A reinforcement learning approach to the design of quantum chains for optimal energy and state transfer

IOP Publishing Mach. Learn.: Sci. Technol. 6 (2025) 015012 https://doi.org/10.1088/2632-2153/ada71d

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
23 March 2024

REVISED
15 December 2024

ACCEPTED FOR PUBLICATION
7 January 2025

PUBLISHED
22 January 2025

PAPER

A reinforcement learning approach to the design of quantum chains for optimal energy and state transfer

S Sgroi[1,2,*], G Zicari[1], A Imparato[3,4] and M Paternostro[1,2]

[1] Centre for Quantum Materials and Technologies, School of Mathematics and Physics, Queen's University Belfast, Belfast, BT7 1NN, United Kingdom
[2] Dipartimento di Fisica e Chimica—Emilio Segrè, Università degli Studi di Palermo, via Archirafi 36, I-90123 Palermo, Italy
[3] Department of Physics, University of Trieste, Strada Costiera 11, 34151 Trieste, Italy
[4] Trieste Section, Istituto Nazionale di Fisica Nucleare, 34127 Trieste, Italy
[*] Author to whom any correspondence should be addressed.

E-mail: sofiasgroi.w@gmail.com

Keywords: reinforcement learning, quantum chains, energy transfer, quantum networks

Abstract

We propose a bottom–up approach, based on reinforcement learning, to the design of a chain achieving efficient excitation-transfer performances. We assume distance-dependent interactions among particles arranged in a chain under tight-binding conditions. Starting from two particles and a localised excitation, we gradually increase the number of constitutents of the system so as to improve the transfer probability. We formulate the problem of finding the optimal locations and numbers of particles as a Markov decision process: we use proximal policy optimization to find the optimal chain-building policies and the optimal chain configurations under different scenarios. We consider both the case in which the target is a sink connected to the end of the chain and the case in which the target is the right-most particle in the chain. We address the problem of disorder in the chain induced by particle positioning errors. We apply our methodology to a simplified model of a relevant physical platform, consisting of trapped ions. We are able to achieve extremely high excitation transfer in all cases, with different chain configurations and properties depending on the specific conditions.

1. Introduction

Studying and optimizing energy, information or state transfer across physical systems are problems of great importance in a multitude of contexts in physics: the field of quantum physics, and quantum technologies in particular, is certainly not an exception. Among the various quantum systems whose transport properties are of special interest, particle chains play a special role for quantum technologies, both for their relative simplicity and for their wide range of applicability. Quantum communications [1, 2] and quantum internet [3] would obviously benefit from a better understanding of transport properties of particle chains together with better tools to design optimal state transfer across them. This would also be beneficial for quantum computing as, for example, particle chains can be used to describe spin-like systems which might be useful to connect distinct quantum processors and registers [4–8]. Implementation of quantum networks simulators [9–12] might in principle also be realised by optimized particle chains with long-range interactions. Even our understanding of biological photosynthetic processes [13–15] might benefit from the study and optimization of quantum transport among relatively simple structures. It is not surprising, then, that given the relevance of the problem, various techniques have been developed to realize transport across these structures [4].

Designing optimal couplings among the particles in such chains would allows us to avoid or minimize the control we have to exert during the system dynamics to achieve effective transfer. While arbitrary couplings engineering between particles in a chain can be a difficult task to accomplish for generic physical systems, some platforms where the couplings can be distance-dependent, such as ion traps [16], could allow some degree of control over their design while, at the same time, potentially avoiding the need to have different kind of physical platforms for processors [17–21] and busses when performing quantum computation.

One possible way to optimizing such couplings is by making use of reinforcement learning (RL) [22]. Machine Learning techniques have been extensively used in recent years to solve different physical problems with great success [23], even in the quantum realm [24]. In particular, RL has been proved especially useful in the context of quantum control, in some cases even clearly outperforming most commonly used optimal control algorithms [25]. RL has also been applied to realise fast transfer across particle chains via magnetic fields control [26]. However, its potential for optimal quantum system design remains largely unexplored.

In this work, we use a RL approach to design optimal particle chains for excitation transfer when the particles interactions depend on their relative position in space. We consider one excitation at maximum in the chain to ease numerical simulations, and we assume dipole-dipole interactions between particles. However, the approach can be easily extended beyond these conditions, and can, in principle, be directly used in an experimental setting without the need of simulating the system dynamics. In particular, we deploy a spatial approach to find the optimal chain design instead of considering arbitrary couplings, making it closer to realistic physical problems. We consider the chain as a fully connected quantum network with distance dependent couplings, hence we do not resort to nearest neighbours interactions or other approximations. Furthermore, we allow for a variable number of particles in the chain (instead of fixing it beforehand), while encouraging the use of less nodes if possible. We discretize the space between the starting particle and the target in a certain number of cells. First we study extensively the simpler case where there can be maximum one particle in each cell, highlighting various cases of interests and finding effective solutions. Then, we specifically consider a simplified model of the more physically relevant platform of trapped ions described in [16], by allowing the chain to have more than one particle in each cell and changing the Hamiltonian accordingly.

Our approach offers multiple advantages compared to most analytical or numerical optimization approaches. The RL approach we have deployed allows us to perform the optimization without fixing the number of particles and it is readily suited for the discrete optimization considered and to solve the problem in the presence of disorder. Moreover it allows us to find an agent's policy that adapts to different disordered configurations, instead of a single solution that only maximize the average performance.

The paper is organized as follows. In section 2 we describe the system of interest and introduce physical problem. In particular, we propose a spatial, bottom-up approach to build a particle chain for optimal excitation transfer, which—in order to address it with RL—we formulate as a decision process. In section 3, we briefly introduce the RL framework and we present our RL approach for optimal chain design. In section 4 we show our numerical study on the effectiveness of such RL approach, along with our optimal chain solutions under different conditions. In particular, we consider the scenario where we are not interested in coherence preservation in section 4.1; we then study the effects of errors and disorder in section 4.2, where we also introduce a possible adaptation of the original technique to further improve the transfer in this case. We address the problem of excitation transfer without coherence loss in section 4.3. In section section 5, we apply our approach to trapped ions, specifically considering the platform in [16]. In section 6, we discuss the geometrical properties of the solutions. Our conclusions are drawn in section 7, together with future outlooks.

2. Physical problem

Let us consider a system of two particles, A and B, coupled through the Hamiltonian

$$\hat{H}_{AB} = \Delta E \sum_{i=A,B} |i\rangle\langle i| + V_{AB}\left(|A\rangle\langle B| + |A\rangle\langle B|\right) \tag{1}$$

with $|i\rangle$ identifying a state where particle $i = A, B$ is in the excited state. We have assumed that the coupling originates from a dipole–dipole-like interaction whose strength scales with the (dimensionless) distance d_{AB} between them as

$$V_{AB} = J/d_{AB}^3, \tag{2}$$

where the coupling constant J is written in units such that V_{AB} has the dimensions of an energy. This implies that d_{AB} is rescaled by a typical distance dictated by the specifics of the implementation of the chain at hand. Note that we assume that both particles have the same energy ΔE and, for the sake of simplicity, we focus on the case in which one excitation at a time is allowed in the whole system.

Let us suppose that A is prepared in the excited state, while B is initially in its ground state. The unitary evolution of the system is governed by equation (1), while we allow for the incoherent transfer of the excitation from B to a sink S through an incoherent damping mechanism. Such evolution is described by a

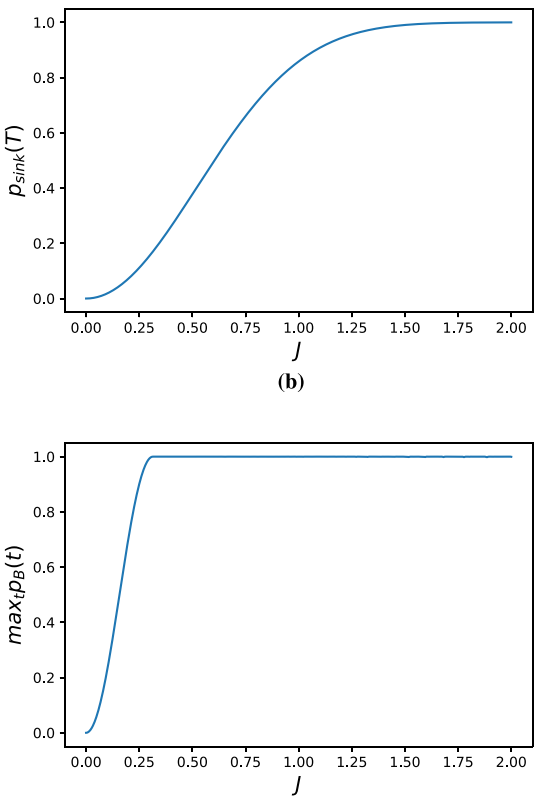

Figure 1. (a) Instance of final excitation transferred to the sink [(b) maximum excitation transferred to B] within the evolution time T, against the coupling constant J (in units of ΔE) in the case of two particles when the evolution is described by equation (1). Here $T = 5/\Delta E$, $d_{AB} = 1$ and $\Gamma_{\text{sink}}/\Delta E = 5$.

master equation in the Lindblad form

$$\dot{\hat{\rho}} = -i\left[\hat{H}_{AB}, \hat{\rho}\right] + D\left[\hat{L}\right]\hat{\rho}\,, \tag{3}$$

where the first term accounts for the unitary evolution while the second involves the dissipator accounting for the incoherent process. The latter is given by $D[\hat{L}]\hat{\rho} \equiv \hat{L}\hat{\rho}\hat{L}^{\dagger} - \left\{\hat{L}^{\dagger}\hat{L}, \hat{\rho}\right\}/2$, with the Lindblad operator $\hat{L} \equiv \hat{L}_{\text{sink}} = \sqrt{\Gamma}|S\rangle\langle B|$ and the damping rate Γ.

Given the distance d_{AB}, the amount of excitation $p_{\text{sink}}(T)$ transferred from A to the sink within a given interval of time T will be determined by the coupling constant J: the stronger the coupling between the two particles, the higher will be the population transferred to the sink. A similar behaviour is observed either when one extends the evolution time T while keeping J and d_{AB} constant, or decreases the distance d_{AB} while having T and J fixed. Alternatively, one can consider a second scenario where we are simply interested in the population transfer from A to B. In such a case we do not need to include the sink S, therefore the system's dynamics is fully unitary, and in equation (3) only the fist term on the right-hand side appears. In this case there is a coherent excitation exchange between the two sites, which results in revivals. Regardless of the formulation being chosen, it is natural to cast the problem in terms of maximum population transferred to B within the time interval T, i.e. $\max_{t\in[0,T]} p_B(t)$. With such a formulation of the problem, $\max_{t\in[0,T]} p_B(t)$ showcases a monotonic behavior against J [cf figure 1]. However, estimating $\max_{t\in[0,T]} p_B(t)$ is computationally more demanding than calculating a sink population at the end of the evolution, as it requires to track the entire dynamics. Therefore, for the sake of simplicity, we will focus on the first scenario for most of this study, while the second scenario will be addressed in section 4.3.

In both cases, given A and B, our goal is to enhance the mutual transfer of excitations by designing a suitable particle chain. For simplicity, we assume the particles to have all the same ΔE. Hence, the system Hamiltonian is a straightforward generalisation of equation (1), i.e.

$$\hat{H} = \Delta E \sum_i |i\rangle\langle i| + \sum_{j\neq i} V_{ij}\left(|i\rangle\langle j| + |j\rangle\langle i|\right), \tag{4}$$

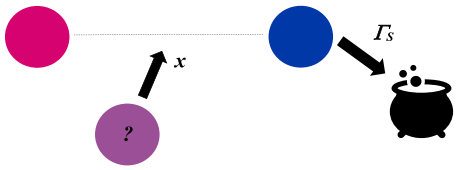

Figure 2. Sketch of the physical problem: two mutually interacting particles, A and B, undergo excitation-transfer processes from A, initially prepared in its excited state, to B. The latter is also incoherently coupled to a sink. We consider this as a starting configuration for the building up of a network: one by one, we add particles between A and B and seek for the potential increase of the excitation-transfer efficiency. The inter-particle interactions depends on their relative positions.

where the sum runs over all the possible particles of the chain, while the hopping potential is

$$V_{ij} = \frac{J}{|x_i - x_j|^3}. \tag{5}$$

The coupling constant J is assumed to be equal across the chain, while x_i and x_j denote the positions of the ith and the jth particle, respectively. The Hamiltonian in equation (4), which straightforwardly generalizes the model in equation (1), represents a tight binding-model where the states $\{|i\rangle\}$ are associated with some spatial degrees of freedom, i.e. the system is found in the state $|i\rangle$ when the excitation is located in the particle in site i while ΔE is the site energy resulting from the presence of such excitation. We stress that, in our model, all particles interact with each other, making the chain a fully connected quantum network.

Designing an optimal chain is equivalent to finding the best number and relative positions of its elements. To achieve this goal, we propose a bottom–up approach: we start with a chain composed of A and B only. Then, for a certain number of steps, we decide if and where to add individual sites to the chain and see how the population transfer is improved [cf figure 2]. We are interested in the cumulative improvement, i.e. in the final or maximal population transfer accomplished at the very end of the building process.

Having formulated the problem of chain design as a decision process, the next natural step is the search of its optimal working point through RL.

3. RL approach

RL problems are characterized by an agent observing and interacting with its environment while being assigned with a specific task. The performance of the agent with respect to the given task must be expressible via a numerical feedback, called *reward*, received as results of its interactions with the environment. The purpose of the agent is to learn how to interact optimally with the environment by trial and error, trying to maximize its (long term) reward.

Such agent-environment, interactions-feedback process can be formalized as a Markov decision process (MDP) [22]: at each interaction step, the agent observe the state of the environment S_i and performs an action A_i based on the current observation. As a result, the environment state is changed (the next observation will be S_{i+1}) and the agent receives a reward R_{i+1} [cf figure 3(a)]. Here we consider an episodic task in which the environment is reset after a certain number of interactions, an episode, or after reaching a terminal state.

The behaviour of the agent can be expressed by the *agent's policy* $\pi(A_i = a|S_i = s)$, which is the probability of performing an action a at step i conditioned on the observation of state s. The agent's goal will be to find the policy $\pi_{\text{opt}}(A_i = a|S_i = s)$ that maximizes the *return function*

$$G_i = R_{i+1} + \gamma R_{i+2} + \gamma^2 R_{i+3} + \dots, \tag{6}$$

where the discount factor $\gamma \in [0, 1]$ ($\gamma = 1$ being included only in episodic tasks) expresses how much we want to weight immediate and long term rewards.

RL provides numerous ways of approaching such problems. One possibility consists in parametrizing the policy $\pi_\theta(A_i = a|S_i = s)$, usually via a Neural Network, and optimizing the expectation value of G, or an estimate of it, with respect to the parameters θ, e.g. via gradient ascent. To perform such optimization, data needs to be gathered by observing the MDP. This can also be done in various ways and multiple algorithms have been proposed. In our work we used proximal policy optimization (PPO) [27], which is one of the most successful and widespread algorithms today.

Regardless of the particular algorithm being chosen, a RL approach to our problem requires the definition of the corresponding MDP, i.e. to define Observations, Actions and Rewards. This can be done

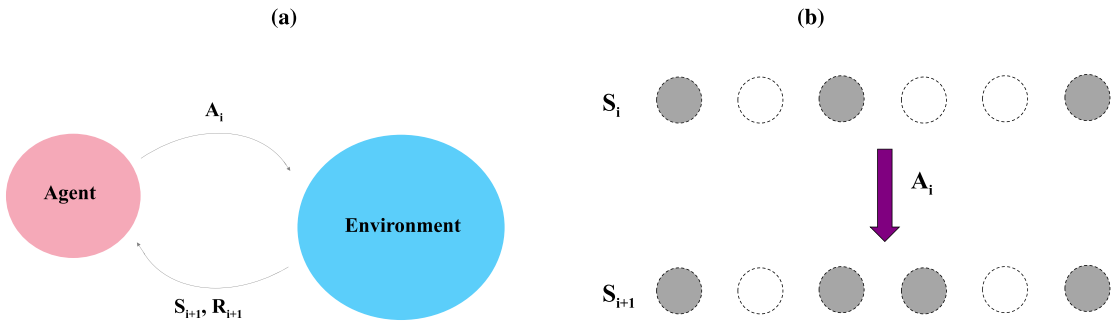

Figure 3. (a) Sketch of a Markov decision process. An Agent interacts with its environment with the purpose of performing a given task. At each interaction step, the Agent observes the state of the environment S_i and, based on this observation, performs an action A_i, which in turn changes the state of the environment and hence the next observation S_{i+1}. Based on its performance relative to the task, the agent receives feedback in the form of a reward R_{i+1}. (b) Agent's operations in the discretized-space configuration. The available spacial cells are illustrated by the dotted circles and the presence of a particle is represented by a filled area. Following observation S_i, the Agent's action is to decide which cell (if any) to populate at step $i+1$.

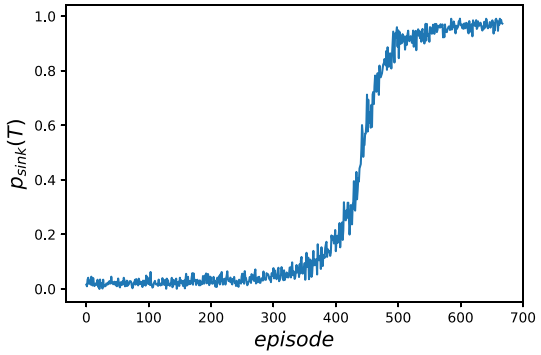

Figure 4. Example of learning process. Notice that $G_0 \approx p_{\text{sink}}(T)$ if $R_{i+1} = p_{\text{sink}}(T)^{i+1} - p_{\text{sink}}(T)^i$ and $\gamma \approx 1$. Here $d_{AB} = 1$, $T\Delta E = 5$, $\Gamma_{\text{sink}}/\Delta E = 5$, $J/\Delta E = 0.05$, while $\gamma/\Delta E = 0.99$.

straightforwardly in our case: at each step, the Observation will be the spatial configuration of the chain, i.e. the relative positions of the particles; the Action will be the absence/presence, and the position, of the next particle to be added to the system; the Reward will be the change in the sink population at time T, $p_{\text{sink}}(T)^{i+1} - p_{\text{sink}}(T)^i$. Alternatively, in the unitary case, the latter is given by the change in the maximum probability to occupy site B within the time interval T, i.e. $\max_{t\in[0,T]} p_B(t)^{i+1} - \max_{t\in[0,T]} p_B(t)^i$.

To describe the physical positions of the particles, we discretize the space between A and B by considering N_{cells} equally spaced cells, all of the same width, so that the state of the environment is described by a binary string of 0s and 1s describing the absence or presence of a particle in the respective cell. The Action will be the index of the next cell to populate with a particle, while no particle will be added if a cell is already populated [cf figure 3(b) for an illustration of the process]. Although such discretization is not strictly required, its use resulted in a better performance of the preliminary numerical experiments that we have performed and, in general, allows us to provide a simplified description of the approach we have taken.

We introduce an upper bound ν_{steps} to the number of steps before ending an episode. Needless to say, this sets a constraint to the maximum number of particles that could be allocated in the chain. We set a second strong constraint by imposing the end of an episode whenever the population being transferred from A to B exceeds 0.99. Both conditions serve the purpose of limiting the physical resources used to build the chain, and can be modified or removed altogether, should it be needed.

Figure 4 shows an example of learning curve for our problem, where it can be seen how the Agent performance, and thus the chain's ability to transfer the excitation, improves episode by episode as a result of the correspondingly improving Agent's policy. It is worth noticing that, iif no sources of errors, disorder or noise affect the dynamics of the chain, we often would not need to find a fine solution for the RL problem, as in this case we are not interested in the actual policy but in the best chain configuration. Such *optimum* will be found during the learning process, before an optimal policy is found.

4. Case studies and results

In this section, we present some numerical results and optimal solutions, i.e. optimal chain designs under various conditions. Unless otherwise specified, we have set $d_{AB} = 1$ (in arbitrary units of length), $T\Delta E = 5$, $\Gamma_{\text{sink}}/\Delta E = 5$, $J/\Delta E = 0.05$. For such a choice of physical parameters, which allow for an effective illustration of the performance of our protocol, the excitation transfer is close to zero when the system reduces to a chain only made of particles A and B. Quantitatively, we find $p_{\text{sink}}(T)^0 \approx 0.005$ when the sink is present, while $\max_t p_B(t)^0 \approx 0.06$ when the system dynamics if fully unitary. These results were obtained by discretizing the space in $N_{\text{cells}} = 21$ cells, and setting a maximum of $\nu_{\text{steps}} = 11$ steps. Details on the numerical simulations of the system evolution can be found in appendix A, PPO algorithm informations, hyperparameters and neural network architectures can be found in appendix B and a brief discussion on the challenges encountered in training the RL agents, along with our strategies to approach them, is reported in appendix C.

4.1. Optimal design

We start by considering the case where the target site is the sink and the Agent makes no errors in placing the particles in the desired locations.

As a first test, we applied the approach described in section 3 when $J = \Delta E$, for which the excitation transfer is already high, as it provides $p_{\text{sink}}(T) \approx 0.86$. With $N_{\text{cells}} = 11$, we find $p_{\text{sink}}(T) > 0.99$ after applying our RL approach. The optimum consists in placing a third particle right half-way between A and B, as it might have been guessed. We also noticed that filling all the cells with particles yields $p_{\text{sink}}(T) \approx 0.97$, making the RL solution more effective with noticeably less resources. For $J/\Delta E = 0.05$, the best configuration for $N_{\text{cells}} = 11$ turns out to be

$$S_{\text{opt}} = 10101010101, \tag{7}$$

where 0 and 1 stand for the absence or presence of a particle in a cell (including A and B). We thus obtain a configuration of equally spaced particles, which allows a population transfer to the sink of $p_{\text{sink}}(T) \approx 0.98$. However, these features are not general. For instance, by increasing the number of cells to $N_{\text{cells}} = 21$ allows us to achieve a population transfer of $p_{\text{sink}}(T) > 0.99$ through the asymmetric configuration

$$S_{\text{opt}} = 100000100010010010001. \tag{8}$$

It is interesting to notice that we only need to add 4 unevenly distributed particles to realize the ideal chain configuration, thus raising the performance of excitation transfer—in the given time—from negligible to nearly perfect. This result is even more noticeable when compared to the naive decision to fill all the available cells with particles, which corresponds to $p_{\text{sink}}(T) \approx 0.97$.

To test the generality of the approach, we performed the same optimization for a different system, for which

$$V_{ij} = \frac{J}{|x_i - x_j|^6}, \tag{9}$$

where again, $J/\Delta E = 0.05$. Once again, the agent was able to realize almost perfect transfer $p_{\text{sink}}(T) \approx 0.992$, with a chain configuration

$$S_{\text{opt}} = 100000001000010000001. \tag{10}$$

4.2. Addressing errors and disorder

So far we have consider the ideal case where no source of disorder is present. Such an assumption can be relaxed to address the effects of imperfections in the performance of our protocol. We thus consider the case in which the positioning of the particles across the chain is affected by static disorder. We can thus associate an uncertainty δx_j to the position of the jth element of a chain as determined by the action of the Agent, who will allocate the particle at position $x_j \pm \delta x_j$. We further assume that

$$\delta x_j < d/2\,, \tag{11}$$

where $d = d_{AB}/(N_{\text{cells}} - 1)$ is the distance between two adjacent spatial cells, so that we can still assign the particle to cell j (as the latter is the closest cell). This implies that the cells are still distinguishable (cf figure 5). These errors will affect the expected dynamics, but will not change the formulation of the MDP. The observation of the Agent can still be expressed as a string of dichotomic variables (being either 0 or 1).

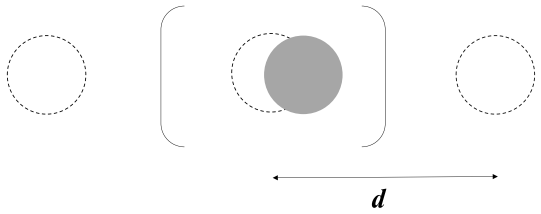

Figure 5. Error made by the agent in the positioning operation. The particle is not found exactly at the desired location. However we assume that the error is not large enough to make the assignation of the particle to a given spatial cell undecidable.

Table 1. From the left: error probability distribution, optimal chain found with the RL approach, average $p_{\text{sink}}(T)$ over 5000 random chain extractions using the corresponding optimal chain and error distribution, average $p_{\text{sink}}(T)$ over 5000 random chain extractions using the chain given in equation (8) perturbed by the corresponding error distribution and average $p_{\text{sink}}(T)$ over 100 random chain extractions when all the cells are filled, perturbed by the corresponding error distribution.

δx_j	Chain	$\langle p^{\text{opt}} \rangle$	$\langle p^{\delta x_j=0} \rangle$	$\langle p^{\text{filled}} \rangle$
$\mathcal{U}([0, d/20])$	Equation (8)	0.998	0.998	0.796
$\mathcal{U}([0, d/8])$	Equation (12)	0.995	0.989	0.300
$\mathcal{U}([0, d/4])$	Equation (13)	0.988	0.943	0.045
$\mathcal{N}(0, d/10)$	Equation (12)	0.979	0.972	0.188

We first considered uniformly distributed errors, i.e. $\delta x_j \in \mathcal{U}([0, rd/2])$, with $0 < r < 1$, where $\mathcal{U}([\alpha, \beta])$ denotes the uniform distribution over the interval $[\alpha, \beta]$. For $r = 0.1$, the optimal chain configuration in equation (8) already yields $p_{\text{sink}}(T) > 0.99$. This is no longer true if we increase the maximum error. For $r = 0.25$, the optimal configuration is

$$S_{\text{opt}} = 10001000100010000 10001, \tag{12}$$

whose Hamming distance with the string in equation (8) is 6. For $r = 0.5$, we have

$$S_{\text{opt}} = 10000100001000010 0001, \tag{13}$$

which has an Hamming distance 5 with the string in equation (8) and 7 with the string in equation (12).

We also considered the case of normally distributed errors, i.e. $\delta x_j \in \mathcal{N}(0, \sigma)$, where $\mathcal{N}(\mu, \sigma)$ denotes the normal distribution with mean μ and standard deviation σ. Note that equation (11) sets a bound on the maximum extracted values, while we consider σ such that $5\sigma = d/2$. We found the optimal configuration to be given by equation (12). All results are reported in table 1, which show that, for different error distributions, we are always able to reach higher population transfers than those obtained by the mere application of equation (8). Interestingly, we gather numerical evidence that, as opposed to case of small or no errors, the best chain configurations for large error are given by equally spaced particles. Moreover, we found that applying either equation (12) or (13) to the case of low or no errors leads to worse results than those achieved through equation (8), making the equally spaced particles solutions characteristic of the moderate-high disorder scenario.

However, we have not leveraged the full potential of RL so far, as we were mostly interested in the optimal chain discovered through policy learning, rather than the agent's policy itself. This change of perspective leaves room for further improvement of the excitation transfer, provided that we are not interested in a single chain configuration that maximizes the average transfer. Alternatively, we might be interested in finding a way to optimize each chain configuration adaptively, taking into account the specific disorder without measuring the errors in the particle positioning. To this end, we can notice that, even if we do not measure the positions of the particles, we already gain some information on the errors made during the particles positioning operations. For the Agent to receive its reward, we need to measure at each step the target population $p_{\text{sink}}(T)$, which is itself implicitly affected by the positions of the particles in the cells. We replace the 1's in the string describing the environment state with a function of $p_{\text{sink}}(T)$ for the chain configurations obtained when adding the particles. Therefore, the Agent can take different actions depending on such values, which in turn depend on the current information available on the disorder. In particular, at each step, after adding a particle to the chain, we change the value of the corresponding cell in the string that represents the environment state from 0 to

$$\eta = \frac{1 + p_{\text{sink}}(T)}{2} \in [1/2, 1]. \tag{14}$$

This choice yields the desired information more effectively, as it is still well separated from the case in which a cell in empty.

We apply this approach to the case of $\delta x_j \in \mathcal{N}(0,\sigma)$, for which the previous method was less effective: we obtained $\langle p_{\text{sink}}(T)\rangle = 0.979$ over 5000 simulations. By contrast, using the whole Agent's policy learned with the new version of the environment state, we were able to visibly improve the excitation transfer, obtaining $\langle p_{\text{sink}}(T)\rangle = 0.987$ over 5000 simulations, with an average of 5 particles in the chain.

4.3. Unitary case

Our discussion has been hitherto focused only on the the case in which the excitation in the system is irreversibly transferred to a sink via spontaneous emission. This results into a monotonic behaviour of the target (i.e. sink) excitation over time, as shown in figure 6(a). Hence, it makes sense to frame the problem as the optimization of the target population at a time T. However, this scenario might be of limited interest for applications, especially for quantum communication purposes, as the irreversibility of the process causes the system to lose coherence. To circumvent the coherence loss, we can remove the sink and directly consider the population transferred to B. This choice results into a different time behaviour of the target excitation: it does not increase monotonically in time, as shown in figure 6(b). Therefore, instead of looking at the excitation transferred at a time T, it is better to optimize the maximal excitation transfer within a time interval T, i.e. $\max_{t\in[0,T]} p_B(t)$.

This requires to sample the dynamics at different instants of time. In our simulations we considered $n_T = 20$ equally spaced points over the time interval $[0, T]$ to calculate the maximum population transfer, hence the reward R_{i+1} at each step. Besides the change in the definition of the Agent's reward, the MDP is identical to the scenario where the target is the sink. For this case, we assume no errors are made in the particle positioning operation as in section 4.1. The optimal chain configuration found during learning is

$$S_{\text{opt}} = 100010111010111010001, \tag{15}$$

for which $\max_{t\in[0,T]} p_B(t) > 0.99$.

Notice that in this instance, despite the particles not being all equally spaced, the chain is symmetrical; furthermore, compared to the sink scenario, we need a larger number of particles to realize almost perfect excitation transfer. In this case, filling all the cells with particles yields $\max_{t\in[0,T]} p_B(t) \approx 0.3$. We also noticed that, using equation (15) for the sink-target case, we could still achieve high final excitation transfer (0.991 compared to the optimum 0.998). Conversely, using equation (8) in the no-sink scenario, we found the maximum excitation transfer to be low, i.e. $\max_{t\in[0,T]} p_B(t) \approx 0.5$.

5. Application to trapped ions

In this section, we apply our methodology to optimize state transport across distant trapped ions. We consider the platform described in [16], consisting of trapped Ca^+ ions, with trap separation of 54 μm. The latter will be the cell-to-cell distance in our discretised space. We aim to realise transport across 10 cells. As in [16], we allow multiple ions to reside in the same trap well (cell), and assume to have cooled down the system all the way to a regime where we have one phonon at most. This simplifies the analysis, as it allows us to decrease the effective system dimensions. Under such conditions, the cell-to-cell interaction is described by

$$V_{ij} = -0.5\hbar\Omega_{\text{C}}\frac{\sqrt{N_i N_j}}{|i-j|^3} \tag{16}$$

with $\Omega_{\text{C}} = 4.5\,\pi$kHz and where N_i (N_j) is the number of ions in the ith (jth) cell.

We can hence approximate the system Hamiltonian with equation (4), where now the index i (j) represents the whole cell with possibly multiple ions in it, instead of a single particle, and we set $\Delta E = 0$. In accordance with [16], the first peak in the average phonon number transferred from one ion to another at a distance of 54 μm is found after $T \approx 0.22$ ms, which we set as the maximum evolution time of the system. We initialize our system with one ion in the first cell and one in the 10th, we prepare it with a single excitation in the first cell and we use our RL approach to maximize $F = \max_{t\in[0,T]} f(t)$, where $f(t) = |\langle 10|e^{-\frac{i}{\hbar}\hat{H}t}|1\rangle|$. Since we do not want the agent to stop to maximum one particle in each cell, we now allow these to accumulate and we introduce a further action that brings the agent to a terminal state (in case it decides to interrupt the episode earlier i.e. not to use all the available ions). We compare the chain found by the RL agents with that in which all the ions are distributed between cell 1 and cell 10 and that in which all the ions are equally distributed among the cells. Results are shown in table 2. We can see that in all cases the RL agent is able to improve the transfer, although a sufficiently high number of ions is required to achieve efficient transfer. Notice that, $f(t)$ uniquely determines the average fidelity of state transfer over all possible qubit states [8] and

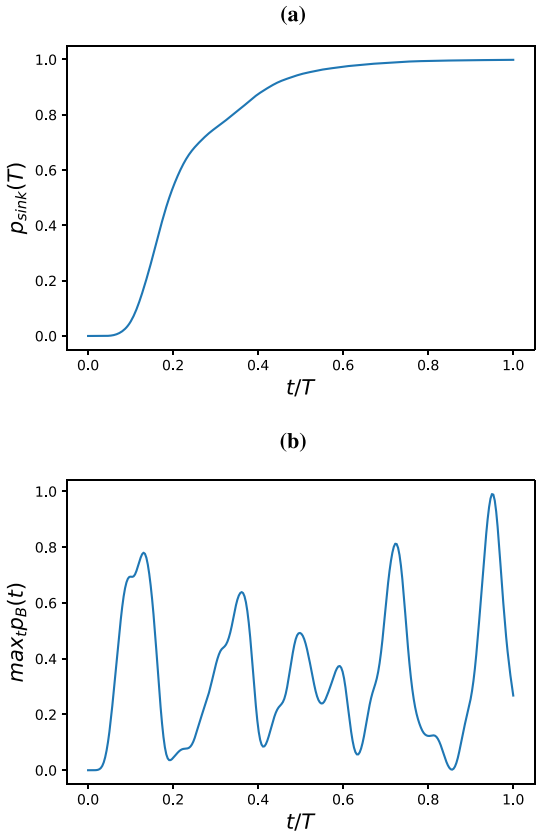

Figure 6. (a) Final excitation transferred to the sink and (b) maximum excitation transferred to B within the evolution time T as function of time when the particle chains are given by the optima found during the RL agent's learning (equations (8) and (15), respectively).

Table 2. Excitation transfer for increasing number of maximum ions. From the left, transfer with all ions equally distributed among the two ends, transfer with all ions equally distributed across the chain, optimal transfer found by the RL agent.

N_{ions}	Chain	F^{NC}	F^{FC}	F^{opt}
30	Equation (17)	0.0320	0.423	0.443
40	Equation (18)	0.0427	0.838	0.904
50	Equation (19)	0.0533	0.840	0.967

not just the energy transfer, making optimal chains potentially useful as quantum wires for state transfer across qubits in quantum communication or quantum computing as quantum busses. The optimal chains found are

$$S_{\text{opt}} = 2433333342, \tag{17}$$

$$S_{\text{opt}} = 2445555442, \tag{18}$$

$$S_{\text{opt}} = 2468548742, \tag{19}$$

for maximum total number of ions equal to 30, 40 and 50 respectively.

While we focused on a specific platform and we made a significant number of simplifications, we want to stress that the methodology presented here can be extended to different systems and can in principle be used with experimental data directly

6. Optimal chain properties

In this small section, we briefly discuss the geometric properties of the optimal configurations found and give a possible physical interpretation for the differences. First, we notice that the chain in equation (8) is the only asymmetrical chain in sections 4.1 and 4.2. This seems reasonable, since when the error in the location

of the particle within the cell is large enough, a symmetrical chain with equally spaced particles might be more robust. It is possible that in those cases the agent is trying to achieve high transfer while simultaneously maximizing the distance between particles, so that the relative error in the particle position within the cell is less relevant. It is also worth noticing that the chain in section 4.3 is also symmetrical, although the particles are not equally spaced. This might be due to the reversibility of the transformation and, in fact, a similar spatial symmetry is also observed in the chains in section 5, except for that in equation (19), which is only slightly asymmetrical. We conjecture that equation (19) is not perfectly symmetrical because the agent got stuck in a local optimum (which is common in RL as in most non-analytical optimization approaches and might be overcome by changing hyperparameters). Having assumed that the agent is not far from the global optimum, we tried to transfer a single ion between cells that break the symmetry. First we moved an ion from cell 8 to cell 6 with no success in improving the transfer. Then, we moved an ion from cell 5 to cell 3, obtaining

$$S_{\text{opt}} = 2478558742, \tag{20}$$

which slightly increased the value of F to 0.973.

7. Conclusions

We have developed a spatial, bottom-up, RL-based approach to the design of particle chains for optimal excitation transfer. We studied the effectiveness of such approach under different conditions. In particular, we considered two different scenarios, i.e. with or without a sink attached to the chain. In the former case, where we we are not interested in preserving coherence, we consider our target to be a sink where excitation is irreversibly transferred from the end of the chain. In the latter case, instead, since we want to avoid coherence loss, our target is the last particle of the chain. We also tested our approach in the presence of agent's errors, adapting our technique to minimize their effects when we build the chain.

We were able to achieve extremely high excitation transfer across particles in all scenarios, resulting into different particle chain design for each case. Our solutions exhibit some interesting properties. In particular, we found an optimal asymmetrical chain with a smaller number of particles in the sink scenario, while the optimal chain found in the no-sink case is symmetrical, presents a peculiar structure, and it is made of a larger number of particles. In the presence of moderate or large agent's errors (which we studied only in the sink case), we found that, to maximize the average excitation transfer, the optimal chains are composed of equally spaced particles, where the spacing dependent of the amount of errors. If we are willing to renounce to a single chain design for all disorder configurations from the error distribution, the full potential of RL can be deployed: we can adaptively build the optimal chain without making additional measurements, further improving the excitation transfer in this case. We applied our methodology to a platform consisting of trapped ions and we were able to optimize state transfer fidelity. Besides being attuned to experimental applications, this further demonstrates the capabilities of our approach to handle slightly more complex tasks than the one considered in the previous scenarios

Our approach presents multiple advantages compared to other techniques. In particular, the spatial dependency of the couplings makes it closer to realistic physical problems and allows us to go beyond some of the usual approximations (e.g. the nearest-neighbour approximation), without rendering the optimization problem extremely complex. Furthermore, when we rely on RL, we do not need to fix the number of particles beforehand, though we can limit such number, hence the resources used to build the optimal chain. Moreover, since the agent searches for an optimal policy and not necessarily a single optimum, it can learn to adapt to disorder introduced by errors or physical limitations and respond with different configurations, in contrast to analytical or numerical optimizations.

Finally, the methodology presented can easily be extended beyond the scenario considered here, as long as the Hamiltonian controlling the interaction between the particles depends on their relative position in space. In principle, one can introduce some changes without substantially affecting the formulation of the problem in terms of MDP. For instance, we could change the specific form of the interaction, we could allow the particles' local energies depend on their positions, we could add some environmental effects, or go beyond the one-excitation approximation. More complex scenarios can be then addressed with the same technique, as long as we are able to simulate the system dynamics or measure the amount of transfer. Its effectiveness for more complex cases is yet to be ascertained, but, given the widespread success of RL for complex tasks, it is reasonable to believe that this approach might still work. It would then be relevant to apply it to a more realistic model of a technologically relevant quantum system, maybe in a real experimental setting. It would also be interesting to extend the use or RL for quantum system design to solve different physical problems.

Data availability statement

The data that support the findings of this study are openly available at the following URL https://github.com/SofiaSgroi/A_Reinforcement_Learning_Approach_to_the_Design_of_Quantum_Chains_for_Optimal_Energy_Transfer.git.

Acknowledgments

AI gratefully acknowledges the hospitality of the Quantum Technology group, the Centre for Quantum Materials and Technologies, and the School of Mathematics and Physics, during his stay at Queen's University Belfast. MP acknowledges support from the Horizon Europe EIC-Pathfinder project QuCoM (101046973), the Leverhulme Trust Research Project Grant UltraQuTe (grant RGP-2018-266), the Royal Society Wolfson Fellowship (RSWF/R3/183013), the UK EPSRC (EP/T028424/1), and the Department for the Economy Northern Ireland under the US-Ireland R&D Partnership Programme.

Appendix A. Numerical simulations

In order to solve the system dynamics, we first vectorise equation (3) [28]. This transforms the density matrix as

$$\rho \to \vec{r} = (\rho_{00}, \rho_{01}, \ldots, \rho_{0K}, \rho_{10}, \ldots, \rho_{KK}) \tag{A1}$$

with $K = N+1$ in the presence of the sink, and $K = N$ in the unitary case. Here, N is the current number of particles in the chain. The unitary part of the master equation becomes

$$[H, \rho] \to \mathcal{L}_U \vec{r} \equiv \left(I \otimes H - H^T \otimes I\right) \vec{r}, \tag{A2}$$

while the dissipative one transforms as

$$\begin{aligned} &L\rho L^\dagger - \frac{1}{2}\left\{L^\dagger L, \rho\right\} \to \\ &\mathcal{L}_\mathcal{D}\vec{r} = \left[\left(L^\dagger\right)^T \otimes L - \frac{1}{2}\left(I \otimes L^\dagger L + \left(L^\dagger L\right)^T \otimes I\right)\right] \vec{r}. \end{aligned} \tag{A3}$$

By defining $\mathcal{L} = \mathcal{L}_\mathcal{D} + \mathcal{L}_\mathcal{U}$, we obtain $\dot{\vec{r}} = \mathcal{L}[\vec{r}(t)]$, hence the state of the system at a time t reads $\vec{r}(t) = \mathrm{e}^{-it\mathcal{L}}\,\vec{r}(0)$, where $\vec{r}(0) = (1, 0, \ldots, 0)$. To calculate the excitation transfer at a time t, we project $\vec{r}(t)$ into the target $r^t_{\text{target}} = \vec{r}(t) \cdot \vec{r}_{\text{target}}$. Note that $\vec{r}_{\text{target}}$ is either the $N+1$ dimensional vector $\vec{r}_S = (1, 0, \ldots, 0)$ in the sink case, or the N dimensional vector $\vec{r}_B = (1, 0, \ldots, 0)$ in the no-sink case. Then, in the former case, we simply have $p_{\text{sink}}(T) = r^T_{\text{target}}$ while in the latter $\max_{t\in[0,T]} p_B(t) \approx \max(\vec{r}(0), \ldots, r_B^{t_n}, r_B^{t_{n+1}}, \ldots, r_B^T)$, where we have divided the time interval $[0, T]$ in n_T equally spaced points, as explained in section 4.3.

We performed all our numerical calculations using Python, in particular the modules NumPy [29] and SciPy [30].

Appendix B. Algorithm hyperparameters and neural network architectures

Throughout this work we used the clipped PPO algorithm described in [27] with clipping parameter $\epsilon = 0.2$, 100 agents and 4 epochs of learning with minibatch size 128 in all cases except for the uniform error distributions in section 4.2, where the minibatch size was 64. The number of episodes considered was always $<$1500. We fixed the discount factor $\gamma = 0.99$ and the parameter $\lambda = 0.95$ for the generalized advantage estimation, while $\lambda = 0.98$ in section 4.3.

We used two separate neural networks for the Actor and the Critic. All hidden layers have a ReLu activation function and the output layer for the Critic has a linear activation function while the ouptut for the Actor is given by a softmax activation function. For the results of section 4 The Actor has 2 hidden layers of 128 neurons in all cases except for the adaptive error case described at the end of section 4.2, where the hidden layers are 3. The Critic has 2 layers of 64 Neurons in all no-error cases and 3 hidden layers of 128 neurons in all other cases except the adaptive case, where the hidden layers are 4. For the results in section 5, both the Actor and the Critic had 5 hidden layers with 128 neurons. The optimizations were performed using Adam [31] with different learning rates lr_A and lr_C for the Actor and the Critic, respectively (cf table B1 for the values being used in each section of the paper).

Table B1. Values of the learning rates used for Actor and Critic in the various case studies reported in the paper.

Section	lr_A	lr_C
Section 4.1	3×10^{-4}	5×10^{-4}
Section 4.2	1×10^{-4}	1×10^{-4}
Section 4.3	8×10^{-5}	1×10^{-4}
Section 5	3×10^{-4}	3×10^{-4}

All neural networks and their parameters optimization were implemented using Tensorflow [32] and Keras [33].

Appendix C. Training challenges

We now briefly discuss the challenges encountered in training our RL agents and how we approached them. Before perfecting the approach illustrated and used in the main body of the paper, we attempted to solve the problem presented in section 2 without discretizing the space between A and B and using a mixed discrete-continuous action space. In such approach, the actor outputted both the probability of adding/not adding a new particle to the chain and the parameters of a Gaussian defining the probability distribution of positioning the new particle at position $x \in]x_A, x_B[$. The performance was significantly lower than that of the method presented in this work, though, making it barely usable with our computational resources, even in the simplest scenario. This was probably due to the complexity of the decision process associated (despite the fact that the physical problem was the same).

In what follows, we describe the process of choosing the PPO and neural networks hyperparameters and the neural network architectures. We started from some of the examples in [27] to chose the PPO parameters and changed some of these parameters only when the transfer was not satisfactory. We did not follow any specific protocol to change such parameters. Unfortunately, to the best of our knowledge, there is currently no known and universally agreed-upon strategy to find the optimal hyperparameters or neural network architecture. We reduced the learning rates when we noticed that the return was rapidly reaching a local maximum and we increased the learning rate of the critic when the actor's policy was converging too fast compared to the critic loss. We progressively increased the number of neurons and layers in the networks when needed to improve performance and we maintained the architecture with the largest number of neuron and layers as a new starting point to address successive problems that we deemed more complex. In particular, our strategy was to start with networks of two layers of 64 neurons for the simplest scenario, increase the number of neurons to 128, and then increase the number of layers. This was sufficient for the problems considered in this work. However, changing the network architecture might prove beneficial when addressing problems with increased number of particles or when a linear array of particles is not sufficient to achieve the desired transfer and two or three dimensions must be considered. In those cases, we conjecture that a Convolutional Neural network might work better than a fully connected one. Training in the unitary case was significantly more time consuming (and hence difficult) than training in the presence of a sink due to the fact that, in the former case, we needed to simulate the whole dynamics within a time T. We did not find any effective strategy to overcome this issue. The attempt to include the optimal transfer time in the agent's action only hindered the performance.

ORCID iDs

S Sgroi https://orcid.org/0000-0002-4903-4020
G Zicari https://orcid.org/0000-0002-2526-9431
A Imparato https://orcid.org/0000-0002-7053-4732
M Paternostro https://orcid.org/0000-0001-8870-9134

References

[1] Gisin N and Thew R 2007 Quantum communication *Nat. Photon.* **1** 165–71
[2] Chen J 2021 Review on quantum communication and quantum computation *J. Phys.: Conf. Ser.* **1865** 022008
[3] Wehner S, Elkouss D and Hanson R 2018 Quantum internet: a vision for the road ahead *Science* **362** eaam9288
[4] Bose S 2007 Quantum communication through spin chain dynamics: an introductory overview *Contemp. Phys.* **48** 13–30
[5] Bayat A, Banchi L, Bose S and Verrucchi P 2011 Initializing an unmodulated spin chain to operate as a high-quality quantum data bus *Phys. Rev.* A **83** 062328
[6] Wang C, Li L, Gong J and Liu Y-X 2022 Arbitrary entangled state transfer via a topological qubit chain *Phys. Rev.* A **106** 052411
[7] Bose S 2003 Quantum communication through an unmodulated spin chain *Phys. Rev. Lett.* **91** 207901

[8] Wójcik A, uczak T, Kurzyński P, Grudka A, Gdala T and Bednarska M 2005 Unmodulated spin chains as universal quantum wires *Phys. Rev.* A **72** 034303
[9] Sameti M, Potočnik A, Browne D E, Wallraff A and Hartmann M J 2017 Superconducting quantum simulator for topological order and the toric code *Phys. Rev.* A **95** 042330
[10] Robens C, Alt W, Meschede D, Emary C and Alberti A 2015 Ideal negative measurements in quantum walks disprove theories based on classical trajectories *Phys. Rev.* X **5** 011003
[11] Qiang X, Loke T, Montanaro A, Aungskunsiri K, Zhou X, O'Brien J, Wang J and Matthews J 2015 Efficient quantum walk on a quantum processor *Nat. Commun.* **7** 11511
[12] Nokkala J, Arzani F, Galve F, Zambrini R, Maniscalco S, Piilo J, Treps N and Parigi V 2018 Reconfigurable optical implementation of quantum complex networks *New J. Phys.* **20** 053024
[13] Engel G S, Calhoun T R, Read E L, Ahn T-K, Mančal T, Cheng Y-C, Blankenship R E and Fleming G R 2007 Evidence for wavelike energy transfer through quantum coherence in photosynthetic systems *Nature* **446** 782–6
[14] Lee H, Cheng Y-C and Fleming G R 2007 Coherence dynamics in photosynthesis: protein protection of excitonic coherence *Science* **316** 1462–5
[15] Hou B and Krems R V 2021 Quantum transfer through small networks coupled to phonons: effects of topology versus phonons *Phys. Rev.* E **104** 045302
[16] Harlander M, Lechner R, Brownnutt M, Blatt R and Hänsel W 2011 Trapped-ion antennae for the transmission of quantum information *Nature* **471** 200–3
[17] Cirac J I and Zoller P 2000 A scalable quantum computer with ions in an array of microtraps *Nature* **404** 579–81
[18] Bruzewicz C D, Chiaverini J, McConnell R and Sage J M 2019 Trapped-ion quantum computing: progress and challenges *Appl. Phys. Rev.* **6** 021314
[19] Benhelm J, Kirchmair G, Roos C F and Blatt R 2008 Towards fault-tolerant quantum computing with trapped ions *Nat. Phys.* **4** 463–6
[20] Häffner H, Roos C F and Blatt R 2008 Quantum computing with trapped ions *Phys. Rep.* **469** 155–203
[21] García-Ripoll J J, Zoller P and Cirac J I 2005 Quantum information processing with cold atoms and trapped ions *J. Phys. B: At. Mol. Opt. Phys.* **38** S567
[22] Sutton R S and Barto A G 2018 *Reinforcement Learning: An Introduction* 2nd edn (The MIT Press)
[23] Carleo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L and Zdeborová L 2019 Machine learning and the physical sciences *Rev. Mod. Phys.* **91** 045002
[24] Krenn M, Landgraf J, Foesel T and Marquardt F 2023 Artificial intelligence and machine learning for quantum technologies *Phys. Rev.* A **107** 010101
[25] Zhang X-M, Wei Z, Asad R, Yang X-C and Wang X 2019 When does reinforcement learning stand out in quantum control? A comparative study on state preparation *npj Quantum Inf.* **5** 85
[26] Zhang X-M, Cui Z-W, Wang X and Yung M-H 2018 Automatic spin-chain learning to explore the quantum speed limit *Phys. Rev.* A **97** 052333
[27] Schulman J, Wolski F, Dhariwal P, Radford A and Klimov O 2017 Proximal policy optimization algorithms (arXiv:1707.06347 [cs.LG])
[28] Am-Shallem M, Levy A, Schaefer I, and Kosloff R 2015 Three approaches for representing Lindblad dynamics by a matrix-vector notation (arXiv:1510.08634 [quant-ph])
[29] Harris C R *et al* 2020 Array programming with NumPy *Nature* **585** 357–62
[30] Virtanen P *et al* SciPy 1.0 Contributors 2020 SciPy 1.0: fundamental algorithms for scientific computing in python *Nat. Methods* **17** 261–72
[31] Kingma D P and Ba J 2017 Adam: a method for stochastic optimization (arXiv: 1412.6980 [cs.LG])
[32] Abadi M et al 2015 TensorFlow: large-scale machine learning on heterogeneous systems software available from tensorflow.org (www.tensorflow.org/)
[33] Chollet F et al 2015 Keras (available at: https://keras.io)