New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG IOP Institute of Physics

PAPER • OPEN ACCESS

Photonic architecture for reinforcement learning

To cite this article: Fulvio Flamini *et al* 2020 *New J. Phys.* **22** 045002

New Journal of Physics
The open access journal at the forefront of physics

Deutsche Physikalische Gesellschaft DPG
IOP Institute of Physics

Published in partnership with: Deutsche Physikalische Gesellschaft and the Institute of Physics

OPEN ACCESS

RECEIVED
7 November 2019

REVISED
13 February 2020

ACCEPTED FOR PUBLICATION
19 February 2020

PUBLISHED
2 April 2020

PAPER

Photonic architecture for reinforcement learning

Fulvio Flamini[1], **Arne Hamann**, **Sofiène Jerbi**, **Lea M Trenkwalder**, **Hendrik Poulsen Nautrup** and **Hans J Briegel**

Institut für Theoretische Physik, Universität Innsbruck, Technikerstraße 25, A-6020 Innsbruck, Austria
[1] Author to whom any correspondence should be addressed

E-mail: fulvio.flamini@uibk.ac.at

Keywords: machine learning, reinforcement learning, quantum photonics, integrated photonic circuits

Supplementary material for this article is available online

Abstract

The last decade has seen an unprecedented growth in artificial intelligence and photonic technologies, both of which drive the limits of modern-day computing devices. In line with these recent developments, this work brings together the state of the art of both fields within the framework of reinforcement learning. We present the blueprint for a photonic implementation of an active learning machine incorporating contemporary algorithms such as SARSA, *Q*-learning, and projective simulation. We numerically investigate its performance within typical reinforcement learning environments, showing that realistic levels of experimental noise can be tolerated or even be beneficial for the learning process. Remarkably, the architecture itself enables mechanisms of abstraction and generalization, two features which are often considered key ingredients for artificial intelligence. The proposed architecture, based on single-photon evolution on a mesh of tunable beamsplitters, is simple, scalable, and a first integration in quantum optical experiments appears to be within the reach of near-term technology.

1. Introduction

Modern computing devices are rapidly evolving from handy resources to autonomous machines [1]. On the brink of this new technological revolution [2], reinforcement learning (RL) has emerged as a powerful and flexible tool to enable problem solving at an unprecedented scale, both in computer science [3–63-6] and in physics research [7–13]. This breakthrough development was in part spurred by the technological achievements of the last decades, which unlocked vast amounts of data and computational power. One of the key ingredients for this advancement was the ultra-large-scale integration [14], which led to the massive capabilities of current portable devices. Meanwhile, in the wake of this technological progress, neuromorphic engineering [15] was developed to mimic neuro-biological systems on application-specific integrated circuits (ASIC) [16]. Their improved performance is rooted in the parallelized operation and in the absence of a clear separation between memory and processing unit, which eliminates off-circuit data transfers. Furthermore, new materials and ASICs are being reported to boost neuromorphic applications [17]. Among them, photonic devices represent a promising technological platform due to their fast switching time, high bandwidth and low crosstalks [18]. For neural networks, for instance, first proof-of-principle demonstrations on optical platforms have already been studied [19, 20] and experimentally tested [21, 22].

Inspired by the outstanding success of both RL and ASICs, here we present a novel photonic architecture for the implementation of active learning agents. More specifically, we consider an RL approach to artificial intelligence [23], where an autonomous agent learns through interactions with an environment. Within this framework, the proposed architecture can operate using any of three learning models: SARSA [24], *Q*-learning [25] and projective simulation (PS) [26]. The main contribution of this Article is twofold. (i) First, we describe a photonic architecture that enables RL algorithms to act directly within optical applications. To this purpose, we focus on linear-optical circuits for their intuitive description, well-developed fabrication techniques and

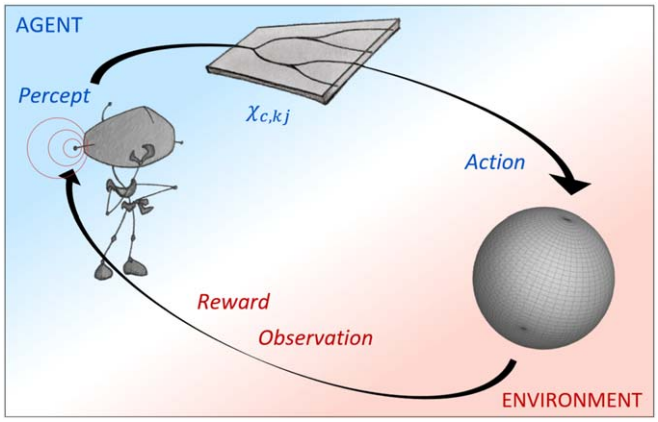

Figure 1. Reinforcement learning in a photonic circuit. In RL, an agent learns by interacting with its environment. Each new observation is internally processed until an action is chosen and performed on the environment. The processing unit, characterized by χ-values, adapts the agent's behavior according to a specific update rule in order to maximize the expected, future reward within a given environment. This unit can be implemented on an integrated photonic circuit.

promising features as compared to electronic processors [27–30]. For instance, nanosecond-scale routing and reconfigurability have already been demonstrated [31–33], while encoding information in photons enables decision-making at the speed of light, only limited by the generation and detection rates. Moreover, the use of phase-change materials for in-memory information processing [34] promises to enhance the energy efficiency, since their properties can be modified without continuous external intervention [35, 36]. Importantly, since the architecture uses single photons, decision-making is fueled by genuine quantum randomness. This feature marks a fundamental departure from pseudorandom number generation in conventional devices. (ii) The second contribution is the development of a specific variant of PS based on binary decision trees (tree-PS, or t-PS for short), which is closely connected to the standard PS and suitable for the implementation on a photonic circuit. Furthermore, we discuss how this variant enables key features of artificial intelligence, namely abstraction and generalization [37, 38].

The Article is structured as follows. In section 2 we summarize the theoretical framework of RL, exemplified by three common approaches: SARSA, *Q*-learning, and PS. In section 3 we describe the blueprint for a fully integrated, photonic RL agent. We then numerically investigate its performance within two standard RL tasks and under realistic experimental imperfections in section 4. Finally, in section 5 we discuss promising features of this architecture within the context of t-PS.

2. Reinforcement learning

In this section, we briefly introduce the RL framework, which is the focus of this work. Within RL, the agent learns through a cyclic interaction with the environment (figure 1). The agent starts with no prior knowledge and randomly probes the environment by performing actions. The environment, in turn, responds to the actions by changing its state, which is observed by the agent through perceptual input, and by providing a reward that quantifies how well the agent is performing. The goal of the agent is then to maximize its long-term expected reward [39]. In the following, we will first describe two standard RL algorithms, SARSA [24] and *Q*-learning [25], before introducing the more recent PS [26].

2.1. SARSA and *Q*-learning

As for all RL algorithms, SARSA (State-Action-Reward-State-Action) and *Q*-learning aim at adjusting the agent's behavior until it performs optimally, in the sense we discuss in the following. The agent's behavior is defined by the policy $\pi_{a|s}$, which governs the choice of an action $a \in A$ given a state $s \in S$. The evolution of the environment under the agent's action can be described by a conditional probability distribution over all state-action-state transitions. Each transition that was taken has an associated reward λ. For a given policy $\pi_{a|s}$, the value of each state is defined by the expected future reward $V_s^{\pi} = \mathbb{E}[\sum_{t=0}^{T} \gamma^t \lambda_t]$, where $\mathbb{E}[x]$ represents the expected value of the random variable x, whose underlying probability distribution depends on the policy and the state-action-state transitions. Here, λ_t is the reward received from the environment at time t, while the so-called discount factor $\gamma \in [0, 1]$ sets the relative importance of immediate rewards over delayed rewards, up to a temporal horizon T. The goal of the agent is to learn the optimal policy $\pi^*_{a|s}$ that maximizes the value V_s^{π} for all states s. The expected future reward is estimated and iteratively updated through the experience gained from its

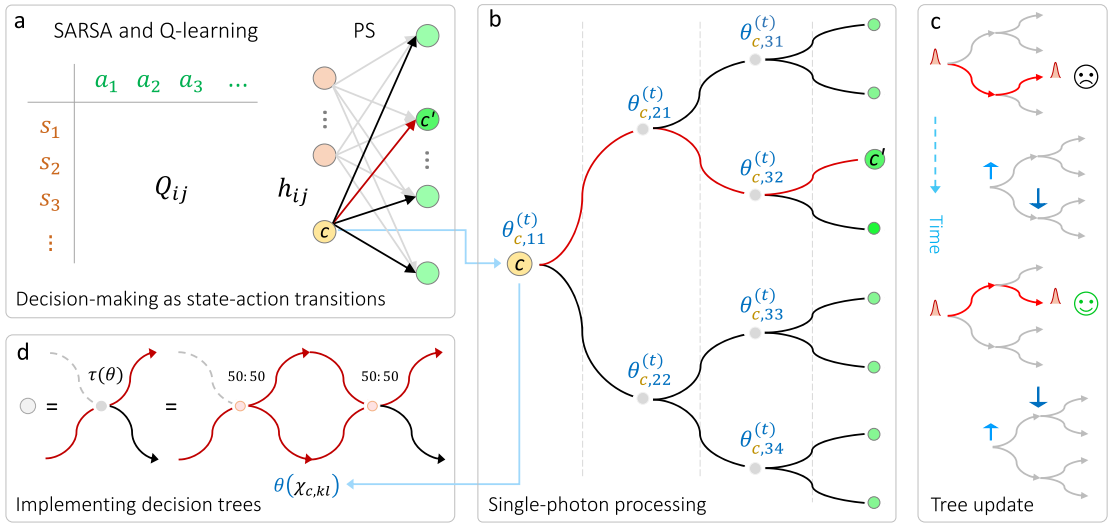

Figure 2. Photonic implementation of reinforcement learning. (a) Learning and decision-making are decomposed into sequences of state-action (SARSA and Q-learning) or clip-to-clip (PS) transitions. Every time a clip (c) is hit, a new clip (c') is chosen according to the model and the current policy. For a two-layer ECM, clip-to-clip transitions are equivalent to state-action pairs. Learning develops by updating suitable quantities: Q-values (SARSA and Q-learning), h-values (PS) and χ-values for their photonic implementation. (b) Arbitrary probabilistic transitions can be implemented on a photonic platform with a cascade of beamsplitters, whose transmissivities τ reproduce the distribution given by the current policy. (c) Beamsplitters transmissivities are adjusted to increase (decrease) the probability of reaching (un)favorable actions. In the panel, the blue arrows describe in what direction the update increases the transition probability. (d) Tunable beamsplitters at each node (k, l) can be implemented using Mach–Zehnder interferometers with tunable phase shifts $\theta_{c,kl}$ [40]. Phases are adjusted according to a quantity $\chi_{c,kl}$, which is updated during the learning process.

interactions with the environment. Instead of the value V_s^{π}, this estimate is more conveniently described by the Q-value, which quantifies the quality of a state-action pair at a given time (figure 2(a)). For both SARSA and Q-learning, this quantity is updated at each step according to

$$Q_{s,a}^{(t+1)} = (1 - \alpha)\, Q_{s,a}^{(t)} + \alpha\, \tilde{Q}_{s',a'}^{(t)}, \tag{1}$$

where $\tilde{Q}_{s',a'}^{(t)} = \lambda_{s',a'} + \gamma\, f(Q_{s',a'}^{(t)})$ is the new estimate due to taking action a' in the state s' observed after s, f is a suitable function that depends on the algorithm and the learning rate α determines to what extent this estimate overrides the old value. Given N actions and as many Q-values for state s, the expected future reward can be estimated as $V_s = \sum_{j=1}^{N} \pi_{j|s} Q_{s,j}$.

In both algorithms, decision-making is usually done by sampling actions according to a probability distribution that depends on the Q-values. In the context of RL, the softmax function is a convenient choice

$$\pi_{a|s}(Q_{s,a}) = \frac{e^{\beta Q_{s,a}}}{\sum_{j=1}^{N} e^{\beta Q_{s,a_j}}}, \tag{2}$$

where the parameter β governs the drive for exploration within the agent. The difference between Q-learning and SARSA lies in the choice of the function f. In SARSA, f updates the value of the current state-action pair $Q_{s,a}$ with the estimate for the following state-action pair $Q_{s',a'}$, i.e. f is the identity function. In state s', the action a' is chosen according to the agent's policy. Thus, SARSA is called an on-policy algorithm. Q-learning, on the other hand, is an off-policy algorithm because, given the state s', f selects the action a' with the maximal value $Q_{s',a'}$, i.e. $f = \max_{a' \in A}$, so that the update is independent of the next action chosen according to the agent's policy.

2.2. Projective simulation

PS is a recent, physically-motivated RL model [26], which has already found several applications ranging from robotics [41] and quantum error correction [7] to the study of collective behavior [42] and automated experiment design [43]. Decision-making in PS occurs in a network of clips that constitutes the agent's episodic and compositional memory (ECM) (figure 2(a)). Each clip represents a remembered percept, a remembered action or a more complex combination thereof. The ECM can accommodate a multilayer structure, where intermediate layers represent abstract clips and connections. Decision-making is carried out by a random walk through the ECM, starting at a percept clip and ending at an action clip which triggers the corresponding action. The random walk is guided by transition probabilities between pairs of clips (c_i, c_j), connected by edges carrying weights h_{c_i,c_j}, by considering probabilities proportional to h_{c_i,c_j} or by using the softmax function $\pi_{c_j|c_i}(h_{c_i,c_j})$ as in equation (2). Learning occurs by updating the clip network in the agent's memory, i.e. by changing its topology or the edge weights h_{c_i,c_j}. In the latter case, the update rule at time t has the form

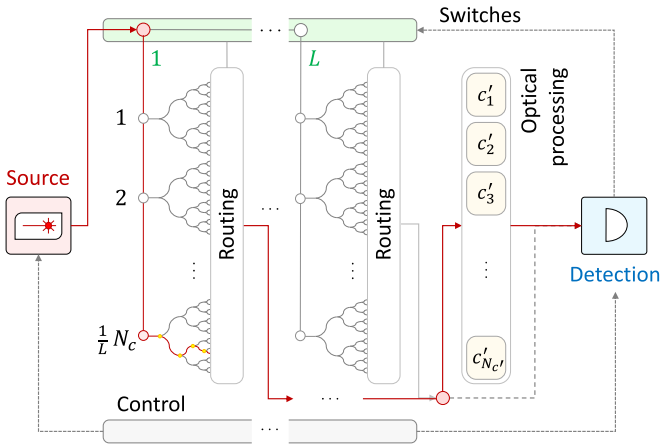

Figure 3. Photonic architecture for learning and decision-making. Single photons can be purposely generated to implement decision-making on a photonic platform, or they can be taken from the optical application wherein the architecture operates. In the latter case, each of N_c input clips leads to $N_{c'}$ output clips corresponding, for instance, to different quantum optical subroutines, as well as to nodes in a network. For each clip-to-clip or state-action transition, a single photon is routed to 1 of the L levels, where it passes through one of the N_c/L decision trees (figure 2). For PS, in this work we only consider acyclic ECMs. At the output of every decision tree, and before the detection stage, the single photons can be routed coherently to different optical subroutines (which act as an optical environment). The system can even be self-stabilized [40, 46, 47].

$$h_{c_i,c_j}^{(t+1)} = h_{c_i,c_j}^{(t)} - \gamma(h_{c_i,c_j}^{(t)} - 1) + g_{c_i,c_j}\lambda, \tag{3}$$

where $\gamma \in [0,1)$ is a damping parameter, λ is the reward and $g_{c_i,c_j} \in [0,\ 1]$ is the so-called edge-glow value or g-value. Here, γ and g_{c_i,c_j} implement mechanisms that take into account forgetting and delayed rewards, respectively. More specifically, the damping parameter γ is essential for environments that change over time, effectively damping h-values at each time step. The edge-glow values serve to backpropagate discounted rewards to earlier sequences of actions. The g-values are updated at each time step: whenever an edge (c_i, c_j) is traversed g_{c_i,c_j} is set to 1, and from then on its value is discounted as $g_{c_i,c_j}^{(t+1)} = (1 - \eta)\, g_{c_i,c_j}^{(t)}$ where $\eta \in [0, 1]$ is the glow parameter.

Consequently, g-values are rescaled according to $g_{c_i,c_j} = (1 - \eta)^{\delta t_{c_i,c_j}}$, where $\delta t_{c_i,c_j}$ is the number of steps between the round when (c_i, c_j) is traversed from the round when a reward is issued. Intuitively, values of η close to 1 reward sequences of actions only in the immediate past, while values close to 0 are used to reward longer sequences. The glow parameter is relevant in environments with delayed rewards such as the GridWorld [23] discussed in section 4. For a more detailed description of PS we refer the reader to [38, 44, 45].

3. Photonic reinforcement learning

In order to implement RL on a photonic platform we need to be able to satisfy two requirements: (i) implement arbitrary probabilistic transitions between clips and (ii) update the corresponding probability distributions in a controlled and effective way. A practical platform has to satisfy further criteria that are crucial for any implementation, such as scalability, ease of fabrication and miniaturizability. In this section, we will describe a linear optical architecture that is tailored to the task at hand, i.e. designing integrated photonic hardware for RL, in the spirit of neuromorphic engineering [18]. While this work focuses on single-photon processing, which makes the approach compatible and easy to integrate with quantum technologies, the scheme can also be operated with laser light, which has broader applications and higher energy efficiency. Besides its simplicity, we choose this approach for a number of practical reasons. (i) In RL, for every state s of the environment (i.e. for every tree in figure 3) the agent must sample only one action a (i.e. light must exit from only one output mode). Using single photons ensures that a measurement can immediately identify which optical mode (i.e. action) a click corresponds to. Coherent states lose this advantage and likely require additional mechanisms to reduce intensity measurements to a single action, e.g. choosing the one with the maximum intensity. (ii) Losing a photon is not critical when there is only one (e.g. if no detector clicks, the process is repeated), but ambiguities arise if the choice depends on the number of photons ad this is not known beforehand. (iii) While single-photon source and detection imply a larger power consumption, these stages can already be present in the apparatus that hosts the architecture, as part of a larger application.

3.1. Decision trees as linear optical circuits

Using a bottom-up approach[2], we focus on the implementation of state-action (SARSA and Q-learning) or clip-to-clip (PS) transitions, as shown in figure 2(a). For PS, each clip-to-clip transition is a building block for the random walk in the agent's memory. For brevity, we will only consider clip-to-clip transitions (c, c'), which are equivalent to state-action pairs (s, a) for a two-layer ECM. Each transition is governed by the probability distribution of detecting a single photon over the output modes. The architecture we present consists of a cascade of reconfigurable beamsplitters arranged in a tree structure (figure 2(b)), which maps a single input mode (associated with a state) to N output modes (corresponding to as many clips). Such an association can be initialized randomly or according to prior knowledge about the environment. Fully-reconfigurable linear-optical interferometers like this one allow to engineer an arbitrary superposition over the optical modes and, given the corresponding probability distribution, it is possible to determine a set of phases that reproduces it exactly (see the supplemental material available online at stacks.iop.org/NJP/22/045002/mmedia). In the next section, we also provide further considerations on various layouts that can be adopted.

To employ this architecture for RL, we consider the following operational scenario: the current policy is stored electronically [28, 30] in the phase shifters that define the single-photon evolution in the circuit and, consequently, the probabilistic decision-making. Each phase-shifter $\theta_{c,kl}$ at node (k, l) is set to implement the transition probabilities for the corresponding clip-to-clip connections. Decision-making (figure 2(a)) is hence realized as a single-photon evolution in a mesh of tunable beamsplitters (figure 2(b)), where the transition to the next state is made by detecting[3] single photons at the output modes. Notably, the probabilistic nature of this process is inherent to its quantum description, and, as such, it is qualitatively superior to standard pseudo-random number generation. Overall, this approach satisfies the requisite for arbitrary probabilistic transitions (i) described at the beginning of this section. Furthermore, it provides a solution that is scalable (one only needs to store the phases that implement a given transition) and that can be fully integrated on a miniaturized photonic chip [30]. Importantly, sensors could be integrated on an optical chip, gyroscopes and magnetometers being first examples in this direction [28].

Concerning the second requirement (ii), to learn an optimal policy we want the agent to autonomously adjust the phases θ according to a suitable update rule. To this end, we first consider the path $\Gamma_{c,c'}$ that connects clips (c, c') and express the phases $\theta_{c,kl}$ in the transition probability

$$p_{c'|c} = \prod_{(k,l)\in\Gamma_{c,c'}} \sin^2\theta_{c,kl} \tag{4}$$

as a function of a quantity χ that is updated during the learning process, namely $\theta(\chi) = \theta_0 + \theta_\chi$. Here, $\theta_0 = \frac{\pi}{4}$ corresponds to the configuration where all transitions are equally probable, while θ_χ spans the whole range of transition probabilities, namely $\theta_\chi \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$. Suitable candidates for θ_χ are the sigmoid functions [50], which are monotonically increasing in a bounded interval and have domain over all real numbers. We then use the function $\theta_\chi = \tanh\chi$, so that

$$\theta(\chi) = \frac{\pi}{4}(1 + \tanh\chi), \tag{5}$$

where the quantity χ is updated according to a suitable update rule within the framework of RL. For SARSA (S) and Q-learning (QL), we update χ according to the rules

$$\begin{aligned}\chi^{S}_{c,kl} &\leftarrow (1-\alpha)\,\chi^{S}_{c,kl} + \alpha(\lambda_c + \gamma R_{c'})\\ \chi^{QL}_{c,kl} &\leftarrow (1-\alpha)\,\chi^{QL}_{c,kl} + \alpha(\lambda_c + \gamma R_{c'} M_{c'}),\end{aligned} \tag{6}$$

where $M_{c'} = \max_{c''}\left(\tanh\sum_{(k,l)\in\Gamma_{c',c''}}\frac{|\chi_{c',kl}|}{n}\right)$, $n = \lceil \log_2 N \rceil$ being the depth of the circuit, and

$$R_c \leftarrow (1-\alpha)\,R_c + \alpha(\lambda_c + \gamma R_{c'}). \tag{7}$$

In the notation used in section 2.1 for SARSA and Q-learning, subscripts in R_c and $M_{c'}$ refer to states. Comparing the original Q-value update rule in equation (1) with the update rule in equation (6), we emphasize that equation (6) does not simply reproduce equation (1) using χ. The reason is that Q- and χ-values provide different information, the former quantifying the quality of a clip-to-clip connection, the latter defining the splitting ratio at each beamsplitter. Indeed, though related once the agent has properly learned the policy, the two quantities are not directly linked during the learning process. For instance, when one clip-to-clip connection

[2] To fulfill all the above desiderata, a top-down approach could be to adopt well-established meshes of optical elements for programmable multifunctional nanophotonics hardware [32]. However, their versatility comes at the cost of higher computational resources (should one iteratively adjust their settings according to externally-processed unitary decompositions [31]) or of a less intuitive dependency of the output on the internal components [48, 49].

[3] Non-ideal detection efficiency can be counteracted with a control feedback, by sending again a photon to the same decision tree if, in the previous time bin, no photon was collected.

(c, c') is favorable (large Q-value) the policy π_c is peaked (i.e. χ-values far from zero), but when multiple (c, c') pairs are favorable (large Q-values) the policy π_c is less peaked (χ-values closer to zero). Therefore, a feature we demand is to keep track of the overall quality of each state, from which the χ-values will reproduce the relative quality of each (c, c') connection. We fulfill this task in equation (7), introducing a new parameter (in addition to the $N-1$ phases) that updates the agent's confidence in the quality of clip c. Also, peakedness of each policy can be quantified by the average deviation from 0 (corresponding to a flat distribution) of the χs in each path, as done in $M_{c'}$ in equation (6).

Besides SARSA and Q-learning, we can choose to operate in the framework of PS. In this case, we evolve χ according to the rule

$$\chi^{\mathrm{PS}}_{c,kl} \leftarrow \gamma\, \chi^{\mathrm{PS}}_{c,kl} + g_{c,kl}\, \lambda_c \tag{8}$$

which is equivalent to the update rule for $h_{c,c'}$ in equation (3), considering that $h_{c,c'}$ ($\chi_{c,kl}$) is initialized to 1 (0). Notably, the choice $\theta_\chi = \tanh \chi$ in equation (5) establishes a formal connection between the proposed architecture and a specific variant of PS, which we call tree-PS (t-PS). This connection is derived in the supplemental material. In t-PS, every clip-to-clip transition is implemented as a binary decision tree between the input and the output clips. In the supplemental material we prove that t-PS can reproduce the operation of the two-layer PS, which has been discussed extensively in the literature. While the two models appear to have the same representational power, t-PS provides an additional structure that can be exploited to enhance the learning process, as we describe below in section 5.1.

3.2. Photonic architecture for the agent's memory

The architecture described in figure 2(b), which represents the building block for decision-making, can take advantage of an efficient design enabled by its fractal geometry [51]. In this section, we will outline three approaches to implement learning and decision making starting from such a building block. First, we can adopt a simple strategy where the circuit consists of a single decision tree: once a photon is detected (thus selecting a clip in the next layer), all phase-shifters are adjusted to implement the next transition and another photon is injected into the same circuit. Similarly, we can devise a loop-based implementation where photons are redirected back to the input while the circuit is reconfigured. Though appealing, this approach is more challenging since it requires non-linearities to detect the presence of a photon in the output modes [52]. Finally, we can conceive a more sophisticated scheme where all building blocks are arranged in a planar structure (figure 3) that represents the memory of the agent (figure 2(a)). In the latter configuration, photons are routed through a bus waveguide and optical switches [40, 53] to one level (out of L), where a clip-to-clip transition is performed in a decision tree. The multi-level architecture is meaningful only for PS, where it represents the L-layer structure of an acyclic ECM, while for SARSA and Q-learning it is a convenient geometry to make the integrated circuit more compact. Fast and efficient routing [33, 54], controlled by a feedback system that also monitors photon losses, guides single photons to the appropriate building block. Photons exit the tree in superposition over $N_{c'}$ waveguides and are routed to either additional photonic processing (which can be seen as a quantum optical environment) or directly to the detection stage. To find out which clip (i.e. output waveguide) was selected, a set of $N_{c'}$ detectors can be used to determine it unambiguously. Since single photons are employed during the decision-making, a combination of delay lines and switches can be arranged to reduce the number of detectors.

An interesting feature of this approach is that it can take advantage of phase-change materials (PCM) [35, 36] to realize the phase-shifters, whose physical properties can be modified in a reversible and controlled way with a single write operation [34]. In fact, only the phases corresponding to traversed paths need to be updated, while the others remain fixed without extra power consumption. Hence, the number of updates scales only logarithmically with the number of output clips. In the supplemental material, we discuss how both computational complexity and energy consumption are even comparable to an electronic ASIC that exploits high locality and specialized data structure. Notably, using the circuit for self-optimization in optical interferometers eliminates the need for a separate generation and detection, since photons can be part of the embedding quantum optical application. In addition, decision-making after learning consumes practically no power since phase-shifters do not need to be adjusted anymore.

4. Testing the architecture

In this section, we employ the proposed architecture in a standard testbed for RL, the GridWorld environment [39]. This task is of broad relevance since any stationary fully-observable environment can be reformulated in this frame [39], notable examples being Atari games [3] and Super Mario Bros. [55]. Henceforth, we will focus on (two-layer) PS, due to its simpler update rule (equation (8)) and to investigate the potential of t-PS. Indeed, GridWorld has been already investigated in the context of PS [44], a relevant example being the design of optical

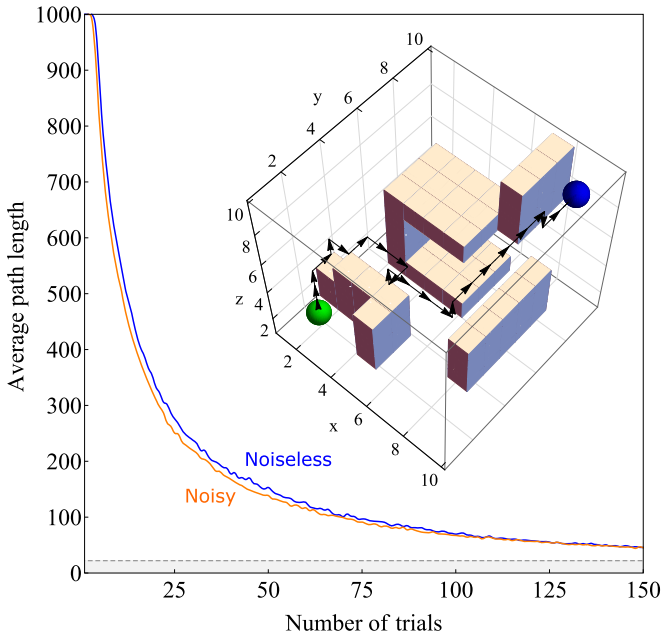

Figure 4. Simulating the photonic architecture in GridWorld. Average path length required by a PS agent to reach the reward in a $10 \times 10 \times 10$ GridWorld, shown in the inset, as a function of the number of trials. The same analysis is carried out for implementations with ideal (blue) and noisy (orange) phase-shifters. See the supplemental material for details on how experimental imperfections were modeled. Curves are averaged over 10^4 agents ($\lambda = 8, \eta = 0.11$ and damping $\gamma = 0.999$ applied every 100 steps), while the gray band excludes lengths below the minimum (19 steps). Inset: Path taken by a single, noisy, random agent after 150 trials. The green sphere ($\vec{p}_A = (3, 1, 4)$) and the blue sphere ($\vec{p}_R = (9, 9, 9)$) represent the agent and the reward, respectively, while blocks represent untraversable 3D walls.

experiments, which was shown to be representable as a generalized GridWorld [43]. Furthermore, note that for both SARSA and Q-learning we numerically observed a performance very similar to PS.

In the simplest formulation of the problem, the goal for the agent is to maximize its long-term expected reward while navigating an environment structured as a planar grid-like maze. The agent starts from a fixed location $\vec{p}_A = (x_A, y_A)$ and is challenged to learn the shortest path that leads to a reward at location $\vec{p}_R = (x_R, y_R)$. Available to the agent is a set of actions $(x^{\pm}, y^{\pm})$, where $x^{\pm}$ corresponds to a movement in the positive/negative x-direction. The learning process is divided in a sequence of episodes, or trials τ, where the agent interacts with the environment until a predetermined condition is met. In our analyses, the agent is reset if the number of interactions in one episode either exceeds 10^3 or a reward is obtained. To account for delayed rewards, the edge-glow mechanism (see section 2.2) rescales the reward λ, assigned to a traversed transition (c_i, c_j), by a quantity that decreases exponentially with the number of steps that pass until a reward is received [39, 44].

The above formulation can be extended to more complex scenarios, which include higher-dimensional mazes with walls, sophisticated moves and/or penalties. For our investigation we employed a 3D GridWorld with walls: whenever the agent tries to move onto the border of the grid or onto a wall, a time step is counted but no movement occurs. We chose a 3D maze, rather than a 2D or a 4D grid, to investigate more complex configurations that could still be visually inspected. As an example (see inset in figure 4), we considered a $10 \times 10 \times 10$ GridWorld where the agent starts at position $\vec{p}_A = (3, 1, 4)$ and a reward is hidden at position $\vec{p}_R = (9, 9, 9)$. Figure 4 shows the average learning curve numerically simulated for a photonic agent navigating this maze. We observe that the average path length rapidly decreases with the number of trials, from $\sim 10^3$ (where the agent behaves like a random walker) to values close to the minimum path length (19 in this case).

The same numerical analysis was carried out simulating a non-ideal implementation of photonic PS, to test to what extent experimental imperfections are expected to spoil the process. To this end, each time phases were adjusted in the simulated device, Gaussian noise was added on top of the ideal value (a more detailed description on how imperfections were modeled is reported in the supplemental material). Remarkably, we find that a realistic amount of noise can even aid the learning process, a feature that can be ascribed to an enhanced tendency of the agent to explore new paths. In the supplemental material, we also expand on this aspect, which is reminiscent of the phenomenon of stochastic resonance[4], providing a visual intuition in support of this

[4] Stochastic resonance occurs when an increase in the level of random fluctuations leads to an increase in the level of order or performance. Since its introduction [56], this phenomenon has been investigated in several physical systems, including biology and neuroscience [57].

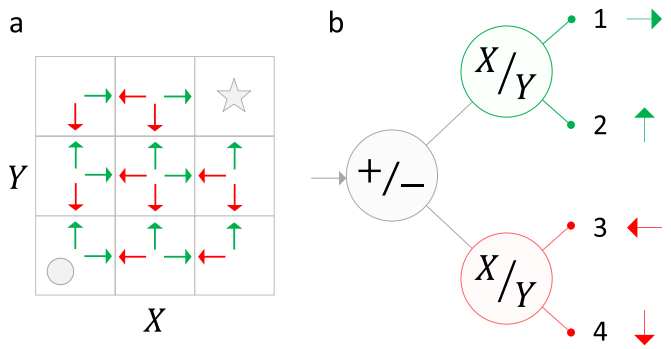

Figure 5. Generalization in GridWorld. t-PS can exploit symmetries in a task environment to boost the learning process. In a 2D GridWorld (a) where the agent (circle) and the reward (star) are initially located at opposite corners, we can associate with modes (1, 2, 3, 4) the actions (→, ↑, ←, ↓), so that the agent can learn to focus on the first two by adjusting just one parameter (b).

interpretation. Eventually, the fact that realistic levels of noise can enhance the agent's learning process makes the present approach even more appealing for a concrete implementation. Indeed, not only the architecture exhibits a natural resilience to noise, but also this very resilience relaxes the (often challenging) technological requirements for isolation and stability.

5. t-PS with generalization and abstraction

While the two-layer and the tree-based implementations of PS have the same representational power (see the supplemental material), t-PS provides an additional structure that can be exploited to boost the learning process. As we will see, this feature allows an agent to exhibit simple forms of abstraction and generalization, which play a central role in artificial intelligence [37]. Abstraction is the ability of an agent to filter out less relevant details, a process that involves a modification in the representation of the object. Generalization corresponds to the ability to identify similarities between objects, without necessarily affecting their representation. In this section, we will describe how an agent can take advantage of these features by suitably ordering the clips over the output modes according to some measure of relevance, such as the reward.

5.1. Generalization and abstraction

To introduce the notions of generalization and abstraction in the present architecture, let us start by considering the simplest case of a 2D GridWorld in the XY plane without walls. Given the tree structure of t-PS, we can expect there to be a beneficial arrangement of action clips over the outputs. Nodes in t-PS can represent meaningful sub-decisions towards a final decision made at the leaf nodes. Since nodes closer to the root are updated more regularly, sub-decisions can, in principle, be learned before the final policy is obtained. Of course, initially, nodes are not necessarily ordered in a way that has a meaningful interpretation. However, the agent can sort them during the learning process such that intermediate nodes obtain meaning which, in turn, guides the agent's decision-making.

Motivated by the above considerations, we propose a simple mechanism, which we call *defragmentation*, that is specifically designed to address this issue, though its benefits are not limited to this scenario. The name defragmentation is inspired by the usual process that occurs in hard-disks, which improves performance by reallocating fragments of memory according to dependencies and usage. The mechanism consists of (1) keeping track of the cumulative reward assigned to each action and (2) sorting actions over the output modes according to their respective cumulative reward. More sophisticated rules can also be designed for step (1), perhaps tailored to capture correlations in time or more intricate patterns between actions. From a practical perspective, step (2) only requires to compute the new phases that produce the reordered probability distribution (see the supplemental material). In any case, whenever there are two or more rewarded actions, this mechanism favors the separation between good and unfavorable actions. It is precisely in its capability of grouping together actions of comparable relevance, e.g. similar collected rewards in the present context, that the agent expresses an elementary form of generalization [38]. For instance, in a 2D GridWorld actions can be conveniently organized according to a hierarchy of criteria (figure 5), e.g. move 'forward' or 'backwards' and move 'along X' or 'along Y', resulting in composite actions such as 'up' ('forward' and 'Y') or 'left' ('backwards' and 'X'). Numerical analyses involving defragmentation on both 2D and 3D GridWorld show that the agent does autonomously discover structures analogous to the one in figure 5, suggesting that this generalization feature is beneficial and informative, and that it can be used in more complex scenarios.

Naturally, defragmentation as a way of knowledge exploitation consumes time that has to be balanced with that reserved for exploration. Nevertheless, in the usual compromise between exploration and exploitation [39], the longer the agent explores the environment to assess the quality of an action, the more its generalization process will be reliable and successful. At a certain time, once a stronger representation is built in its memory, the agent could even perform a sort of abstraction by cutting out the least relevant actions, so as to focus only on those that are deemed more favorable. In RL tasks with large-scale action spaces, this process could even be iterated to progressively reduce the search space for good actions. Indeed, the photonic architecture enables this mechanism to be straightforwardly implemented, by simply setting specific transition probabilities to 0 or 1, which isolates all the subsequent branches of optical components. This feature could, in turn, entail a reduction in computational resources and, possibly, in learning time.

5.2. Exploiting the tree-like structure

To provide quantitative evidence for the above considerations, we numerically applied defragmentation to another standard problem in RL, the multi-armed bandit [39]. In its general formulation, an agent is presented with N bandits (for instance, slot machines) characterized by a probabilistic reward function and, at each time step, the agent is allowed to pull the arm of one of the bandits (which issues a reward drawn from the corresponding distribution). Effectively, this gives an environment with one state and N possible actions. We consider a variant of the problem with additional structure in its action space, referred to in the literature as *combinatorial* multi-armed bandit [58]. In this task, bandits (i.e. actions) are grouped in sub-categories according to a set of features. In the example described above, these features could be the casino, city, country, etc the slot machine is situated in. This structure is provided to the agent at an abstract level (the dependence between features is not specified) by dividing the allowed actions into several sub-actions. As a result, the action space $A = \{1, \ ..., N\}$ factorizes to $A = A_1 \times A_2 \times ... \times A_k$, where $|A_i| = n_i$ is the number of possible choices for sub-action A_i, and $N = \prod n_i$. This kind of factorization is analogous to the decomposition of the state and action space into categories that was considered in [38], except that the structure we consider here is imposed on actions. For simplicity, let us assume that a deterministic reward r_a is associated with each action $a = (a_1,\ldots,a_k)$, but that this reward distribution depends (partially) on the structure of the action space. The agent can then exploit the factorized structure to choose the best sub-actions according to their influence on the reward. In this regard, the proposed architecture can be particularly effective since consecutive levels can separately focus on each A_k. Moreover, a mechanism to rearrange the structure (such as the defragmentation described in section 5.1) can shift the levels associated with the most relevant sub-actions closer to the root, capturing correlations between actions and facilitating learning. In the above example, the agent could learn that the choice of a city is more relevant than the choice of a particular casino in that city, because casinos in a certain city are more lucrative, and choose the city earlier in the deliberation.

We expand on the above considerations in more detail in the supplemental material with a simple example. In the following, we will focus on the performance boost induced by the defragmentation of the action space. Figure 6 shows quantitative evidence of this boost in an instance of the bandit problem where two actions are always rewarded. Analogous advantages can be found in the 3D GridWorld described in section 4, where only a subset of directions is relevant and grouping them is beneficial for the agent. In particular, these numerical results show that defragmentation allows to speed up the learning process, i.e. fewer trials are required to find an optimal policy. This situation is indeed typical in RL, where exploitation of current knowledge allows to reduce the time spent on exploration. From a practical perspective, this feature facilitates learning scenarios where interactions with the environment are costly. For these reasons, the proposed t-PS appears as a promising platform in the framework of RL, being able to support key features for artificial intelligence (in the form of a basic generalization and abstraction) while preserving a good control over its operation and performance.

6. Discussion

The development of autonomous agents capable of learning by interacting with an environment has seen a tremendous surge of interest over the past decade [3, 4, 6]. Similarly, the design of neuromorphic application-specific hardware has attracted massive attention due to its enhanced computational capabilities in terms of speed and energy efficiency [15]. In this work, we propose a blueprint for an application-specific integrated photonic architecture capable of solving problems in RL. Within this framework, the architecture easily accommodates various well-established RL algorithms such as SARSA, Q-learning, and PS. Also, its simple and scalable design warrants a near-term implementation in optical experiments and is apt for embedding in miniaturized devices compatible with quantum technologies. Indeed, all required optical components have already been experimentally demonstrated on integrated circuits [27–32].

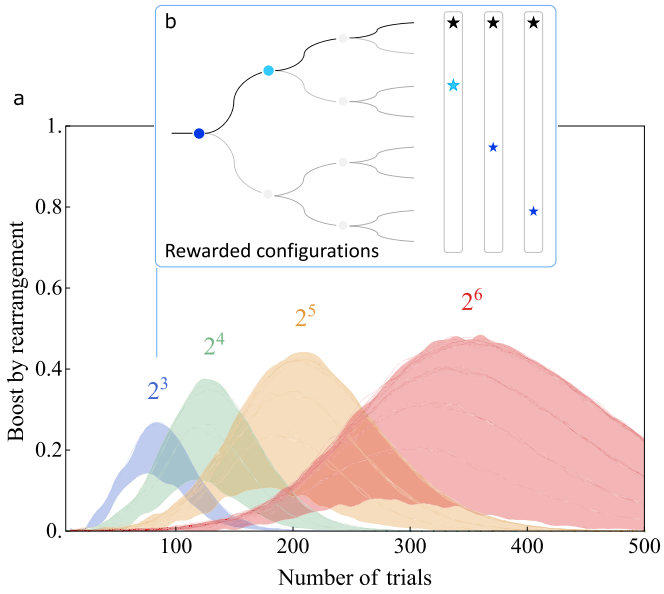

Figure 6. Boosting the learning process in t-PS. Learning can be sped up in tasks with structured action spaces like the combinatorial multi-armed bandit [58], by taking advantage of the tree-like structure. (a) Difference (*boost*) between the average reward collected with and without defragmentation of the agent's memory, i.e. a dynamical rearrangement of the actions over the output modes. The analysis is carried out for actions spaces of size 2^d, with $d = 3, ..., 6$, with only two actions rewarded (b): one fixed on the first output mode, the other one displaced progressively further over the other modes. For each d, the magnitude of the boost depends on the number of levels (from 1 to $d - 1$) where the rewarded paths differ: neighboring (faraway) modes lead to smaller (higher) boosts. For clarity, curves are interpolated connecting one point every 10. Averages are computed over 5×10^3 PS agents ($\lambda = 0.025$, $\gamma = 0.9975$).

We investigated the proposed platform both numerically and analytically, confirming the efficacy of the model also under realistic, imperfect experimental conditions. Besides its efficacy, the architecture enables a novel implementation of PS (t-PS) that is inspired by the geometry of the integrated circuit. This model does not only exhibit some key features of artificial intelligence, namely generalization and abstraction, but can also boost its learning performance via autonomous defragmentation of its memory. Indeed, both numerical and analytical results suggest that t-PS performs at least as well as the simulated standard PS model, which has already found various applications [7, 41–43]. Eventually, we envisage the experimental realization of a photonic RL agent which successfully exploits all these features within a quantum optical environment.

Funding information

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801110 and the Austrian Federal Ministry of Education, Science and Research (BMBWF). It reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains. HPN, LMT, SJ, and HJB acknowledge support from the Austrian Science Fund (FWF) through the projects DK-ALM:W1259-N27 and SFB BeyondC F71. HJB was also supported by the Ministerium für Wissenschaft, Forschung, und Kunst Baden-Württemberg (AZ:33-7533.-30-10/41/1).

ORCID iDs

Fulvio Flamini https://orcid.org/0000-0003-4999-2840
Arne Hamann https://orcid.org/0000-0002-9016-3641
Hendrik Poulsen Nautrup https://orcid.org/0000-0001-7815-7006

References

[1] Iliadis L, Maglogiannis I and Plagianakos V 2018 Artificial intelligence applications and innovations *14th IFIP WG 12.5 Int. Conf. AIAI 2018 (25-27May 2018(Rhodes, Greece: Springer Publishing Company) 1st edn*
[2] Schwab K 2017 *The Fourth Industrial Revolution* (London: Penguin)
[3] Mnih V *et al* 2015 Human-level control through deep reinforcement learning *Nature* **518** 529–33
[4] Silver D *et al* 2016 Mastering the game of Go with deep neural networks and tree search *Nature* **529** 484–9

[5] Silver D *et al* 2018 A general reinforcement learning algorithm that masters chess, shogi, and go through self-play *Science* **362** 1140–4
[6] Arulkumaran K, Cully A and Togelius J 2019 Alphastar: an evolutionary computation perspective arXiv:1902.01724v2
[7] Nautrup H P, Delfosse N, Dunjko V, Briegel H J and Friis N 2019 Optimizing quantum error correction codes with reinforcement learning *Quantum* **3** 215
[8] Sweke R, Kesselring M S, van Nieuwenburg E P L and Eisert J 2018 Reinforcement learning decoders for fault-tolerant quantum computation arXiv:1810.07207
[9] Bukov M, Day A G R, Sels D, Weinberg P, Polkovnikov A and Mehta P 2018 Reinforcement learning in different phases of quantum control *Phys. Rev.* X **8** 031086
[10] Bukov M 2018 Reinforcement learning for autonomous preparation of floquet-engineered states: inverting the quantum kapitza oscillator *Phys. Rev.* B **98** 224305
[11] Niu M Y, Boixo S, Smelyanskiy V N and Neven H 2019 Universal quantum control through deep reinforcement learning *npj Quantum Inf.* **5**
[12] Porotti R, Tamascelli D, Restelli M and Prati E 2019 Coherent transport of quantum states by deep reinforcement learning *Commun. Phys.* **2** 61
[13] Colabrese S, Gustavsson K, Celani A and Biferale L 2017 Flow navigation by smart microswimmers via reinforcement learning *Phys. Rev. Lett.* **118** 158004
[14] Meindl J D 1984 Ultra-large scale integration *IEEE Trans. Electron Devices* **31** 1555–61
[15] Thakur C S *et al* 2018 Large-scale neuromorphic spiking array processors: a quest to mimic the brain *Front. Neurosci.* **12** 891
[16] Mead C 1990 Neuromorphic electronic systems *Proc. IEEE* **78** 1629–36
[17] Islam R, Li H, Chen P-Y, Wan W, Chen H-Y, Gao B, Wu H, Yu S, Saraswat K and Wong H-S P 2019 Device and materials requirements for neuromorphic computing *J. Phys. D: Appl. Phys.* **52** 113001
[18] de Lima T F, Peng H, Tait A N, Nahmias M A, Miller H B, Shastri B J and Prucnal P R 2019 Machine learning with neuromorphic photonics *J. Light. Technol.* **37** 1515–34
[19] Steinbrecher G R, Olson J P, Englund D and Carolan J 2019 Quantum optical neural networks *npj Quantum Inf.* **5** 60
[20] Hughes T W, Minkov M, Shi Y and Fan S 2018 Training of photonic neural networks through *in situ* backpropagation and gradient measurement *Optica* **5** 864–71
[21] Shen Y *et al* 2017 Deep learning with coherent nanophotonic circuits *Nat. Photon.* **11** 441
[22] Zuo Y, Li B, Zhao Y, Jiang Y, Chen Y-C, Chen P, Jo G-B, Liu J and Du S 2019 All-optical neural network with nonlinear activation functions *Optica* **6** 1132–7
[23] Sutton R S and Barto A G 1998 Reinforcement Learning: An Introduction *IEEE Trans. Neural Networks* **16** 285–6
[24] Rummery G A and Niranjan M 1994 *On-line Q-learning using connectionist systems* (https://doi.org/10.1.1.17.2539)
[25] Watkins C J C H 1989 Learning from delayed rewards *PhD Thesis* King's College, Cambridge, UK (https://doi.org/10.1.1.17.2539)
[26] Briegel H J and De las Cuevas G 2012 Projective simulation for artificial intelligence *Sci. Rep.* **2** 400
[27] Sun C *et al* 2015 Single-chip microprocessor that communicates directly using light *Nature* **528** 534–8
[28] Komljenovic T, Davenport M, Hulme J, Liu A Y, Santis C T, Spott A, Srinivasan S, Stanton E J, Zhang C and Bowers J E 2016 Heterogeneous silicon photonic integrated circuits *J. Lightwave Technol.* **34** 20–35
[29] Flamini F, Spagnolo N and Sciarrino F 2018 Photonic quantum information processing: a review *Rep. Prog. Phys.* **82** 016001
[30] Atabaki A *et al* 2018 Integrating photonics with silicon nanoelectronics for the next generation of systems on a chip *Nature* **556** 349–54
[31] Harris N C *et al* 2018 Linear programmable nanophotonic processors *Optica* **5** 1623–31
[32] Pérez D, Gasulla I and Capmany J 2018 Programmable multifunctional integrated nanophotonics *Nanophotonics* **7** 1351–71
[33] Stabile R, Albores-Mejia A, Rohit A and Williams K A 2016 Integrated optical switch matrices for packet data networks *Microsyst. Nanoeng.* **2**
[34] Ríos C, Youngblood N, Cheng Z, Le Gallo M, Pernice W H P, Wright C D, Sebastian A and Bhaskaran H 2019 In-memory computing on a photonic platform *Sci. Adv.* **5**
[35] Wuttig M, Bhaskaran H and Taubner T 2017 Phase-change materials for non-volatile photonic applications *Nat. Photon.* **11** 465–76
[36] Miller K J, Haglund R F and Weiss S M 2018 Optical phase change materials in integrated silicon photonic devices: review *Opt. Mater. Express* **8** 2415–29
[37] Ponsen M, Taylor M E and Tuyls K 2010 Abstraction and Generalization in Reinforcement Learning: A Summary and Framework *Adaptive and Learning Agents* (vol 5924) ed M E Taylor and K Tuyls (Berlin: Springer) (https://doi.org/10.1007/978-3-642-11814-2_1)
[38] Melnikov A A, Makmal A, Dunjko V and Briegel H J 2017 Projective simulation with generalization *Sci. Rep.* **7** 14430
[39] Sutton R S 1990 Integrated architectures for learning, planning, and reacting based on approximating dynamic programming *Machine Learning: Proceedings of the Seventh International Conference (Austin, TX)* ed B Porter and R Mooney (San Francisco (CA): Morgan Kaufmann) pp 216–24
[40] Miller D A B 2013 Self-configuring universal linear optical component *Photon. Res.* **1** 1–15
[41] Hangl S, Ugur E, Szedmak S and Piater J 2016 Robotic playing for hierarchical complex skill learning *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)* pp 2799–804
[42] Ried K, Müller T and Briegel H J 2019 Modelling collective motion based on the principle of agency: general framework and the case of marching locusts *PLoS One* **14** 1–21
[43] Melnikov A A, Poulsen Nautrup H, Krenn M, Dunjko V, Tiersch M, Zeilinger A and Briegel H J 2018 Active learning machine learns to create new quantum experiments *Proc. Natl. Acad. Sci. USA* **115** 1221–6
[44] Melnikov A A, Makmal A and Briegel H J 2018 Benchmarking projective simulation in navigation problems *IEEE Access* **6** 64639–48
[45] Makmal A, Melnikov A A, Dunjko V and Briegel H J 2016 Meta-learning within projective simulation *IEEE Access* **4** 2110–22
[46] Miller D A B 2013 Self-aligning universal beam coupler *Opt. Express* **21** 6360–70
[47] Grillanda S, Carminati M, Morichetti F, Ciccarella P, Annoni A, Ferrari G, Strain M, Sorel M, Sampietro M and Melloni A 2014 Non-invasive monitoring and control in silicon photonics using cmos integrated electronics *Optica* **1** 129–36
[48] Russell N J, Chakhmakhchyan L, O'Brien J L and Laing A 2017 Direct dialling of haar random unitary matrices *New J. Phys.* **19** 033007
[49] Burgwal R, Clements W R, Smith D H, Gates J C, Kolthammer W S, Renema J J and Walmsley I A 2017 Using an imperfect photonic network to implement random unitaries *Opt. Express* **25** 28236–45
[50] Han J and Moraga C 1995 The influence of the sigmoid function parameters on the speed of backpropagation learning *Proc. Int. Workshop on Artificial Neural Networks: From Natural to Artificial Neural Computation, IWANN '96* (Berlin: Springer) pp 195–201
[51] Taylor T 2005 Computational topology and fractal trees *PhD Thesis* Dalhousie University

[52] Imoto N, Haus H A and Yamamoto Y 1985 Quantum nondemolition measurement of the photon number via the optical kerr effect *Phys. Rev.* A **32** 2287–92
[53] Tu X, Song C, Huang T, Chen Z and Fu H 2019 State of the art and perspectives on silicon photonic switches *Micromachines* **10**
[54] Nikolova D, Calhoun D M, Liu Y, Rumley S, Novack A, Baehr-Jones T, Hochberg M and Bergman K 2017 Modular architecture for fully non-blocking silicon photonic switch fabric *Microsyst. Nanoeng.* **3** 16071
[55] Togelius J, Karakovskiy S, Koutník J and Schmidhuber J 2009 Super mario evolution *Proc. 5th Int. Conf. on Computational Intelligence and Games, CIG'09 (Piscataway, NJ, USA)* (IEEE Press) pp 156–61
[56] Benzi R, Sutera A and Vulpiani A 1981 The mechanism of stochastic resonance *J. Phys. A: Math. Gen.* **14** L453–7
[57] McDonnell M D and Abbott D 2009 What is stochastic resonance? Definitions, misconceptions, debates, and its relevance to biology *PLoS Comput. Biol.* **5** 1–9
[58] Wei C, Yajun W and Yang Y 2013 Combinatorial multi-armed bandit: general framework and applications *Proc. 30th Int. Conf. on Machine Learning, volume 28 of Proc. Machine Learning Research (Atlanta, GA)* pp 151–59