Quantum-Enhanced Machine Learning

Vedran Dunjko,[1,*] Jacob M. Taylor,[2,3,†] and Hans J. Briegel[1,‡]

[1]Institut für Theoretische Physik, Universität Innsbruck, Technikerstraße 21a, A-6020 Innsbruck, Austria

[2]Joint Quantum Institute, National Institute of Standards and Technology, Gaithersburg, Maryland 20899, USA

[3]Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, Maryland 20742, USA

(Received 15 April 2016; published 20 September 2016)

The emerging field of quantum machine learning has the potential to substantially aid in the problems and scope of artificial intelligence. This is only enhanced by recent successes in the field of classical machine learning. In this work we propose an approach for the systematic treatment of machine learning, from the perspective of quantum information. Our approach is general and covers all three main branches of machine learning: supervised, unsupervised, and reinforcement learning. While quantum improvements in supervised and unsupervised learning have been reported, reinforcement learning has received much less attention. Within our approach, we tackle the problem of quantum enhancements in reinforcement learning as well, and propose a systematic scheme for providing improvements. As an example, we show that quadratic improvements in learning efficiency, and exponential improvements in performance over limited time periods, can be obtained for a broad class of learning problems.

DOI: 10.1103/PhysRevLett.117.130501

Introduction.—The field of artificial intelligence (AI) has lately had remarkable successes, especially in the area of machine learning [1,2]. A recent milestone, until recently believed to be decades away—a computer beating an expert human player in the game of Go [3]—clearly illustrates the potential of learning machines. In parallel, we are witnessing the emergence of a new field: quantum machine learning (QML), which has a further, profound potential to revolutionize the field of AI, much like quantum information processing has influenced its classical counterpart [4].

The evidence for this is already substantiated with improvements reported in classification and clustering [5–8] problems. Such tasks are representative of two of the three main branches of machine learning. The first, supervised learning, considers the problem of learning the conditional distribution $P(y|x)$ [e.g., a function $y = f(x)$], which assigns labels y to data x (i.e., classifies data), based on correctly labeled examples, called the training set, provided from a distribution $P(x, y)$. The second, unsupervised learning, uses samples to identify a structure in a distribution $P(x)$, e.g., identifies clusters. The quantum analog of the first task corresponds to a tomography-type problem where conditional states ρ_Y^x (states of a partition of a system, given a measurement outcome of another partition) should be reconstructed from the measurement statistics of the joint state ρ_{XY}, which encodes the distribution $P(x, y)$. The unsupervised case is similar.

The third branch, reinforcement learning (RL), constitutes an interactive mode of learning, and is more general. Here, the learning agent (or learning algorithm) learns how to behave correctly through the use of reinforcement signals—rewards, or punishments. RL has been less investigated from a quantum information perspective, although some results have been reported [9,10].

The key question of how quantum processing can help in learning requires us to clarify what constitutes a good learning model. This can be involved, but two characteristics are typically considered. The first is the computational complexity of the algorithm of the learner. The second, sample complexity, is standard for supervised learning, and quantifies how large the training set has to be, for the algorithm to learn the distribution $P(y|x)$. That is, in a tomography context, it counts the number of copies of ρ_{XY} required until the learning algorithm can reconstruct the states ρ_Y^x to the desired confidence.

In RL, sample complexity is usually substituted by learning efficiency—the number of interaction steps needed for the agent to learn to obtain the rewards with high probability. The recent results in QML have focused on improving computational complexity [5–8,10], with only a few recent works considering sample complexity aspects [11] or supervised computational learning [12,13]. However, the broader question of how, and to what extent, AI can ultimately benefit from quantum mechanics, in general learning settings, remains largely open.

In this work we address this question, with emphasis on the more general, and less explored, RL setting. We propose a paradigm for considering QML, which allows us to better understand its limits and its power. Using this, we present a schema for identifying settings where quantum effects can help. To illustrate how the schema works, we provide a method for achieving quantum improvements (polynomial in the required number of interaction rounds and exponential improvements in the success rate) in many RL settings.

A paradigm for QML.—All three learning settings fit in the paradigm of so-called learning agents [14], standard in the field of artificial intelligence. Here, we consider a learning agent A (equivalently a learning program A) that interacts with an unknown environment E (the so-called task environment, or problem setting) via the exchange of messages, interchangeably issued by A (called actions $\mathcal{A} = \{a_i\}$) and E (called percepts $\mathcal{S} = \{s_j\}$). In the quantum extension, these sets become Hilbert spaces, $\mathcal{H}_\mathcal{A} = \mathrm{span}\{|a_i\rangle\}$, $\mathcal{H}_\mathcal{S} = \mathrm{span}\{|s_i\rangle\}$, and form orthonormal bases. The percept and action states, and their mixtures, are referred to as classical states. Any figure of merit $\mathrm{Rate}(\cdot)$ of the performance of an agent A in E is a function of the history of interaction $\mathbf{H} \ni h = (a_1, s_1, \ldots)$, collecting the exchanged percepts and actions. The history of interaction is thus the central concept in learning. The correct quantum generalization of the history is not trivial, and we will deal with this momentarily.

If either A or E is stochastic, the interaction of A and E is described by a distribution over histories (of length t), denoted by $A \leftrightarrow_t E$. Most figures of merit are then extended to such distributions by convex linearity.

To recover, e.g., supervised learning in this paradigm, take E to be characterized by the distribution $P(x, y)$, where the agent is given the training set—n labeled data points [pairs (x, y)] sampled from $P(x, y)$—as the first n percepts. After this, the agent is to respond with the correct labels as actions (responses) to the presented percepts, which are now the unlabeled data points x. Reinforcement learning is naturally phrased as such an agent-environment interaction, where the percept space also contains the reward. We denote the percept space including the reward status as $\overline{\mathcal{S}}$ (e.g., if rewards are binary then $\overline{\mathcal{S}} = \mathcal{S} \times \{0, 1\}$).

Formally, the agent-environment paradigm is a two-party interactive setting, and thus convenient for a quantum information treatment of QML. All the existing results group into four categories [15]: *CC*, *CQ*, *QC*, and *QQ*, depending on whether the agent (first symbol) or the environment (second symbol) are classical (*C*) or quantum (*Q*). The *CC* scenario covers classical machine learning. The *CQ* setting asks how classical learning techniques may aid in quantum tasks, such as quantum control [16,17], quantum metrology [18], adaptive quantum computing [19], and the design of quantum experiments [20]. Here we deal with, for example, nonconvex or nonlinear optimization problems arising in quantum experiments, tackled by machine learning techniques. *QC* corresponds to quantum variants of learning algorithms [7,10,21] facing a classical environment. Figuratively speaking, this studies the potential of a learning robot, enhanced with a "quantum chip." In *QQ* settings, the focus of this work, both A and E are quantum systems. Here, the interaction can be fully quantum, and even the question of what it means "to learn" becomes problematic as, for instance, the agent and environment may become entangled.

Framework.—Since learning constitutes a two-player interaction, standard quantum extensions can be applied: the action and percept sets are represented by the aforementioned Hilbert spaces $\mathcal{H}_\mathcal{A}$, $\mathcal{H}_\mathcal{S}$. The agent and the environment act on a common communication register R_C (capable of representing both percepts and actions). Thus, the agent (environment) is described as a sequence of completely positive trace-preserving maps $\{\mathcal{M}_A^t\}(\{\mathcal{M}_E^t\})$—one for each time-step—that acts on the register R_C, but also a private register R_A (R_E) that constitutes the internal memory of the agent (environment). This is illustrated in Fig. 1 above the dashed line.

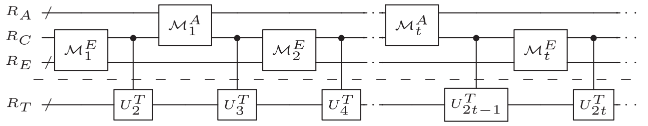

FIG. 1. Tested agent-environment interaction. In general, each map of the tester U_k^T acts on a fresh subsystem of the register R_T, which is not under the control of the agent, nor of the environment. The crossed wires represent multiple systems.

The central object characterizing an interaction, namely, its history, is, for the quantum case, generated by performing periodic measurements on R_C in the classical (often called computational) basis. The generalization of this process for the quantum case is a tested interaction: we define the tester as a sequence of controlled maps of the form

$$U_t^T(|x\rangle_{R_C} \otimes |\psi\rangle_{R_T}) = |x\rangle_{R_C} \otimes U_t^x |\psi\rangle_{R_T},$$

where $x \in \mathcal{S} \cup \mathcal{A}$, and $\{U_t^x\}_x$ are unitary maps acting on the tester register R_T, for all steps t. The history, relative to a given tester, is defined to be the state of the register R_T. A tested interaction is shown in Fig. 1.

The restriction that testers are controlled maps relative to the classical basis guarantees that, for any choice of the local maps U_T^x, the interaction between classical A and E remains unchanged. A classical tester copies the content of R_C relative to the classical basis, which has essentially the same effect as measuring R_C and copying the outcome. In other words, the interface between A and E is then classical. It can be shown that, in the latter case, for any quantum agent and/or environment there exist classical A and E that generate the same history under any tester [22]. In other words, classical agents can, in *QC* settings and, equivalently, in classically tested *QQ* settings, achieve the same performance as quantum agents, in terms of any history-dependent figure of merit. Thus, the only improvements can then be in terms of computational complexity.

Scope and limits of quantum improvements.—What is the ultimate potential of quantum improvements in learning? In the *QC* and classically tested settings, we are bound to computational complexity improvements, which have been achieved in certain cases. Improvements in learning efficiency require a special type of access to the environments, which is not fully tested. Exactly this is done in Refs. [6,8], for the purpose of improving computational complexity, with great success, as the improvement can be exponential. There, the classical source of samples is substituted by a quantum RAM [32] architecture, which allows for the accessing of many samples in superposition. Such a substitution comes naturally in (un)supervised settings, as the basic interaction comprises only two steps and is memoryless—the agent requests M samples, and the environment provides them. However, in more general settings, environments are ill suited for such quantum parallel approaches: in general, the environment stores all the actions of the agent in its memory, never to return them again. This effectively breaks the entanglement in the agent's register R_A, and prohibits all interference effects. Nonetheless, for many environmental settings, it is still possible to "dissect" the maps of the environment, and to provide oracular variants, which we can use to help the agent learn.

*An approach to quantum improvements in reinforcement learning.—*This brings us to our schema for improving RL agents. First, given a classical environment E, we define fair unitary oracular equivalents E^q. Here, fair is meant in the same sense as quantum oracles of boolean functions are fair analogues of classical boolean functions—E^q should not provide more information than E under classical access, which is guaranteed, e.g., when E^q is realizable from a reversible version of E. Second, as access to any quantum environment E^q cannot generically speed up all aspects of an interaction (e.g., while quantum walks can find target vertices faster, the price is that the traversed path is undefined), we identify particular environmental properties that can be more efficiently ascertained using E^q, and that are relevant for learning. Third, we construct an improved agent that uses the properties from the previous points. We now illustrate our approach on a restricted scenario, for the ease of presentation, and show how the examples generalize later.

*Application of the framework.—*Given any task environment, we can separately consider the map that specifies the next percept the environment will present—in general, a stochastic function $f_E\colon \mathbf{H} \to \mathcal{S}$, mapping elapsed histories onto the next percept—and the reward function. The latter is described as the map $\Lambda\colon \mathbf{H} \times \mathcal{S} \to \overline{\mathcal{S}}$, which also depends on the history, and complements the percept by setting its reward status. In environments that are simple and strictly epochal (meaning the environment is reset after M steps and at most one reward is given), although the interaction is turn based, it can be represented as sequences of M-step maps:

$$|a_1, \ldots, a_M\rangle \to |s_1, \ldots, \bar{s_M}\rangle \tag{1}$$

where the "bar" on s_M highlights that it includes a reward status. Moreover, in deterministic environments, the maps f_E and Λ only depend on the actions of the agent, as the percept responses are fixed. For such deterministic, simple strictly epochal environments, the construction of an appropriate oracle is dramatically simplified. The actions can be returned to the agent after each block of M steps, as the next block is independent. Moreover, using phase kickback, the reward map can be modified [22] such as to influence just the global phase of returned action states. This leads to the "phase-flip" oracle realizing

$$|a_1, \ldots, a_M\rangle \xrightarrow{E^q_{\text{oracle}}} (-1)^{\Lambda(a_1,\ldots,a_M)}|a_1, \ldots, a_M\rangle. \tag{2}$$

One use of this environment-specific oracle requires M interaction steps. This constitutes the first step of our proposed schema. Next, we focus on step 2: obtaining a useful property of the environment, and identifying settings where it provably helps. The constructed oracle points towards the use of a Grover-type search to find rewarding action sequences. This alone suffices for improvements only in special environments where learning reduces to searching. We can do better by combining fast searching with a classical learning model. In canonical RL settings, what the agent learns (should learn) is not a correct sequence of moves *per se*, but rather the correct association of actions given percepts. To illustrate this, imagine navigating a maze where the percepts encode correct directions of movement. If the correct association is learned, then the agent will perform well, even when the maze changes. Nonetheless, for the agent to learn the correct association, it first must encounter an instance of rewarding sequences, and here quantum access helps. Thus, we aim at assisting in the exploration phase of the balancing act between exploration (trying out behaviors to find optima) and exploitation (reaping rewards by using learned information) characteristic for RL [33]. This idea can be made fully precise by considering the class of environments where more successful exploration phases are guaranteed to lead to a better overall learning performance. Whether this is the case, however, also depends on the learning model of the agent. Thus, we identify agent-environment pairs, where such better performance in the past (in exploration) implies better performance in the future (on average), which we call luck-favoring settings.

More formally, consider environments E, and agents A, such that if h_t and h'_t are t-length histories, then $\mathrm{Rate}(h_t) > \mathrm{Rate}(h'_t)$ (i.e., h_t is a history with a better performance than h'_t) implies

$$\mathrm{Rate}\big(E(h_t) \leftrightarrow_T A(h_t)\big) \geq \mathrm{Rate}\big(E(h'_t) \leftrightarrow_T A(h'_t)\big), \tag{3}$$

for some future period T. Here, $E(h)$ and $A(h)$ denote the environment and agent, respectively, that have undergone the history h [note that $A(h)$ and A are, technically, different agents].

We will say $A(h_t)$ is luckier than $A(h'_t)$. Such environment-agent pairs (A, E), satisfying the formal conditions above, are thus luck favoring, and we may additionally specify the periods t and T for which the implication (3) holds. This brings us to step 3 of the schema, given as a theorem.

Theorem 1: Let E be a deterministic, strictly epochal environment. Then, there exists an oracular variant E^q of E, such that for any classical learning model A that is luck favoring relative to E, and a figure of merit Rate that is monotonically increasing in the number of rewards in the history, we can construct a quantum agent A^q such that A^q, by interacting with E^q, outperforms A in terms of the figure of merit Rate relative to a chosen tester.

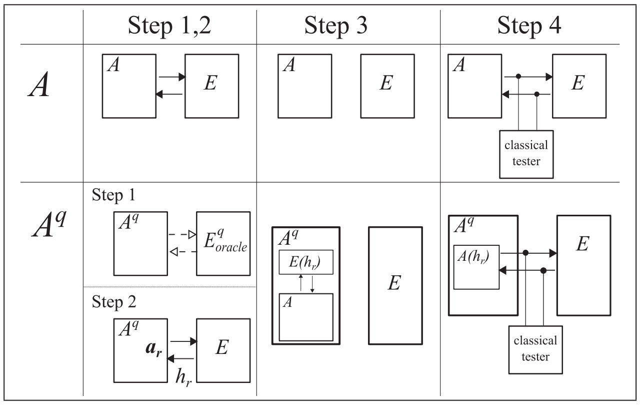

FIG. 2. Differences between the interaction for A and A^q. In steps 1 and 2, A^q uses access to E^q_{oracle}, for $O(t)$ steps, and obtains a rewarding sequence h_r. Step 3: A^q simulates the agent A, and trains the simulation to produce the rewarding sequence. In step 4, A^q uses $A(h_r)$ for the remainder of the now classically tested interaction, with the classical environment E.

This theorem states that, in the restricted settings of deterministic epochal environments, it is possible to generically improve the learning efficiency of all learning agents, provided the environments are luck favoring for those agents. We note that most reasonable learning models are luck favoring relative to most typically considered task environments (see Ref. [22] for a longer discussion). In the statement of Theorem 1, we have omitted additional specifications pertaining to t and T, but it should be understood that if the luck-favoring property holds for t and T, then the improved performance holds relative to these periods.

To prove Theorem 1 we construct A^q, given A. The construction is illustrated step by step in Fig. 2, where for illustrative purposes, the classical interaction of agent A is contrasted against the quantum interaction of agent A^q. In step 1, A^q will use the quantum oracle variant of E (E^q_{oracle}) for time $t \in O(\sqrt{|A|^M})$, where M is the epoch length, and $|A|$ is the number of actions, to find a rewarding action sequence $\mathbf{a}_r$, using a Grover search. During this period the interaction is untested, and the interaction is fully classically tested thereafter. In step 2, A^q will play out one epoch by outputting actions from $\mathbf{a}_r$ sequentially, now with the classical environment, to obtain the responses of the environment (recall, E^q_{oracle} cannot provide these), obtaining the entire rewarding history h_r. Thus far, A^q used $O(M\sqrt{|A|^M})$ interaction steps. In step 3, A^q "trains" an internal simulation of A, simulating the interaction between A and E, and restarting the simulation until the history h_r occurs (we assume such an occurrence has a nonzero probability). This may require many internally simulated interactions, but no interaction with the real environment. In step 4, the internal simulation of $A(h_r)$ corresponds to the luckiest agent possible, and A^q relinquishes control to it.

Finally, we consider what happens with A during the same time periods. Unless additional information about the environment is given, in $O(t)$ steps A has only an exponentially small $\{O(\exp[-M\ln(|A|)/2])\}$ probability of having seen the rewarding sequence. Thus, the quantum agent is luckier than the classical, and in luck-favoring settings this implies that A^q will continue to outperform A after the t steps. The statement of Theorem 1 is not quantitative, due to the generality of the definition of luck-favoring settings. We can, however, trade off generality for exactness. If an agent A employs a variant of ϵ-greedy [33] behavior—that is, it outputs the rewarding sequence (exploits) with probability ϵ and explores with probability $1-\epsilon$, then the ratio of the performances of A^q and A will be exponential in M: the constant reward probability ϵ of A^q versus the exponentially diminishing $O(\exp[-M\ln(|A|)/2])$ of A at step t. This exponential gap holds for time scales $T \in O(t)$. However, the improvement in terms of learning efficiency (number of interaction steps) is quadratic.

Our results achieve solid improvements using simple techniques, at the cost of restricting the task environments. However, our example can be further generalized in two directions.

First, as long as the reset occurs at step M, multiple and multivalued rewards can also be handled by defining oracles that reversibly count the rewards. Highly rewarding sequences can then be found through quantum optimization techniques [34], as worked out in Ref. [22].

Second, under stronger assumptions on E^q, using more involved quantum subroutines, we can deal with stochastic environments. For instance, in the setting with one reward per epoch, the oracle

$$|\mathbf{a}\rangle|0\rangle \overset{\mathcal{U}_E}{\rightarrow} |\mathbf{a}\rangle(\cos\theta_{\mathbf{a}}|0\rangle + \sin\theta_{\mathbf{a}}|1\rangle), \tag{4}$$

where $\sin^2\theta_{\mathbf{a}}$ is the probability of a reward, given the action sequence $\mathbf{a}$, can be constructed from a reversible implementation of the environment where randomness is represented as a subsystem of an entangled state [22].

From here, by using phase kickback and phase estimation the agent can realize the mapping

$$|\mathbf{a}\rangle|\mathbf{0}\rangle \rightarrow |\mathbf{a}\rangle|\tilde{\theta}_{\mathbf{a}}\rangle, \tag{5}$$

where $\tilde{\theta}_{\mathbf{a}}$ is an l-bit precision estimate of the reward probability as specified by the angle $\theta_{\mathbf{a}}$. Next, amplitude amplification is used to amplitude amplify all sequences $\mathbf{a}$ where the reward probability $p_r(\mathbf{a})$, given sequence $\mathbf{a}$, is above a threshold p_{min}.

Given N_{min} such sequences (out of $N_{\mathrm{tot}} \coloneqq |\mathcal{A}|^M$ sequences in total), the overall number of interaction steps multiplies M with the amplitude amplification cost $[O((N_{\mathrm{tot}}/N_{\mathrm{min}})^{1/2})]$, and with phase estimation cost $[O(1/p_{\mathrm{min}})]$. Overall, we have $O(M(N_{\mathrm{tot}}/N_{\mathrm{min}})^{1/2}p_{\mathrm{min}}^{-1})$ interaction steps. The classical agent's interaction cost of the same process is $O(MN_{\mathrm{tot}}/N_{\mathrm{min}})$.

If the minimal relevant success probability is constant for a family of task environments, then this constitutes a quadratic improvement in finding good action sequences. This approach can also be generalized to a wider class of settings [22].

In many settings, e.g., robotics, the classical environments do not allow "oracularization." Nonetheless, the presented constructions can be used in model-based learning [14], where the agent constructs an internal representation of the environment to facilitate better learning through simulation. Then, the quantum chip can help in speeding up internal processing, which is the most that can be done in *QC* settings. A tantalizing exception to this may be nanoscale robots (e.g., intelligent versions of *in situ* probes in Ref. [19]) in future quantum experiments, as on these scales the environment is manifestly quantum and exquisite control becomes a possibility.

Conclusions.—In this work we have extended the general agent-environment framework of artificial intelligence [14] to the quantum domain. Based on this, we have established a schema for quantum improvements in learning, beyond computational complexity. Using this schema, we have given explicit constructions of quantum-enhanced reinforcement learning agents, which outperform their classical counterparts quadratically in terms of learning efficiency, or even exponentially in performance over limited periods. This constitutes an important step towards a systematic investigation of the full potential of quantum machine learning, and the first step in the context of reinforcement learning under quantum interaction.

V. D. and H. J. B. acknowledge support by the Austrian Science Fund (FWF) through Grant No. SFB FoQuS F 4012 and the Templeton World Charity Foundation through Grant No. TWCF0078/AB46. V. D. thanks Christopher Portmann, Petros Wallden, and Peter Wittek for useful discussions that helped in parts of this work.

[*]vedran.dunjko@uibk.ac.at
[†]jmtaylor@umd.edu
[‡]hans.briegel@uibk.ac.at

[1] T. Chouard and L. Venema, Machine intelligence, Nature **521**, 435, 2015.

[2] J. Stajic, R. Stone, G. Chin, and B. Wilbe, Rise of the machines, Science **349**, 248, 2015.

[3] D. Silver *et al.*, Mastering the game of Go with deep neural networks and tree search, Nature (London) **529**, 484 (2016).

[4] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, England, 2000).

[5] P. Wittek, *Quantum Machine Learning: What Quantum Computing Means to Data Mining* (Academic Press, New York, 2014).

[6] S. Lloyd, M. Mohseni, and P. Rebentrost, Quantum algorithms for supervised and unsupervised machine learning, arXiv:1307.0411.

[7] E. Aïmeur, G. Brassard, and S. Gambs, Quantum speed-up for unsupervised learning, Mach. Learn. **90**, 261 (2013).

[8] P. Rebentrost, M. Mohseni, and S. Lloyd, Quantum support vector machine for big data classification, Phys. Rev. Lett. **113**, 130503 (2014).

[9] D. Dong, C. Chen, H. Li, and T. J. Tarn, Quantum reinforcement learning, IEEE Trans. Syst. Man Cybern. B Cybern. **38**, 1207 (2008).

[10] G. D. Paparo, V. Dunjko, A. Makmal, M. A. Martin-Delgado, and H. J. Briegel, Quantum speedup for active learning agents, Phys. Rev. X **4**, 031002 (2014).

[11] S. Arunachalam and R. de Wolf, Optimal quantum sample complexity of learning algorithms, arXiv:1607.00932.

[12] R. A. Servedio and S. J. Gortler, Quantum versus classical learnability, in *16th Annual IEEE Conference on Computational Complexity, Chicago, Illinois, 2001* (IEEE Computer Society, Los Alamitos, 2001), p. 138.

[13] A. Atıcı, Advances in quantum computational learning theory, Ph.D. thesis, Columbia University, 2006.

[14] S. J. Russel and P. Norvig, *Artificial Intelligence—A Modern Approach*, 2nd ed. (Prentice Hall, New Jersey, 2003).

[15] E. Aïmeur, G. Brassard, and S. Gambs, Machine learning in a quantum world, *Advances in Artificial Intelligence: 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006*, (Springer, Berlin, Heidelberg, 2006), pp. 431–442.

[16] E. Zahedinejad, S. Schirmer, and B. C. Sanders, Evolutionary algorithms for hard quantum control, Phys. Rev. A **90**, 032310 (2014).

[17] H. M. Wiseman and G. J. Milburn, *Quantum Measurement and Control* (Cambridge University Press, Cambridge, England, 2010).

[18] N. B. Lovett, C. Crosnier, M. Perarnau-Llobet, and B. C. Sanders, Differential evolution for many-particle adaptive quantum metrology, Phys. Rev. Lett. **110**, 220501 (2013).

[19] M. Tiersch, E. J. Ganahl, and H. J. Briegel, Adaptive quantum computation in changing environments using projective simulation, Sci. Rep. **5**, 12874 (2015).

[20] M. Krenn, M. Malik, R. Fickler, R. Lapkiewicz, and A. Zeilinger, Automated search for new quantum experiments, Phys. Rev. Lett. **116**, 090405 (2016).

[21] M. Schuld, I. Sinayskiy, and F. Petruccione, The quest for a quantum neural network, Quant. Inf. Process. **13**, 2567 (2014).

[22] See Supplemental Material at http://link.aps.org/supplemental/10.1103/PhysRevLett.117.130501, which includes Refs. [23–31], for full detailed proofs of the statements made in the main text, and also extensions of the results presented.

[23] V. Dunjko, J. M. Taylor, and H. J. Briegel, Framework for learning agents in quantum environments, arXiv:1507.08482.

[24] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings, 28th Annual ACM Symposium on the Theory of Computing, Philadelphia, 1996* (ACM Press, Philadelphia, 1996), p. 212.

[25] W. Baritompa, D. W. Bulger, and G. R. Wood, A Grover's quantum algorithm applied to global optimization, SIAM J. Optim. **15**, 1170 (2005).

[26] R. S. Sutton, Integrated architectures for learning, planning, and reacting based on approximating dynamic programming, *Proceedings of the Seventh International Workshop on Machine Learning, Austin TX, 1990* (Morgan Kaufmann, San Francisco, 1990), pp. 216–224.

[27] H. J. Briegel and G. De las Cuevas, Projective simulation for artificial intelligence, Sci. Rep. **2**, 400 (2012).

[28] A. Makmal, A. A. Melnikov, V. Dunjko, and H. J. Briegel, Meta-learning within Projective Simulation, IEEE Access **4**, 2110 (2016).

[29] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, Tight bounds on quantum searching, Fortschr. Phys. **46**, 493 (1998).

[30] T. J. Yoder, G. H. Low, and I. L. Chuang, Fixed-point quantum search with an optimal number of queries, Phys. Rev. Lett. **113**, 210501 (2014).

[31] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, arXiv:quant-ph/0005055.

[32] V. Giovannetti, S. Lloyd, and L. Maccone, Quantum random access memory, Phys. Rev. Lett. **100**, 160501 (2008).

[33] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 1st ed. (MIT Press, Cambridge, Massachusetts, 1998).

[34] C. Durr and P. Hoyer, A quantum algorithm for finding the minimum, arXiv:quant-ph/9607014.