

Learning coherent states by trial and error

M. Bilkis,¹ R. Morral Yepes,¹ M. Rosati,¹ and J. Calsamiglia¹

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 $\sim 10^2$ experiments; (ii) surpass the cumulative reward of the best Gaussian receiver after $\sim 10^3$ experiments; (iii) 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* [61–63], 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 $\sim 10^3$ ***CHECK episodes and attain near-optimal performance ($> 95\%$ optimal) after $\sim 10^4$ episodes, searching a parameter-space of size $3 \sim 10^3$. 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, 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

I. Introduction	1
II. Preliminaries	2
III. Sequential decision-making	4
A. Value functions and the Bellman equation	4
B. Dynamic programming	5
C. Q-learning	5
IV. Model-free reinforcement learning of discrimination strategies	6
A. Benchmarking the success probability via dynamic programming	6
B. Learning a near-optimal receiver via Q-learning	7
C. The multi-armed bandit problem	9
D. Enhancing the agent via UCB and TS	11
E. Noise robustness	12
1. Dark counts	12
2. Phase flip	12
V. Discussion and conclusions	13
VI. Code	13
VII. Acknowledgments	14
A. Optimal state-action values	14
B. Comparison of different UCB strategies	14
References	15

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

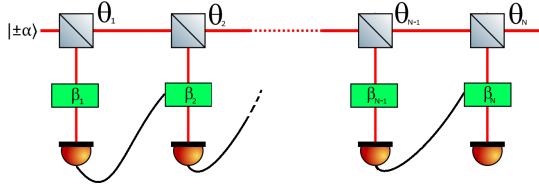


FIG. 1. We depict the experimental setup of the receiver considered. For $L \rightarrow \infty$ one gets Dolinar receiver.

by

$$\begin{aligned} P_s(\alpha, \mathcal{M}) &= \sum_{\hat{k}=0,1} \max p(\alpha^{(k)}, \hat{k}) \\ &= \sum_{\hat{k}=0,1} \max p(\hat{k}|\alpha^{(k)}) p_k. \end{aligned} \quad (1)$$

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

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

where the optimization is carried out over all two-outcome POVMs; note that the Helstrom probability tends to 1/2 for $|\alpha| \rightarrow 0$, 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 $\ell = 0, \dots, L$, as depicted in Fig. 1; this receiver is known to attain the Helstrom probability in the limit $L \rightarrow \infty$ [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

$$|\alpha\rangle|0\rangle \mapsto |\alpha\sqrt{\theta}\rangle_{\text{tr}}|\alpha\sqrt{1-\theta}\rangle_{\text{ref}}, \quad (3)$$

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

$$p(o_{\ell+1} = 0|\alpha, (\beta, \theta)) = |\langle 0|\tilde{\alpha}(\beta, \theta)\rangle|^2 = e^{-|\tilde{\alpha}(\beta, \theta)|^2}, \quad (4)$$

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 $|\alpha_\ell\rangle = |\alpha\sqrt{\theta_0 \dots \theta_{\ell-1}}\rangle$, with $\theta_0 = \emptyset$. Since the experimenter can use all the past history $h_\ell = (a_0, o_1, \dots, a_{\ell-1}, o_\ell)$, 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 $o_{1:L} = (o_1, \dots, o_L)$ can be written as

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

where the total conditional probability of $o_{1:L}$ factors into a product of single-layer probabilities, Eq. (4):

$$p(o_{1:L}|\alpha, \{a(h_{L-1})\}) = \prod_{\ell=1}^L p(o_\ell|\alpha, a(h_{\ell-1})). \quad (6)$$

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 :

$$\{a_\ell^*\} = \arg \max_{\{a_\ell\}} P_s(\alpha, \{a_\ell\}) \quad (7)$$

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

$$P_*^{(L)}(\alpha) = \max_{\{a_\ell\}} P_s(\alpha, \{a_\ell\}), \quad (8)$$

and omitting 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 $\ell = 0, \dots, L$ of each episode $t = 1, \dots, T$, 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 environments state is s . This interaction is ussually known as a Markov decission 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., $h_\ell = (a_0, o_1, \dots, a_{\ell-1}, o_\ell)$. 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. Once episode t is ended, 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,

$$G_\ell^{(t)} = \sum_{i=0}^{L-\ell} \gamma^i r_{i+\ell+1}^{(t)}, \quad (9)$$

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., $L \rightarrow \infty$, 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*:

$$v_\pi(s) = \mathbb{E}_\pi [G_\ell | s_\ell = s], \quad (10)$$

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. (10) 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]:

$$v_\pi(s) = \sum_{a \in \mathcal{A}, s' \in \mathcal{S}, r \in \mathcal{R}} \tau(s', r | s, a) \pi(a | s) (r + \gamma v_\pi(s')) . \quad (11)$$

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:

$$\begin{aligned} v^*(s) &:= v_{\pi^*}(s) = \max_{\pi} v_{\pi}(s) \\ &= \max_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} \tau(s', r | s, a) (r + \gamma v^*(s')) . \end{aligned} \quad (12)$$

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 :

$$Q_\pi(s, a) = \mathbb{E}_\pi [G_\ell | s_\ell = s, a_\ell = a] , \quad (13)$$

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

$$\begin{aligned} Q^*(s, a) &:= Q_{\pi^*}(s, a) = \max_{\pi} Q_\pi(s, a) \\ &= \sum_{s' \in \mathcal{S}, r \in \mathcal{R}} \tau(s', r | s, a) (r + \gamma \max_{a' \in \mathcal{A}} Q^*(s', a')) . \end{aligned} \quad (14)$$

Notice that introducing the Q-function is motivated in particular for model-free settings, in which τ is unknown, since the value function $v_\pi(s)$ retrieves no information on what the quality of a certain action a is. In this setting, the agent can only access to information of the environment via performing actions, and the optimal Q-values have to be estimated from experience.

B. Dynamic programming

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 (12) to find the optimal policy step-by-step. Indeed, 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. (12) at the last time-step,

$$v_L^*(s) = \max_{a \in \mathcal{A}_L} \sum_{r \in \mathcal{R}_{L+1}} \tau(r | s, a) r , \quad (15)$$

where we have omitted the terminal state s_{L+1} and used the fact that $v_{L+1}(s) = 0$. The solution to Eq. (15) 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 $\ell = L, \dots, 0$, 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 it is 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 there afterwards.

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. (11,12,14), 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. (14), 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 episode $t = 1, \dots, T$.

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

$$\begin{aligned} \hat{Q}(s_\ell, a_\ell) &\leftarrow (1 - \lambda_t(s_\ell, a_\ell)) \hat{Q}(s_\ell, a_\ell) \\ &+ \lambda_t(s_\ell, a_\ell) \left(r_{\ell+1} + \gamma \max_{a' \in \mathcal{A}_{\ell+1}} \hat{Q}(s_{\ell+1}, a') \right) , \end{aligned} \quad (16)$$

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 k of iterations of the update rule Eq. (16) 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]:

$$\begin{aligned} \hat{Q}(s, a) &\xrightarrow{k \rightarrow \infty} Q^*(s, a) \quad \forall s \in \mathcal{S}, a \in \mathcal{A}(s) \\ \text{iff } \sum_{t(s,a)} \lambda_t(s, a) &= \infty, \quad \sum_{t(s,a)} \lambda_t(s, a)^2 < \infty , \end{aligned} \quad (17)$$

Algorithm 1: Q-learning pseudo-code.

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

```

for  $t$  in  $1 \dots T$  do
    initialize  $s_0$ 
    for step  $\ell$  in episode do
        Take action  $a_\ell$  according to  $\pi$  (e.g.  $\epsilon$ -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
             $s_\ell \leftarrow s_{\ell+1}$ 

```

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.

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. (16), 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 a policy $\pi(a|s)$ for all episodes, such as ϵ -greedy where with probability ϵ the agent chooses a random action and otherwise he chooses the greedy action that maximizes the current Q -estimate. However, as we will see below one can also consider more general strategies where the policy $\pi(a|s)$ changes or is updated from episode to episode.

IV. MODEL-FREE REINFORCEMENT LEARNING OF DISCRIMINATION STRATEGIES

In the following we describe how we applied RL to achieve near-optimal success probability with the coherent-state receiver described in Sec. II, 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-steps $\ell = 0, \dots, L$, 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 available at time-steps $\ell < L$ are the displacements β_ℓ and BS parameters θ_ℓ available at that layer, conditioned on the history of observations, $a(h_\ell)$, 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

$$\tau(\alpha_{\ell+1}^{(k')} | \alpha_\ell^{(k)}, a_\ell) = \delta(k', k) \quad \forall \ell \leq L, \quad (18)$$

$$\tau(r_{L+1} | \alpha_L^{(k)}, a_L) = \delta(r_{L+1}, 1) \delta(a_L, k), \quad (19)$$

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 transition probabilities. That is the agent is fully knowledgeable of the model that describes the receiver, he even knows the amplitude $|\alpha|$ of the optical signals, and his task is to optimize his setup (including feedback schemes) so as to maximize the success probability of discriminating the two states $|\pm\alpha\rangle$. In this case, as discussed in Sec. III, we can define a belief distribution over the states, with initial value equal to the prior, $b(\alpha_0^{(k)}) = p_k$ and Bayes’ update rule:

$$b_{o'}(\alpha_{\ell+1}^{(k)}) = \frac{p(o' | \alpha_\ell^{(k)}, a) b_o(\alpha_\ell^{(k)})}{\sum_{k' \in \{0, 1\}} p(o' | \alpha_\ell^{(k')}, a) b_o(\alpha_\ell^{(k')}), \quad (20)}$$

given that action a was performed, leading to observation o' with probability $p(o' | \alpha_\ell^{(k)}, a)$ given by Eq. (4). Moreover, the belief’s transition function is simply given by the probability of obtaining outcome o' given the belief state b_o and the action a :

$$\tau(b_{o'} | b_o, a) = \sum_k p(o' | \alpha_\ell^{(k)}, a) b_o(\alpha_\ell^{(k)}) \quad \forall \ell \leq L, \quad (21)$$

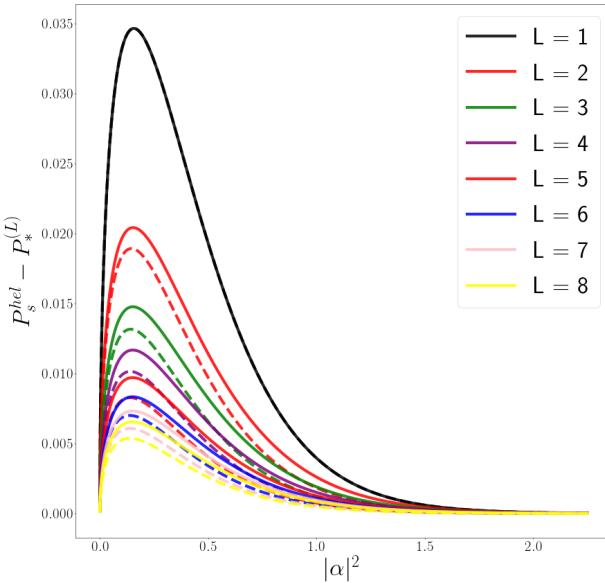


FIG. 2. We show the difference between the best probability of success attainable for a fix L and the optimal probability of success in discriminating two coherent states. The results were obtained by dynamic programming, explained in Sec.III B. Solid lines correspond to fixed attenuations θ_ℓ such that 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.

while the final reward's transition function is

$$\begin{aligned} \tau(r_{L+1}|b_{o_L}, a_L) &= \sum_k \tau(r_{L+1}|\alpha_L^{(k)}, a_L) b_{o_L}(\alpha_L^{(k)}) \\ &= \delta(r_{L+1}, 1) b_{o_L}(\alpha_L^{(a_L)}). \end{aligned} \quad (22)$$

Hence the optimal Bellman equation, Eq. (15), for the state-value function of this POMDP at step L reads

$$v_L^*(b_{o_L}) = \max_{a_L} \{b_{o_L}(\alpha_L^{(a_L)})\}, \quad (23)$$

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

$$v_\ell^*(b_o) = \max_{a \in \mathcal{A}_\ell} \sum_{o' \in \mathcal{O}} \sum_k p(o'|\alpha_\ell^{(k)}, a) b_o(\alpha_\ell^{(k)}) v_{\ell+1}^*(b_{o'}). \quad (24)$$

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. (23). 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. (20), needed to solve Eq. (24) and hence we use interpolation methods to obtain them.

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 $b(\alpha_0^{(k)})$:

$$v_0^*(b) = \mathbb{E}_{\pi^*}[r_{L+1}|b(\alpha_0^{(k)})] = \max_{\{a_\ell\}} P_s(\alpha, \{a_\ell\}), \quad (25)$$

as can be seen by repeated applications of Eqs. (20,24) and detailed in Appendix 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. (16) with $s \rightarrow h$ 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. (17). 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 $\ell = 0$:

$$\max_{a_0} Q^*(a_0) = \max_{a_0} \mathbb{E}_{\pi^*}[r_{L+1}|a_0] = P_*^{(L)}(\alpha), \quad (26)$$

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. (26) 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)

$$\mathbf{R}_t = \frac{1}{t} \sum_{i=1}^t G_0^{(i)} = \frac{1}{t} \sum_{i=1}^t r_{L+1}^{(i)}, \quad (27)$$

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,

$$\mathbf{P}_t = P_s^{(rl)}(\alpha, \{a_\ell^{(t)*}\}), \quad (28)$$

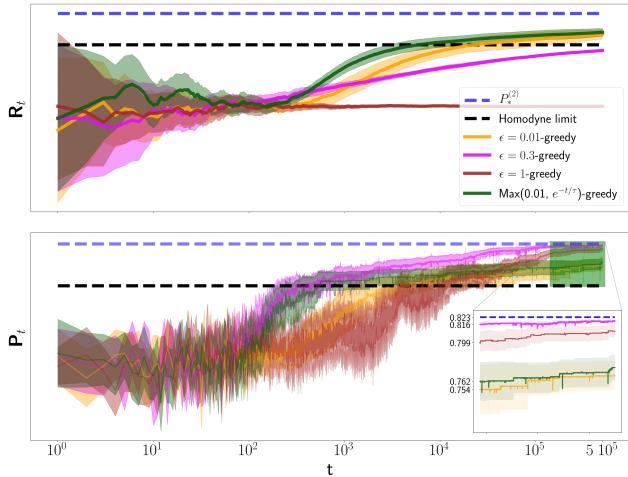


FIG. 3. We benchmark traditional Q-learning with different schedules on ϵ as the episode number increases. The figures of merit are averaged over $A = 48$ agents and show the corresponding uncertainty region.

where the best actions at episode t are obtained by going greedy with respect to the current Q -estimate, i.e.,

$$a_\ell^{(t)*}(h_\ell) = \arg \max_a \hat{Q}^{(t)}((a_0^{(t)*}, o_1, \dots, a_{\ell-1}^{(t)*}, o_\ell), a). \quad (29)$$

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 $t \rightarrow \infty$, for a *good* learner it is expected that $\mathbf{R}_t \rightarrow \mathbf{P}_t$, 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., $\epsilon(t) = \max\{e^{-2 \cdot 10^{-2}t}, 0.01\}$. 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 intrinsic 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. (25), and (ii) the success probability attainable via a standard homodyne measurement, which is optimal among Gaussian receivers [40].

In the first place we note that a completely random choice over the action space (1-greedy policy) leads to the extremely poor cumulative reward (per episode) of $1/2$ for all times, which is expected because a random guess (last action) leads to $p_s = 1/2$, nevertheless \mathbf{P}_t is expected to eventually reach the optimal value at the moment that all actions are sampled a significant amount of times.

does not lead to the best \mathbf{P}_t in a finite number of episodes, for the interaction policies we considered. In turn, a 0.3-greedy policy has—at all episodes—a higher \mathbf{P}_t than the 1-greedy one, being 99.1% the optimal success probability $P_*^{(L=2)}$. Furthermore, for the 0.3-greedy policy, \mathbf{R}_t is at most a 70% of the optimal success probability $P_*^{(L=2)}$, whereas a 1-greedy policy leads to a success rate of 50% of the optimal one. As a matter of fact, the exploration rate should be decrease in the long term in order to \mathbf{R}_t increases; therefore we consider a third type of ϵ -greedy agent, whose exploration rate is exponentially decaying up to a constant: $\epsilon(t) = \max\{e^{-\frac{t}{\tau}}, \epsilon_0\}$, with $\tau = 2 \cdot 10^2$ and $\epsilon_0 = 0.01$. This standard choice assures that at initial episodes the agent favours exploration, whereas at $t \sim \tau \log \frac{1}{\epsilon_0}$, the agent’s behaviour collapses to an ϵ_0 -greedy policy. Hence, the exploration-exploitation trade-off is put in evidence: our exponentially decaying ϵ -greedy policy favours exploitation (i.e. performing potential optimal actions), surpassing the homodyne limit about episode $\sim 5 \cdot 10^3$ (which is comparable with the parameter space size), at the cost of dismissing the exploration of possible better strategies: at episode $5 \cdot 10^5$ \mathbf{P}_t is about $\sim 92\%$ the optimal success probability $P_*^{(L=2)}$.

Our numerical results show that a standard Q-learning schema successfully trains agents that surpass the homodyne limit of optical detection and discover strategies whose error rate is comparable with the optimal receiver. Those agents can only get a binary signal at the end of each episode and, very surprisingly, they completely ignore the details of the experiment (e.g. the energy of the coherent states). Nevertheless, as already stressed, the central figure of merit, \mathbf{R}_t , that reflects the number of correct guesses our agent makes *during* the learning process; is simply not optimal at $t = 5 \cdot 10^5$. At a first glance, this may seem unimportant since the agent

did discovered efficient strategies by this episode number ($\mathbf{P}_t \rightarrow P_*^{(L=2)}$); notice, on the contrary, that this measure of performance is only known by an external oracle: the agents completely ignore \mathbf{P}_t . Hence, the only faithful way to measure agents' learning performance is to evaluate them by their empirical acts, which is exactly captured by \mathbf{R}_t . Thus, the need of tuning the interaction policy in such a way that $\mathbf{R}_t \rightarrow P_*^{(L)}$ arises. In this line of reasoning, we turn to study a simpler Sequential Decision Problem, known as bandit problem, for which several tools have been developed s.t. optimal cumulative reward per episode, \mathbf{R}_t is achieved.

C. The multi-armed bandit problem

In the following we describe a basic instance of sequential decision making, known as multi-armed bandit problem; we introduce smarter interaction policies that are adapted to Q-learning at the last section of the paper. In addition, several theoretical tools are introduced in order to study bandits' learning curves, which are a cornerstone to tackle more challenging situations such as learning optimal policies over a MDP.

In this setting, there's a single default state and the agent faces a fixed set of actions $a \in \mathcal{A}$ at all episodes, 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. In analogy with Sec. III, the agent's objective is to always perform the action whose mean reward is the highest, which we denote as a^* .

The most straightforward policy to use is the ϵ -greedy (already introduced in Sec.III C); in which an estimate of the mean reward $\hat{Q}(a)$ is kept for each action, and the next action is chosen at random with probability ϵ , or greedy w.r.t \hat{Q} otherwise, as described in Algorithm 2.

Algorithm 2: ϵ -greedy for bandit problems.

```

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


```

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

$$\hat{Q}(a) \xrightarrow[t \rightarrow \infty]{} \sum_{r \in \mathcal{R}} r \tau(r|a) \quad \forall a \in \mathcal{A}. \quad (30)$$

The bandit problem is an ideal framework to highlight the aforementioned crucial difference between learning strategies that attain the main goal of obtaining the (near) optimal policy after some time and the more refined strategies that are also procure high cumulative rewards during the learning process. Compared with GO, or even complicated maze problems, the task of identifying the optimal arm to pull is straightforward (from (30) $a^* = \arg \max_a \hat{Q}(a)$). This might seem to put on equal footing all learning strategies. However, the choice of strategy, which rules the sequence by which arms are pulled, will influence a lot shape of actual accumulated rewards in the transient period and the rate at which the optimal strategy 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 [71] where one needs to find the right compromise between advancing in the research the best treatment while effectively treating current patients.

Since most strategies do eventually reach the optimal success rate it is very common in bandit theory to study the so-called *cumulative regret*:

$$\mathcal{L}_t = \sum_{k=1}^t Q(a^*) - Q(a^{(t)}) = t (Q(a^*) - \mathbf{R}_t), \quad (31)$$

which is directly related the expected loss due to not performing the optimal action. The relationship with the cumulative reward per episode, \mathbf{R}_t , as defined in Sec.IV B is shown explicitly.

With this figure of merit in mind, the bandit problem is solved by finding a policy π s.t. $\mathbb{E}_\pi \mathcal{L}_t = o(t)$ (i.e. $\lim_{t \rightarrow \infty} \frac{\mathbb{E}_\pi \mathcal{L}_t}{t} = 0$). asymptotically optimal convergence rate is given by the Lai and Robbins bound [69]

$$\mathbb{E}_\pi \mathcal{L}_t \geq \log t \left(\sum_{a \in \mathcal{A}} \frac{\Delta_a}{\text{KL}(a||*)} + o(1) \right) \quad \forall \pi, \quad (32)$$

with $\Delta_a = Q(a^*) - Q(a)$ and $\text{KL}(a||*)$ the Kullback-Leibler divergence between the reward distributions $\tau(r|a)$ and $\tau(r|a^*)$.

As stressed in Sec.IV B, the ϵ -greedy policy does never solve the problem: with probability ϵ *any* action a is performed at episode t , despite this action leading to a statistically proven sub-optimal reward, as measured by $\hat{Q}(a^{(t)})$. This motivates, at least, two different strategies: the use of Upper Confidence Bounds (UCB) [62, 69, 70] and Thompson sampling (TS) [63, 71–73]. On the one hand, the use of concentration inequalities such as the Hoeffding bound, s.t. for each action at episode t , optimism (on the true value of $Q(a)$) is kept under the face of uncertainty (reflected on the probability that $\hat{Q}(a^{(t)})$ differs from $Q(a^{(t)})$). On the other, the agent is asked to *mimic* the reward distribution of each arm, by using a certain family of distributions that at each episode updates the likelihood of a each arm leading to a certain mean reward.

In UCB, the agent keeps a record of the number of times each action was selected up to episode t , which we denote as $N(a^{(t)})$. Hoeffding's inequality bounds the probability of overestimating $Q(a)$ at episode t :

$$\Pr[Q(a) < \hat{Q}(a) + \bar{\epsilon}(t)] \leq e^{-2N_t(a)\bar{\epsilon}(t)^2} = \underline{\Delta}(t), \quad (33)$$

meaning that with a high probability $1 - \underline{\Delta}(t)$, $Q(a)$ is at most as large as the UCB of $\hat{Q}(a)$

$$\text{ucb}_t(a) := \hat{Q}(a) + \sqrt{\frac{-\log \underline{\Delta}(t)}{2N_t(a)}} \quad \forall a \in \mathcal{A} \text{ s.t. } N_t(a) > 0. \quad (34)$$

Hence, the UCB strategy consists in choosing the action that maximizes the largest plausible value of $Q(a)$: $a^{(t)} = \delta(a, \max_a \text{ucb}_t(a))$, as shown in Algorithm 3. The functional form of $\underline{\Delta}(t)$ can be tuned to balance exploration and exploitation. In particular, by choosing $\underline{\Delta}(t) = t^{-4}$, it can be shown that the asymptotic expected cumulative regret is $O(\log(t))$ (see Appendix B for more details). In this way, actions that have not been visited enough and therefore its Q-estimate is more uncertain, will be more likely to be explored. On the contrary, actions whose Q-estimate is suboptimal with certainty $\underline{\Delta}(t)$, would be more unlikely to be explored again.

Algorithm 3: UCB for bandit problems.

```

input :  $\hat{Q}(a), N(a)$  initialized to zero  $\forall a \in \mathcal{A}$  .  $\underline{\Delta}(t)$ .
for  $t$  in  $1, \dots, T$  do
  if  $t \geq |\mathcal{A}|$  then
    (choose each action once)  $a^{(t)} = t$ 
  else
    choose  $a^{(t)} = \arg \max ucb_t(a)$ , (see Eq.34)
    observe reward  $r$ 
    update Q-value:
     $\hat{Q}(a^{(t)}) \leftarrow \hat{Q}(a^{(t)}) + \lambda(a^{(t)})[r - \hat{Q}(a^{(t)})]$ 
    record visit:
     $N(a^{(t)}) \leftarrow N(a^{(t)}) + 1$ 

```

As mentioned before, TS strategy is based on trying to *mimic* the of each arm. For this, the expected reward \bar{r} obtained by selecting action a is treated as a random variable, with a probability distribution $f_t(\bar{r}|a) \forall a \in \mathcal{A}$. 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} : $a^{(t)} = \delta(a, \max_{a \in \mathcal{A}} \{\bar{r} \sim f_t(\bar{r}|a)\})$. 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 updates, families of distributions that are closed under this update rule are used. In the case of Bernoulli bandits, beta-distributions are employed since those are precisely their conjugate priors. That is, given

$$f_t(\bar{r}|a) \propto \bar{r}^{\mu_t(a)-1} (1-\bar{r})^{1-\nu_t(a)} = \text{Beta}(\mu_t(a), \nu_t(a)), \quad (35)$$

upon obtaining a reward r the prior is updated to a beta distribution with parameters $\mu_{t+1}(a) = \mu_t(a) + r$, $\nu_{t+1}(a) = \nu_t(a) + 1 - r$. The distributions can be chosen to be flat at the first episode, i.e., $\mu_1(a) = \nu_1(a) = 1 \forall a \in \mathcal{A}$. Thus, if a certain action has not been sampled enough at episode t , its reward distribution will still be very broad and when sampled it can easily return a higher value than that obtained from other (more peaked) distributions, hence TS will favour to explore this 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 is sampled again; since the corresponding prior will be highly peaked at low values.

Finally, it is worth mentioning that TS has been proven to attain asymptotically optimal expected cumulative regret [?]. The pseudo-code of TS for Bernoulli bandits is described in Algorithm 4.

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  $1, \dots, T$  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:
     $\mu_{t+1}(a) = \mu_t(a) + r, \nu_{t+1}(a) = \nu_t(a) + 1 - r$ 

```

As defined in IV B, an additional figure of merit, \mathbf{P}_t , can be evaluated, in which the problem of identifying the best arm a^* is casted, without giving importance to the cumulative expected regret \mathcal{L}_t . In the bandit literature this is known as *simple regret* minimization, which at episode t we denote as Λ_t , defined as:

$$\Lambda_t = Q(a^*) - Q(a^{(t)*}), \quad (36)$$

where $a^{(t)*}$ is the agent's *recommendation* of which the optimal action is, available at episode t . In particular, it is known that in order to have asymptotically optimal simple regret, each action should be sampled a linear number of times (nonetheless this does not hold for the non-asymptotic regime t).*** Furthermore, simple and cumulative expected regrets have been related: the smaller the cumulative regret, the larger the simple regret; we refer the interested reader to [?] for more details.

To finish this section, we will consider a numerical example of a 3-armed bandit problem, in which the performance of different policies is benchmarked. To the best knowledge of the authors, theoretical guarantees about the behaviour of non-asymptotic regret for different policies, as well as its extension to Markov Decision Processes is still an open research problem. In addition, a recent work [?] shows that the asymptotic bound

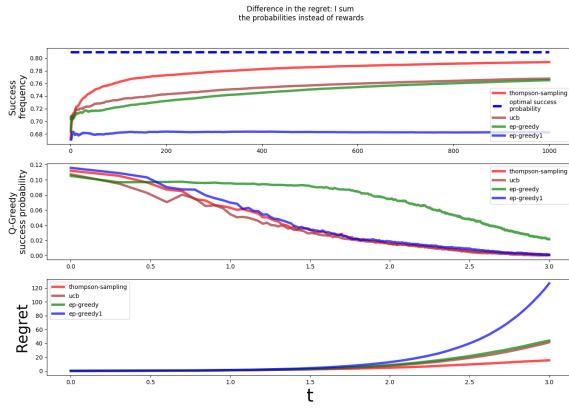


FIG. 4. We show the learning curves for the different policies; we consider the cumulative reward over episode (in the upper plot), the simple regret (middle plot), and the cumulative regret (plot of below), for $\beta \in 0, -\alpha, \beta^*$, for $\alpha = 0.4$ and $\beta^* = -0.74$. All curves are averaged over 10^3 agents. The recommendations J_t for the simple regret are taken to be the actions that maximize the Q-table for the ϵ -greedy and UCB policies, and the action that maximizes the mean value of $f_t(\bar{r}|a)$ for TS.

presented in Eq.(32) must be compensated with a factor problem dependent factor of order, $-O(\log(\log(t)))$ s.t. non-asymptotic behaviours are taken into account.

We consider the receiver described in Sec.??, for the case $L = 1$, but with fixed guessing rule, given by the parity of the outcome observed, *** for optimal beta's? proved by Dolinar to be optimal [?]. With this, each possible displacement β constitutes an action a_0 (recalling the notation used in last section) of a bandit problem, with expected reward $Q(\beta) = \sum_{o_1} p((-1)^{o_1 + \text{sgn}(a_0)} \alpha | o_1; a_0)$. The numerical results are shown in Fig.4

As exemplified in this section with bandit problems, a better balance between exploration and exploitation can be done by considering *UCB* and *TS* strategies, in order to maximize the expected cumulative reward \mathbf{R}_t among episodes. This is achieved either via concentration bounds (e.g. UCB, providing optimism in the face of uncertainty), or by bayesian estimation (e.g. TS, keeping a belief distribution on the mean reward led by each action). In the following section we adapt these strategies in order to design new policies that turn to enhance the traditional Q-learning schema as presented in IV B.

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. (16), but it employs the UCB method to determine the interaction

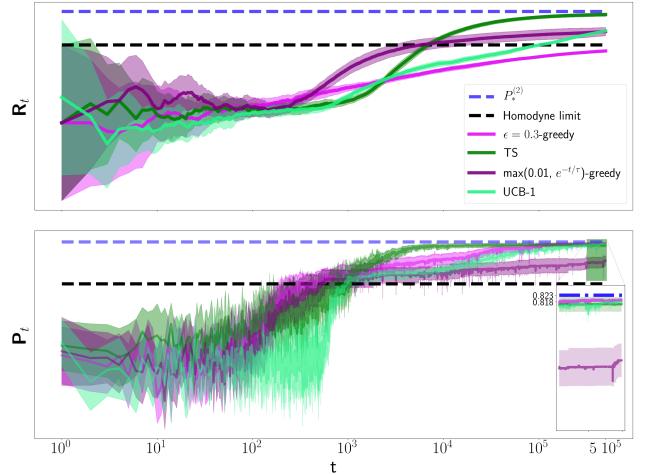


FIG. 5. We show the learning curves for the enhanched Q-learning agents via bandit methods. On the upper plot we despict \mathbf{R}_t , the agent's success rate per episode, whereas on the bottom plot we despict \mathbf{P}_t , the success probability of the agent's recommended actions at episode t ,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

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. (34), 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. (35), 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.

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

$$\hat{Q}_L^{(T)}((\beta, o_1, \beta_{o_1}, o_2), 0) - \hat{Q}_L^{(T)}((\beta, o_1, \beta_{o_1}, o_2), 1). \quad (37)$$

Note that the sign of Eq. (37) 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, obtained from dynamic programming, meaning that the agents are more confident about their guess in these regions.

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 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.

1. Dark counts

Firstly, we consider a common experimental imperfection known as dark counts: due to the presence of background noise, each photodetector of the receiver has a non-zero probability p_{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 $(1 - p_{dc})$.

In Fig. 7 we plot \mathbf{R}_t and \mathbf{P}_t at time $t = 5 \cdot 10^5$ for several RL strategies as a function of $p_{dc} \in [0.5, 1]$, along

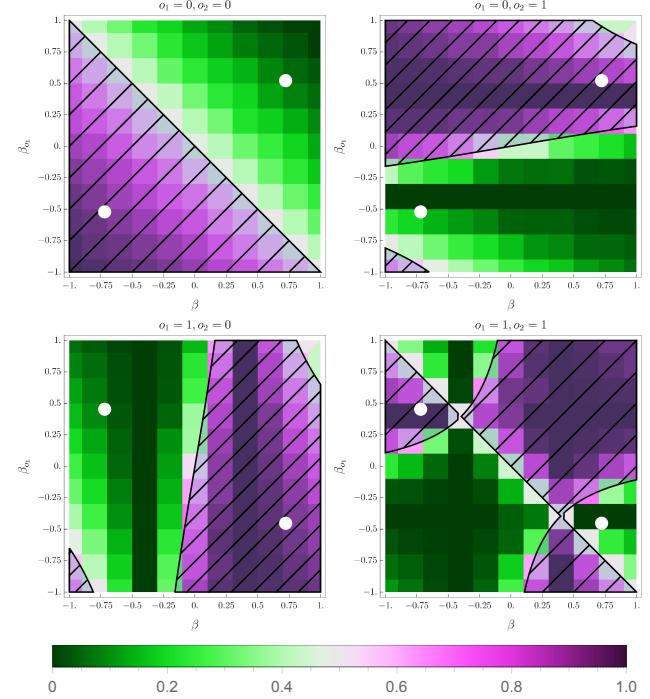


FIG. 6. Density plot of the difference between the estimated Q -values for guessing "plus" and "minus" as a function of the displacements at the first and second layer, for each possible sequence of outcomes, with $\alpha = 0.4$. The shaded areas correspond to the regions where the optimal guess, taken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

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_{dc} , while \mathbf{R}_t seems to be slightly affected in an intermediate region of values of p_{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.

2. Phase flip

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

V. DISCUSSION AND CONCLUSIONS

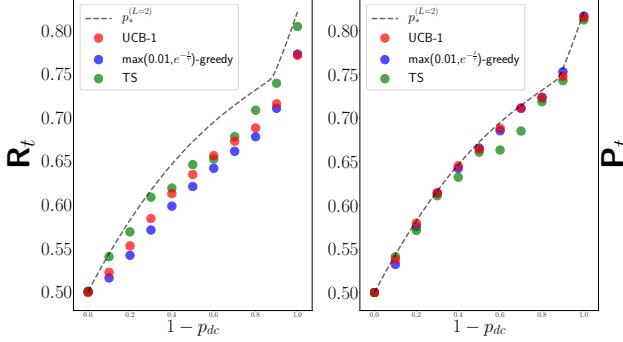


FIG. 7. The performance at episode $t = 5 \cdot 10^5$ of three different RL agents (the same considered in 5) is evaluated as a function of photo-detection noise. The amplitude of the coherent states is fixed to $\alpha = 0.4$; all data points are averaged over 24 agents.

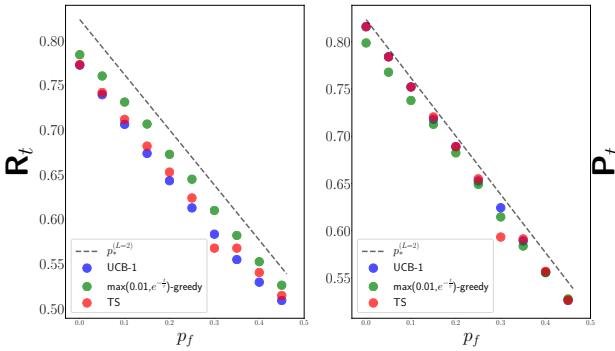


FIG. 8. The performance at episode $t = 5 \cdot 10^5$ of three different RL agents (the same considered in 5) is evaluated as a function of phase flipping probability before the signal arrives to the receiver. The amplitude of the coherent states is fixed to $\alpha = 0.4$; all data points are averaged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

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

$$\begin{aligned} p(o_{1:L}|\alpha, \{a(h_{L-1})\}) &\rightarrow (1 - p_f)p(o_{1:L}|\alpha, \{a(h_{L-1})\}) \\ &+ p_f p(o_{1:L} - \alpha, \{a(h_{L-1})\}) \end{aligned} \quad (38)$$

In Fig. 8 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.

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 are based on the interplay of several simple quantum gates and measurements that are combined to create a complex structure. In practice, it would be convenient to optimize such structure in real time, based on the actual experimental conditions and limitations of the communication channel and of the receiver. The RL methods that we implemented and analyzed thus 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 studied methods mainly rely on tabular functions, i.e., keeping individual scores of each visited state and action, and not on function approximation or neural networks, which allow to approximate the value of unseen instances through those in their vicinity, thus increasing the size of the parameter space. Hence, in future works we will focus on the extension to deep RL methods, which could be employed to control receivers of multiple and/or multi-mode coherent states, whose best performance is still to be determined at present.

Finally, we would like to stress that the RL agents we studied are able to learn from intrinsically stochastic or noisy rewards. In other words, 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 in the best case, there is a non-zero probability of discrimination error; (ii) our simulations reproduce faithfully the experimental setting, in the sense that, at each episode, each agent receives does not obtain a conclusive answer on the value of the chosen actions but rather a single bit of information about it. Hence, the agent has to sample repeatedly a sequence of actions in order to estimate their value, and these sampling steps are explicitly counted in the total number of runs of the experiment. Still, the best among our agents are able to reach good configurations and start exploiting them in a number of runs which is comparable with the number of actions.

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 Unions 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). To Gaggia, for all those coffees provided before passing away; you will always have a place deep in our hearts.

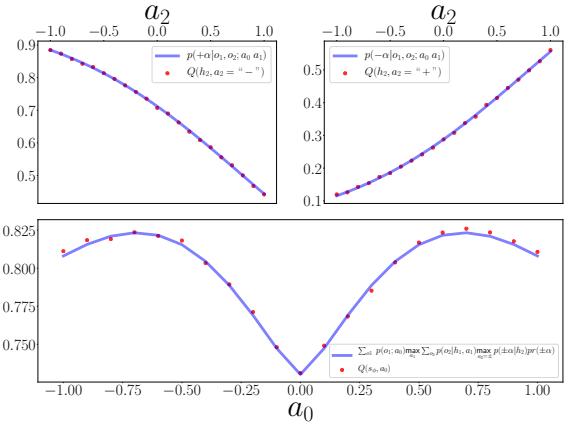


FIG. 9. We plot different values of the Q-estimates, after 10^8 episodes of random exploration ($\epsilon = 1$), updating the Q-estimates according to Q-learning (see Algorithm 1). The random exploration is used in order to ensure that, at finite number of episodes - all state-action pairs were equally visited on average.

Appendix A: Optimal state-action values

In this section we verify that - by construction - the optimal policy leads to the maximum success probability $p_*^{(L)}$. We assume action a_ℓ corresponds to displacing the attenuated signal at step ℓ , i.e. the attenuations are fixed. In addition, 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 (by following the optimal policy), 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 step equal the mean return $\mathbb{E}[G_L|h_L, a_L] = p(r_{L+1} = 1|h_L, a_L)$:

$$Q(h_L, a_L) = p(\alpha^{(a_L)}|o_{1:L}; a_{0:(L-1)}),$$

$o_{0:\ell} = \{o_1, o_2, \dots, o_\ell\}$ the observations obtained up to the $(\ell + 1)^{\text{th}}$ photodetector, and $a_{0:\ell} = \{a_0, a_1, \dots, a_\ell\}$ the actions done up to step ℓ . Hence, 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. (14), the optimal action to take for a given history h_{L-1} at step $L - 1$ is

$$\begin{aligned} a_{L-1}^* &= \arg \max_{a_{L-1}} Q(h_{L-1}, a_{L-1}) \\ &= \arg \max_{a_{L-1}} \sum_{o_L} p(o_L|o_{1:(L-1)}; a_{1:(L-1)}) \max_{a_L} Q(h_L, a_L) \\ &= \arg \max_{a_{L-1}} \sum_{o_L} p(o_L|o_{1:(L-1)}; a_{1:(L-1)}) \max_k p(\alpha^{(k)}|o_{1:L}; a_{0:(L-1)}) \\ &= \arg \max_{a_{L-1}} \sum_{o_L} p^{-1}(o_{1:(L-1)}; a_{1:(L-1)}) \max_k p(o_{1:L}|\alpha^{(k)}; a_{0:(L-1)}) \end{aligned} \quad (\text{A1})$$

Following this line of reasoning, up to the first step

$\ell = 0$, it follows that

$$\begin{aligned} Q(h_0, a_0) &= \sum_{o_1} p(o_1|a_0) \max_{a_1} Q(h_1, a_1) \\ &= \sum_{o_1} p(o_1|a_0) \max_{a_1} \sum_{o_2} p(o_2|o_1; a_1) \max_{a_2} Q(h_2, a_2) \\ &= \sum_{o_1} p(o_1|a_0) \max_{a_1} \sum_{o_2} p(o_2|o_1; a_1) \max_{a_2} \sum_{o_3} \dots (\\ &\quad \dots \sum_{o_L} p(o_L|o_{1:(L-1)}; a_{1:(L-1)}) \max_{a_L} Q(h_L, a_L) \Big) \\ &= \sum_{o_1} \max_{a_1} \sum_{o_2} \max_{a_2} \sum_{o_3} \dots (\\ &\quad \dots \sum_{o_L} \max_k p(o_{1:L}|\alpha^{(k)}; a_{0:(L-1)}) p_k \Big). \end{aligned} \quad (\text{A2})$$

Therefore, taking $\max_{a_0} Q(h_0, a_0)$ leads to the optimal success probability $p_*^{(L)}$ as defined in 8 and optimized over $\{a_L\}$.

In Fig. 9 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 $\Delta(t)$ for the UCB strategy can lead to policies whose learning curves for the case $L = 2$ exhibit different results. As explained in ??, the upper bound in the probability of overestimate the state-action value can be bounded by Hoeffdring's inequality. This probability can be forced to depend on the experiment instance. In figure 10 we show the performance of three

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

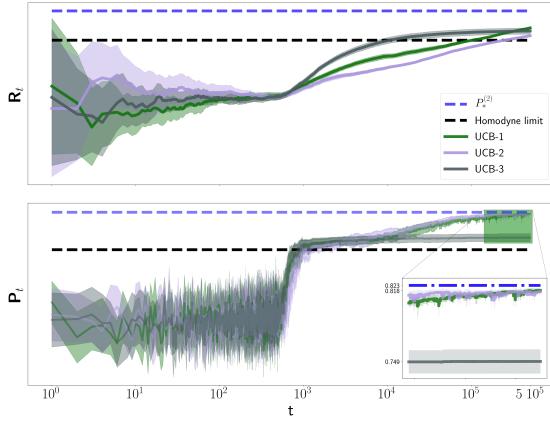


FIG. 10. We compare two different variants of UCB showing that the exploration-exploitation trade-off is an intrinsic feature of our problem.

different choices of $\underline{\Delta}(t)$ on our receiver, for the same receiver considered in IV B. Firstly we consider UCB-1, the *standard* choice of $\underline{\Delta} = t^{-4}$, which is proven to have asymptotically cumulative regret - in bandit problems - $\mathcal{L}_t = O(\log(t))$, but not with optimal convergence rate. Secondly, we consider UCB-2, with a choice of $\underline{\Delta}(t)$ proved to be asymptotically optimal in bandit problems [?]. Lastly, an heuristic and instance dependent variation of $\underline{\Delta}(t)$, 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).

-
- [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. Zwolinski, 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. Gu, and G. Adesso, EPJ Quantum Technol. **2** (2015), 10.1140/epjqt/s40507-015-0030-4, arXiv:1410.8700.
- [21] M. Gu and W. Kotlowski, New J. Phys. **12**, 123032 (2010), arXiv:1004.2468.
- [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] T. L. Lai and H. Robbins, *Adv. Appl. Math.* **6**, 4 (1985).
- [70] R. Agrawal, *Adv. Appl. Probab.* **27**, 1054 (1995).
- [71] W. R. Thompson, *Biometrika* **25**, 285 (1933).
- [72] W. R. Thompson, *Am. J. Math.* **57**, 450 (1935).
- [73] S. L. Scott, *Appl. Stoch. Model. Bus. Ind.* **26**, 639 (2010).