Learning coherent states by trial and error

M. Bilkis,[1] R. Morral Yepes,[1] M. Rosati,[1] and J. Calsamiglia[1]

[1] *Física Teòrica: Informació i Fenòmens Quàntics, Departament de Física, Universitat Autònoma de Barcelona, ES-08193 Bellaterra (Barcelona), Spain*

The optimal discrimination of coherent states of light with current technology is a key problem in classical and quantum communication, whose solution would enable the realization of efficient receivers for long-distance communications in free-space and optical fiber channels. In this article, we show that reinforcement learning (RL) protocols allow an agent to learn near-optimal coherent-state receivers made of passive linear optics, photodetectors and classical adaptive control. Each agent is trained and tested in real time over several runs of independent discrimination experiments and has no knowledge about the energy of the states nor the receiver setup nor the quantum-mechanical laws governing the experiments. Based exclusively on the observed photodetector outcomes, the agent adaptively chooses among a set of $\sim 10^3$ possible receiver setups, and obtains a reward at the end of each experiment if its guess is correct. At variance with previous applications of RL in quantum physics, the information gathered in each run is intrinsically stochastic and thus insufficient to evaluate exactly the performance of the chosen receiver. Nevertheless, we present families of agents that: (i) discover a receiver beating the best Gaussian receiver after $\sim 3\ 10^2$ experiments; (ii) surpass the cumulative reward of the best Gaussian receiver after $\sim 10^3$ experiments; (iii) simultaneously discover a near-optimal receiver and attain its cumulative reward after $\sim 10^5$ experiments. Our results show that RL techniques are suitable for on-line control of quantum receivers and can be employed for long-distance communications over potentially unknown channels.

I. INTRODUCTION

Quantum state discrimination (QSD) is the problem of determining the state of a quantum system among a set of possible candidates. It constitutes a fundamental primitive in quantum information processing, with applications ranging from long-distance communication [1–9], cryptography [10–17] and, recently, quantum machine learning [18–26].

In the past few years, the use of machine learning methods to deepen the understanding of fundamental physics has become a standard technique [27–36]. Machine learning can be classified as supervised, unsupervised and reinforcement learning (RL). In particular, RL studies the behaviour of an agent interacting with an environment via observations, actions and rewards. The goal is to optimize such interactions in order to maximize a suitable figure of merit, e.g., the expected reward over time. Combinations of these three machine-learning classes have recently led to out-performing the best human GO player, discovering strategies never played before [37]. Recently, RL techniques have also been proved successful in quantum information, e.g., in the design of novel quantum experiments [32], quantum error-correction codes [33], quantum communication protocols [34] and optimal control of quantum systems [35, 36].

In the present work we consider the discrimination of two coherent states with passive linear optics, photodetectors and discrete-time classical adaptive control. This is a prototypical problem in quantum information theory [1, 38, 39], of great technological significance for long-distance communication [4–8, 40]: the optimal measurement to discriminate two coherent states is known [1, 38] but its implementation is demanding at the state of the art, i.e., via the so-called Dolinar receiver [40–50] that requires asymptotically many control rounds. Moreover, its extension to multiple states is not fully understood [48, 51, 52], although it may bring us a step closer to achieving the Holevo communication capacity of real-world channels.

We propose an innovative and experimentally appealing approach to the problem: the search for optimal discrimination strategies is cast as a test-bed for RL, by studying how well an agent can perform in calibrating a receiver by means of *model-free* methods. The nature of our approach is particularly appealing for scenarios where an accurate description of the system is not possible, e.g., due to intrinsic complexity, experimental constraints or imperfections, untrusted devices or simply lack of knowledge. This is precisely the case for applications of coherent-state discrimination in communication scenarios, where discriminating multiple hypotheses may require tuning long sequences of gate parameters [53–56], the detectors may be affected by losses and dark-counts [43, 52], the actual communication channel may add different kinds of noise depending on the physical implementation [4, 5, 57, 58] and device-independent security may be additionally required [13, 17].

In this article we show that a RL agent can achieve near-optimal control of a coherent-state detector when it has zero prior knowledge of: (i) the energy of the coherent states themselves; (ii) the actual operations that the detector performs; (iii) the underlying quantum-mechanical laws governing the system. By trial and error, the RL agent has to sequentially press buttons and select actions according to previous measurement outcomes and at a final stage guess for one of the possible hypotheses. A non-zero reward is given only if the guess is correct. By repeating the procedure over several episodes (or runs), the agent earns experience and *learns a near-optimal dis-*

crimination protocol and guessing rule with the resources at its disposal.

Our approach differs from previous applications of RL in quantum information [32–36] at least in three crucial aspects: (i) our agents can simultaneously learn and be tested in a completely model-free setting; (ii) each reward is obtained directly from a single-shot experiment and not indirectly inferred from a known model or from several runs of the experiment as in the case were the reward is, say, a target fidelity or a success probability; (iii) we will not only be concerned about finding near-optimal detectors but also, importantly, on the actual on-line success rate of the agents as measured by the cumulative reward.

We tackle the problem in three stages of increasing complexity: first, in the model-aware setting, where the outcome probability function of the receiver is known, we find the optimal action sequence by solving the Bellman equation via dynamic programming [43, 59]; second, in the model-free setting, where the receiver is completely unknown, we apply Watkins' Q-Learning [60, 61], a standard RL method whose update rule approximates the optimal Bellman equation; third, in the model-free setting we study the trade-off between exploiting potential optimal strategies and exploring new ones, by applying two state-of-the-art methods adapted from *bandit theory* [62, 63, 75], thus enhancing the learning speed or accuracy of our agents. With these methods, in the model-aware setting we are able to compute numerically the optimal success probability and set of actions for several control rounds. Moreover, in the model-free setting we are able to construct agents that surpass the performance of the best Gaussian receiver [40] after $\sim 3\ 10^2$ episodes and attain near-optimal performance ($> 97\%$ optimal) after $\sim 10^5$ episodes, searching a parameter-space of size $\sim 3\ 10^3$. Our results provide a flexible and comprehensive ensemble of methods both in the model-aware and model-free settings that enable the on-line optimization of small quantum devices and the benchmarking of their performance. Furthermore, the methods we propose can be enhanced by the use of deep-learning techniques [37], which would allow their application to more complex problems and devices, e.g., multi-state QSD and the study of generalization performance.

The article is organized as follows. In Sec. II we introduce our QSD problem, the receiver architecture and the target function for a RL agent controlling the receiver. In Sec. III we present the theoretical framework of standard RL methods, introducing the state-action value function, the Bellman equation and Q-Learning. In Sec. IV we describe the implementation of these methods and analyze their performance in terms of the cumulative reward. The bandit problem is introduced here as a basic framework to study, quantify and optimize the real-time performance of agents over sequential learning strategies. We analyse and compare the performance of standard and bandit-inspired learning strategies in a variety of experimentally relevant settings. We conclude in Sec. V by mentioning possible extensions of our work.

CONTENTS

II. PRELIMINARIES

We consider the discrimination of two electromagnetic signals with opposite phases, described by two coherent states of the field, $|\pm\alpha\rangle$, whose energy is proportional to $|\alpha|^2$. When the energy of the signals approaches zero, i.e., $|\alpha|^2 \ll 1$, quantum effects become evident and it becomes impossible to discriminate between them perfectly.

Any binary discrimination protocol is described compactly by a quantum positive-operator-valued measurement (POVM), $\mathcal{M} = \{M_0, M_1\}$ with $M_{1,2} \geq 0$ and $M_1 + M_2 = \mathbb{I}$. Defining the k-th hypothesis as $\alpha^{(k)} = (-1)^k\alpha$, with prior probability p_k, the probability of obtaining outcome $\hat{k}$ given that hypothesis k was true is $p(\hat{k}|\alpha^{(k)}) = \langle\alpha^{(k)}| M_{\hat{k}} |\alpha^{(k)}\rangle$ and the best guess is given by the most likely hypothesis given that outcome. Thus the average success probability over all outcomes is given

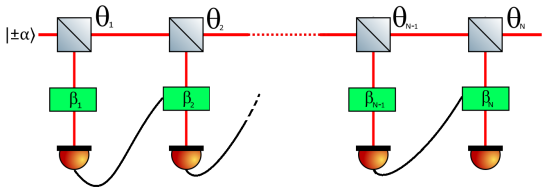

FIG. 1. We depict the experimental setup of the receiver considered. For $L \to \infty$ one gets Dolinar reciever.

by

$$P_s(\alpha, \mathcal{M}) = \sum_{\hat{k}=0,1} \max_{k=0,1} p(\alpha^{(k)}, \hat{k}) = \sum_{\hat{k}=0,1} \max_{k=0,1} p(\hat{k}|\alpha^{(k)}) p_k. \tag{1}$$

For non-orthogonal quantum states, this quantity is bounded below 1 by the so-called Helstrom bound [1], which in our case reads

$$P_s^{(hel)}(\alpha) = \max_{\mathcal{M}} P_s(\alpha, \mathcal{M}) = \frac{1}{2}\left(1 + \sqrt{1 - e^{-4|\alpha|^2}}\right), \tag{2}$$

where the optimization is carried out over all two-outcome POVMs; note that the Helstrom probability tends to 1/2 for $|\alpha| \to 0$, i.e., the states become indistinguishable at very low energies. The optimal Helstrom measurement that attains Eq. (2) is a difficult projection on a superposition of $|\pm\alpha\rangle$, i.e., a Schrödinger-cat-like state, which cannot be realized with simple linear-optical operations [44]. Quite surprisingly, Dolinar [41] showed that Eq. (2) can be asymptotically attained by continuous-time control of a displacement operator; his receiver has since been extended to the discrete-time scenario by Takeoka et al. [44]. Nevertheless, the practical implementation of these receivers still proves demanding at present [46, 51], due to various experimental limitations. Moreover, in a general communication scenario, the states will be transferred through a noisy channel and could be subject to various kinds of noise, e.g., loss, thermal noise and phase diffusion [52, 58].

Based on these premises, we aim to construct a model-free RL agent that, without any knowledge of the problem at hand nor of the receiver setup, learns to tune the receiver's parameters in order to maximize its success probability. In this way, when placed in a real-life situation, the agent will be able to train and optimize the receiver for the specific experimental conditions it encounters in real time. The receiver we consider comprises passive linear optics, photodetectors and classical feed-forward, structured into successive processing layers $\ell = 0, \cdots, L$, as depicted in Fig. 1; this receiver is known to attain the Helstrom probability in the limit $L \to \infty$ [44]. For each layer $\ell < L$, the following operations are applied:

1. The input signal $|\alpha\rangle$ is split on a beamsplitter (BS) of transmissivity θ, effectively extracting a fraction $1-\theta$ of the energy for detection. The BS transforms the input signal and vacuum states as
$$|\alpha\rangle\,|0\rangle \mapsto |\alpha\sqrt{\theta}\rangle_{\rm tr}|\alpha\sqrt{1-\theta}\rangle_{\rm ref}, \tag{3}$$
where the added phase of the second mode has been corrected via a proper phase-shift, not shown in the figure.

2. The reflected part of the signal undergoes a displacement operation $D(\beta)$, realizable via interference with a strong coherent signal on a small-reflectivity BS, not shown in the figure. The resulting state is $|\tilde{\alpha}(\beta,\theta)\rangle = |\alpha\sqrt{1-\theta} + \beta\rangle$.

3. The displaced signal is measured via a on/off photodetector, which detects no photon, i.e., outcome $o_{\ell+1} = 0$, with conditional probability
$$p(o_{\ell+1} = 0|\alpha, (\beta,\theta)) = |\langle 0|\tilde{\alpha}(\beta,\theta)\rangle|^2 = e^{-|\tilde{\alpha}(\beta,\theta)|^2}, \tag{4}$$
and detects one or more photons, i.e., $o_{\ell+1} = 1$, with probability $1 - p(o_{\ell+1} = 0|\alpha, (\beta,\theta))$.

4. The transmitted part of the signal enters layer $\ell+1$.

Finally, the last processing layer $\ell = L$ consists in elaborating a guess $\hat{k}$ of the true hypothesis k, based on previous measurement outcomes and parameter choices.

For an initial coherent state $|\alpha\rangle$, the input state at the ℓ-th layer is $|\alpha_\ell\rangle = \left|\alpha\sqrt{\theta_0 \cdots \theta_{\ell-1}}\right\rangle$, with $\theta_0 = \emptyset$. Since the experimenter can use all the past history $h_\ell = (a_0, o_1, \cdots, a_{\ell-1}, o_\ell)$, with $h_0 = \emptyset$, to decide the next value of (β,θ) and the final guess, the total set of parameters over all possible histories is of exponential size in L. We label them compactly as $a_\ell(h_\ell) = (\beta_{h_\ell}, \theta_{h_\ell})$ and $a_L(h_L) = \hat{k}$, omitting the label ℓ or the dependence on h_ℓ when it is clear from the context. Hence, the average success probability of this strategy over all possible outcomes' sequences $o_{1:L} = (o_1, \cdots, o_L)$ can be written as

$$P_s(\alpha, \{a_\ell\}) = \sum_{o_{1:L}} \prod_{\ell=1}^{L} p(o_\ell|\alpha^{(k)}, a(h_{\ell-1}))\; p_k\Big|_{k=a(h_L)}, \tag{5}$$

where the total conditional probability of $o_{1:L}$ factors into a product of single-layer probabilities.

In the model-aware setting, this expression can be optimized using dynamic programming, as we show in Sec. IV A, finding the set of optimal parameters $\{a_\ell^*\}$ for any given α and L:

$$\{a_\ell^*\} = \arg\max_{\{a_\ell\}} P_s(\alpha, \{a_\ell\}) \tag{6}$$

As a shorthand we denote the optimal success probability (over the available actions) as

$$P_*^{(L)}(\alpha) = \max_{\{a_\ell\}} P_s(\alpha, \{a_\ell\}), \tag{7}$$

and omit the label L when it is clear from the context.

In the model-free setting instead, the agent has no knowledge of Eqs. (4, 5), so it must resort to exploring the set of possible parameters and sample from the probability of (5) during several runs of the experiment to discover an optimal choice of parameters and guessing rule by trial and error.

III. SEQUENTIAL DECISION-MAKING

The framework of RL is based on the interaction between an agent and an environment during several episodes [61]. At each time-step $\ell = 0, \cdots, L$ of each episode $t = 1, \cdots, T$, the agent observes the environment in a *state* $s_\ell^{(t)} \in \mathcal{S}$ and chooses an *action* $a_\ell^{(t)} \in \mathcal{A}$; as a consequence, the agent enjoys a *reward* $r_{\ell+1}^{(t)} \in \mathcal{R}$ and observes a new state of the environment, $s_{\ell+1}^{(t)} \in \mathcal{S}$; where $\mathcal{S}$, $\mathcal{A}$ and $\mathcal{R}$ stand for the sets of states, actions and rewards the agent may experience; nonetheless the accessible future states/actions at a given state s_ℓ may be restricted to a subsets $\mathcal{S}(s_\ell) \subseteq \mathcal{S}, \mathcal{A}(s_\ell) \subseteq \mathcal{A}$.

The environment is usually modeled to be Markovian: its dynamics is completely determined by the last time-step via the *transition function* $\tau(s', r|s, a)$, i.e., the conditional probability of ending up in a state s' and conferring a reward r, given that the previous state was s and the agent took an action a. The agent does not have control of nor access to the transition function, but it will influence the dynamics of the environment by choosing actions according to an interaction *policy* $\pi(a|s)$, i.e., the conditional probability of performing an action a when the observed environment's state is s. This setting is ussually known as a Markov decision process (MDP).

Informally, the agent's objective is to interact with the environment through an *optimal policy* π^*, such that the total reward acquired during an episode is as high as possible. To achieve this goal, a *value function* is assigned to each state and optimized over all possible policies, as further explained below III A.

The Markov assumption is justified whenever the agent's observations provide a complete description of the state of the environment s_ℓ. However, in general this is not the case, and the agent only has access to *partial observations* $o_\ell \in \mathcal{O}$ at each time-step. Such observations would not allow to determine the dynamics even if τ was known, and they are generated from the current state and the previous action. In RL literature this is called a partially-observable MDP (POMDP) and developing methods to solve it efficiently constitutes an active area of research [64–68]; usually, the problem is tackled by first reducing it to an effective MDP. The most straightforward approach is to define an effective state that contains all the past history of observations and actions up to a given time-step, i.e., $h_\ell = (a_0, o_1, \cdots, a_{\ell-1}, o_\ell)$. In this way, the dynamics observed by the agent can always be described by an effective MDP with transition function $\tau(h', r|h, a)$, which is unknown to the agent and determined by the underlying environmental transition function. Clearly, this approach makes the problem intractable for large time-steps, since the number of states increases exponentially in L. In the model-aware setting, one can condense the history in a belief distribution over the states, $b_{o'}(s') = p(s'|o', a, b_o)$, i.e., the probability that the environment is in state s' given the current observation o', the previous action a and the belief at the previous time-step b_o. The belief has an initial value $b(s)$ equal to the prior distribution over the initial states and at each time-step it is updated using Bayes' rule. In the following parts of this Section we will introduce several tools for MDPs, which can be immediately adapted to POMDPs by exchanging the unknown state with the history h or the belief $b_o(s)$.

A. Value functions and the Bellman equation

The agent's objective is to acquire as much reward as possible during an episode. As a matter of fact this strongly depends on the agent's policy. At the end of episode t, in which a sequence of L tuples $\{(s_\ell, a_\ell, r_{\ell+1})\}_{\ell=0}^{L}$ has been experienced (with s_{L+1} a *terminal state*, and L generally varying among different episodes), the agent's performance after each time-step ℓ can be evaluated using the so-called return,

$$G_\ell^{(t)} = \sum_{i=0}^{L-\ell} \gamma^i r_{i+\ell+1}^{(t)}, \tag{8}$$

i.e., the weighted sum of rewards obtained at all future time-steps, with a *discount factor* $\gamma \in (0, 1]$, which weighs more the rewards that are closer in the future. Note that for infinite-horizon MDPs, i.e., $L \to \infty$, it must hold $\gamma < 1$ to ensure that G_ℓ remains finite.

By introducing the return, it is straightforward to assign a value to a state s for a given interaction policy π, via the so-called *value function*:

$$v_\pi(s) = \mathbb{E}_\pi \left[G_\ell | s_\ell = s \right], \tag{9}$$

which is the expected return over all possible trajectories that start from state s, take actions according to policy π and whose dynamics is governed by τ. In other words, the value function measures how convenient it is to visit state s when policy π is being followed. Note that this quantity is completely determined by the future trajectories accessible from s and hence its dependence on the time-step ℓ can have at most the effect of restricting the set of states on which $v_\pi(s)$ is supported at that time; we keep this dependence implicit unless otherwise stated. By writing explicitly the expected value for the first future time-step in Eq. (9) and then applying the definition of v recursively, it is easy to show that the state-value function satisfies, for any policy, the following Bellman equation [59]:

$$v_\pi(s) = \sum_{a \in \mathcal{A}, s' \in \mathcal{S}, r \in \mathcal{R}} \tau(s', r|s, a) \pi(a|s) \left(r + \gamma v_\pi(s') \right). \tag{10}$$

This equation relates the value of a state s with that of its nearest neighbours s', which can be reached with a single action from s, and with the corresponding reward obtained by performing such action.

The problem can then be solved by finding an optimal policy π^*, namely one that maximizes the state-value function for each s and thus satisfies the optimal Bellman equation:

$$\begin{aligned} v^*(s) &:= v_{\pi^*}(s) = \max_{\pi} v_\pi(s) \\ &= \max_{a\in\mathcal{A}} \sum_{s'\in\mathcal{S}, r\in\mathcal{R}} \tau(s', r|s, a)\left(r + \gamma v^*(s')\right). \end{aligned} \tag{11}$$

Similarly, one can define the state-action-value function (or Q-function) as the expected return when starting from state s and performing action a:

$$Q_\pi(s, a) = \mathbb{E}_\pi\left[G_\ell | s_\ell = s, a_\ell = a\right], \tag{12}$$

which is related to the state-value function by $v_\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) Q_\pi(s, a)$. The optimal policy π^* can also be obtained by maximizing the Q-function, with a corresponding optimal Bellman equation

$$\begin{aligned} Q^*(s, a) &:= Q_{\pi^*}(s, a) = \max_{\pi} Q_\pi(s, a) \\ &= \sum_{s'\in\mathcal{S}, r\in\mathcal{R}} \tau(s', r|s, a)(r + \gamma \max_{a'\in\mathcal{A}} Q^*(s', a')). \end{aligned} \tag{13}$$

B. Model-aware learning

In the model-aware setting, where the transition function is known, an optimal policy can be efficiently found off-line by optimizing the corresponding state-value function. This problem, known as planning [61], can be solved for finite-horizon MDPs via dynamic programming methods. We follow the method introduced by Bellman [59], which makes use of the recursive relation (11) to find the optimal policy step by step; for this we assume that episodes deterministically end at a fixed time-step L, and denote by $v^*_\ell(s)$ to the value function of state s at time-step ℓ (for scenarios in which finite and fixed L cannot be assumed, other approaches can be used, such as value iteration [61]).

Since the optimal policy consists in taking the best possible action from any given state, it can be constructed by concatenation of the optimal policies at each time-step: we start by solving Eq. (11) at the last time-step,

$$v^*_L(s) = \max_{a\in\mathcal{A}_L} \sum_{r\in\mathcal{R}_{L+1}} \tau(r|s, a) r, \tag{14}$$

where we have omitted the terminal state s_{L+1} and used the fact that $v_{L+1}(s) = 0$. The solution to Eq. (14) provides the optimal action at step L for each s and the optimal value function $v^*_L(s)$. Then we plug the latter into the optimal Bellman equation for the previous time-step, which in turn can be solved to obtain the optimal action and value function $v^*_{L-1}(s)$. By repeating this procedure iteratively for each time-step $\ell = L, \cdots, 0$, we can obtain the optimal sequence of actions and value functions for any state at any time-step.

C. Q-learning

In the model-free setting, the agent not only has to find an optimal policy by exploiting valuable actions, but also needs to characterize the environment in the first place by exploring possibly advantageous configurations. This is known as the exploration-exploitation trade-off [61] and lies at the core of RL problems. In this setting, the Q-function is quite helpful since it associates a value to the transitions determined by taking action a from state s and following policy π thereafter.

Q-learning was first proposed by Watkins [60], and it is often used as a basis for more advanced RL algorithms [65]. It is based on the observation that any Bellman operator, i.e., the operator describing the evolution of a value function as in Eqs. (10,11,13), is contractive [59]. This implies that, under repeated applications of a Bellman operator, any value function converges to a fixed point, which by construction satisfies the corresponding Bellman equation. Thus, in order to find $Q^*(s, a)$, Q-learning turns the optimal Bellman equation for Q, Eq. (13), into an update rule for $\hat{Q}(s_\ell, a_\ell)$, i.e., the Q-function's estimate available to the agent at a given time-step ℓ of any episode $t = 1, \cdots, T$.

After an interaction step $s_\ell \to a_\ell \to r_{\ell+1} \to s_{\ell+1}$ is experienced, the update rule for the Q-estimate is

$$\begin{aligned} \hat{Q}(s_\ell, a_\ell) \leftarrow &(1 - \lambda_t(s_\ell, a_\ell))\hat{Q}(s_\ell, a_\ell) \\ &+ \lambda_t(s_\ell, a_\ell)\left(r_{\ell+1} + \gamma \max_{a'\in\mathcal{A}_{\ell+1}} \hat{Q}(s_{\ell+1}, a')\right), \end{aligned} \tag{15}$$

where $\lambda_t(s, a)$ is the learning rate, which depends on the number of times the state-action pair (s_ℓ, a_ℓ) has been visited. Note that in order to do the updates at each time-step ℓ, it is only necessary to enjoy the next immediate reward $r_{\ell+1}$ and observe the next state $s_{\ell+1}$; this method thereby allows an on-line learning of the MDP. A pseudo-code of the algorithm is given in Algorithm 1.

After a large number n of iterations of the update rule Eq. (15) for all state-action couples, the convergence of the Q-estimate to the optimal Q-function is guaranteed by two general conditions on the learning rate (also known as Robinson conditions) [60, 61]:

$$\begin{aligned} \hat{Q}(s, a) &\underset{k\to\infty}{\to} Q^*(s, a) \ \ \forall s \in \mathcal{S}, a \in \mathcal{A}(s) \\ &\text{iff } \sum_{t(s,a)} \lambda_t(s, a) = \infty, \ \sum_{t(s,a)} \lambda_t(s, a)^2 < \infty, \end{aligned} \tag{16}$$

where the sums are taken over all interactions at which a given state-action couple is visited. Once the optimal Q-function is obtained, an optimal deterministic policy can be constructed by "going greedy" with respect to it, i.e., $\pi^*(a|s) = \delta(a, \arg\max_{a\in\mathcal{A}} Q^*(s, a))$ for all $s \in \mathcal{S}$, where $\delta(x, y)$ is a Kronecker delta.

Algorithm 1: Q-learning pseudo-code.

input : $\hat{Q}(s,a)$ arbitrarly initialized
$\forall s \in \mathcal{S}\ \forall a \in \mathcal{A}(s)$; learning rates
$\lambda_t(s_\ell, a_\ell) \in (0,1]$, $\epsilon > 0$
output: $\hat{Q}(s,a) \sim Q^*(s,a)$

for *t in* 1 ... *T* **do**
  initialize s_0
  for *step* ℓ *in episode* **do**
   take action a_ℓ according to π (e.g. ϵ-greedy)
   observe reward $r_{\ell+1}$ and next state $s_{\ell+1}$
   update $\hat{Q}(s_\ell, a_\ell)$ according to:
   $\hat{Q}(s_\ell, a_\ell) \leftarrow \hat{Q}(s_\ell, a_\ell) + \lambda(s_\ell, a_\ell)[r_{\ell+1} +$
   $\gamma \max_{a'} \hat{Q}(s_{\ell+1}, a') - \hat{Q}(s_\ell, a_\ell)]$
   if $s_{\ell+1}$ *is terminal state* **then**
     break
   else
     $s_\ell \leftarrow s_{\ell+1}$

Q-learning is an off-policy method [61] that employs two distinct policies: (i) a *learning policy* to update the Q-estimate given by Eq. (15), that efficiently encapsulates the information gathered in previous experience; (ii) an *interaction policy* which provides a prescription to choose the next action actually taken by the agent. The most basic Q-learning method is to do this by committing to an interaction policy $\pi(a|s)$ for all episodes, such as ϵ-greedy where with probability ϵ the agent chooses a random action and otherwise it chooses the greedy action that maximizes the current Q-estimate. However, as we will see below, more general strategies can be considered.

IV. MODEL-FREE REINFORCEMENT LEARNING OF DISCRIMINATION STRATEGIES

In the following we frame the calibration of the receiver described in Sec. II into a RL context, in which an agent has to attain optimal reward-per-episode rate (success rate) by departing from a situation of complete ignorance of the experiment. For simplicity, we assume that the sender and receiver have a shared reference frame, so that we can take the states and displacements to be real, $\alpha, \beta \in \mathbb{R}$, without loss of generality.

The notation introduced in Sec. II is straightforward to translate into the RL notation of Sec. III:

- Each episode t corresponds to an independent discrimination experiment, with a new default state $s_0 = \alpha^{(k)}$ sampled from p_k, $k \in \{0,1\}$; we set $\gamma = 1$ since the process has finite horizon;
- Each episode consists of $L+1$ time-step $\ell = 0, \cdots, L$, corresponding to the L detection layers followed by the final guessing stage;
- The possible states of the environment at time-step ℓ are $s_\ell = \alpha_\ell^{(k)}$, i.e., the transmitted part of s_0 at that layer;
- The agent is not aware of the state s_ℓ, in particular it does not know which hypothesis is true, but it can observe the measurement outcome o_ℓ, $0 < \ell \leq L$;
- The actions a_ℓ available at time-step $\ell < L$ are the displacements β_ℓ and BS parameters θ_ℓ available at that layer, conditioned on the history of observations, h_ℓ, while at the last step they constitute the guess, $a(h_L) = \hat{k} \in \{0,1\}$;
- The reward $r \in \{0,1\}$ is non-zero only at the end of the episode and provided that the guess is correct, hence the transition function for the environment is

$$\tau(\alpha_{\ell+1}^{(k')}|\alpha_\ell^{(k)}, a_\ell) = \delta(k',k)\ \ \forall \ell \leq L, \tag{17}$$

$$\tau(r_{L+1}|\alpha_L^{(k)}, a_L) = \delta(r_{L+1}, 1)\delta(a_L, k), \tag{18}$$

were we omitted the trivial reward for $\ell \leq L$.

A. Benchmarking the success probability via dynamic programming

In order to benchmark the performance of our RL agent, we start by considering a model-aware POMDP where the agent knows the amplitude $|\alpha|$ of the optical signals and the transition probabilities; its task is to optimize the success probability of Eq. (5). In this case, we define $b_\ell(k) = p(\alpha^{(k)}|o_\ell)$ to be the belief distribution over the states $\alpha^{(k)}$, afer outcome o_ℓ is observed, with initial value equal to the prior $b_0(k) = p_k$, and update rule following Bayes' theorem:

$$b_{o_{\ell+1}}(k) = \frac{p(\alpha^{(k)}|o_{\ell+1}, a_\ell)\ b_\ell(k)}{\sum_k p(\alpha^{(k)}|o_{\ell+1}, a_\ell)\ b_\ell(k)}, \tag{19}$$

given that action a_ℓ was performed. The optimal Bellman equation, Eq. (14), for the state-value function of this POMDP at step L reads

$$v_L^*(b_{o_L}) = \max_k b_{o_L}(k), \tag{20}$$

which means that at the last step, if the final belief distribution over the states is known, the best guess is the hypothesis with maximum likelihood. The optimal Bellman equation at step $\ell < L$ instead reads

$$v_\ell^*(b_o) = \max_{a \in \mathcal{A}_\ell} \sum_{o' \in \mathcal{O}} \sum_k p(o'|\alpha_\ell^{(k)}, a) b_o(\alpha_\ell^{(k)}) v_{\ell+1}^*(b_{o'}). \tag{21}$$

These equations can be solved iteratively by inserting the solution $v_{\ell+1}^*(b_{o'})$ into the equation for $v_\ell^*(b_o)$, starting with $\ell = L-1$ and $v_L^*(b_{o_L})$ found in Eq. (20). Note that, since $v_{\ell+1}^*(b_{o'})$ is computed for a discrete set of values of the belief distribution, these cannot always coincide with the values, determined by Eq. (19), needed to solve Eq. (21) and hence we use interpolation methods to obtain them.

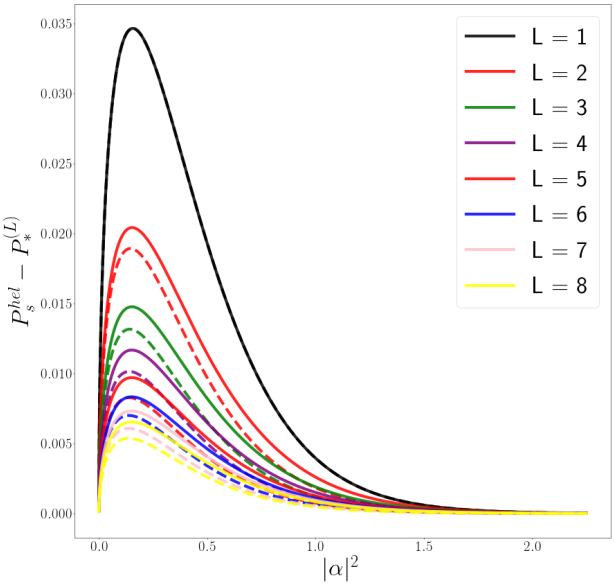

FIG. 2. We show the difference between the best probability of success attainable for a fix L and the optimal probability of success in discriminating two coherent states. The results were obtained by dynamic programming, exaplained in Sec.III B. Solid lines correspond to fixed attenuations θ_ℓ such that the input state of each layer has equal amplitude $\alpha_\ell^{(k)} = \frac{\alpha^{(k)}}{\sqrt{L-1}}$ for all ℓ, whereas dashed lines correspond to the probability of success optimized also on conditional attenuations.

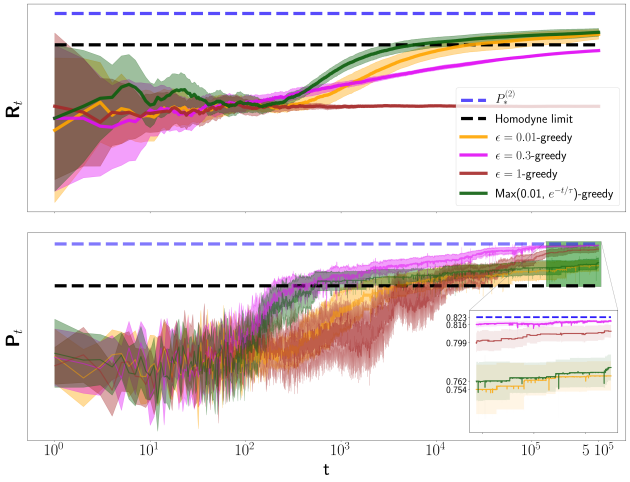

FIG. 3. We benchmark traditional Q-learning with different *schedules* on ϵ as the episode number increases. The figures of merit are averaged over $A = 48$ agents and show the corresponding uncertainty region.

The maximum success probability attainable with the receiver is equal to the optimal value function at step $\ell = 0$, since the latter corresponds to the expected reward starting from the initial belief distribution $b(\alpha_0^{(k)})$:

$$v_0^*(b) = \mathbb{E}_{\pi^*}[r_{L+1}|b(\alpha_0^{(k)})] = P_*^{(L)}(\alpha) \tag{22}$$

as can be seen by repeated applications of Eqs. (19,21) and detailed in Sec. IV A.

In Fig. 2 we show the optimal success probability obtained with this method as a function of $|\alpha|^2$ and for up to $L = 8$ layers. We also show the results at fixed θ_ℓ such that the input state of each layer has equal amplitude $\alpha_\ell^{(k)} = \frac{\alpha^{(k)}}{\sqrt{L-1}}$ for all ℓ (dashed lines). We observe that for all $L \geq 2$ there is an energy threshold above which allowing adaptive optimization of the attenuations gives a better success probability than adding one layer with fixed attenuations.

B. Learning a near-optimal receiver via Q-learning

In this Subsection we present the results obtained by a RL agent based on Q-Learning with ϵ-greedy interaction policy. The experiment is modelled as a POMDP, which can be reduced to an effective MDP for the history of observations and actions h_ℓ, as explained in Sec. III. The update rule for the Q-function is given by Eq. (15) with $s \to h$ and learning rates $\lambda_t(h,a) = N_t(h,a)^{-1}$, the inverse of the number of times a state-action pair has been visited. This standard choice guarantees convergence as per Eq. (16). As for dynamic programming, the optimal value of the success probability of Eq. (5) is obtained by maximizing the optimal Q-function at time-step $\ell = 0$:

$$\max_{a_0} Q^*(a_0) = \max_{a_0} \mathbb{E}_{\pi^*}[r_{L+1}|a_0] = P_*^{(L)}(\alpha), \tag{23}$$

where we have omitted the default history state $h_0 = \emptyset$; this is detailed in Appendix A. At variance with the model-aware case, where the guessing rule was obtained straightforwardly from the Bellman equation at the last time-step, the optimization of Eq. (23) includes a non-trivial search for the optimal guessing rule, determined by the optimal Q-function at the last time-step.

We evaluate the performance of the agent using two figures of merit as a function of the number of episodes elapsed so far, t: (i) the cumulative return per episode (also called average reward per episode)

$$\mathbf{R}_t = \frac{1}{t}\sum_{i=1}^{t} G_0^{(i)} = \frac{1}{t}\sum_{i=1}^{t} r_{L+1}^{(i)}, \tag{24}$$

where $r_{L+1}^{(i)} = \{1, 0\}$ stands for the correctness of the guess made at episode i, and (ii) the success probability of the best actions according to the agent, at the current episode,

$$\mathbf{P}_t = P_s(\alpha, \{a_\ell^{(t)*}\}), \tag{25}$$

where the best actions $\{a_\ell^{(t)*}\}$ at episode t are obtained by going greedy with respect to the current Q-estimate,

i.e.,

$$a_0^{(t)*}(h_0) = \underset{a\in\mathcal{A}(h_0)}{\arg\max}\, \hat{Q}(h_0, a) \to h_1^* = (o_1, a_0^{(t)*}) \tag{26}$$
$$a_1^{(t)*}(h_1^*) = \underset{a\in\mathcal{A}(h_1^*)}{\arg\max}\, \hat{Q}(h_1^*, a) \to h_2^* = \left(o_1, o_2, a_0^{(t)*}, a_1^{(t)*}(h_1^*)\right)$$
$$\ldots$$
$$a_L^{(t)*}(h_L^*) = \underset{a\in\mathcal{A}(h_L^*)}{\arg\max}\, \hat{Q}(h_L^*, a).$$

The first figure of merit, $\mathbf{R}_t$, is usually employed to describe the learning process in RL and it evaluates the success rate of the agent so far. On the other hand, the second figure of merit, $\mathbf{P}_t$, is standard in QSD and in our context it evaluates the best strategy discovered by the agent so far.

As $t \to \infty$, for a *good* learner it is expected that $\mathbf{R}_t \to \mathbf{P}_t$, i.e. with enough learning time the average reward should tend to the success probability for the best actions found by the agent, which in turn should converge to the optimal success probability $P_*^{(L)}$. Therefore, the learner is not only expected to find a good discrimination strategy, but to also follow it: the interaction policy should tend to the optimal policy. This feature is captured by the evolution of $\mathbf{R}_t$ over different episodes: a good learner is asked to obtain as much reward as possible *during* the learning process.

In Fig. 3 we plot these two figures of merit for Q-learning agents with three different ϵ-greedy interaction policies: (i) a completely random one, i.e., $\epsilon = 1$, (ii) a 0.3-greedy one, i.e., $\epsilon = 0.3$, and (iii) a dynamic one (exp-greedy) that becomes exponentially greedier as time passes, i.e., $\epsilon(t) = \max\{e^{-2\cdot 10^{-2}t}, 0.01\}$; this standard choice assures that at initial episodes the agent favours exploration, whereas at $t = \tau \log \frac{1}{\epsilon_0}$ the agent's behaviour collapses to an ϵ_0-greedy policy.

Here and in the rest of the article, we restrict to $L = 2$ interaction layers and fix the attenuation coefficients to give equal amplitude at each layer, since the difference in success probability is small compared with the additional number of episodes one would need to learn it, as shown in Sec. IV A. We choose a resolution of 21 points for each displacement, each one ranging from -1 to 1 with step 0.1, leading to a fairly large action space: the agent can choose among a total of 3528 possible actions at each episode (including final decision rule). We note that each discretized displacement is an independent action or "button" in the eyes of the agent —the agent is dispossessed of any notion of closeness between buttons corresponding to similar values of β. As the behaviour of the RL agent strongly depends on the actions chosen at early episodes, we averaged the learning curves over 24 agents. Our results are compared with: (i) the maximum success probability attainable with this number of layers, as benchmarked by dynamic programming, Eq. (22), and (ii) the success probability attainable via a standard homodyne measurement, which is optimal among Gaussian receivers [40].

In the first place we note in Fig. 3 that a fully random search over the action space (1-greedy policy) leads to the extremely poor cumulative reward per episode of $\mathbf{R}_t \approx 1/2$, even for long times, which is expected because a random guess (last action) leads to $P_s(\alpha, \{a_\ell\}) = 1/2$. Instead, since all the actions will be sampled enough times for the agent to learn the optimal policy, $\mathbf{P}_t$ will converge to optimal value at long enough episode number. Nevertheless, if the action space is large, the fully random strategy will require a large number of episodes to explore each action a significant number of times, and for moderate times a ϵ-greedy strategy might reach a better strategy. Indeed, Fig. 3 shows that the 0.3-greedy policy has at all episodes a higher $\mathbf{P}_t$ than the 1-greedy one, being 99% the optimal success probability $P_*^{(L=2)}$ at episode $t = 10^5$. Of course, for 0.3-greedy policy the agent collects many more rewards (actual correct guesses) than for the 1-greedy but it is still limited to $\mathbf{R}_t \approx 0.7 P_*^{(L=2)}$. In order to reach a better exploration-exploitation trade-off, it is customary to consider an episode-dependent ϵ, e.g. $\epsilon(t) = \max\{e^{-\frac{t}{\tau}}, \epsilon_0\}$. Fig. 3 shows the results for this tunable interaction policy with $\tau = 2\cdot 10^2$ and $\epsilon_0 = 0.01$. This allows the agent's $\mathbf{R}_t$ to surpass the homodyne limit at about episode $\sim 5\cdot 10^3$ (which is comparable with the size of the action space), while at later times the performance converges to that of the 0.01-greedy policy. Notice finally that 0.3-greedy discovers a strategy whose $\mathbf{P}_t$ surpasses the homodyne limit at episode $\sim 310^2$.

Our numerical results show that standard Q-learning successfully trains agents that surpass the homodyne limit of optical detection and discover strategies whose error rate is comparable with that of the optimal receiver. This is remarkable specially taking into consideration that the agents are not particularly trained for this task, and run in a model-free setting entirely based on the feedback they get (correct/incorrect) on their guess. As mentioned above, although many RL schemes focus on extracting the optimal policy from the agent (as measured e.g. by $\mathbf{P}_t$), our central figure of merit, $\mathbf{R}_t$ captures the real performance of the agent, and can actually be assessed by the agent itself. It is hence important to design strategies that not only aim at finding the optimal policy within an episode, but also maximize the cumulative reward per episode, reaching $\mathbf{R}_t \to P_*^{(L)}$ as fast as possible. We have seen above an example of such strategy (the exponential greedy) and in the next sections we will study more advanced ones. For this purpose we will first study a simplified setting, called the multi-armed bandit problem, where the intra-episode dynamics is trivial, and the main focus is drawn on how to optimize the inter-episode learning strategy. On passing, we introduce some theoretical tools in order to study the bandits' learning curves, which are a cornerstone to tackle more challenging situations such as learning optimal policies over a MDP.

C. The multi-armed bandit problem

Multi-armed bandit problems have a single default state and the agent faces a fixed set of actions $a \in \mathcal{A}$ each one leading to a reward $r \in \mathcal{R}$ with an unknown probability $\tau(r|a)$; after action a is performed, the reward r is enjoyed and the episode finishes. The situation models a gambler at a row of slot machines (also known as "one-armed bandits") that has to decide which arms to pull, how many times to pull each one and in which order with the aim of maximizing the earned rewards.

The bandit problem is an ideal framework to highlight the aforementioned crucial difference between learning strategies that accomplish the main goal of learning the (near) optimal policy after some time and the more refined strategies that also procure high *de facto* cumulative rewards during the learning process. Compared with other RL problems with a rich intra-episode dynamics, such as a maze problems or GO, the task of identifying the optimal arm to pull in the bandit problem is straightforward (from (29) $a^* = \arg\max_a \hat{Q}(a)$). This might seem to put on equal footing all bandit learning strategies. However, the choice of strategy, which rules the sequence by which different arms are pulled, will influence a lot the shape of actual accumulated rewards in the transient period and the rate at which the optimal performance is reached. This is why bandit problems are very relevant in real-life applications where the final success rate is not the only figure of merit, as for example in clinical trials [69] where one needs to find the right compromise between advancing in the search of the best treatment while effectively treating current patients.

Looking at the general traits of the cumulative reward per episode $\mathbf{R}_t$ in Fig. 3 could inspire several ways of quantifying the performance of the learning agent, e.g., (a) the onset time at which $\mathbf{R}_t$ starts exceeding the random policy; (b) transient time at which $\mathbf{R}_t$ reaches a given fraction of the optimal success rate; (c) the learning rate as quantified by the slope of the $\mathbf{R}_t$ after the onset time; (d) the learning rate at which $\mathbf{R}_t$ converges to the optimal value. Unfortunately, very little is known about these or alternative ways to characterize the learning curves. Bandit theory provides us with a framework were some of these notions can be defined and rigorously studied. In particular, related to (d), bandit theory defines the so-called *cumulative regret*:

$$\mathcal{L}_t = \sum_{k=1}^{t} Q(a^*) - Q(a^{(t)}) = t\,\big(Q(a^*) - \mathbb{E}\mathbf{R}_t\big), \qquad (27)$$

where $a^{(t)}$ is the action actually taken at episode t. The cumulative regret is closely related to the (averaged) cumulative reward per episode $\mathbf{R}_t$, and quantifies the price to pay, or loss, for taking actions different from the optimal one (a^*). In other words, it quantifies the difference in earnings of the agent with respect to those of a distribution-aware super-agent. One of the most fundamental results in bandit theory is the Lai and Robbins bound [70] for the asymptotic expected cumulative regret:

$$\mathbb{E}\,\mathcal{L}_t \underset{t\gg 1}{\gtrsim} \log t\Big(\sum_{a\in\mathcal{A}\setminus\{a^*\}} \frac{\Delta_a}{\mathrm{KL}(a||*)} + o(1)\Big) := C_{\mathrm{LR}} \log t \qquad (28)$$

with $\Delta_a = Q(a^*) - Q(a)$ and $\mathrm{KL}(a||*)$ the Kullback-Leibler divergence between the reward distributions $\tau(r|a)$ and $\tau(r|a^*)$. Recalling the definition in (27) we note that the Lai and Robbins bound characterizes the learning curve in the asymptotic regime, i.e. $\mathbf{R}_t$ (averaged over many agents that follow the same strategy) can approach the optimal value no faster than $\mathbf{R}_t \lesssim Q(a^*) - C_{\mathrm{LR}}\frac{\log t}{t}$. Let us now briefly present some possible bandit strategies in light of this ultimate performance bound.

The most straightforward policy to use is the ϵ-greedy (already introduced in Sec.III C), as described in Algorithm 2.

Algorithm 2: ϵ-greedy for bandit problems.

input : $\hat{Q}(a)$ arbitrarily initialized and learning rates $\lambda_t(a) \in (0,1]\ \forall a \in \mathcal{A}$, $\epsilon \in (0,1]$
for *t in 1 ... T* **do**
  `generate a random number j`
  if $j \le \epsilon$ **then**
    `choose` $a^{(t)}$ `at random`
  else
    `choose` $a^{(t)} = \arg\max_{a\in\mathcal{A}} \hat{Q}(a)$
  `observe` r
  `update` $\hat{Q}$:
    $\hat{Q}(a^{(t)}) \leftarrow \hat{Q}(a^{(t)}) + \lambda_t(a^{(t)})[r - \hat{Q}(a^{(t)})]$

Note that by choosing the learning rates $\lambda_t(a)$ to be the inverse of the number of times action a was visited up to time t, then

$$\hat{Q}(a) \underset{t\to\infty}{\to} \sum_{r\in\mathcal{R}} r\ \tau(r|a)\ = Q(a)\ \ \forall a \in \mathcal{A}. \qquad (29)$$

As stressed in Sec.IV B, the ϵ-greedy policy never attains the optimal success rate, $\mathbf{R}_t < Q(a^*)$ for all t, since at all episodes there is a finite probability ϵ that the agent does a suboptimal action. For this reason the expected cumulative regret grows linearly with time as can bee seen in Fig. 4. It is then clear than there is a lot of room for improvement before reaching the logarithmic scaling of the ultimate limit (28). We will present two strategies, one based on Upper Confidence Bounds (UCB) [62, 70, 71] and the other called Thompson sampling (TS) [63, 69, 72, 73], which substantially improve the performance of ϵ-greedy and may even attain the Lai and Robbins ultimate bound [74, 75].

In UCB, the agent keeps a record of the number of times each action a was selected up to episode t, which we denote as $N_t(a)$. Hoeffding's inequality bounds the probability that the $\hat{Q}(a)$ underestimates the true value

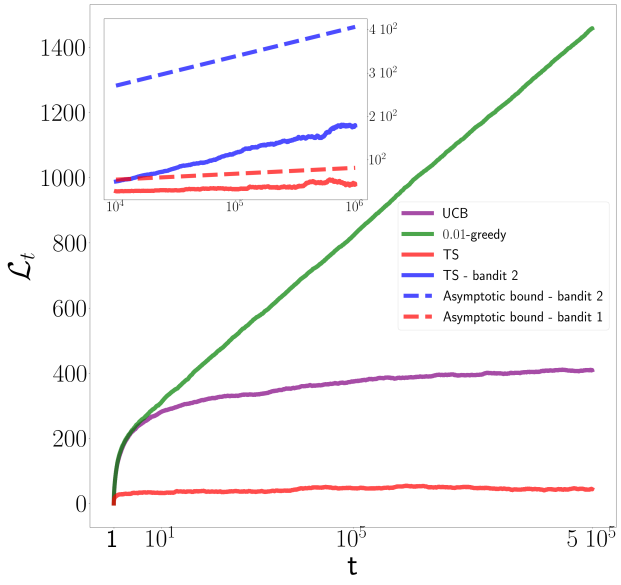

FIG. 4. We show the evolution of the cumulative regret for three different strategies. In this case, *bandit problem 1*, the displacements considered are $\beta \in \{0, -\alpha, \beta^*\}$, for $\alpha = 0.4$ and $\beta^* = -0.74$. All curves are averaged over 10^3 agents. Furthermore, we compare the asymptotic behaviour of TS, studying *bandit problem 2*, with $\beta \in \{-\alpha, \beta^*, -1.5\alpha\}$

of mean $Q(a)$ by more than $\varepsilon(t) > 0$, as

$$\Pr[\hat{Q}(a) < Q(a) - \varepsilon(t)] \leq e^{-2N_t(a)\varepsilon(t)^2} =: \mathcal{P}(t), \qquad (30)$$

where here and in the rest of this section we assume that $r \in [0, 1]$. Then, for $N_t(a) > 0$, the upper confidence bound, defined as

$$\mathrm{ucb}_t(a) := \hat{Q}(a) + \varepsilon(t) = \hat{Q}(a) + \sqrt{\frac{-\log \mathcal{P}(t)}{2N_t(a)}}, \qquad (31)$$

represents an upper bound to the true value $Q(a)$ with (high) probability $1 - \mathcal{P}(t)$. This is the value that is used to compare and choose among the different actions, i.e. $a^{(t)} = \arg\max_{a\in\mathcal{A}} \mathrm{ucb}_t(a)$, and responds to the motto "optimism under the face of uncertainty": actions that have not been visited enough are assigned an "optimistic" value of the return and hence more chances of being picked; in addition, actions whose Q-estimate is accurate but sub-optimal will have little chances to be picked again. The functional form of $\mathcal{P}(t)$ can be tuned to balance exploration and exploitation. In particular, for the standard choice $\mathcal{P}(t) = t^{-4}$ (often called UCB-1) it can be easily seen that the expected cumulative regret follows the logarithmic scaling. In Appendix B we also discuss a different choice, which is known to attain the Lai and Robbins bound, but that performs worst than UCB-1 for our setting and moderately small times we consider.

Algorithm 3: UCB for bandit problems.

input : choose $\mathcal{P}(t)$ and $\lambda_t(a)$; initialize to zero $\hat{Q}(a), N(a)\ \forall a \in \mathcal{A}$.
for t *in* $1, ..., T$ **do**
  if $t \leq |\mathcal{A}|$ **then**
    `(choose each action once)` $a^{(t)} = t$
  else
    `choose` $a^{(t)} = \arg\max_{a\in\mathcal{A}} ucb_t(a)$, (Eq. (31))
  `observe reward` r
  `update Q-value:`
    $\hat{Q}(a^{(t)}) \leftarrow \hat{Q}(a^{(t)}) + \lambda_t(a^{(t)})[r - \hat{Q}(a^{(t)})]$
  `record visit:`
    $N(a^{(t)}) \leftarrow N(a^{(t)}) + 1$

Thompson sampling (TS) departs from the standard Q-learning paradigm, which is based on keeping track and be updating the Q-table.Instead, TS follows a Bayesian approach and at every episode assigns a full prior distribution (not just an expectation value) for the *expected reward* $\bar{r}$ of every arm a, $f_t(\bar{r}|a)\ \forall a \in \mathcal{A}$. This distribution characterizes the knowledge the agent has about the expected earnings of each arm, $Q(a)$, and at the first episode can be taken to be flat over the whole interval $[0, 1]$. The policy then consists in sampling an expected reward $\bar{r} \sim f_{t-1}(\bar{r}|a)$ *for each* possible action a and choosing the action with the largest sample $\bar{r}$: $a^{(t)} = \arg\max_{a\in\mathcal{A}}\{\bar{r} \sim f_t(\bar{r}|a)\}$). Finally, the distribution for the chosen action is updated according to the true reward r obtained, using Bayes' theorem.

In order to avoid computationally-expensive Bayesian updates, families of distributions that are closed under the update rule are used. In the case of Bernoulli bandits, beta-distributions are employed since those are precisely their conjugate priors. That is, given

$$f_t(\bar{r}|a) = \mathrm{Beta}(\mu_t(a), \nu_t(a)) \propto \bar{r}^{\mu_t(a)-1}(1-\bar{r})^{1-\nu_t(a)}, \qquad (32)$$

upon obtaining a reward r the prior is updated to a beta distribution with parameters $\mu_{t+1}(a) = \mu_t(a) + r$, $\nu_{t+1}(a) = \nu_t(a) + 1 - r$, where at the first episode it is $\mu_1(a) = \nu_1(a) = 1\ \ \forall a \in \mathcal{A}$ (flat prior). If a certain action has not been sampled enough at episode t, its reward distribution will still be very broad and, when sampled, can easily return a higher value of $\bar{r}$ than that obtained from other (more peaked) distributions. Hence TS will favour to explore such action. At the same time if for some reason a clearly sub-optimal action has been sampled for many episodes, it will be very unlikely that it is sampled again, since the corresponding prior will be highly peaked at low values. The pseudo-code of TS for Bernoulli bandits is described in Algorithm 4.

Figure 4 shows the performance of different strategies in a 3-armed bandit problem. For this purpose we have considered a simple optical receiver as described in Sec.II with a single layer $L = 1$. Since there is only a single detector with binary outcome, we have assumed a fixed decision rule. With this, each possible displacement β

Algorithm 4: TS for Bernoulli bandit problems.

input : $\mu_1(a), \nu_1(a)$ initialized to one $\forall a \in \mathcal{A}$

for t *in* $1, ..., T$ **do**

  for a *in* $\mathcal{A}$ **do**
    draw $\bar{r}_a$ according to $\text{Beta}(\mu_t(a), \nu_t(a))$

  choose $a^{(t)} = \arg\max_a \bar{r}_a$
  observe reward r
  update Beta distribution:
    $\mu_{t+1}(a) = \mu_t(a) + r$, $\nu_{t+1}(a) = \nu_t(a) + 1 - r$

constitutes an action a_0 (recalling the notation used in last section) of a bandit problem. The figure shows that the cumulative regret scales linearly with time for the ϵ-greedy strategy, while it has a logarithmic scaling for the UCB and TS strategies. The inset shows the cumulative regret as a function of $\log t$ together with the ultimate bound given by Lai and Robbins bound. The achievability of this bound is hard to observe in simulations because the convergence to the asymptotic results is very slow [77], i.e. there are sub-leading constant or order $\log(\log t)$ terms might be important. Nevertheless, in the setting of Fig. 4 we see that leading term captured by the slope of the curve is consistent with the ultimate bound.

Let us conclude this overview of bandit theory by introducing the *simple regret*, another widely used figure of merit that, as $\mathbf{P}_t$, quantifies how well has the agent learned so far, regardless of his actual performance:

$$\Lambda_t = Q(a^*) - Q(a^{(t)*}), \tag{33}$$

where $a^{(t)*}$ is the agent's *recommendation* of which the optimal action is at episode t, which, e.g., in Q-learning would be given by $a^{(t)*} = \arg\max_a \hat{Q}(a)$. Good policy-learning strategies will eventually learn what is the optimal arm to pull, and the probability that the agent confuses the optimal arm by a sub-optimal one will be exponentially small, hence Λ_t will converge to zero exponentially fast. Recent results [76] show that exploitation-exploration trade-off manifests itself in the asymptotic scaling of the simple and cumulative regret in the sense that one imposes lower and upper-bounds on the other, and therefore optimizing one usually affects the performance of the other.

*** To conclusions: The characterization of the cumulative or simple regret for intermediate times, through non-asymptotic general upper and lower bounds, as well as its extension from Bandits to more general Markov Decision Processes is still an open and active field of research. Quantum technologies can benefit from this progress, and non-trivial quantum features might appear in more general quantum learning scenarios.

D. Enhancing the agent via UCB and TS

In this Subsection we consider two enhanced RL strategies, inspired by the advanced bandit methods introduced in Sec. IV C, and adapted to our MDP problem.

The first strategy employs the standard Q-learning update rule for the estimate $\hat{Q}$, described in Eq. (15), but it employs the UCB method to determine the interaction policy at each time-step of each episode, as described in Sec. IV C, with $\Delta(t) = t^{-4}$ (see Appendix B for a comparison of different learning parameters). The UCB policy is implemented by keeping count of the number of visits of each history-action couple up to the current episode t, i.e., $N_t(h_\ell, a_\ell)$, which is then used to compute an upper confidence bound, $\text{ucb}_t(h_\ell, a_\ell)$ as in Eq. (31), for each action a_ℓ and history h_ℓ. Finally, at time-step ℓ the agent chooses the greedy action w.r.t. the UCB, i.e., $a_\ell^{(t)} = \delta(a, \max_a \text{ucb}_t(h_\ell, a))$.

The second strategy instead is based entirely on TS, considering each action conditioned on the past history as a bandit problem and rewarding each sequence of actions that led to a successful experiment. In detail, the agent keeps a beta-distribution, Eq. (32), of the mean reward obtainable at each time-step ℓ from each action a_ℓ given each possible history h_ℓ, i.e., $f_t(\bar{r}|h_\ell, a_\ell)$. In order to choose a new action at time-step ℓ given history h_ℓ, the agent samples an expected reward $\bar{r} \sim f_t(\bar{r}|h_\ell, a_\ell)$ for each a_ℓ and selects the action with the largest sample $\bar{r}$. At the end of the episode a reward is obtained as usual, and $f_t(\bar{r}|h_\ell, a_\ell)$ is updated in a Bayesian way for all the history-action couples visited at the episode. In this case, when computing $\mathbf{P}_t$, the best actions are chosen by going greedy w.r.t. their mean reward distribution $f_t(\bar{r}|h_\ell, a_\ell)$.

In Fig. 5 we plot the two figures of merit $\mathbf{R}_t$, $\mathbf{P}_t$ for agents trained using these two enhanced strategies, as well as for those based on the exp-greedy and 0.3-greedy strategies, considered in Sec. IV B, which had respectively the largest final $\mathbf{R}_t$ and $\mathbf{P}_t$ out of all the analyzed strategies. We observe that UCB performs a thorough exploration of the action space and indeed it is able to attain a value of $\mathbf{P}_t$ close to that of 0.3-greedy. This result comes at the price of a small $\mathbf{R}_t$ value, which nevertheless shows that UCB has better exploitation properties than 0.3-greedy; in particular it has a strikingly larger slope than the latter at long times. As for TS, we observe that this strategy attains the best $\mathbf{R}_t$ values, surpassing exp-greedy at intermediate times. Moreover, TS also radically improves the values of $\mathbf{P}_t$ w.r.t. exp-greedy and it is even able to attain the performance of the other two strategies that favour exploration. Overall, it appears that for our problem TS provides the most profitable balance of exploration and exploitation.

In Fig. 6 we study the guessing rule discovered by the UCB agent at the final episode. For each sequence of outcomes o_1, o_2, we plot the difference between the Q-values of guessing for $|+\alpha\rangle$, i.e., $a_L = 0$, and $|-\alpha\rangle$, i.e.,

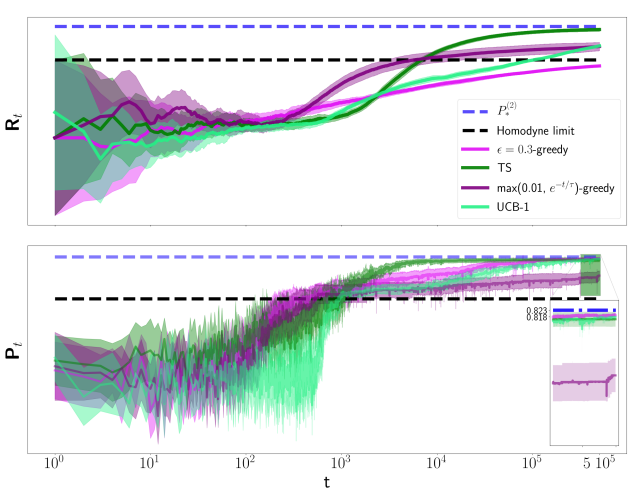

FIG. 5. We show the learning curves for the enhanced Q-learning agents via bandit methods. On the upper plot we despict $\mathbf{R}_t$, the agent's success rate per episode, whereas on the bottom plot we despict $\mathbf{P}_t$, the success probability of the agent's recommended actions at episode t, $\{a_\ell^{(t),*}\}$. Each of the learning curves is averaged over 24 agents; the amplitude was fixed to $\alpha = 0.4$.

$a_L = 1$, as a function of the past actions:

$$\hat{Q}_L^{(T)}((\beta, o_1, \beta_{o_1}, o_2), 0) - \hat{Q}_L^{(T)}((\beta, o_1, \beta_{o_1}, o_2), 1). \quad (34)$$

Note that the sign of Eq. (34) corresponds to the agent's best guess for the true hypothesis, since the latter is obtained by "going greedy" towards $\hat{Q}(h_L, a_L)$. We compare these results with the optimal guessing rule in the model-aware setting, plotting a shaded region when the maximum-likelihood guess is $|\pm\alpha\rangle$. The plot shows that UCB agents perfectly learn the guessing rule at the given resolution. Moreover, the difference between the two Q-values is more pronounced in the surroundings of the optimal β values, meaning that the agents are more confident about their guess in these regions.

Finally, we show that RL agent's performance is independent of the coherent states' energy. For this we evaluate $\mathbf{R}_t$ and $\mathbf{P}_t$ at episode $t = 5\ 10^5$ for a range of different amplitudes $|\alpha|$, and compare them with the optimal success probability $P_*^{(L=2)}$, as can be seen in 7. In the following we turn to test model-free methods in realistic experimental scenarios, where the ultimate success probability is affected by the presence of noise.

E. Noise robustness

In the previous subsections we have shown that our RL agents are able to learn near-optimal discrimination strategies and — most importantly — exploit them in real time, employing exclusively the detectors' outcomes and the rewards at the end of each episode. Here we show

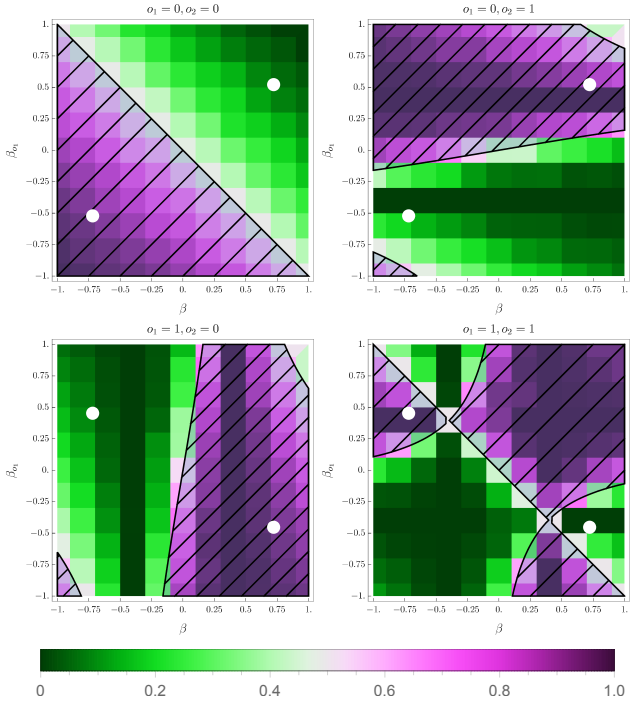

FIG. 6. Density plot of the difference between the estimated Q-values for guessing "plus" and "minus" as a function of the displacements at the first and second layer, for each possible sequence of outcomes, with $\alpha = 0.4$. The shaded areas correspond to the regions where the optimal guess, taken according to maximum-likelihood, is "plus". The white dots corresponds to the optimal values of the displacements for the proper discretization.

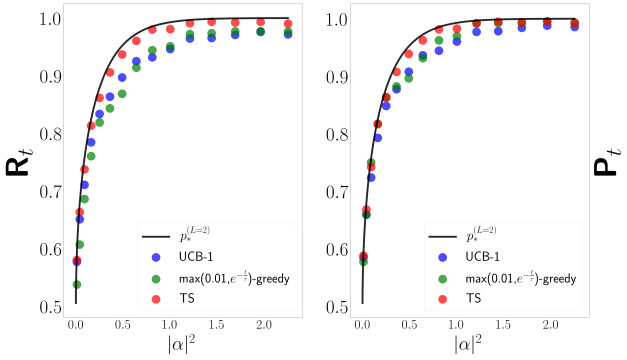

FIG. 7. The performance at episode $t = 5 \cdot 10^5$ of three different RL agents is evaluated as the energy of the coherent states increase. All data points are averaged over 24 agents.

that these results do not sensibly change in the presence of noise, i.e., that the same agents are able to attain near-optimal performance even when unknown errors affect the experiment and hence the learning process.

Firstly, we consider a common experimental imperfection known as dark counts: due to the presence of background noise, each photodetector of the receiver has

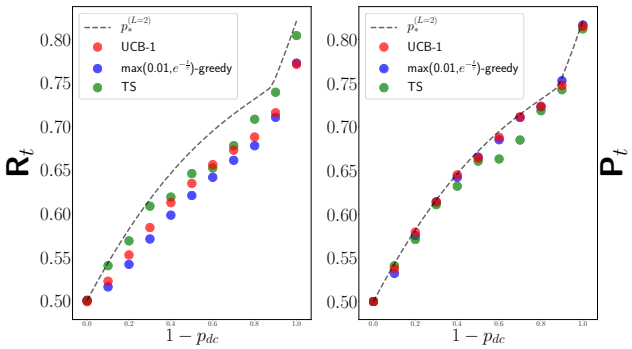

FIG. 8. The performance at episode $t = 5 \cdot 10^5$ of three different RL agents (the same considered in 5) is evaluated as a function of photo-detection noise. The amplitude of the coherent states is fixed to $\alpha = 0.4$; all data points are averaged over 24 agents.

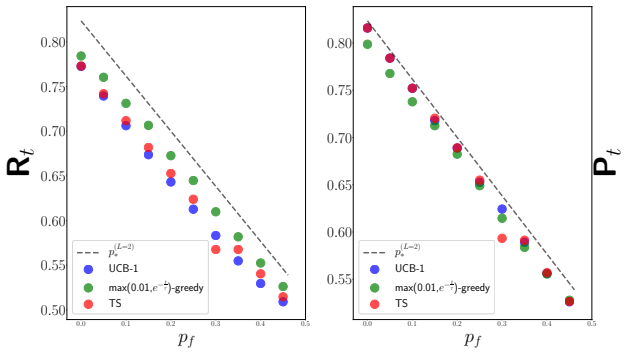

FIG. 9. The performance at episode $t = 5 \cdot 10^5$ of three different RL agents (the same considered in 5) is evaluated as a function of phase flipping probability before the signal arrives to the receiver. The amplitude of the coherent states is fixed to $\alpha = 0.4$; all data points are averaged over 24 agents.

a non-zero probability $p_{\rm dc}$ of detecting a photon even when it receives a vacuum signal. Accordingly, the conditional probability of obtaining an outcome 0 given an input state $|\alpha\rangle$, Eq. (4), is modified by a multiplicative factor $(1 - p_{\rm dc})$.

In Fig. 8 we plot $\mathbf{R}_t$ and $\mathbf{P}_t$ at time $t = 5 \cdot 10^5$ for several RL strategies as a function of $p_{\rm dc} \in [0.5, 1]$, along with the maximum success probability attainable by the corresponding receiver. We see that the final values of $\mathbf{P}_t$ are near-optimal for all values of $p_{\rm dc}$, while $\mathbf{R}_t$ seems to be slightly affected in an intermediate region of values of $p_{\rm dc}$. Since the agents operate on a completely model-free basis and the reward system has been chosen to ensure convergence of the value function to the true success probability, it can be expected that they are still be able to learn in the long term, as shown by the high values of $\mathbf{P}_t$ attained. However, since a dark count effectively increases the chance of (not) obtaining a reward for a (correct) wrong action, the time it takes to learn a near-optimal strategy and to start exploiting it might increase, as shown by the behaviour of $\mathbf{R}_t$. Note that for $p_{dc} \sim 0.5$ the best guess is almost random and thus is easier to learn.

Next, we consider the case where the phase of the incoming signal is flipped before arriving to the receiver, with probability p_f. In this scenario, if the agent guesses for the correct received phase, the corresponding reward will be zero since the phase initially sent was opposite than the received one. In particular, the probability of observing a string of outcomes $p(o_{1:L}|\alpha, \{a(h_{L-1})\})$ in Eq. (4) is modified such that

$$p(o_{1:L}|\alpha, \{a(h_{L-1})\}) \to (1 - p_f)p(o_{1:L}|\alpha, \{a(h_{L-1})\}) + p_f p(o_{1:L}| - \alpha, \{a(h_{L-1})\}) \tag{35}$$

In Fig. 9 we plot $\mathbf{R}_t$ and $\mathbf{P}_t$ at $t = 5 \cdot 10^5$ for several agents as a function of $p_f \in [0.5, 1]$, along with the maximum success probability attainable by the corresponding receiver. As in the case of dark-counts, we see that for all values of p_f, the agents are able to converge to near-optimal $\mathbf{P}_t$ values and they exhibit very small variations in $\mathbf{R}_t$ as p_f increases.

V. DISCUSSION AND CONCLUSIONS

In this article we provided an in-depth study of RL methods for the on-line optimization of coherent-state receivers based on current technology. Such receivers are crucial for the deployment of high-data-rate long-distance communications in free space or optical fiber and they are based on the interplay of several simple quantum gates and measurements, combined to create a complex structure. The RL methods that we analyzed enable to optimize such structure based on the actual experimental conditions and limitations of the communication channel and of the receiver. Thus, they possess a high potential for increasing the flexibility and effectiveness of current receivers and provide a useful addition to the current experimental toolbox. This is even more so the case if we consider that the methods we studied are relatively simple and do not rely only on "shallow" RL techniques, i.e., they are not based on the use of neural networks, which would allow to consider larger state-action spaces, possibly at the cost of a longer training time. We expect that the use of "deep" RL methods could allow to control receivers of multiple and/or multi-mode coherent states, whose best performance is still to be determined at present.

The characterization of the cumulative or simple regret for intermediate times, through non-asymptotic general upper and lower bounds, as well as its extension from Bandits to more general Markov Decision Processes is still an open and active field of research. Quantum technologies can benefit from this progress, and non-trivial

quantum features might appear in more general quantum learning scenarios.

Finally, we would like to stress that the RL problem induced by real-time state discrimination is characterized by intrinsically noisy and stochastic rewards. As such, it stands out from other instances where RL has been applied to quantum physics. In particular, even when performing a good set of actions and guessing rule, an agent might still not be rewarded. This is due to two crucial factors: (i) the intrinsic indistinguishability of quantum states, i.e., even the best receiver has a non-zero probability of discrimination error; (ii) our methods can be applied in real-time to the experiment, hence the binary reward received for a given set of actions is not sufficient to estimate the success probability of the corresponding receiver. Still, the best among our agents are able to reach good configurations and start exploiting them in a number of experiments which is roughly sufficient to try each set of actions only once. This is a key signature of the agent's intelligent behaviour, showing that they make the most out of each reward rather than blindly trying actions at random. Hence we believe that the discrimination problem provides an interesting, rich and flexible sandbox for testing RL in quantum-physics-inspired scenarios and will constitute an interesting works at the intersection between these two fields.

VI. CODE

The code developed to obtain the numerical results of this research can be found at `github.com/matibilkis/marek.git`. Any suggestions, comments and even collaborations are welcome.

VII. ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 845255 and by the Catalan Government for the project QuantumCAT 001-P-001644 (RIS3CAT comunitats) co-financed by the European Regional Development Fund (FEDER).

Appendix A: Optimal state-action values

In this section we verify that — by construction — the optimal policy leads to the maximum success probability $P_*^{(L)}(\alpha)$. It is assumed that Q is always associated with the optimal policy π^*, hence $Q = Q_{\pi^*}$.

At step L, a given history h_L was obtained, and the actions available to the agent are $\hat{k} = a_L$, i.e. guessing for one of the possible phases of the coherent state. The Q-values at this time-step read as

$$\begin{aligned} Q(h_L, a_L) = \mathbb{E}[G_L|h_L, a_L] &= \sum_{r_{L+1}} r_{L+1}\, p(r_{L+1}=1|h_L, a_L) \\ &= p(\alpha^{(a_L)}|o_{1:L}; a_{0:(L-1)}), \end{aligned} \tag{A1}$$

with $o_{1:\ell} = \{o_1, o_2, ..., o_\ell\}$ the observations obtained up to the $(\ell+1)^{\underline{\text{th}}}$ photodetector, and $a_{0:\ell} = \{a_0, a_1, ..., a_\ell\}$ the actions done up to step ℓ. Hence, recalling that the optimal policy given h_ℓ is obtained from Q as $\pi^*(h_\ell) = \arg\max_{a_\ell} Q(h_\ell, a_\ell)$, the optimal guess a_L^* is the one of maximum-likelihood:

$$a_L^* = \arg\max_{a_L} p(\alpha^{(k)}|o_{1:L}; a_{0:(L-1)})\Big|_{k=a_L},$$

By definition of the optimal policy and because optimal Bellman equation Eq. (13), the optimal action to take for a given history h_{L-1} at step $L-1$ is

$$\begin{aligned} a_{L-1}^* &= \arg\max_{a_{L-1}}\ Q(h_{L-1}, a_{L-1}) \\ &= \arg\max_{a_{L-1}} \sum_{o_L} p(o_L|o_{1:(L-1)}; a_{0:(L-1)})\ \max_{a_L}\ Q(h_L, a_L) \\ &= \arg\max_{a_{L-1}} \sum_{o_L} p(o_L|o_{1:(L-1)}; a_{0:(L-1)})\ \max_{k}\ p(\alpha^{(k)}|o_{1:L}; a_{0:(L-1)}) \\ &= \arg\max_{a_{L-1}} \sum_{o_L} \frac{\max_k\ p(o_{1:L}|\alpha^{(k)}; a_{0:(L-1)}) p_k}{p(o_{1:(L-1)}; a_{0:(L-1)})}. \end{aligned} \tag{A2}$$

Following this line of reasoning, we can obtain the optimal actions $a_\ell^*\ \forall \ell$. In particular, for $\ell = 0$, by recursively applying the optimal Bellman equation (13) we have

$$\begin{aligned} Q(h_0, a_0) &= \sum_{o_1} p(o_1; a_0) \max_{a_1} Q(h_1, a_1) \\ &= \sum_{o_1} p(o_1; a_0) \max_{a_1} \sum_{o_2} p(o_2|o_1; a_1) \max_{a_2} Q(h_2, a_2) \\ &= \sum_{o_1} p(o_1; a_0) \max_{a_1} \sum_{o_2} p(o_2|o_1; a_1) \max_{a_2} \sum_{o_3} (...) \Big(\\ &\quad (...) \sum_{o_L} p(o_L|o_{1:(L-1)}; a_{1:(L-1)}) \max_{a_L} Q(h_L, a_L) \Big) \\ &= \sum_{o_1} \max_{a_1} \sum_{o_2} \max_{a_2} \sum_{o_3} (...) \Big(\\ &\quad (...) \sum_{o_L} \max_k p(o_{1:L}|\alpha^{(k)}; a_{0:(L-1)})\, p_k \Big). \end{aligned} \tag{A3}$$

Therefore, by taking the optimal action $a_0^* = \arg\max_{a_0} Q(h_0, a_0)$, we obtain

$$\max_{a_0} Q(h_0, a_0) = p_*^{(L)}. \tag{A4}$$

As pointed out in the main text, the value and action-value functions are related as $v_\pi(s) = \sum_a \pi(a|s) Q_\pi(s, a)$. Therefore, the optimal value function for the initial state is the optimal success probability:

$$v_*(h_0) = \sum_a \delta\big(a, \arg\max_a Q(h_0, a)\big) = Q(h_0, a_0^*) = P_*^L(\alpha). \tag{A5}$$

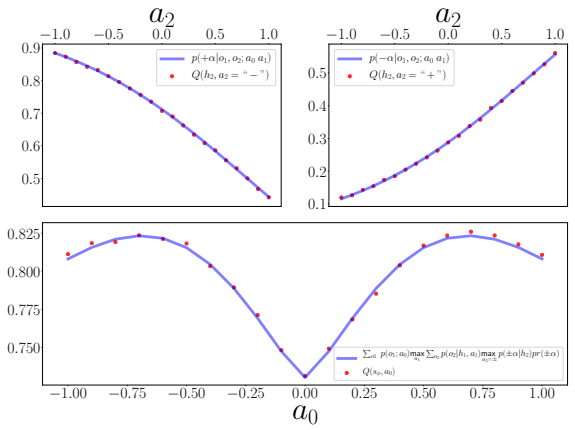

FIG. 10. We plot different values of the Q-estimates, after 10^8 episodes of random exploration ($\epsilon = 1$), updating the Q-estimates according to Q-learning (see Algorithm 1). The random exploration is used in order to ensure that, at *finite* number of episodes, all state-action pairs were equally visited on average.

In Fig. 10 some sections of the Q-estimates $\hat{Q}$ are shown for the 1-greedy interaction policy, after step $t = 10^8$, each update made according to Algorithm 1.

Appendix B: Comparison of different UCB strategies

In this section we show numerical studies on how different choices of $\mathcal{P}(t)$ for the UCB strategy can lead to policies whose learning curves for the case $L = 2$ exhibit different results.As explained in IV C, the upper bound in the probability of overestimate the state-action value can be bounded by Hoeffding's inequality. This probability can be forced to depend on the epispode. In figure 11 we show the performance of three different choices of $\mathcal{P}(t)$ on our receiver, for the same receiver considered in IV B. Firstly we consider UCB-1, the *standard* choice of $\mathcal{P} = t^{-4}$, which for a bandit problem with $K = |\mathcal{A}|$ arms can be easilly proven to have a cumulative regret upperbounded by [75]

$$\mathbb{E}\,\mathcal{L}_t \leq 8 \sum_{a \in \mathcal{A}\setminus\{a^*\}} \frac{\log t}{\Delta_k} + \frac{K\pi^2}{3} \tag{B1}$$

which together with the Lai and Robbins (28) implies that $\mathcal{L}_t = O(\log(t))$. Secondly, we consider UCB-2, with a choice of $\mathcal{P}(t)$ proved to be asymptotically optimal in bandit problems [75]. Lastly, an heuristic and instance dependent variation of $\mathcal{P}(t)$, UCB-3, leads to better $\mathbf{R}_t$ only in the short-term, as exploration is damped too fast (which is also reflected in sub-optimal $\mathbf{P}_t$ even in the long term).

Algorithm's name:	UCB-1	UCB-2	UCB-3
$\mathcal{P}(t)$	t^{-4}	$\frac{1}{1+t\log^2 t}$	$t\,\frac{1}{N_t[s,a]}$

[1] C. W. Helstrom, "Quantum Detection and Estimation Theory," (1976).
[2] A. S. Holevo, *Quantum Systems, Channels, Information* (De Gruyter, Berlin, Boston, 2012).
[3] M. Takeoka and S. Guha, IEEE Int. Symp. Inf. Theory - Proc. **042309**, 2799 (2014), arXiv:1401.5132.
[4] A. Waseda, M. Takeoka, M. Sasaki, M. Fujiwara, and H. Tanaka, J. Opt. Soc. Am. B **27**, 259 (2010).
[5] A. Waseda, M. Sasaki, M. Takeoka, M. Fujiwara, M. Toyoshima, and A. Assalini, J. Opt. Commun. Netw. **3**, 514 (2011).
[6] S. Guha, Phys. Rev. Lett. **106**, 1 (2011), arXiv:1101.1550.
[7] H. Krovi, S. Guha, Z. Dutton, and M. P. Da Silva, IEEE Int. Symp. Inf. Theory - Proc. **062333**, 336 (2014), arXiv:1507.04737.
[8] M. Rosati, A. Mari, and V. Giovannetti, Phys. Rev. A **94**, 062325 (2016).
[9] W. Zwoliński, M. Jarzyna, and K. Banaszek, Opt. Express **26**, 25827 (2018), arXiv:1806.08401.
[10] B. Huttner, N. Imoto, N. Gisin, and T. Mor, Phys. Rev. A **51**, 1863 (1995), arXiv:9502020 [quant-ph].
[11] M. Dušek, M. Jahma, and N. Lütkenhaus, Phys. Rev. A - At. Mol. Opt. Phys. **62**, 9 (2000).
[12] F. Grosshans and P. Grangier, Phys. Rev. Lett. **88**, 4 (2002), arXiv:0109084 [quant-ph].
[13] N. Gisin, G. Ribordy, W. Tittel, and H. Zbinden, Rev. Mod. Phys. **74**, 145 (2002), arXiv:0101098 [quant-ph].
[14] J. A. Bergou, J. Phys. Conf. Ser. **84** (2007), 10.1088/1742-6596/84/1/012001.
[15] C. H. Bennett and G. Brassard, Theor. Comput. Sci. **560**, 7 (2014).
[16] U. Chabaud, T. Douce, F. Grosshans, E. Kashefi, and D. Markham, (2019), arXiv:1905.12700.
[17] S. Pirandola, U. L. Andersen, L. Banchi, M. Berta, D. Bunandar, R. Colbeck, D. Englund, T. Gehring, C. Lupo, C. Ottaviani, J. Pereira, M. Razavi, J. S. Shaari, M. Tomamichel, V. C. Usenko, G. Vallone, P. Villoresi, and P. Wallden, (2019), arXiv:1906.01645.
[18] M. Schuld and F. Petruccione, *Supervised Learning with Quantum Computers* (SPRINGER, 2018).
[19] G. Sentís, E. Bagan, J. Calsamiglia, and R. Muñoz-Tapia, Phys. Rev. A - At. Mol. Opt. Phys. **82** (2010), 10.1103/PhysRevA.82.042312.
[20] G. Sentís, M. Guţă, and G. Adesso, EPJ Quantum Technol. **2** (2015), 10.1140/epjqt/s40507-015-0030-4, arXiv:1410.8700.
[21] M. Guţă and W. Kotłowski, New J. Phys. **12**, 123032 (2010), arXiv:1004.2468.

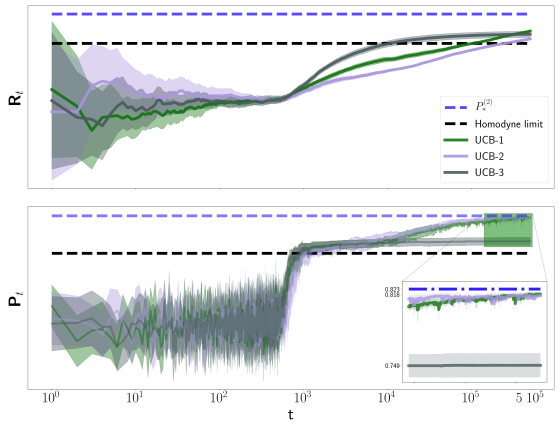

FIG. 11. We compare two different variants of UCB showing that the exploration-exploitation trade-off is an intrinsic feature of our problem.

[22] S. Lloyd and C. Weedbrook, Phys. Rev. Lett. **121** (2018), 10.1103/PhysRevLett.121.040502, arXiv:1804.09139.

[23] M. Fanizza, A. Mari, and V. Giovannetti, IEEE Trans. Inf. Theory **65**, 5931 (2019), arXiv:1805.03477.

[24] C. Blank, D. K. Park, J.-K. K. Rhee, and F. Petruccione, (2019), arXiv:1909.02611.

[25] G. Sergioli, R. Giuntini, and H. Freytes, PLoS One **14** (2019), 10.1371/journal.pone.0216224.

[26] M. Benedetti, E. Grant, L. Wossnig, and S. Severini, New J. Phys. **21** (2019), 10.1088/1367-2630/ab14b5, arXiv:1806.00463.

[27] G. Carleo and M. Troyer, Science **355**, 602 (2016), arXiv:1606.02318.

[28] G. Torlai and R. G. Melko, Phys. Rev. Lett. **119** (2017), 10.1103/PhysRevLett.119.030501, arXiv:1610.04238.

[29] E. P. Van Nieuwenburg, Y. H. Liu, and S. D. Huber, Nat. Phys. **13**, 435 (2017), arXiv:1610.02048.

[30] J. Carrasquilla and R. G. Melko, Nat. Phys. **13**, 431 (2017), arXiv:1605.01735.

[31] G. Torlai, G. Mazzola, J. Carrasquilla, M. Troyer, R. Melko, and G. Carleo, Nat. Phys. **14**, 447 (2018), arXiv:1703.05334.

[32] A. A. Melnikov, H. Poulsen Nautrup, M. Krenn, V. Dunjko, M. Tiersch, A. Zeilinger, and H. J. Briegel, Proc. Natl. Acad. Sci. U. S. A. **115**, 1221 (2018), arXiv:1706.00868.

[33] T. Fösel, P. Tighineanu, T. Weiss, and F. Marquardt, Phys. Rev. X **8** (2018), 10.1103/PhysRevX.8.031084, arXiv:1802.05267.

[34] J. Wallnöfer, A. A. Melnikov, W. Dür, and H. J. Briegel, (2019), arXiv:1904.10797.

[35] M. Bukov, A. G. Day, D. Sels, P. Weinberg, A. Polkovnikov, and P. Mehta, Phys. Rev. X **8** (2018), 10.1103/PhysRevX.8.031086, arXiv:1705.00565.

[36] M. Y. Niu, S. Boixo, V. Smelyanskiy, and H. Neven, in *AIAA Scitech 2019 Forum*, Vol. 5 (Nature Publishing Group, 2019) p. 33, arXiv:1803.01857.

[37] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, Nature **529**, 484 (2016).

[38] M. Osaki, M. Ban, and O. Hirota, Phys. Rev. A **54**, 1691 (1996).

[39] C. Weedbrook, S. Pirandola, R. García-Patrón, N. J. Cerf, T. C. Ralph, J. H. Shapiro, and S. Lloyd, Rev. Mod. Phys. **84**, 621 (2012), arXiv:1110.3234.

[40] M. Takeoka and M. Sasaki, Phys. Rev. A - At. Mol. Opt. Phys. **78**, 1 (2008), arXiv:0706.1038.

[41] S. J. Dolinar, Q. Prog. Rep. (Research Lab. Electron. **111**, 115 (1973).

[42] R. S. Kennedy, MIT Res. Lab. Electron. Q. Prog. Rep. **108**, 219 (1973).

[43] J. Geremia, Phys. Rev. A **70**, 062303 (2004), arXiv:0407205 [quant-ph].

[44] M. Takeoka, M. Sasaki, P. Van Loock, and N. Lütkenhaus, Phys. Rev. A - At. Mol. Opt. Phys. **71**, 1 (2005), arXiv:0410133 [quant-ph].

[45] M. Takeoka, M. Sasaki, and N. Lütkenhaus, Phys. Rev. Lett. **97**, 040502 (2006).

[46] R. L. Cook, P. J. Martin, J. M. Geremia, B. A. Chase, and J. M. Geremia, Nature **446**, 774 (2007).

[47] M. P. Da Silva, S. Guha, and Z. Dutton, Phys. Rev. A - At. Mol. Opt. Phys. **87** (2013), 10.1103/PhysRevA.87.052320, arXiv:arXiv:1201.6625.

[48] R. Nair, S. Guha, and S. H. Tan, Phys. Rev. A - At. Mol. Opt. Phys. **89**, 1 (2014), arXiv:1212.2048.

[49] M. Rosati, A. Mari, and V. Giovannetti, Phys. Rev. A **93**, 062315 (2016), arXiv:1602.03989.

[50] M. T. Dimario and F. E. Becerra, Phys. Rev. Lett. **121**, 023603 (2018).

[51] F. E. Becerra, J. Fan, G. Baumgartner, J. Goldhar, J. T. Kosloski, and a. Migdall, Nat. Photonics **7**, 147 (2013).

[52] C. R. Müller and C. Marquardt, New J. Phys. **17**, 1 (2015), arXiv:1412.6242.

[53] S. Guha and M. M. Wilde, IEEE Int. Symp. Inf. Theory - Proc. , 546 (2012), arXiv:1202.0533.

[54] M. M. Wilde and S. Guha, IEEE Trans. Inf. Theory **59**, 1175 (2013), arXiv:1109.2591.

[55] M. Rosati and V. Giovannetti, J. Math. Phys. **57**, 062204 (2016), arXiv:1506.04999.

[56] M. Rosati, A. Mari, and V. Giovannetti, Phys. Rev. A **96**, 012317 (2017), arXiv:1703.05701.

[57] Z. L. Xiang, M. Zhang, L. Jiang, and P. Rabl, Phys. Rev. X **7**, 011035 (2017), arXiv:1611.10241.

[58] M. T. DiMario, L. Kunz, K. Banaszek, and F. E. Becerra, npj Quantum Inf. **5** (2019), 10.1038/s41534-019-0177-4, arXiv:1907.12515.

[59] R. Bellman, *Dynamic Programming (Reprinted version)* (Dover Publications, 2003) p. 384.

[60] C. Watkins, *"Learning from delayed rewards". PhD thesis*, Ph.D. thesis, Cambridge (1989).

[61] R. Sutton and A. G. Barto, *Reinforcement Learning Sutton* (MIT Press, 2018).

[62] P. Auer, N. Cesa-Bianchi, and P. Fischer, Mach. Learn. **47**, 235 (2002).

[63] D. J. Russo, B. Van Roy, A. Kazerouni, I. Osband, and Z. Wen, "A tutorial on Thompson sampling," (2018), arXiv:1707.02038.

[64] S. P. Singh, T. Jaakkola, and M. I. Jordan, in *Mach. Learn. Proc. 1994* (1994) pp. 284–292.

[65] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, (2013),

arXiv:1312.5602.
[66] G. Shani, J. Pineau, and R. Kaplow, Auton. Agent. Multi. Agent. Syst. **27**, 1 (2013).
[67] M. Egorov, *Deep reinforcement learning with pomdps*, Tech. Rep. (Technical Report, Stanford University, 2015) arXiv:1903.07765.
[68] P. Zhu, X. Li, P. Poupart, and G. Miao, (2018), arXiv:1804.06309.
[69] W. R. Thompson, Biometrika **25**, 285 (1933).
[70] T. L. Lai and H. Robbins, Adv. Appl. Math. **6**, 4 (1985).
[71] R. Agrawal, Adv. Appl. Probab. **27**, 1054 (1995).
[72] W. R. Thompson, Am. J. Math. **57**, 450 (1935).
[73] S. L. Scott, Appl. Stoch. Model. Bus. Ind. **26**, 639 (2010).
[74] T. L. Lai and H. Robbins, Adv. Appl. Math. **6**, 4 (1985).
[75] T. Lattimore and C. Szepesvari, Cambridge Univ. Press , 542 (2018).
[76] S. Bubeck, R. Munos, and G. Stoltz, Theor. Comput. Sci. **412**, 1832 (2011).
[77] A. Garivier, P. Ménard, and G. Stoltz, Math. Oper. Res. **44**, 377 (2019), arXiv:1602.07182.
[78] S. J. Dolinar, *02830334-MIT.pdf*, Ph.D. thesis.