Ivanova-Rohling et al. *EPJ Quantum Technology* (2025) 12:53
https://doi.org/10.1140/epjqt/s40507-025-00363-0

EPJ Quantum Technology
a SpringerOpen Journal

RESEARCH Open Access

Reinforcement learning approach for finding exchange-only gate sequences for CNOT with optimized gate time

Violeta N. Ivanova-Rohling[1,2*], Niklas Rohling[1,3] and Guido Burkard[1]

*Correspondence: violeta.ivanova-rohling@uni-konstanz.de
[1]Department of Physics, University of Konstanz, D-78457 Konstanz, Germany
[2]Zukunftskolleg, University of Konstanz, D-78457 Konstanz, Germany
Full list of author information is available at the end of the article

Abstract

Exchange-only quantum computation is a version of spin-based quantum computation that entirely avoids the difficulty of controlling individual spins by a magnetic field and instead functions by sequences of exchange pulses. The challenge for exchange-only quantum computation is to find short sequences that generate the required logical quantum gates. A reduction of the total gate time of such synthesized quantum gates can help to minimize the effects of decoherence and control errors during the gate operation and thus increase the total gate fidelity. We apply reinforcement learning to the optimization of exchange-gate sequences realizing the CNOT and CZ two-qubit gates which lend themselves to the construction of universal gate sets for quantum computation. We obtain a significant improvement regarding the total gate time compared to previously published results.

Keywords: Exchange-only qubits; Reinforcement learning; Optimal gate sequences

1 Introduction

Quantum computing has been a strongly growing field in the last years due to its potential to solve certain problems efficiently that are hard on a classical computer. The growth of the field is driven by advances in gate fidelity and the number of qubits for scalable quantum computing platforms. Among those platforms are superconducting qubits [1, 2], Rydberg atoms [3], trapped ions [4], and spin qubits in semiconductor quantum dots [5]. Specifically, spin qubits are promising with respect to scaling due to their small size and synergy with silicon-based technology and due to recent advances in gate fidelity [6–8]. In the original spin-qubit setting [9], each qubit is represented by the spin of an electron or hole trapped in a semiconductor quantum dot. Computations in such single-electron or single-hole spin qubits are based on controlling the exchange interaction between the spins and the local time-dependent magnetic fields acting on individual spins. A logical two-qubit universal gate, such as the controlled-NOT (CNOT) gate, is implemented by using exchange-interaction-based SWAP$^\alpha$ operations controlled by inter-dot voltage combined with magnetic-field-controlled single-qubit gates. These physical SWAP$^\alpha$ gates are the result of the exchange operation, where α is the normalized time parameter for which the exchange interaction is pulsed and the SWAP gate is switched on. This means that

α denotes the gate time in units of π/J where J is the strength of the exchange interactions when switched on. One of the challenges for quantum computing based on spins in quantum dots is the single-spin control that necessitates the modulation of a strongly non-homogeneous magnetic field on short time scales or the realization of a strongly inhomogeneous magnetic field using on-chip micro-magnets [10]. The necessity for this is completely avoided in an alternative approach which encodes one logical qubit using three physical spins. For this encoding, the exchange interaction is sufficient to implement universal quantum gates [11, 12], and thus the control of the local magnetic field is no longer necessary. This paradigm of quantum computation is referred to as spin *exchange-only* computation and has been subject to great experimental advances recently [13].

Various approaches to exchange-only computation exist, described in [14]. This physical platform has since been theoretically and experimentally investigated, and a large number of practical implementations of quantum dot systems for three-spin qubits have been developed, for more detail refer to [5].

We will consider the exchange-only computational model, described in Ref. [11], where each qubit is encoded using three physical spins (spin-$\frac{1}{2}$ particles), and where one- and two-qubit quantum gates on the logical qubits are implemented by sequences of SWAP^{α} gates (switching on and off the exchange interaction between pairs of spin particles) applied to the physical qubits. The exchange interaction can be completely switched off by a sufficiently large voltage barrier between the quantum dots (then the gate is off), and only through pulsing the voltage, the exchange interaction is switched on.

The cost of disposing of single-spin rotations in exchange-only quantum computation is twofold, (1) the necessity of an extended physical system, i.e., a larger number of physical spins to represent the qubit register, and (2) logical quantum gates that need to be synthesized by a sequence of several physical exchange operations, rather than a single application of exchange in the standard spin qubit paradigm. More specifically, in the case of two-qubit gates, several applications of the exchange interaction between at least five pairs of spins are involved in controlling the system of two logical qubits, see Fig. 1. A vital aspect for exchange-only computation to be practically relevant is thus to optimize the efficiency of the gates needed for quantum computation. This is where this paper provides a novel method that is shown to allow for a substantial improvement by applying reinforcement learning (RL). Efficiency can be looked at in several terms: minimizing the number of pulses, time steps, or total time necessary. In this paper, we will focus on minimizing the total gate time for a fixed value J of the exchange coupling when switched on. We will consider both the case of exchange gates applied in parallel when possible and exchange gates applied sequentially, which can be advantageous for avoiding cross-talk [13]. Here, *parallel* refers to the simultaneous application of commuting exchange pulses, i.e., for disjoint pairs of physical qubits, while *sequential* refers to the application of only one exchange interaction at a time. Minimizing the time needed for a desired gate is beneficial with respect to gate fidelity as noise acts on the system for a shorter time while the gate sequence is performed. For pulses of fixed duration with varied exchange strength, as in [13], the actual time needed for the sequence will depend only on the number of pulses (sequential) or number of time steps for pulses applied in parallel. However, we note that there is a lower limit to the gate time set by the maximum available value of J. Addition-

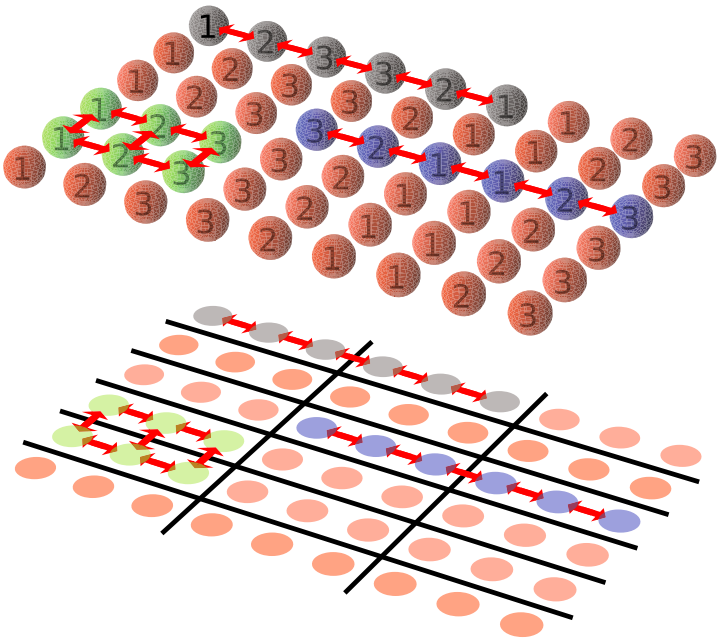

Figure 1 A possible two-dimensional arrangement of physical spins where each rectangle with three spins (labeled 1,2,3) defines one logical qubit. The definition of the logical qubit depends on the order of the physical qubit because each logical qubit, $a|0\rangle_l + b|1\rangle_l$ is defined for the total-spin-0 subspace as $|0\rangle_l = (|010\rangle - |100\rangle)/\sqrt{2}$, $|1\rangle_l = (2|001\rangle - |010\rangle - |100\rangle)/\sqrt{6}$, for the total-spin-1 subspace, see [16]. Here, $|ijk\rangle = |i\rangle_1|j\rangle_2|k\rangle_3$ where the numbers 1,2, and 3, correspond to the labels in the figure. We clearly see that the definition depends on the order of the physical qubits (spins), specifically the spins labeled 1 and 2 host a singlet in state $|0\rangle_l$ and a triplet in state $|1\rangle_l$. Further, note that the coupling of neighboring logical qubits depends on the arrangement and on the order of the physical qubits (within one row or within neighboring rows). This results in three distinct coupling scenarios between pairs of logical qubits (highlighted in gray, blue, and green) which we label "33" (gray), "11" (blue), and 2D (green) which are all considered in this paper. The exchange couplings that can be used in each scenario are indicated with red arrows

ally, minimizing the total time normalized to a fixed J [1] then corresponds to operating at smaller exchange coupling which can reduce charge noise [13].

Moreover, when arranging the physical qubits in a two-dimensional square lattice, different connections between neighboring logical qubits are present, see Fig. 1. Each of these different arrangements yields a distinct optimization problem. In quantum computing, a CNOT gate is universal when combined with single-qubit gates, which makes finding (efficient) exchange-only sequences to realize the CNOT gate an important problem. In [11], the first exchange-only universal gate set consisting of single-qubit rotations and a CNOT was presented. In [15], an exact specification of a universal logical gate-set using four spins to encode a single qubit was presented. The authors use extensive numerical optimization in order to obtain an optimized CNOT gate sequence with 27 parallel nearest-neighbor exchange interactions or 50 serial gates.

Different approaches have been utilized to find optimized sequences numerically [11, 15, 16]. The sequence for a CNOT found by Fong and Wandzura, via the use of genetic algorithms [16], see Fig. 2, is currently the most efficient known exact CNOT sequence for physical qubits connected via nearest-neighbor interactions in a linear chain archi-

[1] The quantity we refer to as normalized sequential time is termed *exchange angle* in [13].

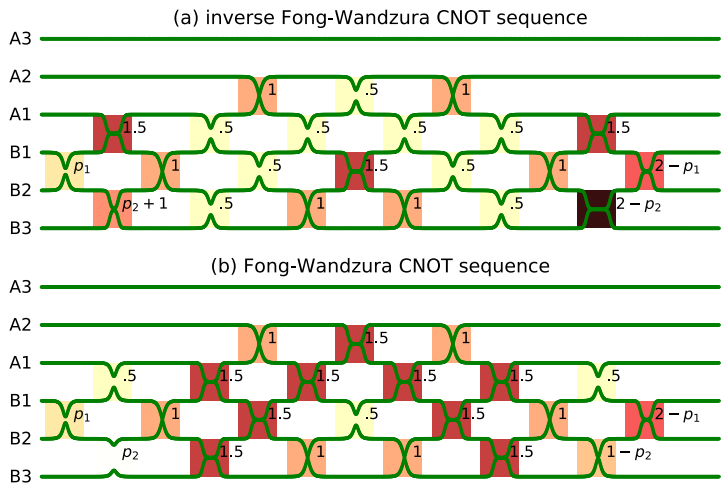

Figure 2 (a) Inverse FW sequence compared to (b) the FW sequence from Ref. [16]. Both sequences require twenty-two pulse and 13 time steps, while their parallel times are $T_p = 13.89\,\pi/J$ and $T_p = 15.89\,\pi/J$, respectively, and their sequential times are $T_s = 20$ and $T_s = 24$, respectively. Logical qubits A and B are arranged as shown. Numbers after A and B label the physical qubits. The SWAP powers α are displayed explicitly in the gates. $\alpha = 1$ corresponds to a full SWAP operation, up to a global phase. Here, we have introduced $p_1 = \arccos(-1/\sqrt{3})/\pi$ and $p_2 = \arcsin(1/3)/\pi$. The representation of the SWAP$^\alpha$ gates in this figure is inspired by the representation in [13]. The colors of the rectangular boxes at the exchange pulses are mapped to the values of α for each SWAP$^\alpha$ according to a color map having black for $\alpha = 2$ via red (around $\alpha = 1.5$), orange (around $\alpha = 1$), and yellow (around $\alpha = 0.5$) to white ($\alpha = 0$, i.e. no pulse)

tecture, see the area in Fig. 1 highlighted in blue. Importantly, even though this solution has been discovered numerically, it has a precise analytical description. In [17], gate sequences were found for a logical two-qubit gate locally equivalent to CNOT for various connectivities by applying exhaustive search under the condition that all exchange gates are $\sqrt{\text{SWAP}}$ or products thereof. Note that it is possible to search for gates from a set of gates that are locally equivalent to each other using the Makhlin invariants [18] for the reward function as done in Refs. [11, 17] which is harnessing the fact that two-qubit gates up to single-qubit operations can be expressed by three real parameters [19]. However, the application in actual quantum circuits requires exact and efficient gate implementations, and thus we follow here the work by Fong and Wanzura [16] using the distance to CNOT (or CZ) in the reward function. Aside from a large number of purely numerical approaches, it has been possible to come up with an analytic derivation of the optimal Fong-Wandzura (FW) CNOT sequence [20]. Furthermore, analytical considerations with regards to *leakage* were utilized in combination with numerics [21] to simplify the search problem and construct another set of gate sequences realizing the CNOT gate. Under certain assumptions, the solution presented in Ref. [21] is more efficient than the FW sequence if one considers total time as the efficiency criterion. Other efficient universal two-qubit gates have also been investigated, such as a gate locally equivalent to the CPHASE gate [22] that is potentially valuable in the currently available NISQ quantum devices. Leakage errors in exchange-only spin qubits can be approached by a reset-if-leaked procedure and, via numerical optimization, by a leakage correcting gate sequence [23].

Numerous advances in implementing quantum dot systems for three-spin qubits have been made [24–32]. Recently, Weinstein *et al.* [13] presented a two-qubit exchange-only system implemented using an array of six ^{28}Si/SiGe quantum dots to achieve universal

gates of very high operational fidelity. The fidelity of universal control of two encoded qubits was evaluated to be $96.3\% \pm 0.7\%$ for encoded CNOT operations, and even higher ($99.3\% \pm 0.5\%$) for encoded SWAP, demonstrating substantial progress towards achieving fault tolerance and computational acceleration with this approach. Below, we present our main results showing improvement in total time over previously published results for two different arrangements.

2 Results

2.1 Main results

The problem of optimizing gate sequences for two-qubit logical gates is high-dimensional. In our work, we use an intelligent optimization [33] approach enhanced by an RL algorithm, suitable for continuous search spaces. This allows us to explore a vastly larger search space by enforcing much fewer assumptions on the optimization problem in comparison to [17]. We aim to optimize the total time of the exchange-only gate sequences representing exact CNOT and exact CZ gates with varying connection topologies and find gate sequences for all three arrangements shown in Fig. 1. For the linear "11" arrangement highlighted in blue in Fig. 1, we find gate sequences representing CNOT gates. Notably one of the sequences we find, presented in Fig. 2 (a), has a shorter total time than the original FW sequence and the RL approach found it from scratch. We discuss the relation to other known gate sequences in Sect. 3. Furthermore, we discover a CNOT sequence in the "33" arrangement that is faster than the one implemented in Ref. [13], see Fig. 3.

Importantly, we demonstrate the usefulness of a reinforcement-learning-based approach for optimizing exchange-only sequences, which can be seamlessly extended to optimize different universal gates, and gate sequences with different architectures and different types of exchange interactions by simply redefining the cost function. The main aspects of the RL approach we use for gate sequence optimization are visualized in Fig. 4, and full details of the approach are given in Sect. 5, as well as in the Supplementary Information (Sec. A) and Refs. [34–36].

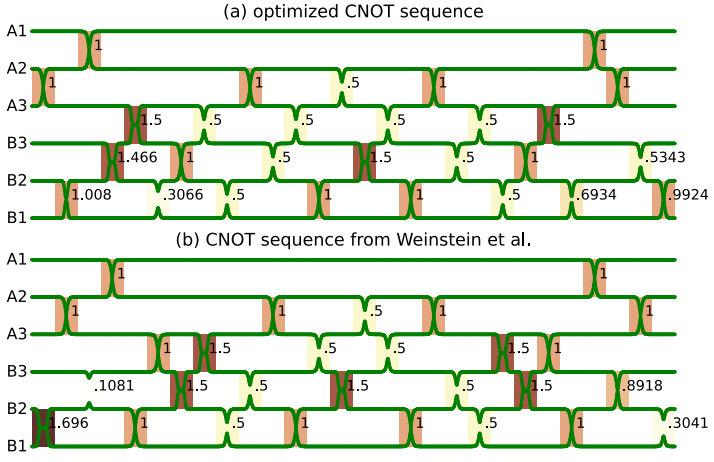

Figure 3 CNOT gate sequences for the linear arrangement with the singlet-triplet pairs on the outside (highlighted in gray in Fig. 1): (a) The RL-optimized sequence of sequential time $T_s = 24\pi/J$ and (b) the sequence from [13] of $T_s = 26\pi/J$. The representation of the SWAP$^\alpha$ pulses is the same as in Fig. 2

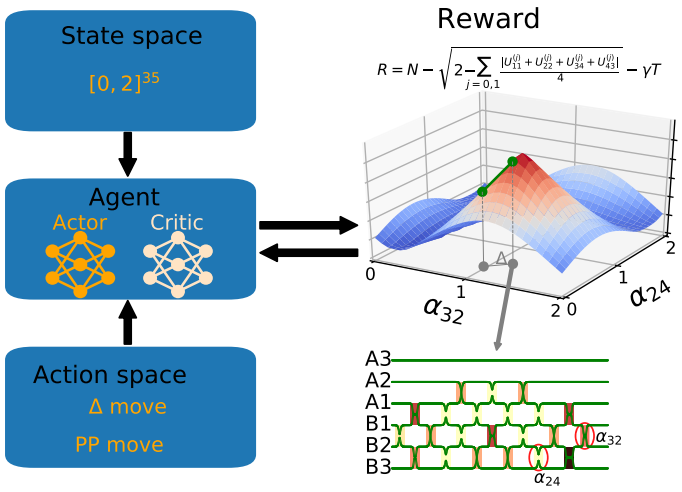

Figure 4 The RL algorithm used for CNOT gate sequence optimization. A deep deterministic policy gradient (DDPG) actor-critic algorithm was used as an agent. The state space is defined as 35 normalized time parameters $\boldsymbol{\alpha}_i \in [0, 2]$ of SWAP$^{\alpha_i}$ gates, operating on six spins A_k and B_k ($k = 1, 2, 3$) representing two logical qubits A and B, and arranged in the five-brick structure. Each state corresponds to a sequence of exchange pulses with a length of up to 14 time steps using parallelism. The number of pulses can be smaller than 35 when some of the $\boldsymbol{\alpha}_i$ are zero. The action space is defined as two possible actions – a Δ step, that potentially changes all parameters $\boldsymbol{\alpha}_i$, and a PP-move, i.e., a partial Powell local optimization step, where a local optimization with a fixed number of iterations is performed. The number of iterations is small, ranging from 0 (no local optimization is performed) to 12. The reward, see Eq. (2), is based on the distance to the CNOT gate used by Fong and Wandzura [16] and the total time, which can be either the time $T = T_p$ for applying the SWAP$^{\alpha_i}$ gates in parallel where possible, or the time $T = T_s$ for applying the SWAP$^{\alpha_i}$ gates sequentially. We plot the reward function here in dependence of two ($\boldsymbol{\alpha}_{24}$ and $\boldsymbol{\alpha}_{32}$) out of the 35 parameters. The representation of the SWAP$^{\alpha}$ pulses for the pulse sequence is the same as in Fig. 2

Additionally, we apply the RL algorithm to find optimized CZ gate sequences, see Sect. 2.2, CNOT sequences for a linear arrangement with the singlet-triplet qubit part on the edges ("33" arrangement, highlighted in gray in Fig. 1) and obtain a sequence beating the one actually implemented in [13] with respect to sequential total gate time, see Sect. 2.4, Sect. 5.5, and the Supplementary Information (Sec. B) for details. Furthermore, we search for optimized gate sequences for CNOT gates in the 2D arrangement of spins, highlighted in green in Fig. 1, where seven pairs of spins can be coupled by exchange interactions, see Sect. 2.3 and the Supplementary Information (Sec. C). We summarize the found optimized times T_p and T_s in comparison to values from the literature in Table 1.

2.2 CNOT and CZ gates in "11" arrangement

Multiple exact CNOT gate sequences of 14 time steps were discovered using the RL algorithm for gate sequence optimization. The search was done in a sequence of blocks of five pulses ordered in a brick-like manner, see Fig. 5 and Sect. 5.2 for details. Most of the discovered sequences comprise 14 time steps, but several, including the FW gate sequence and the improved FW gate sequences, required only 13 time steps. The total times T_s and T_p for performing the exchange pulses sequentially and in parallelized form, respectively, of some of the discovered CNOT gate sequences are plotted as a function of the used training steps in Fig. 6. We find that with an increasing number of training steps, the solution discovered by the algorithm improves. The best solution, discovered by the algorithm, has

Table 1 Comparison of parallel and sequential times, T_p and T_s, from the literature and those found by our approach. Note that (i) with deviations in the local gates at the beginning and end of the sequence, what we refer to as *inverse FW sequence*, was referred to as FW sequence in [13, 17] (ii) the sequence in [13] differs by the order of the SWAP operations, which translate between different arrangements, (iii) for the linear "33", we consider only the sequential time, and (iv) in [17], a sequence for the 2D arrangement for gates equivalent to CNOT was considered, as this does not provide directly T_p or T_s for the explicit CNOT we refer her to the times of the FW sequence as an upper bound for the time-optimal solution

Arrangement/gate	Literature		Our results	
	T_p in π/J	T_s in π/J	T_p in π/J	T_s in π/J
linear "11" CNOT	FW sequence [16]		inverse FW [(i)]	
	15.89	24	13.89	20
linear "11" CZ	Weinstein et al. [13][(ii)]		same/rediscovered	
	11.5	16	11.5	16
linear "33" CNOT	Weinstein et al. [13]		our solution	
	(iii)	26	(iii)	24
2D CNOT	bound by FW[(iv)]		our solution	
	$\leq$15.89	$\leq$24	16.01	20.63

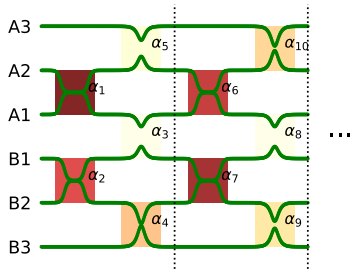

Figure 5 The five-brick structure used to define the observation space of the RL algorithm. The representation of the SWAP$^\alpha$ pulses is the same as in Fig. 2

shorter total sequential and parallel times than the original sequence published by Fong and Wandzura in [16].

In addition to the results for the CNOT gate, the RL algorithm also produced several exact CZ sequences of length 14 time steps, discovered by the RL algorithm, see Fig. 2 of the Supplementary Information. The dotted lines correspond to the total times required for the parallel and sequential operation of the CZ gate sequence described in [13]. The shortest sequence has total times $T_p = 11.5\,\pi/J$ and $T_s = 16.0\,\pi/J$, respectively, for parallel and sequential execution. This sequence (shown in Fig. 1 of the Supplementary Information) is equivalent to the CZ gate described in [13]. As the number of RL training steps increases, the corresponding best solutions discovered by the algorithm improve in efficiency. For details on the results for the CZ gate sequences, see the Supplementary Information (Sec. B).

2.3 CNOT gate for 2D arrangement

We also use the RL algorithm to optimize the CNOT gate with a different connecting topology, namely for a two-dimensional (2D) topology where each spin is exchange-coupled to a spin of the other qubit. The constraints of the 2D arrangement, highlighted in green in Fig. 1, lead to a modification of the 5-brick structure used in the FW optimization, yielding a seven-component structure as shown in Fig. 7. In each block, we first apply the exchange gates between the logical qubits A1 and A2 as well as B1 and B2 in parallel. Second, we apply the gates between A2 and A3 as well as between B2 and B3 in parallel.

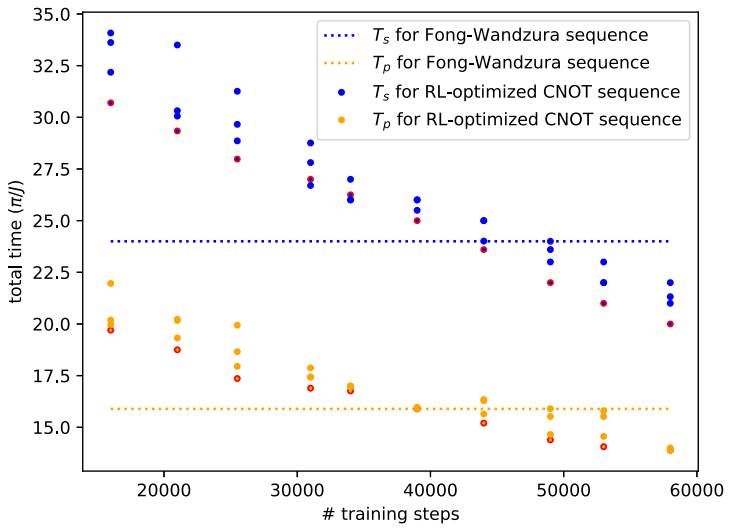

Figure 6 Improvement of the total sequential time $T = T_s$ (in blue) and total parallel time $T = T_p$ (in orange) of the CNOT gate sequences discovered by the RL algorithm for the linear "11" arrangement depending on the number of passed training steps (filled circles). The total gate time T in units of π/J, where J is the constant strength of the exchange interaction, is plotted as a function of the number of training steps. The solutions with optimal T_p for each investigated number of training steps are highlighted with a red circumference. Note that optimal T_s does not necessarily lead to optimal T_p. For comparison, we show T_p (T_s) of the FW gate sequence as an orange (blue) dotted horizontal line. The RL procedure is used as described in Sect. 5.4 with a reward described in Eq. (2). We find that after about 45,000 training steps, the sequences obtained from the algorithm trained using RL become more time-efficient than previously published sequences

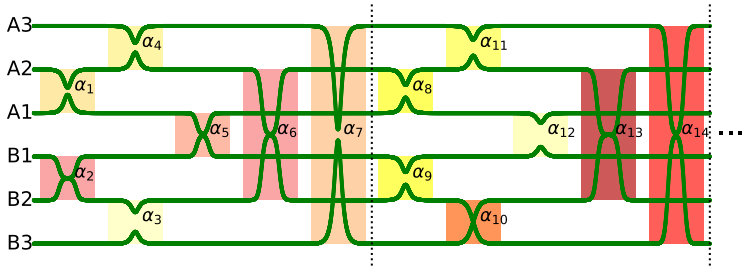

Figure 7 The seven-parameter block of quantum gates used for the physical qubits being arranged in a two-by-three rectangle with nearest-neighbor coupling. Note that the exchange gates between two spins where one belongs to logical qubit A and the other to logical qubit B (corresponding normalized gate times are $\alpha_5, \alpha_6, \alpha_7$ or $\alpha_{12}, \alpha_{13}, \alpha_{14}$) can be performed in parallel. The representation of the SWAP$^\alpha$ pulses is the same as in Fig. 2

Finally, we apply the interactions J_{AjBj} with $j = 1, 2, 3$ in parallel. We again discover many exact solutions of a length of seven five-brick blocks.

Again, the efficiency of the discovered gates depends on the number of training steps of the RL algorithm. We find a gate sequence for CNOT with total time $T_s = 20.63\,\pi/J$ ($T_p = 16.09\,\pi/J$), see Fig. 3 of the Supplementary Information. For this 2D topology, we could not find an improvement compared to the linear "11" topology.

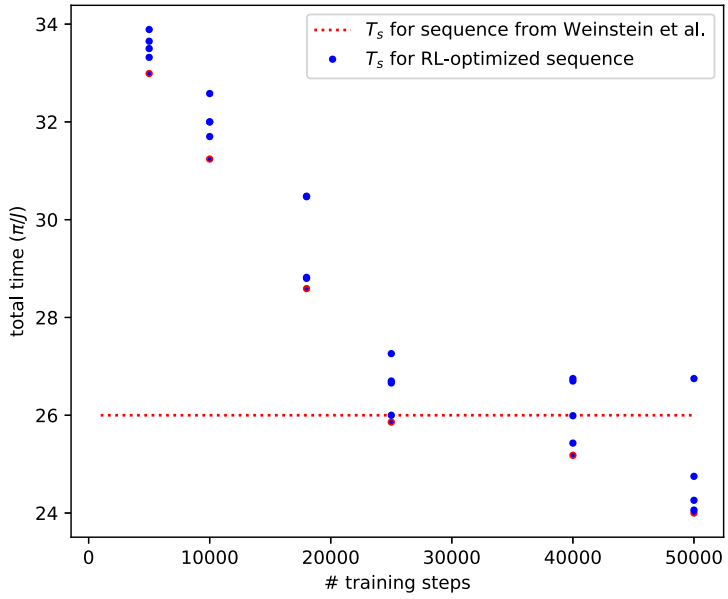

Figure 8 Total times of the resulting CNOT gate sequences with fixed ordered gates discovered by the RL algorithm and the sufficient number of training steps, needed to obtain them

2.4 CNOT gate in "33" arrangement

Finally, we also optimize a sequential CNOT gate sequence for the linear arrangement as in Ref. [13], with singlet-triplet pairs at the edges (the "33" arrangement shown in gray in Fig. 1). Under these constraints, we again discover multiple exact CNOT gates of various efficiencies. In this situation, we only evaluate the sequential total time T_s. The results are shown in Fig. 8. Importantly, with our RL approach, we rediscover the gate sequence implemented in [13]. This sequence has additional SWAP gates which are in some sense switching between two distinct linear orders (highlighted in blue and gray in Fig. 1). The total sequential time of the Weinstein CNOT sequence, see Fig. 3, is $26\pi/J$ assuming a constant exchange coupling J for each of the exchange gates in contrast to the actual implementation in [13], while the total time of the improved FW sequence, see Fig. 2a together with eight ordering SWAP gates is $28\pi/J$. However, this is an unfair comparison, as some gates at the beginning and the end are shifted relative to the ordering SWAP gates to transform them in a more efficient way. Most importantly, in addition, we discover a few solutions that are more efficient than the sequence in [13] with respect to total sequential time T_s. The best solution is shown in Fig. 3. This solution is identical to the Weinstein CNOT sequence with respect to the locally-equivalent part but with optimized local gates at the beginning and at the end of the sequence. The efficiency of the discovered CNOT gates again heavily depends on the training steps of the algorithm.

To be able to fairly compare to the sequence discussed in [13], we set specific constraints for the gate sequence – sequential execution of the physical gates and we use the same pairs spins coupled by exchange gates as in [13]. RL was used to optimize a CNOT sequence that is sequential and with order gates using the form from [13]. We achieved sequences with better total sequential time than the one in [13], see Fig. 3. More sequences found by the RL approach representing CNOT in the "11", "33", and 2D arrangement, as well as CZ in the "11" arrangement are available at [37].

3 Discussion

For the linear arrangement with singlet-triplet pairs at the inside of the chain ("11" arrangement), our RL approach finds the realization of an exact CNOT gate which improves previously published results regarding the total gate times T_s and T_p. For the arrangement with singlet-triplet pairs on the outside of a chain ("33" arrangement) as in [13] we found a sequence with reduced total sequential time T_s compared to the one implemented in [13]. These results demonstrate the power of RL applied to the optimization of quantum gates and quantum gate sequences.

We observed that some of the solutions for the CNOT gate sequence found by our RL approach are related to each other by symmetry operations. Those symmetries are presented in the Supplementary Information (Sec. D) in the form of mathematical lemmas. Importantly, explicitly implementing these symmetries in the future can boost the performance of the optimization strategy. These operations themselves are not difficult to understand and are implicitly used already in the literature at least partially, given that what we refer to as an 'inverse' FW sequence (or the non-local part of it) is also referred to as 'FW sequence' [17]. In general, the term 'FW sequence' refers to different sequences in the literature [13, 17, 22], see Fig. 2, which can be either an explicit CNOT sequence as in the original work by Fong and Wandzura [16] or locally equivalent [17] and which can be either using the same SWAP$^\alpha$ gates presented in [16] as in [22] or the inverse operations [13, 17]. Remarkably, our RL approach found both versions from scratch for the exact CNOT for the same linear arrangement of physical spin qubits as in [16], i.e., the chain highlighted in blue in Fig. 1, see Sect. 5 for the details of our optimization procedure. We note that the CNOT sequence presented here in Fig. 2 (a) requires a shorter gate time than the original FW sequence [16] and in contrast to the one presented in [17], it provides the full CNOT sequence rather than a sequence which is locally equivalent to CNOT. We further note that we did not impose any restrictions on the values of α for the SWAP$^\alpha$ gates in contrast to [17]. While a restriction to $\sqrt{\text{SWAP}}$ gates and products as made in [17] cannot yield an exact CNOT gate, it can, however, provide a gate sequence that is locally equivalent to the CNOT or CZ gate. The independence from such restrictions on the gate sequence demonstrates the flexibility of our approach. Regarding the more complex optimization problem for the 2D connectivity, we note that while the RL algorithm can tackle also this problem, it is challenging to obtain a solution comparable in total time to the most efficient sequence for linear connectivity. Specifically, the RL algorithm did not reproduce the best solution for the "11" arrangement and those sequences found for 2D are slightly longer in sequential total time.

4 Conclusions

We have shown that machine learning and intelligent optimization through RL are working approaches for finding optimal exchange-only gate sequences. Specifically, we have discovered optimized solutions for a variety of gates and connectivities. In this work, RL has demonstrated its flexibility and usefulness in optimizing the total times of exchange-only CZ and CNOT gate sequences. The results demonstrate that RL helps find such sequences and improves the total gate times of state-of-the-art solutions with fewer prior assumptions compared to other approaches. In the optimization problems considered in this work, we have used a brick (base) structure that encodes the commutation relations of the exchange coupling. By enforcing a fixed connectivity we have turned the problem

into a continuous optimization problem, for which efficient methods exist. We then use the RL as a tool for intelligent optimization that learns the appropriate starting points of a local optimizer. We find optimized solutions that are better or equivalent in terms of total times to known state-of-the-art solutions.

A limitation of our approach is the use of a fixed brick structure, which captures the commutation relations of operators but is not unique. Ideally, different brick structures could be used in optimization. For a more flexible approach, the symmetries that follow from the commutator relations can be encoded in an equivariant neural network, instead of using a fixed brick structure. This is meaningful also for other symmetries in the search space. Moreover, symmetries arising from the commutation relationships, as well as the other discovered symmetries, could be exploited by directly incorporating them in various ways in the optimization problem. As the approach is flexible, it allows for the investigation of different connection topologies in future work. Instead of optimizing the total gate time, the objectives could also be to minimize the number of exchange gates or the number of time steps. This might be particularly promising for gates other than CNOT and CZ. Additionally, one could extend the problem of optimal exchange-only gate sequences to a more realistic scenario where the gate fidelity is optimized in the e presence of state leakage and noise.

We note that the availability of high-performance hardware for quantum computing based on spin qubits in semiconductor quantum dots [38] increases the potential for experimental improvement on exchange-only qubits. The RL approach described here might be extended in future work in order to include real-life complications of the optimization problem described in this paper, including (i) longer sequences in terms of the number of pulses, (ii) the maximum value of the exchange interaction depending on the qubits involved, and (iii) non-commuting exchange interactions being switched on at the same time. The RL-based approach presented here is by no means limited to spin exchange-only qubits. In contrast, it can be broadly applied for finding sequences for various quantum computing hardware platforms or for optimizing compiling sequences of quantum gates.

5 Methods

5.1 Reinforcement learning for optimization problems

RL is a class of machine learning algorithms, where an agent interacts with an environment and gets back a reward based on its actions. The goal of the agent is to learn a behavior that optimizes the total reward obtained. RL that uses neural networks as agents to learn the optimal policy is referred to as *deep RL*. Recently, RL, and especially deep RL have been used with great success for numerous problems in various areas of physics, in general, [39], as well as quantum computing [40], in particular. The reward function that is maximized by RL varies depending on the specific problem, e.g., in [41] a quantity called recoverable quantum information was introduced in order to apply RL to quantum error correction, basically maximizing the ability to obtain a quantum state at the end of a long sequence of quantum gates and measurements. More recently, RL has been used to learn appropriate optimizers that solve difficult optimization problems, or to *learn to optimize*, examples include [42–47]. In the present work, we use RL to solve an optimization problem as well.

The RL approaches to optimization show advantages over automating and accelerating the optimization of complicated problems. Instead of manually crafting classical optimizers, one can parameterize and learn optimization rules in a data-driven fashion.

Yet another application of RL for optimization is to use the RL agent as a hybrid aspect of the optimizer to automatically guide the behavior of the optimizer in an intelligent way, suitable in particular for the problem at hand. This does not involve "learning" to optimize on a similar task prior to the optimization task, but using the machinery of RL, and the stored experiences during the optimization procedure (the experience replay [48]), to select the appropriate next steps in the optimization search. Based on the agent's prior experience and obtained reward, the next optimization behaviors are selected, instead of being encoded, such as selecting exploration vs. exploitation behaviors, or parameter values. Examples include [49, 50], where different global optimization heuristics were combined with a simple Q-learning approach to intelligently choose between exploitation or exploration behavior of the heuristic, as well as intelligently set other parameters of the optimization heuristics. These intelligent optimization approaches were tested on known hard mathematical functions as benchmarks and were found to outperform other state-of-the-art methods that were not enhanced by RL. In [51], a review of hybrid approaches for optimization that use RL as well as metaheuristics for combinatorial optimization is presented. In [52], a memetic particle swarm optimizer that uses RL to control optimization operations, related to choosing local search behavior and particle selection, was introduced. The method turned out to be successful on several benchmark optimization problems. In this work, we use a deep RL approach to intelligently guide an optimizer to better optimize a gate sequence. The RL agent, based on previous experience, recorded in an experience buffer, and on previous rewards from the environment, predicts the optimality of an action. In this case, the action is a behavior of the optimizer. The Deep Deterministic Policy Gradient (DDPG) is an actor-critic RL approach, which is suitable for continuous action spaces.

Namely, the actor is a neural network with parameters θ, which takes observation data S as an input and returns the action A that maximizes the long-term reward, based on its learned parameters. Each actor learns a *policy*. Often, the policy is a probability distribution of actions which the agent should take given a specific state; however, due to the deterministic nature of DDPG, the actor outputs a specific optimal action for a given state s.

The critic is a neural network with parameters ϕ, which, for given observation S and action A as inputs, returns the corresponding *expectation* of the return, or cumulative reward. In essence, the actor is choosing the action, and the critic is evaluating the action of the actor. In this specific case, the actor is learning the optimal action given a position in the search space – either a jump to another position in the search space or a partial local Powell optimisation. The critic is evaluating the action in expectation. The DDPG algorithm uses an experience replay buffer to store and learn from past experiences. Using both the actor network to predict the optimal action for a particular state and the critic network to predict how good the policy of the actor network is, allows the agent to learn effectively in continuous environments.

5.2 The five-brick structure

In the search for the reset-if-leaked sequence [23], a brick-like structure of repeated patterns of physical exchange gates (SWAP-α) was used. The brick structure is taking advantage of the commutation relation between the exchange interactions between the qubits in different subsystems. Two exchange interaction operators commute if the exchange interaction is applied to pairs of physical qubits that do not share any common spins. Then

the gate sequence is invariant under the interchange of the order of these operators. Here, since we are not trying to reproduce the FW gate sequence, but aim to improve it, we loosen the four-brick pattern structure to a general five-brick structure (Fig. 5) that allows all six physical qubits to be affected by the sequence, which in principle enables a generalized search for other, potentially better sequences.

5.3 Reward function for the CNOT-gate

In order to assess how well a gate sequence approximates a logical CNOT, the distance from CNOT is measured using the FW distance function introduced in [16],

$$d_{\mathrm{FW}}(U(\boldsymbol{\alpha})) = \left[2 - \frac{1}{4}\left|U_{11}^{(0)} + U_{22}^{(0)} + U_{34}^{(0)} + U_{43}^{(0)}\right| - \frac{1}{4}\left|U_{11}^{(1)} + U_{22}^{(1)} + U_{34}^{(1)} + U_{43}^{(1)}\right|\right]^{1/2}, \quad (1)$$

where $U^{(0/1)}$ is the $\boldsymbol{\alpha}$-dependent unitary matrix describing the overall gate sequence on the subspace for total spin zero or total spin one, respectively. Here, $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots)$ represents the list of exchange time parameters. The function d_{FW} is a distance measure between a unitary matrix and the desired CNOT gate, taking advantage of the fact that CNOT as the target gate comprises only ones and zeros as matrix elements in the computational basis. Furthermore, note that while the unitarity of $U^{(0/1)}$ is used, d_{FW} allows for different phase factors in the spin-0 and in the spin-1 subspaces. The reward is given by

$$R(\boldsymbol{\alpha}) = N - d_{\mathrm{FW}}(U) - \gamma T, \quad (2)$$

where T denotes the total gate time. The parameter γ is a penalty parameter that penalizes the total time needed for the gate sequence. N is a number added for technical reasons, as negative rewards do not perform well with the DDPG algorithm.

5.4 Reinforcement learning for gate sequence optimization

RL can use the memory hash that is built from the learning experience in order to achieve intelligent optimization. The reward feedback, provided from the environment in the RL setting can improve the optimizer's behavior, and instead of choosing parameters of the optimization heuristics manually, the RL machinery can be used to guide the optimizer parameters in high-reward areas, with the actor-critic used to learn to predict the behavior of the optimizer that will optimize the reward.

A visual representation of our RL approach is shown in Fig. 4, where the observation space consists of the possible values for the normalized times α_i for gate sequences of fixed length 35, the action space consists of two types of actions, a small change of the normalized times of the sequence, and a partial local optimization (the derivative-free Powell's method) of a fixed small number of iterations. The RL agent learns the best way to optimize the total time of the sequence in an actor-critic approach, where both the actor and the critic are neural networks. The reward obtained by the agent at each step is based on the sequence distance to the exact CNOT or CZ gate and the total time of the gate sequence. To optimize the gate sequence, given the difficulty of the problem, we utilize RL to learn strategies to optimize the sequence, instead of manually selecting and parametrizing an optimizer. We use the deep deterministic policy gradient (DDPG) [53] algorithm, which is an actor-critic algorithm [54] for RL with continuous state space, where the gate

sequence is assumed to be constructed by a sequence of repeating the five-brick structure of a fixed length of 35 pulses, and the state space consists of the values of the normalized times α_i of the different SWAP^{α_i} gates, $i = 0, \ldots, 34$. We use hybrid control, namely, the action space has both continuous as well as discrete components. The continuous components are values that change the normalized times α_i at a given step, while the discrete component determines the number of iterations of a partial derivative-free optimization (partial use of Powell's method, [55]). The number of possible iterations can be 0, which allows for the case where no derivative-free partial optimization takes place, and only the values of the normalized times are varied. By *partial optimization*, here we mean that we fix the number of optimization iterations without the necessity of a local optimum to be achieved. The goal is to learn a sequence of parameter values (starting points of the partial Powell algorithm) that result in the best gate sequence. As a reward, we use a function based on the FW measure for the distance from CNOT combined with a penalty for the total time, see Eq. (2).

5.5 Optimizing T_s for CNOT with linear "33" arrangement

We investigate the performance of our approach for optimizing the CNOT gate sequence, imposing the same constraints on qubit arrangement as in [13] in order to be able to compare the resulting gate sequences to the one used in Weinstein *et al.* [13]. We enforce connectivity constraints of the physical qubits so that the singlet-triplet part of the logical qubit is on the outside of the gate sequence chain. This yields the order of the linear-chain arrangement highlighted in gray in Fig. 1 which is A1, A2, A3, B3, B2, B1 (the "33" arrangement). This allows us to compare solutions discovered by our approach to the solution expressed in [13], where such additional requirement was imposed. We present the results in Sect. 2.

Abbreviations

CNOT, controlled-NOT; CPHASE, controlled-phase; CZ, controlled-Z; DDPG, deep deterministic policy gradient; FW, Fong-Wandzura; NISQ, noisy intermediate-scale quantum; RL, reinforcement learning.

Supplementary information

Supplementary information accompanies this paper at https://doi.org/10.1140/epjqt/s40507-025-00363-0.

Additional file 1. (PDF 1.1 MB)

Acknowledgements

We thank Thaddeus D. Ladd for useful correspondence.

Author contributions

GB and VNIR developed the idea to apply RL learning to the search for quantum gates for exchange-only qubits. NR implemented the numerics for the quantum gates, which VNIR used to implement a cost function. VNIR developed and applied the RL-based method to solve the resulting optimization problem. All authors contributed to the discussion of the results as well as to writing the manuscript.

Funding information

Open Access funding enabled and organized by Projekt DEAL. Violeta N. Ivanova-Rohling acknowledges funding by the German Research Foundation (DFG) under project no. 527263720 and by the Zukunftskolleg at the University of Konstanz.

Data availability

The obtained data is available under https://doi.org/10.5281/zenodo.12786663.

Declarations

Competing interests
The authors declare no competing interests.

Author details
[1]Department of Physics, University of Konstanz, D-78457 Konstanz, Germany. [2]Zukunftskolleg, University of Konstanz, D-78457 Konstanz, Germany. [3]*Present address:* Institute for Quantum Inspired and Quantum Optimization, Hamburg University of Technology, D-21071, Hamburg, Germany.

Received: 14 January 2025 Accepted: 6 May 2025 Published online: 16 May 2025

References

1. Arute F, Arya K, Babbush R, Bacon D, Bardin JC, Barends R, et al. Quantum supremacy using a programmable superconducting processor. Nature. 2019;574(7779):505–10.
2. Kjaergaard M, Schwartz ME, Braumüller J, Krantz P, Wang JIJ, Gustavsson S, et al. Superconducting qubits: current state of play. Annu Rev Condens Matter Phys. 2020;11:369–95.
3. Evered SJ, Bluvstein D, Kalinowski M, Ebadi S, Manovitz T, Zhou H, et al. High-fidelity parallel entangling gates on a neutral atom quantum computer. Nature. 2023;622:268–72.
4. Bruzewicz CD, Chiaverini J, McConnell R, Sage JM. Trapped-ion quantum computing: progress and challenges. Appl Phys Rev. 2019;6(2):021314.
5. Burkard G, Ladd TD, Pan A, Nichol JM, Petta JR. Semiconductor spin qubits. Rev Mod Phys. 2023;95:025003.
6. Xue X, Russ M, Samkharadze N, Undseth B, Sammak A, Scappucci G, et al. Quantum logic with spin qubits crossing the surface code threshold. Nature. 2022;601(7893):343–7.
7. Noiri A, Takeda K, Nakajima T, Kobayashi T, Sammak A, Scappucci G, et al. Fast universal quantum gate above the fault-tolerance threshold in silicon. Nature. 2022;601(7893):338–42.
8. Madzik MT, Asaad S, Youssry A, Joecker B, Rudinger KM, Nielsen E, et al. Precision tomography of a three-qubit donor quantum processor in silicon. Nature. 2022;601(7893):348–53.
9. Loss D, DiVincenzo DP. Quantum computation with quantum dots. Phys Rev A. 1998;57(1):120.
10. Pioro-Ladriere M, Obata T, Tokura Y, Shin YS, Kubo T, Yoshida K, et al. Electrically driven single-electron spin resonance in a slanting Zeeman field. Nat Phys. 2008;4(10):776–9.
11. DiVincenzo DP, Bacon D, Kempe J, Burkard G, Whaley KB. Universal quantum computation with the exchange interaction. Nature. 2000;408(6810):339–42.
12. Bacon D, Kempe J, Lidar DA, Whaley KB. Universal fault-tolerant quantum computation on decoherence-free subspaces. Phys Rev Lett. 2000;85:1758–61.
13. Weinstein AJ, Reed MD, Jones AM, Andrews RW, Barnes D, Blumoff JZ, et al. Universal logic with encoded spin qubits in silicon. Nature. 2023;615(7954):817–22.
14. Russ M, Burkard G. Three-electron spin qubits. J Phys Condens Matter. 2017;29(39):393001.
15. Hsieh M, Kempe J, Myrgren S, Whaley KB. An explicit universal gate-set for exchange-only quantum computation. Quantum Inf Process. 2003;2:289–307.
16. Fong BH, Wandzura SM. Universal quantum computation and leakage reduction in the 3-qubit decoherence free subsystem. Quantum Inf Comput. 2011;11(11–12):1003–18.
17. Setiawan F, Hui HY, Kestner JP, Wang X, Das Sarma S. Robust two-qubit gates for exchange-coupled qubits. Phys Rev B. 2014;89:085314.
18. Makhlin Y. Nonlocal properties of two-qubit gates and mixed states, and the optimization of quantum computations. Quantum Inf Process. 2002;1:243–52.
19. Zhang J, Vala J, Sastry S, Whaley KB. Geometric theory of nonlocal two-qubit operations. Phys Rev A. 2003;67:042313.
20. Zeuch D, Bonesteel N. Simple derivation of the Fong-Wandzura pulse sequence. Phys Rev A. 2016;93(1):010303.
21. van Meter JR, Knill E. Approximate exchange-only entangling gates for the three-spin-1/2 decoherence-free subsystem. Phys Rev A. 2019;99(4):042331.
22. Zeuch D, Bonesteel N. Efficient two-qubit pulse sequences beyond CNOT. Phys Rev B. 2020;102(7):075311.
23. Langrock V, DiVincenzo DP. A reset-if-leaked procedure for encoded spin qubits. 2020. arXiv preprint. arXiv:2012.09517.
24. Laird EA, Taylor JM, DiVincenzo DP, Marcus CM, Hanson MP, Gossard AC. Coherent spin manipulation in an exchange-only qubit. Phys Rev B. 2010;82:075403.
25. Gaudreau L, Granger G, Kam A, Aers G, Studenikin S, Zawadzki P, et al. Coherent control of three-spin states in a triple quantum dot. Nat Phys. 2012;8(1):54–8.
26. Medford J, Beil J, Taylor J, Bartlett S, Doherty A, Rashba E, et al. Self-consistent measurement and state tomography of an exchange-only spin qubit. Nat Nanotechnol. 2013;8(9):654–9.
27. Medford J, Beil J, Taylor J, Rashba E, Lu H, Gossard A, et al. Quantum-dot-based resonant exchange qubit. Phys Rev Lett. 2013;111(5):050501.
28. Kim D, Shi Z, Simmons C, Ward D, Prance J, Koh TS, et al. Quantum control and process tomography of a semiconductor quantum dot hybrid qubit. Nature. 2014;511(7507):70–4.
29. Eng K, Ladd TD, Smith A, Borselli MG, Kiselev AA, Fong BH, et al. Isotopically enhanced triple-quantum-dot qubit. Sci Adv. 2015;1(4):e1500214.
30. Reed M, Maune B, Andrews R, Borselli M, Eng K, Jura M, et al. Reduced sensitivity to charge noise in semiconductor spin qubits via symmetric operation. Phys Rev Lett. 2016;116(11):110402.
31. Cao G, Li HO, Yu GD, Wang BC, Chen BB, Song XX, et al. Tunable hybrid qubit in a GaAs double quantum dot. Phys Rev Lett. 2016;116(8):086801.
32. Thorgrimsson B, Kim D, Yang YC, Smith L, Simmons C, Ward DR, et al. Extending the coherence of a quantum dot hybrid qubit. npj Quantum Inf. 2017;3(1):32.
33. Pham D, Karaboga D. Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks. Berlin: Springer; 2012.

34. Raffin A, Hill A, Gleave A, Kanervisto A, Ernestus M, Dormann N. Stable-Baselines3: reliable reinforcement learning implementations. J Mach Learn Res. 2021;22(268):1–8.
35. Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, et al. Openai gym. 2016. arXiv preprint. arXiv:1606.01540.
36. Smith LN. A disciplined approach to neural network hyper-parameters: part 1–learning rate, batch size, momentum, and weight decay. 2018. arXiv preprint. arXiv:1803.09820.
37. Ivanova-Rohling VN, Rohling N, Burkard G. Reinforcement learning approach for finding exchange-only gate sequences for CNOT with record-low gate time: data. Zenodo. 2024;12786663. Available from: https://doi.org/10.5281/zenodo.12786663.
38. Neyens S, Zietz OK, Watson TF, Luthi F, Nethwewala A, George HC, et al. Probing single electrons across 300-mm spin qubit wafers. Nature. 2024;629(8010):80–5.
39. Martín-Guerrero JD, Lamata L. Reinforcement learning and physics. Appl Sci. 2021;11(18):8589.
40. Krenn M, Landgraf J, Foesel T, Marquardt F. Artificial intelligence and machine learning for quantum technologies. Phys Rev A. 2023;107:010101.
41. Fösel T, Tighineanu P, Weiss T, Marquardt F. Reinforcement learning with neural networks for quantum feedback. Phys Rev X. 2018;8(3):031084.
42. Chen T, Chen X, Chen W, Wang Z, Heaton H, Liu J, et al. Learning to Optimize: a Primer and a Benchmark. J Mach Learn Res. 2022;23(1).
43. Gregor K, LeCun Y. Learning fast approximations of sparse coding. In: Proceedings of the 27th international conference on International Conference on Machine Learning. ICML'10. Madison: Omnipress; 2010. p. 399–406.
44. Li K, Malik J. Learning to optimize. In: International conference on learning representations. 2017. Available from: https://openreview.net/forum?id=ry4Vrt5gl.
45. Chen X, Chen T, Cheng Y, Chen W, Awadallah A, Wang Z. Scalable learning to optimize: a learned optimizer can train big models. In: Computer vision – ECCV 2022: 17th European conference, Tel Aviv, Israel, October 23–27, 2022, proceedings, part XXIII. Berlin: Springer; 2022. p. 389–405. Available from:https://doi.org/10.1007/978-3-031-20050-2_23.
46. Chen Y, Hoffman MW, Colmenarejo SG, Denil M, Lillicrap TP, Botvinick M, et al. Learning to learn without gradient descent by gradient descent. In: Precup D, Teh YW, editors. Proceedings of the 34th international conference on machine learning. Proceedings of machine learning research. vol. 70. PMLR; 2017. p. 748–56.
47. Dai H, Khalil EB, Zhang Y, Dilkina B, Song L. Learning combinatorial optimization algorithms over graphs. In: Proceedings of the 31st international conference on Neural Information Processing Systems. NIPS'17. Red Hook: Curran Associates Inc.; 2017. p. 6351–61.
48. Fedus W, Ramachandran P, Agarwal R, Bengio Y, Larochelle H, Rowland M, et al. Revisiting fundamentals of experience replay. In: International conference on machine learning. PMLR; 2020. p. 3061–71.
49. Seyyedabbasi A. A reinforcement learning-based metaheuristic algorithm for solving global optimization problems. Adv Eng Softw. 2023;178:103411.
50. Seyyedabbasi A, Aliyev R, Kiani F, Gulle MU, Basyildiz H, Shah MA. Hybrid algorithms based on combining reinforcement learning and metaheuristic methods to solve global optimization problems. Knowl-Based Syst. 2021;223:107044.
51. Karimi-Mamaghan M, Mohammadi M, Meyer P, Karimi-Mamaghan AM, Talbi EG. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: a state-of-the-art. Eur J Oper Res. 2022;296(2):393–422.
52. Samma H, Lim CP, Saleh JM. A new reinforcement learning-based memetic particle swarm optimizer. Appl Soft Comput. 2016;43:276–97.
53. Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, et al. Continuous control with deep reinforcement learning. 2015. arXiv preprint. arXiv:1509.02971.
54. Konda V, Tsitsiklis J. Actor-critic algorithms. Adv Neural Inf Process Syst. 1999;12.
55. Fletcher R, Powell MJ. A rapidly convergent descent method for minimization. Comput J. 1963;6(2):163–8.

Publisher's note