IOP Publishing Quantum Sci. Technol. 9 (2024) 045011 https://doi.org/10.1088/2058-9565/ad5907

Quantum Science and Technology

RECEIVED
29 July 2023

REVISED
8 March 2024

ACCEPTED FOR PUBLICATION
17 June 2024

PUBLISHED
15 July 2024

PAPER

Towards interpretable quantum machine learning via single-photon quantum walks

Fulvio Flamini[1,3], **Marius Krumm**[1,3,*], **Lukas J Fiderer**[1], **Thomas Müller**[2] **and Hans J Briegel**[1]

[1] Universität Innsbruck, Institut für Theoretische Physik, Technikerstraße 21a, A-6020 Innsbruck, Austria
[2] Department of Philosophy, University of Konstanz, Universitätsstraße 10, 78464 Konstanz, Germany
[3] These authors contributed equally to this work.
[*] Author to whom any correspondence should be addressed.

E-mail: marius.krumm@uibk.ac.at

Keywords: quantum machine learning, integrated photonics, quantum walks, artificial intelligence

Supplementary material for this article is available online

Abstract

Variational quantum algorithms represent a promising approach to quantum machine learning where classical neural networks are replaced by parametrized quantum circuits. However, both approaches suffer from a clear limitation, that is a lack of interpretability. Here, we present a variational method to quantize projective simulation (PS), a reinforcement learning model aimed at interpretable artificial intelligence. Decision making in PS is modeled as a random walk on a graph describing the agent's memory. To implement the quantized model, we consider quantum walks of single photons in a lattice of tunable Mach–Zehnder interferometers trained via variational algorithms. Using an example from transfer learning, we show that the quantized PS model can exploit quantum interference to acquire capabilities beyond those of its classical counterpart. Finally, we discuss the role of quantum interference for training and tracing the decision making process, paving the way for realizations of interpretable quantum learning agents.

1. Introduction

Classical machine learning methods are revolutionizing science and technology, with applications ranging from drug discovery [1] to the study of quantum phases of matter [2]. Consequently, the exploitation of quantum effects to enhance machine learning has emerged as a rapidly evolving research domain [3], showcasing early successes such as provable learning advantages [4] and the development of variational quantum circuits tailored to noisy intermediate-scale quantum devices [5]. These circuits can serve as replacements for classical neural networks within traditional algorithms. Nevertheless, amidst the rapid progress made in both classical and quantum machine learning, it is essential to bear in mind the inherent limitations of classical neural networks, which may persist or even be exacerbated in their quantum counterparts.

In particular, the opaque nature of classical neural networks can prove challenging, especially in scenarios necessitating delicate decision-making directly impacting human lives. Similarly, the quest for deeper comprehension of natural phenomena extends beyond a mere oracle for specific problems: Often the process leading to a solution yields insights that are as valuable as the solution itself. While various approaches have been explored in the machine learning literature [6], the most principled approach appears to be the design of transparent models that enable the tracing of decision processes. However, with the inclusion of quantum effects, the prospect of interpreting decision processes appears daunting. Interesting first steps in this direction, using Shapley values to measure feature importance [7] or mutual information to trace the flow of information [8], still rely on black-box methods from classical explainability research.

In this work, we aim to dispel such reservations by quantizing a transparent classical reinforcement learning model known as projective simulation (PS) [9, 10]. By undertaking this quantization, we lay the groundwork for investigating the interpretability aspects inherent in quantum machine learning models. In

PS, the agent's decision making process is modeled as a random walk of an excitation in an episodic memory. Following an approach already investigated by some of the authors [11], we replace this classical random walk with a single-photon quantum walk in an optical interferometer. On the one hand, we are able to preserve the core features of classical PS, so that the main ideas developed in the field can be transferred to the quantum domain. On the other hand, this approach can be tailored to end applications, and even extended to other platforms such as trapped ions [12, 13] and superconducting quantum circuits [14, 15]. These two points link to another main motivation of this work, that is, the development of an interpretable learning agent that operates in distributed quantum optical networks [16], and that can interact with other agents by exchanging quantum information encoded in light.

The article is structured as follows: In section 2, we review the PS model that provides the background of our work. In section 3, we present our approach to quantizing PS and, based on that, to designing a photonic implementation. In section 4, we introduce two variational training algorithms tailored to the proposed framework, with applications that go even beyond the scope of this work. In section 5, we provide numerical evidence that such quantum PS agents can generalize better than their classical counterparts. Most of our analyses can be tested with a package we make available online [17], and can be further extended in various directions (using the same package; see also supplementary sections I–III).

2. PS

In this section, we review the basics of RL and PS. RL is a paradigm of machine learning that involves an agent interacting with an environment (see figure 1). The agent receives from the environment an input $s \in S$, called *state* or *percept*, and decides what action $a \in A$ to take next. RL typically considers Markov decision problems, for which the agent's *policy* can be described by a conditional probability distribution $\pi(a|s)$. The agent receives immediate feedback on its actions through a reward R; its goal is to find the policy that maximizes the long-term expected reward.

In this work we focus on PS, a physically motivated RL model that has been applied to the study of collective behavior [18], robotics [19], automated design of quantum experiments [20] and protocols [21], and quantum error correction [22]. PS relies on a memory structure called *episodic and compositional memory* (ECM), described by a directed graph (see figure 1). The vertices of this graph, called *clips*, are associated with memories of the agent such as remembered percepts, actions or intermediate, more complex steps in the decision making process. Importantly, this means that a clip carries semantic information [23]. When the environment delivers a percept s, the corresponding clip is excited, and a random walk of a single excitation is initiated in its clip network (see figure 2), until the excitation hits an action clip a. This random walk in the ECM can be interpreted as a chain-of-thought that led to the decision. Each directed edge from clip c_i to clip c_j is associated with an unnormalized transition probability h_{ij} called h-value. The transition probability $p(c_j|c_i)$ is computed by normalizing over all clips that can be reached from c_i:

$$p_{ij} \equiv p\left(c_j|c_i\right) = \frac{h_{ij}}{\sum\limits_{j'} h_{ij'}}. \tag{1}$$

Learning occurs by changing the topology or the edge weights of the clip network, i.e. by tuning the h-values, which are initially set to 1. To this end, each edge with h-value h_{ij} is additionally equipped with a so-called glow factor g_{ij}, which contributes to the standard PS update rule [10]:

$$h_{ij}^{(\mathrm{t}+1)} = 1 + (1-\gamma)\left(h_{ij}^{(\mathrm{t})} - 1\right) + g_{ij}^{(\mathrm{t})} R^{(\mathrm{t})}. \tag{2}$$

Here, the role of the *glow* factor $g_{ij}^{(\mathrm{t})}$ is to remember transitions $(c_i \to c_j)$ taken in past interactions with the environment, so that they are reinforced when a reward $R^{(\mathrm{t})}$ is received. Further, $\gamma \in [0,1]$ is a *forgetting* factor: its main purpose is to gradually dampen previously learned transition probabilities, to let the agent adapt to changing environments. γ is also used as a soft cut-off to prevent the h-values to overflow. Each $g_{ij}^{(\mathrm{t})}$ is updated as:

$$g_{ij}^{(\mathrm{t})} = \begin{cases} 1 & \text{if } c_i \to c_j \text{ was taken at } t, \\ (1-\eta)\, g_{ij}^{(\mathrm{t}-1)} & \text{otherwise,} \end{cases} \tag{3}$$

where $g_{ij}^{(0)} = 0$ and $\eta \in [0,1]$ is a discount factor for the actions taken further in the past. We refer to the literature on PS for a more detailed description and additional mechanisms, such as association [24] and reflection [25, 26].

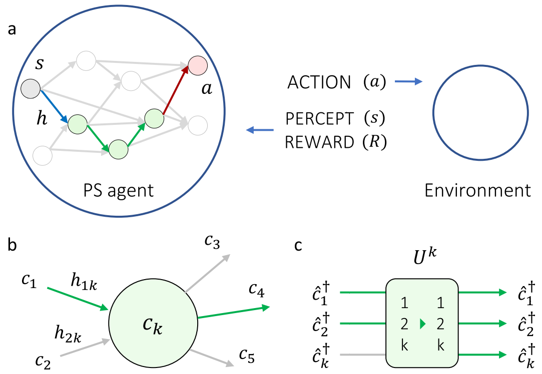

Figure 1. Projective Simulation (PS). This work uses PS as a framework for reinforcement learning and its quantization. (a) When a PS agent receives a percept s from the environment, it initiates a random walk in the graph that describes its memory, until it hits a vertex representing an action a. The performed action may be rewarded, leading to an update of the parameters (h-values) that control its decision making process. (b) In classical PS, the vertices of the graph represent *clips* (c_k). Directed edges are associated with transition probabilities that depend on the h-values. (c) In the quantized PS, clips are represented by mode operators $\hat{c}_k^\dagger$ describing a single excitation, while the connectivity is governed by mode-mixing transformations U^k. Each U^k acts on the mode operators associated with the parent clips of c_k (here c_1, c_2), and its complex elements are the transition amplitudes that replace the classical probabilities of transitioning to c_k. All panels: all edges that contributed to the decision making process are in green.

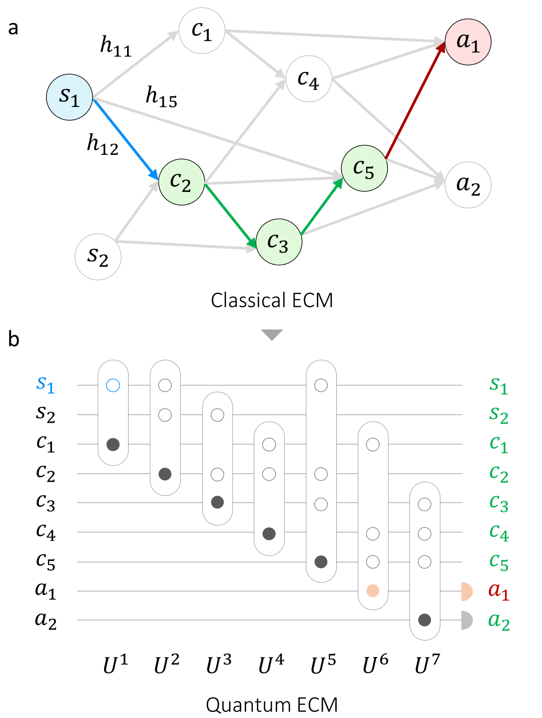

Figure 2. Decision making in classical and quantum projective simulation (PS). (a) In PS, decision making occurs via a classical random walk over a network of clips $\{c_1, \ldots, c_{|C|}\}$ that describes the agent's episodic and compositional memory (ECM) [23]. When the agent receives a percept s (blue), an excitation triggers and performs a random walk over the network of intermediate clips $\{c_1, \ldots, c_{|I|}\}$ (green) until it hits one of the action clips $\{a_1, \ldots, a_{|A|}\}$ (red), which is immediately coupled out. (b) In the proposed quantized PS, these random walks become quantum walks of single excitations through layers implementing the transformations U^k, which mix sets of clips in a certain order (see Supplementary section V). Here, clip-to-clip transition probabilities, classically described by the h-values, are replaced by transition amplitudes described by the matrix elements of U^k. The evolution can be seen as hopping between eigenstates (of the modes) in the Hilbert space (empty disks: in/out modes; filled disks: output modes). This panel reproduces the quantized evolution in the classical ECM in panel (a). Clips are now visited in superposition (green).

3. Quantum PS

Since decision making in PS is generated via a random walk of an excitation, it is possible to quantize it by giving learning agents access to a quantum walk in a quantum memory. In the literature, several variants of this idea have been investigated. In [25], an algorithm based on discrete quantum walks was presented which provides a quadratic speedup. However, this algorithm relies on sophisticated oracles that likely will not be available in the near future. Furthermore, since the algorithm is described as a quantum circuit, it is not clear how it can be interpreted as a random walk of a physical excitation. On the other hand, in [9], it was proposed to quantize PS by considering a quantum walk of an excitation in a quantum many-body system. Since Hamiltonians are hermitian, in such systems the probability amplitude of a transition is as large as the amplitude of the backwards transition. To break this symmetry, the authors proposed to use dissipative effects ('quantum jumps'), at the cost of coherence.

In this section, we propose a quantum extension of PS that allows for an optical implementation. Our guiding principle is to design a PS agent *natively* as a quantum walk of a physical excitation. In particular, our proposal does not rely on the availability of quantum oracles. Our scheme uses an optical interferometer that induces a natural directionality of transitions, without the need for dissipation. We will first describe how to implement the main elements of PS in the quantum domain in general and then, in section 3.2, on an optical platform.

3.1. From classical to quantum PS

In order to quantize PS and facilitate an optical implementation, we need to quantize its main elements: the episodic memory, realized by the clip network, and the decision-making process, described by the propagation of an excitation through the clip network. A physical realization of a clip network could be a system of harmonic oscillators (modes) with non-trivial many-body interactions between sets of oscillators [9]. In case of an optical implementation, clips $c_i \in C$ (including percept clips, action clips and intermediate clips) can then correspond to individual modes in an optical interferometric network [11] (see section 3.2). This choice proves especially useful, as we will see in section 3.2, to allow a natural translation from the classical model to its quantized implementation. For the quantization, we associate with each clip c_i a pair of creation and annihilation operators of the corresponding (optical) mode, $\{\hat{c}_i^\dagger, \hat{c}_i\}$. An excitation of memory clip c_i at time τ_0 is then described by the quantum state $\hat{c}_i^\dagger(\tau_0)|\mathrm{vac}\rangle$, where $|\mathrm{vac}\rangle$ denotes the vacuum state, describing the inactive state of the agent's memory. Here we use the Heisenberg picture where the operators carry the agent's internal time dependence. Note that we index RL steps with t, while physical time within a step is denoted by τ_k.

A decision-making process is initiated by the excitation of some clip c_i at time τ_0. The subsequent evolution of the agent's memory will be described by a unitary of the form

$$\hat{c}_j^\dagger(\tau_n) = \mathcal{U}^\dagger(\tau_n, \tau_0)\, \hat{c}_j^\dagger(\tau_0)\, \mathcal{U}(\tau_n, \tau_0) = \sum_{i=1}^{|C|} U_{ji}^{\mathrm{ECM}}\, \hat{c}_i^\dagger(\tau_0). \tag{4}$$

which relates the excitation operators of the clip network at different times. The Heisenberg time evolution of the ECM $\mathcal{U}(\tau_n, \tau_0)$ is thus realized by the interferometric scattering matrix U^{ECM} connecting the modes, which, in turn, is composed of a sequence of n intermediate transformations U^k,

$$U^{\mathrm{ECM}}\left(\vec{\theta}\right) = \prod_{k=1}^{n} U^k\left(\vec{\theta}\right). \tag{5}$$

The U^k are updated during the learning process via the set of parameters $\vec{\theta}$ (see figure 2). The precise role of the $U^k(\vec{\theta})$ will be discussed in more detail below. For convenience and to facilitate an optical implementation, in this work we will consider ECMs described by a directed acyclic graph (DAG).

Percept encoding—When the agent receives a classical percept at time τ_0, this leads to the excitation of a percept clip s. The quantum state of the memory is then described by $c_s^\dagger(\tau_0)|\mathrm{vac}\rangle$. Subsequently, a quantum walk through the clip network starts [9].

Quantum walk in the ECM—To connect the random walk in the ECM to the quantum walk considered in this work, we refer to the examples in figures 1 and 2. As shown in panel figure 1(b), a clip effectively acts as a node that routes incoming edges from sets of clips to other clips in the next step of the random walk. In the quantum ECM sketched in figures 1(c) and 2(b), this mechanism translates into the application of a transformation U^k, which mixes creation operators of incoming clips with $\hat{c}_k^\dagger$. A general clip-to-clip transition $c_i \to c_j$ is then modeled by transformations U^k that mix subsets of creation operators, so that the stochastic nature of PS manifests itself at the level of quantum amplitudes.

A crucial departure from classical PS is that clips are now visited in parallel (in superposition, see figure 1(c), and all regions of the ECM that are reachable from a percept via a sequence of U^k contribute to the decision making process. As a consequence, the probability to find an excitation in action clip a at time τ_n (described by the quantum state $c_a^\dagger(\tau_n)|\mathrm{vac}\rangle$) when starting with an excitation of percept clip s at time τ_0 (described by $c_s^\dagger(\tau_0)|\mathrm{vac}\rangle$) is given by the modulus square of the corresponding transition amplitude

Table 1. Links between the main elements of projective simulation (PS) and its proposed optical quantization.

Element	Classical PS	Quantum PS (→ Optical implementation)
Decision making	Random walk of one excitation	Quantum walk of a single excitation (photon)
Agent's memory	Directed graph of clips	Variational (optical) circuit
Clips	Vertices of the graph	(Optical) mode operators
Clip-to-clip transitions	h-values	Quantum scattering amplitudes
Input/Output encoding	Percept/Action clips	Input/Output modes
Learning mechanism	Update of h-values via equation (2)	Update of variational parameters (phase shifters) via equation (8)

$$p_{sa} = |\langle a|s\rangle|^2 \equiv |\langle a;\tau_n|s;\tau_0\rangle|^2 = \left|U^{\mathrm{ECM}}_{as}\left(\vec{\theta}\right)\right|^2, \tag{6}$$

where $|c_i;\tau_k\rangle = \hat{c}_i^\dagger(\tau_k)|\mathrm{vac}\rangle$ are the time-dependent eigenstates of the number operators $\hat{c}_i^\dagger(\tau_k)\hat{c}_i(\tau_k)$ corresponding to an excitation being present in clip c_i at time τ_k. Equation (6) entails a quantum interference of all paths. While in classical PS the excitation has a well-defined trajectory in the ECM, now the excitation can be delocalized over all the allowed modes (i.e. the quantum amplitude is in general non-zero for more than one mode at the same time). This aspect bears an interesting potential for the learning model, which has been discussed in the literature [9, 27] and that is further discussed in section 4.

Action decoding—The decision-making process ends when U^n is applied and the output states are measured. When the excitation is detected in an output state associated with an action, this action is coupled out. In all other cases, the excitation can be (i) routed back to other regions of the reachable ECM; (ii) discarded, exciting again the same percept. Solution (ii) can be interpreted as the agent's capacity to *not* take action during this decision-making process (see also the notion of *suspension of judgment* [28]).

Practically, the agent takes action by post-selecting over the outcomes that correspond to actions. In general, one can associate multiple output states with a single action. This makes it possible to avoid post-selection in settings with an action space smaller than the percept space, or to implement quantum channels that are more general than unitaries.

Training the quantum PS agent—To train the agent, we consider a variational approach based on a loss function $\mathcal{L}$ that is inspired by classical PS and aims to reproduce its learning mechanism. $\mathcal{L}$ depends on the probabilities $p_{sa}(\vec{\theta})$ to take action a given a percept s (that is, the policy) and the variational parameters $\vec{\theta}$. These probabilities can be estimated either numerically (by simulating the quantum walk) or experimentally (via multiple measurements with fixed input). In section 4.1 we introduce and motivate our choice for $\mathcal{L}$ (see equation (8)), along with several modifications and training methods.

3.2. Quantum optical PS

In this section, we describe our approach to implement quantum PS on a photonic platform. Table 1 summarizes the main connections with classical PS.

Related works—Photonic circuits provide a promising platform in both classical and quantum domains [29, 30], with results of increasing complexity being continuously reported [31, 32], in particular for (un)supervised learning [33–38] and RL [11, 27, 39] based on path encoding.

In this work, we focus on RL and aim to design a quantum PS agent that operates with a native quantum walk of physical excitations. Hence, we will replace the single excitation with a single photon propagating in a linear-optical circuit. This approach, while retracing some of the intuitions in [11] where the p_{sa} were reproduced by optical binary trees, promises the following advantages: (1) it requires only a single (possibly larger) circuit for all percepts, instead of one binary tree per percept and layer, with no explicit need for fast active switching; (2) it allows to take advantage of quantum coherence and interference in the decision making process; (3) it allows for more advanced training algorithms, which are applicable beyond the scope of this work (see section 4); (4) it suggests concrete directions for further developments (see Supplementary Sections II-III), such as a framework for learning with quantum data. Below we outline how the proposed quantized version of PS allows for a natural implementation on a photonic circuit that is feasible with current state-of-the-art technology.

Photonic architecture—For the implementation of PS on quantum hardware, we want to identify quantum extensions of the Markov chain process that govern the evolution of a classical excitation. These extensions, usually referred to as quantum channels (completely positive trace-preserving maps), would have to reproduce the dynamics in classical PS as a limiting case. A direct implementation of classical PS would require the $\{U^k\}$ to reproduce the stochastic maps that form the random walk in the ECM graph described in section 2. Specifically, the real-valued matrices P associated with these transformations need to (i) reproduce the probabilities p_{ij} induced by the h-values, and (ii) fulfill the connectivity in the ECM (P can be

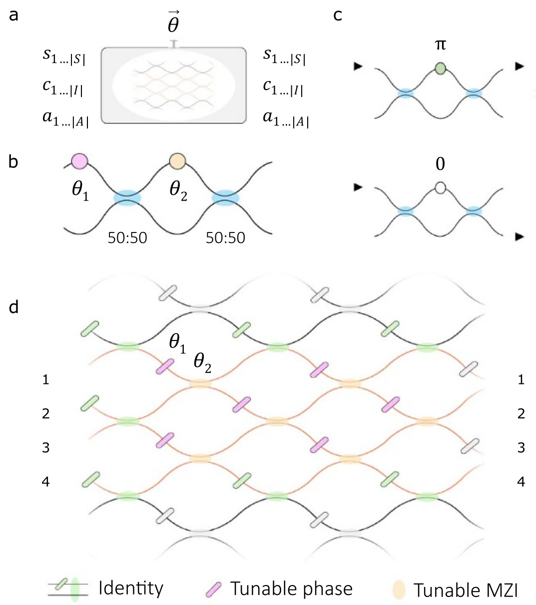

Figure 3. Optical implementation of PS. To enable PS on photonic circuits, we use as variational ansatz the square architecture [40]. (a) The update of the agent's memory translates into an update of the phase shifters $\vec{\theta}$. (b) Any unitary U can be implemented with a regular mesh of Mach–Zehnder interferometers (MZI), consisting of two 50:50 beam splitters (BS, blue) and two phase shifters (θ_1, θ_2) [40, 41]. (c) When $\theta = \pi$ all light entering a MZI is reflected (*bar* state), while when $\theta = 0$ it is transmitted (*cross* state). (d) Square, universal architectures can be carved inside larger ones by tuning the phases around them. In this panel, a universal 4×4 circuit (in/out modes are labeled) is carved in a larger mesh. Tilted elements represent tunable phase shifters (those inside the carved structure are in fuchsia). BSs with arbitrary transmissivities (orange) are implemented using tunable MZIs. Light can be confined to the pink waveguides by setting the green optical elements.

seen as a weighted adjacency matrix). Such a class of transformations can in principle be realized on a photonic platform by recalling that any real-valued matrix P can be decomposed as $P = V_1 \Sigma V_2^\dagger$ via singular value decomposition, where Σ is a diagonal matrix with non-negative real entries (i.e. optical attenuators) and $V_{1,2}$ are unitary transformations [33]. Hence, the architecture we use for the variational circuit is the canonical square one [40], since it can implement any unitary transformation by suitably tuning its phase shifters. Indeed, we recall that any such U^k can be implemented by a homogeneous mesh of Mach–Zehnder interferometers, each comprising two 50:50 beam splitters and two phase shifters (see figure 3 and section B) [40, 41]. The variational parameters $\vec{\theta}$ correspond to the phase settings that control U^k.

In the fully coherent case, without additional attenuators, one can set $\Sigma = I$ in the singular value decomposition, meaning that the U^k become unitary. In this case, due to unitarity constraints and since each U^k acts as an adjacency matrix, additional edges between clips are created that are not present in the original ECM. This limitation represents an additional departure from classical PS, which can open up interesting opportunities in the context of learning, for instance leveraging the constraints to improve generalization.

Percept encoding—We associate each percept with a photon input in a spatial mode of the circuit. The size of the Hilbert space grows linearly in the number of percepts.

ECM structure—We associate each clip with one optical mode of the architecture. The unnormalized transition probabilities h_{ij} that describe the transitions $c_i \to c_j$ are now replaced by transition amplitudes between optical modes, described by a complex amplitude U^k_{ji}.

Our chosen architecture is especially useful for reasons that go beyond its universality, noise resilience and compactness [40], since it also promises to support key features of PS:

- *Clip creation/deletion*: new clips (i.e. optical modes) can be added during the learning process, while clips that are rarely traversed can be removed.
- *Edge creation/deletion*: the random walk can be manipulated by creating or deleting links between clips.
- *Clip composition*: fictitious clips can also be created, by random variation or merging of existing clips.

We observe that this mesh allows one to *carve* substructures that are in turn universal for unitary transformations, by adjusting the surrounding phases (without the need to modify the hardware; see figure 3(d). This means that the Hilbert space describing the ECM can be dynamically enlarged to accommodate new clips, and the U^k_{ji} can be controlled by intervening on the regions of the circuit that implement U^k.

Action decoding—In PS, the decision-making process ends when an action clip is hit. In the proposed implementation, this corresponds to detecting photons from one of the output modes that correspond to an action clip. Whenever a photon is detected in a mode that does *not* correspond to an action, that event is discarded. In the future, these photons could be routed back into the circuit, or associated with mechanisms of suspension of belief [28].

4. Training the architecture

In this section, we discuss two algorithms to train the quantum optical PS agent. To this end, we first recall that the standard update rule in PS has no straightforward quantum analogue: while only one path is taken by the classical excitation and rewarded by the update rule (technically, the h-values associated with its edges), all paths from percept to action are visited by the quantum excitation, and the very notion of path loses its meaning (see supplementary section IV for a discussion on the impact of superposition in the quantized model). Hence, also motivated by the advances in quantum machine learning, here we take a different, variational approach to update the parameters $\vec{\theta}$.

4.1. Variational approach to mimic classical PS

The first algorithm to train the quantum PS agent is based on a variational optimization of a loss function $\mathcal{L}$ inspired by classical standard PS.

Since a quantum circuit does not have access to the intermediate states of the computation, the update rule we propose adapts that of a classical 2-layer ECM. This does not mean that the quantum agent is necessarily 2-layered (in fact, the architecture supports multi-layer ECMs, as shown in figure 2); rather, it means that we update the observable transition probabilities $p_{sa}(\vec{\theta})$. In turn, also the intermediate transition amplitudes in the circuit are affected. However, the update rule of PS requires the use of unnormalized h-values instead of (normalized) probabilities. In the following, we will sometimes use the shorthand notation $p_{sa}^{(t)}$ for $p_{sa}(\vec{\theta}^{(t)})$, where t denotes the time index. Let us first introduce the classical normalization factor of the h-values h_s, i.e. $h_s := \sum_a h_{sa}$, from which we get $h_{sa} = p_{sa}h_s$ and the classical 2-layer PS update rule

$$p_{sa}^{(t+1)}h_s^{(t+1)} = 1 + (1-\gamma)\left(p_{sa}^{(t)}h_s^{(t)} - 1\right) + g_{sa}^{(t)}R^{(t)}. \tag{7}$$

A loss function $\mathcal{L}$ that implements equation (7) is

$$\mathcal{L}\left(\vec{\theta}^{(t+1)}, \left\{h_s^{(t+1)}\right\}_s\right) = \sum_s D\left[p_{sa}\left(\vec{\theta}^{(t+1)}\right)h_s^{(t+1)} - 1,\ (1-\gamma)\left(p_{sa}^{(t)}h_s^{(t)} - 1\right) + g_{sa}^{(t)}R^{(t)}\right]. \tag{8}$$

Here, $\vec{\theta}^{(t+1)}$ and $\left\{h_s^{(t+1)}\right\}_s$ are variational parameters, while D is a suitable distance measure such as the Kullback–Leibler divergence (after rearranging the arguments of D), $R^{(t)}$ is the reward, and $g_{sa}^{(t)}$ is the glow factor described in equation (3). The expression in equation (3) is still valid in the proposed quantum extension, since the update rule underlying the loss function is taken from a classical two-layer PS, and both input (s) and output (a) are classical. However, we emphasize again that the quantum ECM itself is multi-layered. Since uncontrolled output phases can spoil the quantum interference, if the agent is used as a module in a larger photonic setup, we choose to add the ℓ^1-norm of these phases to the loss function.

Note that only the first argument of D in equation (8) contains tunable parameters, while the right entry contains the target values calculated from the previous state of the agent, as a reference, similar to deep-Q learning. Similarly, one can use *experience replay* [42, 43] to collect rewards and glows for a batch of percept-action pairs, and add up or merge the loss functions in appropriate ways. That is because continuous online learning (i.e. updating after each action) has the disadvantage that the unitarity constraints will also affect the transition probabilities of undetected percept-action pairs, without taking their rewards into account.

In the following, we discuss suitable heuristic simplifications of this loss function, which make it more amenable to an actual implementation. First, by approximating $h_s^{(t)} \approx h_s$ and introducing an effective reward $r_s^{(t)} := \frac{R^{(t)}}{h_s}$ in the exact update rule in equation (7), we obtain

$$p_{sa}^{(t+1)} - \frac{1}{h_s} \approx (1-\gamma)\left(p_{sa}^{(t)} - \frac{1}{h_s}\right) + g_{sa}^{(t)}r_s^{(t)}. \tag{9}$$

Without a reward (i.e. $r_s = 0$), the forgetting mechanism after the approximation seeks to reset $p_{sa}^{(t)}$ to $\frac{1}{h_s}$. However, in information theory, the no-information distribution is the uniform distribution. This motivates us to replace $\frac{1}{h_s}$ with a uniform distribution $\frac{1}{|A|}$,

$$p_{sa}^{(t+1)} = \frac{1}{|A|} + (1-\gamma)\left(p_{sa}^{(t)} - \frac{1}{|A|}\right) + g_{sa}^{(t)} r_s^{(t)}. \quad (10)$$

So far, we did not make any choices concerning the source of the rewards $R^{(t)}$. For our simplifications, we now take the point of view that the reward $R^{(t)}$ is designed by a programmer to achieve an intended agent behavior. Then, a further simplification can be made by directly modelling the effective reward $r^{(t)}$, rather than R. Therefore, we replace $r_s^{(t)}$ with $r^{(t)}$. With this change in mind, equation (10) now looks like an analogue of the exact update rule in equation (7), but for probabilities instead of h-values. To ensure that the agent updates get smaller over time, it is recommended to use an 'annealing schedule' for the reward, i.e. to put an additional time dependence into the reward of the form $r^{(t)} := f(t) \cdot r_0$. Here, $f(t)$ is a function that monotonically decreases from 1 to 0. This allows the training algorithm to make finer and finer updates to the probabilities.

In order to stay as close as possible to classical PS, the update rule would have to be applied to all percept-action pairs. This would require implementing the forgetting mechanism ($\gamma > 0$) for unvisited percept-action pairs, or enforcing the transition probability of unvisited pairs to remain unperturbed ($\gamma = 0$), leading to a complicated loss function. To avoid this overhead, we include only the actually observed percept-action pairs in the loss function, while the transition probabilities of unobserved pairs are allowed to change according to the unitarity constraints. Combining all of the above simplifications, we obtain the loss function

$$\mathcal{L}(\vec{\theta}) = \sum_{(s,a)\,|\,g_{sa}^{(t)}>0} D\left[p_{sa}(\vec{\theta}^{(t+1)})\,, C\left(\frac{1}{|A|} + (1-\gamma)\left(p_{sa}^{(t)} - \frac{1}{|A|}\right) + g_{sa}^{(t)} r^{(t)}\right)\right] \quad (11)$$

where C is a (smooth) cutoff function to enforce that probabilities are in $[0,1]$, and only (s,a)-pairs with non-zero glow are considered. We point out that summing over all states is necessary only in the general case, due to the forgetting and glow mechanisms. However, when the environment is described by a Markov decision process (with no forgetting) or the glow has a finite horizon, the number of states in the sum gets significantly reduced and the process faster.

4.2. Variational approach based on causal diamonds

In the previous section, we implicitly assume that at each step one tunes *all* phase shifters (i.e. parameters $\vec{\theta}$) in the circuit. Indeed, this is possible and can be beneficial to train the agent faster and with a better performance. At the same time, updating all phases at each step might not be necessary for the following reason.

In a square architecture with $|C|$ input/output modes, the number of phase shifters is quadratic in $|C|$. However, for a fixed transition probability $p_{sa}(\vec{\theta})$, one can see that only some phases play a role: Those at the intersection of the future and past light cones of s and a, respectively (see figure 4(a). In the literature on relativity, this intersection is called *causal diamond*. A number of considerations follow:

i. To update $p_{sa}(\vec{\theta})$, one could simply tune the phase shifters inside the causal diamond.
ii. For light to reach a from s, what matters is that light does not leak out of their causal diamond. Hence, one can also just tune the $O(|C|)$ phases that lie on the *surface* of the causal diamond.
iii. Since leakage only occurs on the surface of the past light cone, it is sufficient to focus on its intersection with the causal diamond (see figure 4). We call these phase shifters *leaking nodes*.

It is also interesting from a learning perspective to discuss the apparent inconsistency between the spread of the wave packet during the quantum walk (influenced by $O(|C|^2)$ parameters) and the update of just $O(|C|)$ parameters in the causal diamonds. This is due to the fact that the quantum walk relates to the decision-making process, while the update relates to its training. Intuitively, it is as if the decision-making process looked at the remembered history of the agent's interaction (which involves all parameters), while individual updates only look at the last in/out transitions.

Overall, in terms of reconfiguration time, stability and power consumption, the above observations allow for progressively simpler experimental requirements. The price to pay is that the surfaces of different causal diamonds (for different (s, a) pairs) intersect and, for this reason and due to unitarity constraints, updating one transition probability also affects the other transitions. Also, the fewer phase shifters are tuned, the smaller the set of values that solve the optimization problem when multiple (s, a) pairs are rewarded. The desired trade-off can be found with the experimental conditions in mind.

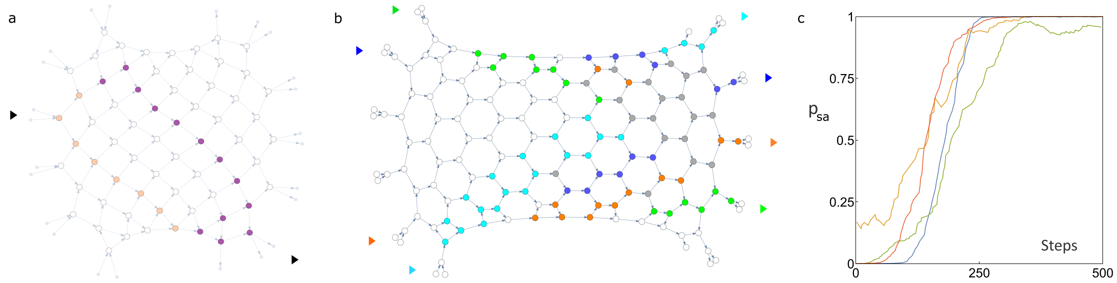

Figure 4. Update via causal diamonds. To reinforce a transition probability p_{sa} in a 2-layer PS, it is not necessary to update all phase shifters at each step. (a) Only the phases inside the causal diamond of (s, a) influence p_{sa} (colored nodes indicate the surface of the causal diamond). Since light can only leave the causal diamond from the purple nodes (*leaking nodes*), it is sufficient to tune their phases. (b) The approach works also with multiple transitions, by iteratively tuning different sets of leaking nodes (here shown with five different colors, including gray for the overlaps, and representing pairs of adjacent phase shifters and beam splitters as a single node), even if the updates compete with each other (some leaking nodes belong to multiple causal diamonds). (c) The agent learns to maximize p_{sa} for the four (s, a) pairs in panel (b), by tuning only the leaking nodes. In some cases, it may be necessary to tune more nodes to maximize all probabilities. Each percept/action is encoded in a pair of adjacent input/output modes.

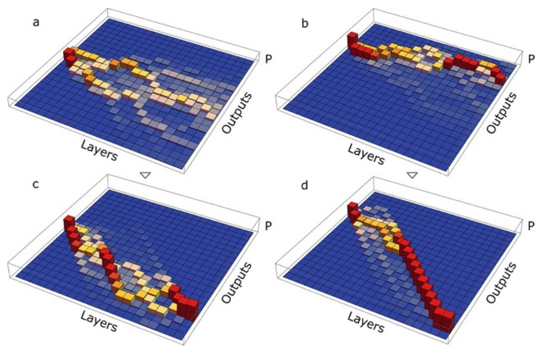

Figure 5. Partial traceability in the decision making process of quantum PS agents. (a) and (b) Two quantum walks, associated with as many 2-layer ECMs, are numerically simulated with a square optical architecture. Each panel shows the probability P to measure a photon in one of the output modes before training. Each column (label: *Layers*) displays the squared column (for a fixed percept) of the unitary matrix obtained by adding a new layer of optical elements, from left (input layer) to right (full circuit). The rightmost column of each plot corresponds to the output probabilities of a quantum walk over the full ECM. (c) and (d) Same analysis after training with the causal diamond. Panels (c) and (d) correspond to the evolutions in panels (a) and (b), respectively. We observe that a notion of partial trace emerges inside the circuit, which can be leveraged to study interpretability in quantum learning agents.

We consider two main strategies to tune the sets of relevant phases described by points (i-iii) above: (a) phases are independently updated by gradient descent, and (b) phases are updated sequentially (from left to right, each phase in a layer is set to a new value before the phases in the next layers are adjusted). The latter strategy is possible because we assume that light propagates only forward, and it leads to a faster and smoother training. A quantitative comparison between the two strategies, and a study of the most suitable one with multiple rewarded percept/action pairs, can be done with our shared package, keeping experimental requirements in mind [17]. An illustrative analysis for strategy (b) is shown in figure 4(c). Additional information and analyses can be found in section A.2 and in the supplementary section VI, respectively. Importantly, we point out that this approach is not restricted to the regular mesh considered in this work. Rather, similar considerations also apply to circuits with arbitrary mode connectivity and even higher-level structures in the ECM [17].

Interpretability—Regarding the aspect of interpretability, we have seen how PS offers a viable framework towards achieving this goal, by allowing one to retrace the path taken between percepts and actions. We also know that this is no longer true in the quantum domain, since all paths contribute to the decision making process at once. However, while the individual path taken by a photon is not well-defined, the fact that its wave packet can be concentrated in some regions of the interferometer can give rise to a notion of partial traceability (see, e.g. figure 5). Interpretability in the proposed architecture is also facilitated by its modularity, a key feature at the core of hierarchical RL and known to enhance the explainability of artificial neural networks [44, 45]. While a modular approach could also be applied to qubit-based variational circuits, the proposed framework benefits from its synergy with partial traceability. Specifically, here one

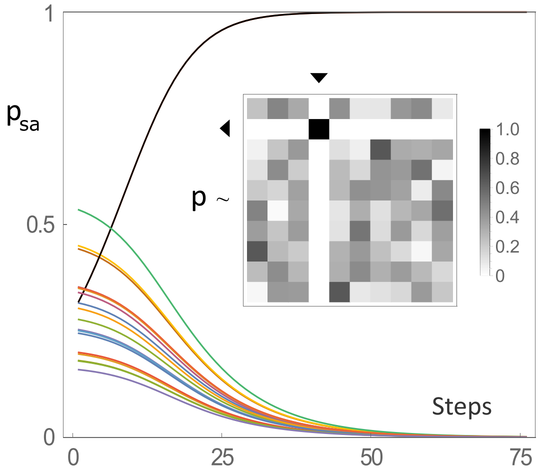

Figure 6. Update via a Gram–Schmidt process. We can directly update the unitary transformation U that describes the ECM using a Gram–Schmidt orthonormalization. Here, a learning curve (black) is shown for a single rewarded (s, a) pair in a 2-layer PS described by a 10×10 Haar-random U. All other transition probabilities (other colors), from s and to a, decrease to 0. Curves correspond to a single agent, without average. Inset: $(|U_{ij}|^2)_{ij}$ after training (arrows indicate (s, a)).

could search for specific reasons behind a failure, by looking at which modules (i.e. sets of modes) contributed the most to a decision (i.e. constructing a trace with a higher amplitude concentration).

We expect that this feature can be extended to multi-photon encoding schemes, with no need for a process tomography or reconstruction of the output probability distribution. To estimate the amplitude of a multi-photon configuration over the modes at a given layer (corresponding to a group of clips), one can tune all previous layers to implement the ECM up to that point, setting the others to implement the identity channel. One can also compute the likelihood that an n-photon coincidence event would be measured at each layer. While the number of such combinations grows exponentially and the task to compute each likelihood is hard [46], the task would still be feasible for $n \sim 10$ photons in $O(n^2)$ modes [47]: a number sufficiently large to hope for interesting many-body quantum interference effects, e.g. reminiscent of so-called suppression laws and bunching effects reported in Boson Sampling experiments [48].

In the context of this work, these phenomena could lead to the suppression of clip-to-clip transitions that are not favorable during the training stage, and/or to the enhancement of specific transitions above a predefined threshold. For instance, it could be possible to find correlations between sets of modes (clips), or correlations between clips and percepts/actions modes, which carry semantics that aid interpretability. Additional considerations are provided in Supplementary section I.

4.3. Direct updating via a Gram–Schmidt process

The previous sections consider a variational approach to the optimization problem, where phases are gradually adjusted to produce a new transformation that behaves better under a given figure of merit. We now consider a complementary approach, where we first update the unitary evolution U of the ECM, and then use a well-known decomposition [40] to retrieve the $\vec{\theta}$ that produce the desired U. In this case, one directly adjusts the matrix elements of U to implement an update rule such as equations (7) or (11): the new matrix is no longer unitary; however, a Gram–Schmidt orthonormalization (GSO) process yields a new U' that approximates the intended U'_{as}.

We add a few remarks on this approach (see also section A.1). First, if U'_{as} is close to the old value U_{as}, one step of GSO only causes a small update of the other entries of U. In this sense, this approach allows for a smooth, flexible and controlled update of U (see figure 6). Also, both the GSO and the unitary decomposition are efficient and relatively fast subroutines, which could even be faster than collecting statistics for other variational approaches. In other words, the overhead is outsourced to the classical CPU. The limitation of this approach is that it assumes perfect knowledge of U and a precise control of the experimental settings. The impact of imperfections can be estimated using the shared package [17].

5. Quantum versus classical PS in a transfer-learning scenario

In this section, we apply the proposed quantum PS to an actual learning problem. The PS update rule of equations (7) and (8) has already been used in various applications [18–22]. Here, we consider a scenario which pronounces the differences between classical and quantum PS. We consider a transfer-learning

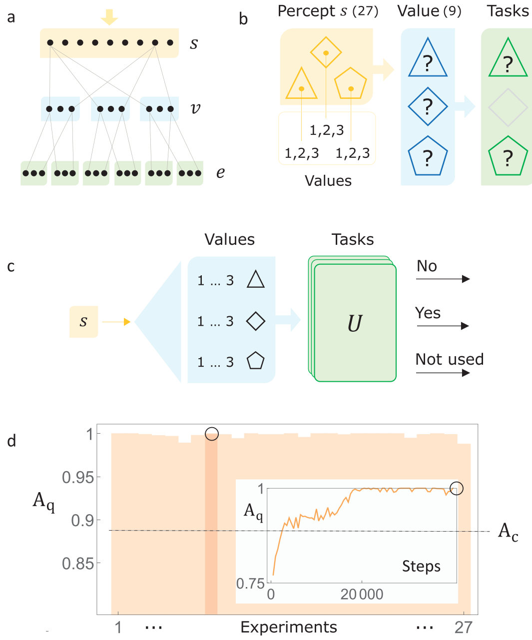

Figure 7. A transfer-learning scenario for PS agents. (a) Structure of the ideal, classical ECM considered in the transfer-learning scenario. Panel adapted from [50]. (b) Percepts s consist of the values of three observables which characterize particles. The experiment consists of two stages. (i) In the first stage, the PS agent is set up to learn a middle layer of nodes, in which each node corresponds to one observable and one value of that observable. (ii) In the second stage, this middle layer is kept fixed. The task layer is exchangeable and it represents one yes/no question about pairs of observables. Since the excitation of the classical ECM is localized, in the middle layer the classical agent can only use the value of one observable. Interference allows the quantum PS agent to combine knowledge of several observables at once. (c) Layout of the quantum PS agent. To fulfill the unitarity constraints, the connection from percept to middle layer is an interferometer tree with 9 outputs (one tree per percept). The connection from the middle layer to the task layer is a 9×9 square architecture, shared by all percepts but different for each task layer. Two output modes correspond to the yes/no answers, the others are discarded. (d) Prediction accuracy A_Q of the quantum PS agent for each of the 27 experiments (yes/no question). Inset: Learning curve for a single experiment (dark orange). A classical agent can achieve at most an accuracy $A_C = 8/9$ (horizontal line, for both plots), always lower than A_Q.

scenario [49]: a machine learning setting where a model trained for one task can solve another task. The scenario we choose is that of Eva *et al* [50] (see figure 7).

In this scenario, one considers a list of particles characterized by discretized observables, and asks whether two observables have two particular values. To answer these questions, or tasks, in our adapted version of this thought experiment the agent attempts to build a meaningful representation of the particles in two stages (corresponding to the middle layer and a task layer): In stage 1, the agent learns a representation from the percept layer to the middle layer, where each node corresponds to one observable and one value of that observable. While Eva *et al* intended that this representation is learned implicitly, in our case we directly train the middle layer to achieve this representation. In stage 2, the connections from percept to middle layer are kept fixed. The task layer has two nodes, which correspond to yes/no-answers. For each question (task), we train a separate set of connections from the middle layer to the corresponding task layer.

The contrast between classical and quantum learning is rooted in the way the decision-making process handles the structured representation of the PS agents: Since the excitation of the classical PS agent is localized, it can only use knowledge about one observable and, therefore, it cannot reliably answer questions involving two observables. The quantum PS agent, however, can use interference to combine knowledge about two observables.

In our proof-of-concept analyses, described in detail in Supplementary section IV and in the shared code [17], the quantum agent consistently achieved over 97% (weighted) accuracy for all task layers.

Conversely, one can use analytical reasoning to see that the classical PS agent cannot achieve near perfect accuracy: since the classical excitation is always localized in one node, in the middle layer the classical PS agent can at most know the value of one relevant observable. Therefore, the best thing the agent can do is to guess about the other observable. Since two out of three possible values of that other observable lead to a right no-answer, the best policy the classical PS agent can represent is to always answer no. Hence, the performance of the classical PS agent is upper bounded by $\frac{8}{9}$, thereby showing yet again the impact of quantum interference for learning agents.

6. Discussion

Significant endeavors are currently underway to leverage quantum effects for future AI applications. At the same time, it is becoming increasingly evident that there are certain use cases where interpretability in the decision-making process is crucial. In this work, we introduced a quantum learning framework obtained through the quantization of a transparent and interpretable classical learning model, aiming to explore the intricate relationship between quantum effects and interpretable decision processes. We extend the theoretical framework by proposing an implementation based on integrated photonics that is specifically tailored for quantum reinforcement learning.

Diverging from conventional, gate-based variational quantum algorithms, our approach draws inspiration not from neural networks but from the quantization of PS, an interpretable classical machine learning model. The proposed model shares the graph structure with classical PS, where decision-making is realized through a traceable and interpretable random walk. In our implementation, this random walk becomes a quantum walk of a single photon undergoing unitary transformations, and the variational parameters become the phases of a homogeneous mesh of Mach–Zehnder interferometers. In contrast to classical PS, here quantum interference effects and the constraints imposed by unitarity come into play. As we discuss, this leads to an interesting interplay between the interpretable elements inherited from classical PS and the genuine quantum effects. We also provide evidence of the enhanced learning capabilities achieved by quantum PS. To this end we investigate a transfer learning problem where quantum PS showcases its capacity to leverage quantum interference, surpassing the capabilities of its classical counterpart.

By bridging the gap between quantum effects and interpretability, the proposed quantum learning model paves the way for future investigations and applications that benefit from the combined strengths of interpretability and quantum effects. For example, some of the ideas include (i) the use of restricted connectivity to implement domain knowledge, (ii) the search for relevant regions of the trained model exhibiting amplitude concentration (see figure 5), and (iii) the link to the classical field of neuro-symbolic logic [51], motivated by the similarity between interference effects and Boolean logic. Additional follow-up investigations regarding the photonic implementation are already outlined in the Supplementary Information with numerical analyses. There, Secs. II-III discuss the potential offered by multi-photon (i.e. multi-percept) and multi-frequency quantum information processing, respectively. All results can be tested with a package we release online [17].

Data availability statement

Data for all numerical analyses is available at https://github.com/MariusKrumm/PhotonicsPS.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement, Grant Nos. 801110 and 885567. It reflects only the author's view, the EU Agency is not responsible for any use that may be made of the information it contains. ESQ has received funding from the Austrian Federal Ministry of Education, Science and Research (BMBWF). This research was funded in part by the Austrian Science Fund (FWF) [Grant-DOI 10.55776/F71]. For open access purposes, the authors have applied a CC BY public copyright license to any author accepted manuscript version arising from this submission. This work was supported in part by the Volkswagen Foundation (Az:97721). This work was also co-funded by the European Research Council (ERC) under Project No. 101055129. Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

Author contributions

F F , M K and H J B conceived the project, L J F , T M and H J B contributed to the development of the work. F F, M K and L J F conceived the analyses related to photonic circuits and to episodic memories, then F F carried out the analyses writing a Mathematica software. M K conceived the experiment on the transfer-learning scenario and the variational approach to mimic PS based on loss functions, then carried out the analyses writing a Python software. All authors contributed to discussing the results and writing the manuscript.

Conflict of interest

The authors declare no competing interests.

Appendix A. Variational algorithms

In this section, we provide additional information on the two variational algorithms presented in section 4.

The following algorithms naturally take into account the unitarity constraint inherent to the architecture. We point out that this constraint is not necessarily a limitation and, rather, it could be advantageous to improve the agent's performance. In fact, since a machine learning ansatz with more freedom is also more prone to overfitting, it is reasonable to believe that a specialized ansatz with restrictions and symmetries might generalize better. A reduced number of degrees of freedom might also help during the training stage.

A.1. Update via Gram–Schmidt process

Here we outline the update via a Gram–Schmidt process. The algorithm consists of four steps (i–iv):

i. Compute $U^{(\mathrm{t}+1)}_{a_0 s_0}$ from the current $U^{(\mathrm{t})}_{as}$ according to a desired update rule. Here, s_0 and a_0 are the observed percept and action of the current round, respectively. In the examples in section 4.3, we considered a constant rescaling factor α, i.e. $U^{(\mathrm{t}+1)}_{a_0 s_0} = \alpha\, U^{(\mathrm{t})}_{a_0 s_0}$, which acts like a learning rate.
ii. Normalize the row $U^{(\mathrm{t}+1)}_{a_0,\bullet}$.
iii. Use GSO on the other rows to obtain a new unitary, in the following way: For each row $a \neq a_0$, define

$$\tilde{U}^{(\mathrm{t}+1)}_{as} := U^{(\mathrm{t})}_{as} - \sum_{\tilde{s}\in S}\sum_{\tilde{a}\in A_a} \overline{U^{(\mathrm{t}+1)}_{\tilde{a}\tilde{s}}}\; U^{(\mathrm{t})}_{a\tilde{s}}\; U^{(\mathrm{t}+1)}_{\tilde{a}s}, \tag{12}$$

where the overbar denotes the complex conjugation, and A_a is the set of rows $\tilde{a}$ for which we have already defined $U^{(\mathrm{t}+1)}_{\tilde{a},\bullet}$, i.e. the previous $\tilde{a}$ in the loop. Determine $U^{(\mathrm{t}+1)}_{a,\bullet}$ from $\tilde{U}^{(\mathrm{t}+1)}_{a,\bullet}$ via normalization.
iv. Use the decomposition [40] of canonical square architectures to find the phase settings that implement the new unitary. The decomposition is exact and requires a number of operations quadratic in the size of U.

The main advantage of this approach is that it allows one to control any theoretical aspect of the learning process, from the update rule to the learning speed. One limitation is that it assumes perfect knowledge and control of the experimental settings. Appending to it one step of variational fine-tuning, as the one we describe in the following section, can help mitigate this issue and speed up the learning process.

A.2. Update via causal diamonds

Here we provide additional considerations on the update based on causal diamonds. The advantage of this approach is that it is efficient (only few phase shifters are adjusted at each step) and that it is variational (it takes into account the imperfect control over the experimental settings).

We consider two main approaches (a,b) to update a given subset $\Theta = (\theta_1, \ldots, \theta_N)$ of N phase shifters at each training step, albeit more sophisticated strategies can be devised and tailored to the hardware. These phases can correspond to the set of leaking nodes described in section 4.2, or to a larger set selected via other criteria.

a. *Gradient ascent*—Each phase $\theta \in \Theta$ is independently tuned to maximize a predefined figure of merit $\mathcal{F}$.
b. *Sequential*—Each phase $\theta \in \Theta$, in the order given by the light propagation, is tuned to independently maximize a figure of merit $\mathcal{F}$, and is immediately set to the new value before the next phase is probed.

The latter approach (b) (see figure 4) is usually faster than (a), since each phase update starts from a progressively higher value of $\mathcal{F}$, and the individual updates within a training step do not sabotage each other. However, an update based on gradient ascent as in (a) can be beneficial to escape local minima, which are more likely to cause problems in (b). In the Supplementary Information we provide additional considerations on this method, together with a description of the strategy used to improve the efficiency of the algorithm.

Appendix B. Optical architecture

Here we review the basics behind the operation of the optical architecture discussed in the main text. In section 3.2, we proposed to consider the square canonical layout [40] for several reasons, above all because it supports the implementation of any unitary transformation U, with no modifications at the hardware level. This is possible because any $m \times m$ U can be decomposed as $U = D\prod_b U_b$, i.e. the product of a diagonal matrix D (whose elements are complex and have modulo 1) and $m(m-1)/2$ unitaries U_b implementing complex rotations in the plane spanned by two adjacent modes:

$$U_b = \begin{pmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & u_{11} & u_{12} & \cdots & 0 \\ 0 & \cdots & u_{21} & u_{22} & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 1 \end{pmatrix}, \tag{13}$$

where the u_{ij} $(i,j \in \{1,2\})$ describe the transformation acting on the two modes. This transformation is implemented by means of a pair of phase shifter (θ_1) and imbalanced beam splitter, whose transformation U^{BS} is implemented as a Mach–Zehnder interferometer with tunable phase shifter (θ_2):

$$U^{\mathrm{BS}}(\theta_2) = \frac{1}{2}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}\begin{pmatrix} e^{i\theta_2} & 0 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix}. \tag{14}$$

Overall, the proposed ansatz can be implemented using well-established, integrated photonic components that only involve tunable phase shifters and fixed, balanced beam splitters (directional couplers).

ORCID iDs

Fulvio Flamini https://orcid.org/0000-0003-4999-2840
Thomas Müller https://orcid.org/0000-0003-1225-1483

References

[1] Jumper J M *et al* 2021 Highly accurate protein structure prediction with AlphaFold *Nature* **596** 583–9
[2] Venderley J, Khemani V and Kim E-A 2018 Machine learning out-of-equilibrium phases of matter *Phys. Rev. Lett.* **120** 257204
[3] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 Quantum machine learning *Nature* **549** 195–202
[4] Liu Y, Arunachalam S and Temme K 2021 A rigorous and robust quantum speed-up in supervised machine learning *Nat. Phys.* **17** 1013–7
[5] Cerezo M *et al* 2021 Variational quantum algorithms *Nat. Rev. Phys.* **3** 625–44
[6] Burkart N and Huber M F 2021 A survey on the explainability of supervised machine learning *J. Artif. Intell. Res.* **70** 245–317
[7] Heese R, Gerlach T, Mücke S, Müller S, Jakobs M and Piatkowski N 2023 Explaining quantum circuits with shapley values: towards explainable quantum machine learning (arXiv:2301.09138)
[8] Zheng P-L, Wang J-B, and Zhang Y 2022 A quantum neural network with efficient optimization and interpretability (arXiv:2211.05793)
[9] Briegel H J and De las Cuevas G 2012 Projective simulation for artificial intelligence *Sci. Rep.* **2** 400
[10] Mautner J, Makmal A, Manzano D, Tiersch M and Briegel H J 2015 Projective simulation for classical learning agents: a comprehensive investigation *New Gener. Comput.* **33** 69–114
[11] Flamini F, Hamann A, Jerbi S, Trenkwalder L M, Poulsen Nautrup H and Briegel H J 2020 Photonic architecture for reinforcement learning *New J. Phys.* **22** 045002
[12] Friis N, Melnikov A A, Kirchmair G and Briegel H J 2015 Coherent controlization using superconducting qubits *Sci. Rep.* **5** 18036
[13] Sriarunothai T, Wölk S, Giri G S, Friis N, Dunjko V, Briegel H J and Wunderlich C 2018 Speeding-up the decision making of a learning agent using an ion trap quantum processor *Quantum Sci. Technol.* **4** 015014
[14] Lamata L 2017 Basic protocols in quantum reinforcement learning with superconducting circuits *Sci. Rep.* **7** 1609
[15] Cárdenas-López F A, Lamata L, Retamal J C and Solano E 2018 Multiqubit and multilevel quantum reinforcement learning with quantum technologies *PLoS One* **13** 1–25
[16] Wei S *et al* 2022 Towards real-world quantum networks: a review *Laser Photon. Rev.* **16** 2100219
[17] Code available at https://github.com/MariusKrumm/PhotonicsPS.
[18] López-Incera A, Ried K, Müller T and Briegel H J 2020 Development of swarm behavior in artificial learning agents that adapt to different foraging environments *PLoS One* **15** 1–38
[19] Hangl S, Ugur E, Szedmak S and Piater J 2016 Robotic playing for hierarchical complex skill learning 2799–804
[20] 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 *PNAS* **115** 1221
[21] Wallnöfer J, Melnikov A A, Dür W and Briegel H J 2020 Machine learning for long-distance quantum communication *PRX Quantum* **1** 01030
[22] Poulsen Nautrup H, Delfosse N, Dunjko V, Briegel H J and Friis N 2019 Optimizing quantum error correction codes with reinforcement learning *Quantum* **3** 215
[23] Despite the apparent similarity of their representation, the operation of projective simulation (PS), based on the random walk of a single excitation, is in stark contrast with that of neural networks: (i) in neural networks one has multiple excitations of various magnitude adding up to produce an output; (ii) the clip network has arbitrary and dynamic connectivity and (iii) clips carry semantics, making PS a genuinely interpretable model
[24] Melnikov A A, Makmal A, Dunjko V and Briegel H J 2017 Projective simulation with generalization *Sci. Rep.* **7** 14430

[25] Paparo G D, Dunjko V, Makmal A, Martin-Delgado M A and Briegel H J 2014 Quantum speedup for active learning agents *Phys. Rev.* X **4** 031002
[26] Dunjko V, Friis N and Briegel H J 2015 Quantum-enhanced deliberation of learning agents using trapped ions *New J. Phys.* **17** 023006
[27] Saggio V *et al* 2021 Experimental quantum speed-up in reinforcement learning agents *Nature* **591** 229–33
[28] Wagner V 2022 Agnosticism as settled indecision *Phil. Stud.* **179** 1–27
[29] Wang J, Sciarrino F, Laing A and Thompson M G 2020 Integrated photonic quantum technologies *Nat. Photon.* **14** 273–84
[30] Bogaerts W, Pérez D, Capmany J, Miller D A B, Poon J K S, Englund D R, Morichetti F and Melloni A 2020 Programmable photonic circuits *Nature* **586** 207–16
[31] Taballione C, Anguita M C, de Goede M, Venderbosch P, Kassenberg B, Snijders H, Kannan N, Smith D, Epping J P, van der Meer R, Pinkse P W H, v. d. Vlekkert H, Renema J J 2022 20-mode universal quantum photonic processor (arXiv:2203.01801)
[32] Arrazola J *et al* 2021 Quantum circuits with many photons on a programmable nanophotonic chip *Nature* **591** 54–60
[33] Shen Y *et al* 2017 Deep learning with coherent nanophotonic circuits *Nat. Photon.* **11** 441–6
[34] Steinbrecher G, Olson J, Englund D and Carolan J 2019 Quantum optical neural networks *npj Quantum Inf.* **5** 60
[35] Williamson I, Hughes T, Minkov M, Bartlett B, Pai S and Fan S 2020 Reprogrammable electro-optic nonlinear activation functions for optical neural networks *IEEE J. Sel. Top. Quantum Electron.* **26** 7700412
[36] Marquez B A, Guo Z, Morison H, Shekhar S, Chrostowski L, Prucnal P and Shastri B J 2021 Photonic pattern reconstruction enabled by on-chip online learning and inference *J. Phys. Photon.* **3** 024006
[37] Zhang H *et al* 2021 An optical neural chip for implementing complex-valued neural network *Nat. Commun.* **12** 457
[38] Xu X *et al* 2021 11 tops photonic convolutional accelerator for optical neural networks *Nature* **589** 44–51
[39] Lamata L 2021 Quantum reinforcement learning with quantum photonics *Photonics* **8** 33
[40] Clements W R, Humphreys P C, Metcalf B J, Kolthammer W S and Walmsley I A 2016 Optimal design for universal multiport interferometers *Optica* **3** 1460–5
[41] Reck M, Zeilinger A, Bernstein H J and Bertani P 1994 Experimental realization of any discrete unitary operator *Phys. Rev. Lett.* **73** 58–61
[42] Lin L-J 1992 Self-improving reactive agents based on reinforcement learning, planning and teaching *Mach. Learn.* **8** 293–321
[43] Mnih V *et al* 2015 Ostrovski, human-level control through deep reinforcement learning *Nature* **518** 529–33
[44] Sun H and Guyon I 2023 Modularity in deep learning: a survey *Science and Information Conf.* (Springer) pp 561–95
[45] Amer M and Maul T 2019 A review of modularization techniques in artificial neural networks *Artif. Intell. Rev.* **52** 527–61
[46] Valiant L G 1979 The complexity of computing the permanent *Theor. Comput. Sci.* **8** 189–201
[47] Wu J, Liu Y, Zhang B, Jin X, Wang Y, Wang H and Yang X 2018 A benchmark test of boson sampling on Tianhe-2 supercomputer *Nat. Sci. Rev.* **5** 715–20
[48] Brod D J, Galvão E F, Crespi A, Osellame R, Spagnolo N and Sciarrino F 2019 Photonic implementation of boson sampling: a review *Adv. Photon.* **1** 034001
[49] Bozinovski S 2020 Reminder of the first paper on transfer learning in neural networks, 1976 *Informatica (Slovenia)* **44** 291–302
[50] Eva B, Ried K, Müller T and Briegel H J 2022 How a minimal learning agent can infer the existence of unobserved variables in a complex environment *Minds Mach.* 33 185–219
[51] Gibaut W, Pereira L, Grassiotto F, Osorio A, Gadioli E, Munoz A, Gomes S and Santos C d 2023 Neurosymbolic AI and its taxonomy: a survey (arXiv:2305.08876)