^*\rangle |j\rangle \}\).

The quantum walk operator can be easily realized through four applications of the quantum diffusion operators, see e.g. [26] for details. Thus, it too can be considered a primitive process of the agent.

Another operation which both the classical and the quantum agents do is checking whether the clip found is flagged. The quantum check operator is defined as follows.

Definition 5. (Check). The quantum check operator is the reflection denoted as \(\text{ref}(f(s))\) performing

\[ |i\rangle |j\rangle \mapsto \begin{cases} -|i\rangle |j\rangle, & \text{if } i \in f(s) \\ |i\rangle |j\rangle, & \text{otherwise} \end{cases} \] (11)

where \(f(s)\) denotes the set of flagged actions corresponding to the percept \(s\).

In order to prove our main theorems we will be using the ideas introduced in the context of quantum searching [26]. We will now briefly expose some of the ideas introduced in [25, 26] in that context. In the quantum walk over graphs approach to searching, one defines an
initial state, which encodes the stationary distribution of a MC,
\[ |\pi\rangle = \sum_i \sqrt{\pi(i)} |i\rangle |p_i\rangle , \quad (12) \]
and performs a rotation onto the state containing the 'marked items'
\[ |\tilde{\pi}\rangle := \frac{\Pi^f(s) |\pi\rangle}{\|\Pi^f(s) |\pi\rangle\|} , \quad (13) \]
where \( \Pi^f(s) \) is the projector on the space of marked items i.e \( \Pi^f(s) = \sum_{i \in f(s)} |i\rangle \langle i| \otimes I \). Let us point out that \( \text{Span} \{ |\tilde{\pi}\rangle, |\pi\rangle \} \subseteq \text{Span} \{ |i\rangle |p_i\rangle \} + \text{Span} \{ |p^*_j\rangle |j\rangle \} \).

In order to achieve this rotation one makes use of two reflections. The first is the reflection over \( |\tilde{\pi}\perp\rangle \) (denoted \( \text{ref}(\tilde{\pi}^-) \)), the state orthogonal to \( |\pi\rangle \) in \( \text{Span} \{ |\tilde{\pi}\rangle, |\pi\rangle \} \). This operator can be realized using the primitive of checking. Indeed, we have the following claim (stated in [26]) given by:

**Lemma 6.** Restricted on the subspace \( \text{Span} \{ |\tilde{\pi}\rangle, |\pi\rangle \} \), the action of \( \text{ref}(\tilde{\pi}^-) \) is identical to \( -\text{ref}(f(s)) \).

**Proof.** Let \( \alpha |\pi\rangle + \beta |\tilde{\pi}\rangle \) be a vector in \( \text{Span} \{ |\tilde{\pi}\rangle, |\pi\rangle \} \).
We have that:
\[ -\text{ref}(f(s)) (\alpha |\pi\rangle + \beta |\tilde{\pi}\rangle) = \quad (14) \]
\[ = \alpha (I - 2\Pi^f(s) |\pi\rangle - \beta |\tilde{\pi}\rangle) \]
where \( \Pi^f(s) \) is the projector on the set \( f(s) \). The result easily follows by noting that
\[ \text{ref}(\tilde{\pi}^-) (\alpha |\pi\rangle + \beta |\tilde{\pi}\rangle) = \quad (15) \]
\[ = \alpha \left( I - 2\frac{\Pi^f(s) |\pi\rangle \Pi^f(s) |\pi\rangle}{\|\Pi^f(s) |\pi\rangle\|^2} \right) |\pi\rangle - \beta |\tilde{\pi}\rangle , \]
and that
\[ \frac{\|\Pi^f(s) |\pi\rangle \Pi^f(s) |\pi\rangle}{\|\Pi^f(s) |\pi\rangle\|^2} |\pi\rangle = \frac{\langle \pi | \Pi^f(s) |\pi\rangle |\pi\rangle}{\|\Pi^f(s) |\pi\rangle\|^2} \Pi^f(s) |\pi\rangle \]
\[ = \Pi^f(s) |\pi\rangle , \]
since \( \frac{\langle \pi | \Pi^f(s) |\pi\rangle}{\|\Pi^f(s) |\pi\rangle\|^2} = 1 . \]

On the other hand, the reflection over \( |\pi\rangle \) is not straightforward. One can devise an approximate scheme to implement this reflection using the phase estimation algorithm. Indeed, one can build a unitary operator, using phase estimation applied to the quantum walk operators, which approximates the reflection over \( |\pi\rangle \). Before we state the theorem regarding this approximate reflection operator (constructively proven in [26]), we will first give another result regarding the spectrum of the quantum walk operator, which will be relevant to us presently.

**Theorem 3.** (Szegedy [25]) Let \( P \) be an irreducible, reversible MC with stationary distribution \( \pi \). Then the quantum walk operator \( W(P) \) is such that:

1. \( W(P) |\pi\rangle = |\pi\rangle . \)
2. \( W(P) |\psi\rangle = \exp(\pm 2i\theta) |\psi\rangle , \) where \( \cos(\theta) = \lambda \) is an eigenvalue of \( P \) and \( |\psi\rangle \in \text{Span} \{ |i\rangle |p_i\rangle \} + \text{Span} \{ |p^*_j\rangle |j\rangle \} \).
3. \( W(P) \) has no other eigenvalue in \( \text{Span} \{ |i\rangle |p_i\rangle \} + \text{Span} \{ |p^*_j\rangle |j\rangle \} \).

Let us note that the phase gap \( \Delta \) i.e. the minimum nonzero \( 2\theta \) is such that \( \cos \Delta = |\lambda_2| \). One can then, with some algebra, conclude that \( \Delta \geq 2\sqrt{3} \).

Let us note that any unitary able to approximately detect whether the eigenvalue of \( W(P) \) of a state in \( \text{Span} \{ |i\rangle |p_i\rangle \} + \text{Span} \{ |p^*_j\rangle |j\rangle \} \) is different from one (or equivalently, its eigenphase is different from zero) and conditionally flip the state, will do. We will use such a unitary to approximate \( \text{ref}(\pi) \). Let us use this intuition to build such a unitary, \( R(P) \), that takes as a parameter the precision \( s \) and refer to it in the following as the approximate reflection operator \( R(P) \) [26]:

**Theorem 4.** (Approximate Reflection Operator [26]).
Let \( P \) be an ergodic, irreducible Markov chain on a space of size \( |X| \) with (unique) stationary distribution \( \pi \). Let \( W(P) \) be the corresponding quantum Markov chain with phase gap \( \Delta \). Then, if \( s \) is chosen in \( O(\log_2(1/\Delta)) \), for every integer \( k \) there exist a unitary \( R(P) \) that acts on \( 2|\log_2|X|| + ks \) qubits, such that:

1. \( R(P) \) makes at most \( ks2^{s+1} \) calls to the (controlled) \( W(P) \) and \( W(P)^\dagger \).
2. \( R(P) |\pi\rangle |0\rangle^{\otimes ks} = |\pi\rangle |0\rangle^{\otimes ks} \).
3. If \( |\psi\rangle \in \text{Span} \{ |i\rangle |p_i\rangle \} + \text{Span} \{ |p^*_j\rangle |j\rangle \} \) and is orthogonal to \( |\pi\rangle \), then \( || (R(P) + 1) |\psi\rangle |0\rangle^{\otimes ks} || \leq 2^{1-k} \).

By the approximate reflection theorem above, there exists a subroutine \( R(P)(q,k) \), where \( s \) from the statement of Theorem 4 is taken as \( \log_2(q) \), and \( k \) explicitly controls the fidelity of the reflection. Note that, in the definition of the quantum reflecting agent from the main text, the parameter \( q \) was chosen in \( O(1/\sqrt{\delta}) \), and since by a Theorem of Szegedy [25], as we have commented, it holds that \( 1/\sqrt{\delta} \in O(1/\Delta) \), we have that the fidelity of the approximation reflection approaches unity exponentially quickly in \( k \). For the explicit construction of the approximate reflection operators, we refer the reader to [26].
C. The PS model

The PS model is a reinforcement learning agent model, thus it formally fits within the specification provided with Def. 1. Here we will recap the formalization standard PS model introduced in [19] but note that the philosophy of the projective simulation-based agents is not firmly confined in the formal setting we will provide here, as it is more general. PS agents are defined on a more conceptual level as agents whose internal states represent episodic and compositional memory and whose deliberation comprises an association driven hops between memory sequences - so called clips. Nonetheless, the formal definitions we give here allow us to precisely state our main claims. Following this, we provide a formal treatment of a slight generalization of the standard PS model which subsumes both the standard and the reflecting agent model we refer to in the main text, and formally treat in this Appendix later.

The PS model comprises the percept and action spaces as given in Def. 1. The central component of PS is the so-called episodic and compositional memory (ECM), and it comprises the internal states of the agent. The ECM is a directed weighted network (formally represented as a directed weighted graph) the vertices of which are called clips.

Each clip $c$ represents fragments of episodic experiences, which are formally tuples

$$c = (c^{(1)}, c^{(2)}, \ldots, c^{(L)}), \quad (16)$$

where each $c_k$ is an internal representation of a percept or an action, so

$$c^{(k)} = \mu(i), \quad i \in \mathcal{S} \cup \mathcal{A}, \quad (17)$$

where $\mu$ is a mapping from real percepts and actions to the internal representations. We will assume that each ECM always contains all the unit-length clips denoting elementary percepts and actions.

Within the ECM, each edge between two clips $c_i$ and $c_j$ is assigned a weight $h(c_i, c_j) \geq 1$ and the weights are collected in the $h$-matrix.

The elementary process of the PS agent is a Markov chain, in which the excitations of the ECM hop from one clip to another, where the transition probabilities are defined by the $h$-matrix:

$$p(c_j | c_i) = \frac{h(c_i, c_j)}{\sum_k h(c_i, c_k)}, \quad (18)$$

thus the $h$-matrix is just the non-normalized transition matrix.

In the standard PS model, the decision function is realized as follows: given a percept $s$, the corresponding clip in the ECM is excited and hopping according to the ECM network is commenced. In the simplest case, the hopping process is terminated once a unit-length action clip is encountered, and this action is coupled out and output by the actuator (see Fig. 1). The moment when an action is coupled out can be defined in a more involved way, as we explain presently.

Finally, the update rule, in the standard model necessarily involves the re-definition of the weights in the $h$-matrix.

A prototypical update rule, for a fully classical agent, defining an update from external time-step $t$ to $t+1$ depends on whether an action has been rewarded. If the previous action has been rewarded, and the transition between clips $c_i, c_j$ had actually occurred in the hopping process then the update is as follows:

$$h^{(t+1)}(c_i, c_j) = h^{(t)}(c_i, c_j) - \gamma(h^{(t)}(c_i, c_j) - 1) + \lambda \quad (19)$$

where $\lambda$ is the reward (often $\lambda = 1$) and $0 \leq \gamma \leq 1$ a dissipation (forgetfulness) parameter. If the action had not been rewarded, or the clips $c_i, c_j$ had not played a part in the hopping process then the weights are updated as follows:

$$h^{(t+1)}(c_i, c_j) = h^{(t)}(c_i, c_j) - \gamma(h^{(t)}(c_i, c_j) - 1). \quad (20)$$

The update rule can also be defined such that the update only requires the initial and terminal clip of the hopping process, which is always the case in the simple PS model, where all the clips are just actions or percepts, and hopping always involves a transition from a percept to an action. This example was used in section IV. For that particular example, the update function can be exactly defined by the rules above. As mentioned, aside from the basic structural and diffusion rules, the PS model allows for additional structures, which we repeat here. 1) Emoticons - the agents short term memory, i.e. flags which notify the agent whether the currently found action, given a percept was previously rewarded or not. For our purposes, we shall use only the very rudimentary mode of flags, which designate that the particular action (given a particular percept) was not already unsuccessfully tried before. If it was, the agent can ‘reflect on its decision’ and re-evaluate its strategies, by re-starting the diffusion process. This is an example of a more complicated out-coupling rule we have mentioned previously.

2) Edge and clip glow - mechanisms which allow for the establishing of additional temporal correlations. 3) Clip composition - the PS model based on episodic and compositional memory allows the creation of new clips under certain variational and compositional principles. These allow the agent to develop new behavioral patterns under certain conditions, and allow for a dynamic reconfiguration of the agent itself. For more details we refer the reader to [19, 35].

As illustrated, the PS model allows for a great flexibility. A straightforward generalization would allow for the ECM network to be percept-specific, which is the view we adopt in the definition of reflecting agents.
However, here we will demonstrate that the same notion can be formalized without introducing multiple networks (one for every percept).

In particular, the ECM network allows for action and percept clips to occur a multiple number of times. Thus the ECM network can be represented as $|S|$ disjoint networks, each of which comprises all elementary action clips, and only one elementary percept clip. This structure is clearly within the standard PS model, and it captures all the features of the reflecting PS agent model. A simple case of such a network, relative to the standard picture, is illustrated in Fig. 2, part b). Thus, the reflecting PS agent model is, structurally, a standard PS model as well.

However, we will additionally fix a particular process of deliberation for the reflecting PS agents, which we fully formally treat next.

**D. Reflecting agents**

Here we give the formal definition of the ideal reflecting agent. The classical and quantum realizations of the reflecting agent below differ in the actual method of approximating the decision function of the ideal reflecting agent.

**Definition 7.** (Ideal reflecting agent with flags) An ideal reflecting agent is a reinforcement learning agent as defined in Def. 1 where

- The set of internal states of the agent is

  $$C = \{(g,f)|g \in G, f \in F\},$$

  (21)

  each $g$ is a set of ergodic regular Markov chains over the set of all actions and some subset of percepts, each specified for every percept: $g = \{P_s\}_{s \in S}$, and $P_s$ is an ergodic, regular Markov chain over $S' \cup A$, for $S' \subseteq S$, and $f$ is the set of configurations of flagged actions assigned to each percept:

  $$f = \{f(s)|f(s) \subseteq A|s \in S\}.$$

  (22)

  We will use the shorthand $g(s) = P_s$ in the following. $G$ and $F$ are just sets of all such sets of Markov chains and sets of percept-specific flagged actions.

- The decision function $D$ given a percept $s$ and an internal state $C = (g,f)$ returns a distribution (given as a probability mass function) over the actions $f(s)$ defined as follows. Let $\pi_s$ be the stationary distribution of the Markov chain $g(s)$, and let $\pi_s(i)$, for every $i \in S' \cup A$ denote the probability of sampling the element $i$ from $\pi_s$. Then, for $i \in f(s)$ we define

  $$[D(s,C)](i) = \frac{\pi_s(i)}{\sum_{j \in f(s)} \pi_s(j)}$$

  (23)

  If no flagged actions exist, the agent will respond with an action chosen uniformly at random.

In the definition above we do not specify any particular update function as our results hold for all update functions. While the definition above may seem convoluted, conceptually it is relatively simple and intuitive. For each percept, the agent has an internal network/graph with actions, and other clips (some of the percepts) as vertices. They represent the associative memory of the agent. Instead of simply performing a random walk over this network until an action clip is hit (like in the standard PS model), this agent ‘reflects deeper’ and lets the walk fully mix - that is, obtains the stationary distribution over its clip network. Following this the agent samples from this distribution until a flagged item is hit. Recall, flagged actions correspond to actions that were either not tried or were rewarded by the environment in the previous steps the agent was presented with that same percept, and thus they represent short term memory which allows the agent to be efficient on the scales where the environment does not change its policies.

The output distribution realized by this process maintains the relative probabilities of flagged actions unperturbed. Formally, it can be seen as a normalized projection of the stationary distribution probability vector onto only flagged actions.

**E. Proofs of main theorems**

The definitions of the internal processes of classical and quantum reflecting agents are given in section III B of the main body of this paper. Here, we prove separately that both the classical and the quantum reflecting agent arbitrarily well approximate the ideal reflecting agent in terms of the passive behavior, which we used to define passive behavioral equivalence classes.

From these two theorems, the main theorem given in section III B immediately follows. In the following, when $\pi$ and $\pi'$ are distributions then $||\pi - \pi'||$ denotes the standard variational distance (Kolmogorov distance) on distributions, so

$$||\pi - \pi'|| = \frac{1}{2} \sum_x |\pi(x) - \pi'(x)|.$$

**Theorem 5.** (classical reflecting agents)

Let $P_s$ be the transition matrix of the Markov chain associated to percept $s$, let $f(s)$ be the (non-empty) set of flagged action clips. Furthermore, let $\pi_s(x)$ be the
Let $\kappa$ be the probability distribution over the clips as outputted by the classical reflecting agent, upon receiving $s$. Then the distance $||\kappa - \tilde{\pi}_s||$ is constant (ignoring logarithmic factors), and can be efficiently made arbitrarily small.

Proof. Note that, since the Markov chain is regular, the distribution $P_s^{O(1/\delta)} \pi_0$, for any initial distribution $\pi_0$ is arbitrarily close to the stationary distribution of $P_s$, more precisely,

$$\| P_s^{(k_0+\log(1/\epsilon))/\delta} \pi_0 - \pi \| \leq \epsilon', \tag{25}$$

or, equivalently

$$\| P_s^{(k_0+k_1)/\delta} \pi_0 - \pi \| \leq e^{-k_1}, \tag{26}$$

where $k_0 = \max, \log(\pi(i)^{-1})$. While $\pi(i)^{-1}$ can in principle be very large, it only logarithmically contributes to the overhead, so it can be effectively bounded, and omitted from the analysis. Thus, we can achieve an exponentially good approximation of the stationary distribution with $O(1/\delta)$ iterations. In the remainder we will denote $\pi' = P_s^{O(1/\delta)} \pi_0$.

The reflecting agent mixes its Markov chain (achieving $\pi'$) and then samples from this distribution, iteratively until a flagged action is hit, but at most $t_2$ times. If no flagged action is hit, the agent outputs the uniform distribution over all actions (which we denote $u$).

Thus $\kappa = (1 - p_{\text{fail}}) \pi' + p_{\text{fail}} u$, where $p_{\text{fail}}$ is the probability the agent fails to hit a flagged action when sampling from $\pi'$ on all $t_2$ attempts, and $\tilde{\pi}'$ is the tailed $\pi'$ distribution in the sense of Eq. (24) (substituting $\pi_s$ with $\pi'$).

Hence we have that

$$||\kappa - \tilde{\pi}'|| = ||p_{\text{fail}} u + (1 - p_{\text{fail}}) \pi' - \tilde{\pi}'|| \leq p_{\text{fail}} \| u - \tilde{\pi}' \| + (1 - p_{\text{fail}}) \| \pi' - \tilde{\pi}' \| \leq (27)$$

Next, we shall bound $||\tilde{\pi}' - \tilde{\pi}_s||$. Note that

$$||\tilde{\pi}' - \tilde{\pi}_s|| = \| \frac{1}{\epsilon_{\pi'}} \pi'_\text{sub} - \frac{1}{\epsilon} (\pi_s)_\text{sub} \| \tag{28}$$

where $\epsilon$ and $\epsilon_{\pi'}$ are the probabilities of sampling a flagged action from $\pi_s$ and $\pi'$, respectively, and $(\pi_s)_\text{sub}$ and $(\pi_s)_\text{sub}$ are the sub-normalized distributions $\pi_s = \epsilon_{\pi'} \pi'$ and $(\pi_s)_\text{sub} = \epsilon \tilde{\pi}_s$. Note that it holds that $\| (\pi_s)_\text{sub} - (\pi_s)_\text{sub} \| \leq \| \pi' - \pi_s \|$. So we have that

$$\| \frac{1}{\epsilon_{\pi'}} \pi'_\text{sub} - \frac{1}{\epsilon} (\pi_s)_\text{sub} \| = \| \frac{1}{\epsilon_{\pi'}} \pi'_\text{sub} - \frac{1}{\epsilon} \pi'_\text{sub} + \frac{1}{\epsilon_{\pi'}} \pi'_\text{sub} - \frac{1}{\epsilon} (\pi_s)_\text{sub} \| \leq \| \frac{1}{\epsilon_{\pi'}} - \frac{1}{\epsilon} \| \pi'_\text{sub} \| + \frac{1}{\epsilon} \| \pi'_\text{sub} - (\pi_s)_\text{sub} \| \leq $$

$$\| \frac{1}{\epsilon_{\pi'}} - \frac{1}{\epsilon} \| \pi'_\text{sub} \| + \frac{1}{\epsilon} \| \pi'_\text{sub} - (\pi_s)_\text{sub} \| \leq \| \frac{1}{\epsilon_{\pi'}} - \frac{1}{\epsilon} \| + \| \pi' - \pi_s \| \leq \frac{1}{\epsilon_{\pi'}} - \frac{1}{\epsilon} + e^{-k_1}. \tag{29}$$

Next note that $e^{-k_1} \geq \| \pi' - \pi_s \| \geq |\epsilon_{\pi'} - \epsilon|$, so finally we have

$$\| \tilde{\pi}' - \tilde{\pi}_s \| \leq 2 \frac{e^{-k_1}}{\epsilon} = 2 e^{-k_1 + \log(1/\epsilon)}. \tag{30}$$

Since we are interested in the analysis at the $\tilde{O}$ level, we will, in the end, omit the logarithmically contributing factor above.

Thus we have:

$$||\kappa - \tilde{\pi}_s|| \leq p_{\text{fail}} + (1 - p_{\text{fail}}) 2e^{-k_1 + \log(1/\epsilon)}. \tag{31}$$

Next, we will upper bound $p_{\text{fail}}$. Note that $p_{\text{fail}} = (1 - \epsilon_{\pi'})^2$. Since $\epsilon_{\pi'} = \epsilon + (\epsilon_{\pi'} - \epsilon)$, and $|\epsilon_{\pi'} - \epsilon| \leq e^{-k_1}$, we have that

$$p_{\text{fail}} \leq (1 - \epsilon + e^{-k_1})^2. \tag{32}$$

We can already see that, for a choice of $k_1$ such that $k_1 \geq \log(2/\epsilon)$, we have that $p_{\text{fail}} \leq (1 - \epsilon/2)^2$, which clearly decays exponentially quickly in $t_2$. In particular, if $t_2$ is chosen in $O(2/\epsilon)$ (which is equal to $O(1/\epsilon)$) say $t_2 = 2k_2/\epsilon$ we have

$$p_{\text{fail}} \leq ((1 - \epsilon/2)^2)^{k_2} \leq e^{-k_2}. \tag{33}$$

Hence, with a choice of $k_1 \geq \log(2/\epsilon)$ we have that

$$||\kappa - \tilde{\pi}_s|| \leq e^{-k_2} + 2e^{-k_1 + \log(1/\epsilon)}. \tag{34}$$

Thus, with choices of $t_1 \in \tilde{O}(1/\delta)$ and $t_2 \in \tilde{O}(1/\epsilon)$ we can have the tailed distribution of the classical reflecting agent efficiently arbitrarily close to the (ideal) tailed distribution, which proves this theorem. 

In the proof above, as in the proof which follows, we have assumed that the flagged sets of actions are non-empty, as only in that case the tailed distribution is defined. Operatively, having no flagged actions would mean that the agent tried all the possible actions for a given percept, and none were rewarded. In this case, the agent should conclude (via a simple check of its memory) that the environment has changed, and redefine the flag status. In particular, it may reset the flags to all actions, but this depends on the details of the construction of the particular reflecting agent. Next, we prove an analogous theorem for the quantum agents.

probability mass function of the stationary distribution $\pi_s$ of $P_s$ and let $\tilde{\pi}_s$ be the renormalized distribution of $\pi_s$, where the support is retained only over the flagged actions, so:

$$\tilde{\pi}_s(i) = \frac{\pi_s(i)}{\sum_{j \in f(s)} \pi_s(j)} \text{ if } i \in f(s), \tag{24}$$

0, otherwise

Thus, with choices of $t_1 \in \tilde{O}(1/\delta)$ and $t_2 \in \tilde{O}(1/\epsilon)$ we can have the tailed distribution of the classical reflecting agent efficiently arbitrarily close to the (ideal) tailed distribution, which proves this theorem. 

In the proof above, as in the proof which follows, we have assumed that the flagged sets of actions are non-empty, as only in that case the tailed distribution is defined. Operatively, having no flagged actions would mean that the agent tried all the possible actions for a given percept, and none were rewarded. In this case, the agent should conclude (via a simple check of its memory) that the environment has changed, and redefine the flag status. In particular, it may reset the flags to all actions, but this depends on the details of the construction of the particular reflecting agent. Next, we prove an analogous theorem for the quantum agents.
Theorem 6. (quantum reflecting agents) Let $P_s$ be the transition matrix of the Markov chain associated to percept $s$, let $f(s)$ be the (non-empty) set of flagged action clips. Furthermore, let $\tilde{\pi}_s(i)$ be the probability mass function of the stationary distribution $\pi_s$ of $P^s$ and let $\tilde{\pi}_s$ be the renormalized distribution of $\pi_s$, where the support is retained only over the flagged actions, so:

$$\tilde{\pi}_s(i) = \begin{cases} 
\frac{\pi_s(i)}{\sum_{j \in f(s)} \pi_s(j)}, & \text{if } i \in f(s) \\
0, & \text{otherwise}
\end{cases}$$

(35)

Let $\kappa$ be the probability distribution over the clips as output by the quantum reflecting agent, upon receiving $s$. Then the distance $\|\kappa - \tilde{\pi}\|$ is constant (up to logarithmically contributing terms), and can be efficiently made arbitrarily small.

Proof. The proof consists of two parts. First, we prove that, if the reflection operators (used in the diffusion step of the quantum agent) are ideal, then the claim follows. The remainder of the proof, which considers imperfect reflection operators which are actually used, follows from the proof of Theorem 7 in [48].

Assuming the procedure the agent follows starts from the perfect stationary distribution, and that the reflections over $|\sigma_s\rangle$ are perfect then, by Lemma 6 we have that the state of the system never leaves the span of $|\tilde{\pi}_s\rangle$ and $|\tilde{\pi}_s^\perp\rangle$ where $|\tilde{\pi}_s^\perp\rangle$ is the state orthogonal to $|\tilde{\pi}_s\rangle$ in $\text{Span}\{|\tilde{\pi}_s\rangle, |\sigma_s\rangle\}$.

Note that the agent outputs an action only conditional on the result being a flagged action. This is equivalent to first projecting the state of the system onto the subspace of flagged actions, using the projector $\Pi_{f(s)} = \sum x \in f(s) |x\rangle \langle x|$, followed by renormalization and a measurement. The state after this projection is, for any state in $\text{Span}\{|\tilde{\pi}_s\rangle, |\sigma_s\rangle\}$ a (sub-)normalized state $|\tilde{\pi}_s\rangle$, hence, after normalization, measurement outcomes always follow the distribution $\tilde{\pi}_s$.

However, any state in $\text{Span}\{|\tilde{\pi}_s\rangle, |\sigma_s\rangle\}$ which is not exactly $|\tilde{\pi}_s\rangle$ still has a non-zero support on the non-flagged clips. However, it is the key feature of Grover-like search algorithms (like the search algorithm in [48], the reflections of which make up the decision process of the quantum agent) that the reflection iterations produce a state which has a constant overlap with the target state (in our case $|\tilde{\pi}_s\rangle$), which has support only over flagged actions. This implies that the probability of failing to hit a flagged action is at most some constant $\beta < 1$. Now, if the the deliberation process is iterated some constant $k_3$ number of times it is straightforward to see that $\|\kappa - \tilde{\pi}_s\|$ $\leq 3^{k_3}$, thus decays exponentially quickly as desired.

Next, we need to consider the errors induced by the approximate reflections.

The analysis in [48] (proof of Theorem 7) directly shows that the terminal quantum (after $t_2' \in [0, t_2]$ iterations) state $|\psi\rangle$ is close to the state $|\phi\rangle$ the algorithm would produce had the reflections been perfect, formally,

$$\|\psi - |\phi\rangle\| \leq 2^{1-c},$$

(36)

where $c$ is a constant only additively increasing the internal parameter $k$ of the approximate reflection subroutine.

Thus the error on the final state produced by the quantum agent induced by the approximate reflection algorithm can be made arbitrarily small within the allowed cost $O(1/\sqrt{\lambda})$, which also means that all distributions obtained by measurements of these states will not differ by more than $4 \times 2^{1-c}$.

Returning to the inequality proven for the perfect reflections, we see that the error of magnitude $4 \times 2^{1-c}$ can only increase the probability of failing to output a flagged action from $\beta$ to $4 \times 2^{1-c} + \beta$. Thus, by tuning the parameters only linearly, we can make sure the agent produces an action with an exponentially high probability in terms of the parameter $k_3$ (the number of iterations of the deliberation process). If an action has been produced, then, as we have shown above, it holds that it has been sampled from a distribution within $4 \times 2^{1-c}$ distance of the tailed distribution $\tilde{\pi}_s$, which again can be efficiently made exponentially small. Thus the theorem holds.

We note that if the repetition of the deliberation process is required then a way of re-creating the desired initial coherent encoding of the stationary distribution can be achieved by ‘inverting’ the quantum search algorithm in [26] by ‘unsearching’ for all all non-action clips. This will, with high fidelity, again recreate a good approximation of the initial stationary distribution.

A corollary of the two theorems above is that the classical and quantum agents are passively approximately equal. But then, from the definitions of these embodied agents we can see that the quantum agent exhibits a consistent quadratic speed up, in terms of the required number of applications of their elementary operations. That is, the quantum agent is quadratically faster as claimed. This proves the main theorem stated in the main text. 

\[\text{\footnotesize \[\text{\footnotesize \[\text{\footnotesize \[\text{\footnotesize \footnote{Additionally the quality of the approximation can be arbitrarily raised by one round of phase estimation algorithm for the walk operator, provided no phase is detected. In the case this fails, the agent simply outputs the uniform distribution over actions. Crucially, this occurs only with exponentially small probability.}}\]}}\]}}\]
the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA ’05, 1109–1117 (Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2005). URL http://dl.acm.org/citation.cfm?id=1070452.1070591.

[37] Temme, K., Osborne, T. J., Vollbrecht, K. G. H., Poulin, D. & Verstraete, F. Quantum metropolis sampling 471, 87–90 (2011).

[38] Yung, M.-H. & Aspuru-Guzik, A. A quantum quantum metropolis algorithm. Proceedings of the National Academy of Sciences (2012).

[39] Somma, R. D., Boixo, S., Barnum, H. & Knill, E. Quantum simulations of classical annealing processes. Phys. Rev. Lett. 101, 130504 (2008). URL http://link.aps.org/doi/10.1103/PhysRevLett.101.130504.

[40] Wocjan, P. & Abeyesinghe, A. Speedup via quantum sampling. Phys. Rev. A 78, 042336 (2008). URL http://link.aps.org/doi/10.1103/PhysRevA.78.042336.

[41] Paparo, G. D. & Martin-Delgado, M. Google in a quantum network. Scientific reports 2 (2012).

[42] Paparo, G., Mueller, M., Comellas, F. & Martin-Delgado, M. Quantum google in a complex network. Scientific reports 3 (2013).

[43] Boyer, M., Brassard, G., Hoeyer, P. & Tapp, A. Tight bounds on quantum searching. Fortsch. Phys. 46, 493–506 (1996). arXiv:quant-ph/9605034.

[44] Aspuru-Guzik, A. & Walthier, P. Photonic quantum simulators. Nature Physics 8, 285–291 (2012). URL http://dx.doi.org/10.1038/nphys2253.

[45] Blatt, R. & Roos, C. F. Quantum simulations with trapped ions. Nature Physics 8, 277–283 (2012). URL http://dx.doi.org/10.1038/nphys2252.

[46] Schaetz, T., Monroe, C. R. & Esslinger, T. Focus on quantum simulation. New Journal of Physics 15, 085009 (2013). URL http://stacks.iop.org/1367-2630/15/i=8/a=085009.

[47] Sinclair, A. Algorithms for random generation and counting: a Markov chain approach, vol. 7 (Springer, 1993).

[48] Magniez, F., Santha, M. & Szegedy, M. Quantum algorithms for the triangle problem. SIAM Journal on Computing 37, 413–424 (2007). URL http://epubs.siam.org/doi/abs/10.1137/050643684. http://epubs.siam.org/doi/pdf/10.1137/050643684.

[49] Bernstein, E. & Vazirani, U. Quantum complexity theory. In in Proc. 25th Annual ACM Symposium on Theory of Computing, ACM, 11–20 (1993).