npj | Quantum Information

www.nature.com/npjqi

ARTICLE OPEN

Check for updates

Deep reinforcement learning for efficient measurement of quantum devices

V. Nguyen[1,5], S. B. Orbell[1,5], D. T. Lennon[1], H. Moon[1], F. Vigneau[1], L. C. Camenzind[2], L. Yu[2], D. M. Zumbühl[2], G. A. D. Briggs[1], M. A. Osborne[3], D. Sejdinovic[4] and N. Ares[1]✉

Deep reinforcement learning is an emerging machine-learning approach that can teach a computer to learn from their actions and rewards similar to the way humans learn from experience. It offers many advantages in automating decision processes to navigate large parameter spaces. This paper proposes an approach to the efficient measurement of quantum devices based on deep reinforcement learning. We focus on double quantum dot devices, demonstrating the fully automatic identification of specific transport features called bias triangles. Measurements targeting these features are difficult to automate, since bias triangles are found in otherwise featureless regions of the parameter space. Our algorithm identifies bias triangles in a mean time of <30 min, and sometimes as little as 1 min. This approach, based on dueling deep Q-networks, can be adapted to a broad range of devices and target transport features. This is a crucial demonstration of the utility of deep reinforcement learning for decision making in the measurement and operation of quantum devices.

npj Quantum Information (2021)7:100 ; https://doi.org/10.1038/s41534-021-00434-x

INTRODUCTION

Reinforcement learning (RL) is a neurobiologically inspired machine-learning paradigm where an RL agent will learn policies to successfully navigate or influence the environment. Neural network-based deep reinforcement learning (DRL) algorithms have proven to be very successful by surpassing human experts in domains such as the popular Atari 2600 games[1], chess[2], and Go[3]. RL algorithms are expected to advance the control of quantum devices[4–21], because the models can be robust against noise and stochastic elements present in many physical systems and they can be trained without labelled data. However, the potential of deep reinforcement learning for the efficient measurement of quantum devices is still unexplored.

Semiconductor quantum dot devices are a promising candidate technology for the development of scalable quantum computing architectures. Singlet–triplet qubits encoded in double quantum dots[22] have demonstrably long coherence times[23,24], as well as high one-[25] and two-qubit[26–28] gate fidelities. Promising qubit performance was also demonstrated in single-spin qubits[29–31], and exchange only qubits[32,33]. However, quantum dot devices are subject to variability, and many measurements are required to characterise each device and find the conditions for qubit operation. Machine learning has been used to automate the tuning of devices from scratch, known as super coarse tuning[34–36], the identification of single or double quantum dot regimes, known as coarse tuning[37,38], and the tuning of the inter-dot tunnel couplings and other device parameters, referred to as fine tuning[39–41].

The efficient measurement and characterisation of quantum devices has been less explored so far. We have previously developed an efficient measurement algorithm for quantum dot devices combining a deep-generative model and an information-theoretic approach[42]. Other approaches have developed classification tools that are used in conjunction with numerical optimisation routines to navigate quantum dot current maps[37,40,43]. These methods, however, fail when there are large areas in parameter space that do not exhibit transport features. To perform efficient measurements in these areas, which are often good for qubit operation, requires prior knowledge of the measurement landscape and a procedure to avoid over-fitting, i.e., a regularisation method.

In this paper, we propose to use DRL for the efficient measurement of a double quantum dot device. Our algorithm is capable of finding specific transport features, in particular bias triangles, surrounded by featureless areas in a current map. The state-of-the-art DRL decision agent is embedded within an efficient algorithmic workflow, resulting in significant reduction of the measurement time in comparison to existing methods. A convolutional neural network (CNN), a popular image classification tool[44,45], is used to identify the bias triangles. This optimal decision process allows for the identification of promising areas of the parameter space without the need for human intervention. Fully automated approaches, such as the measurement algorithm presented here, could help to realise the full potential of spin qubits by addressing key difficulties in their scalability.

We focus on quantum dot devices that are electrostatically defined by Ti/Au gate electrodes fabricated on a GaAs/AlGaAs heterostructure (Fig. 1a)[46,47]. All the experiments were performed using GaAs double quantum dot devices at dilution refrigerator temperatures of ~30 mK. The two-dimensional electron gas created at the interface of the two semiconductor materials is depleted by applying negative voltages to the gate electrodes. The confinement potential defines a double quantum dot, which is controlled by these gate voltages and coupled to the electron reservoirs (the source and drain contacts). Depending on the combination of gate voltages, the double quantum dot can be in the 'open', the 'pinch-off' or the 'single-electron transport' regime. In the 'open' regime, an unimpeded current flows through the device. Conversely, when the current is completely blocked, the device is said to be in the 'pinch-off' regime. In the 'single-electron

[1]Department of Materials, University of Oxford, Oxford, UK. [2]Department of Physics, University of Basel, Basel, Switzerland. [3]Department of Engineering, University of Oxford, Oxford, UK. [4]Department of Statistics, University of Oxford, Oxford, UK. [5]These authors contributed equally: V. Nguyen, S. B. Orbell. ✉email: natalia.ares@materials.ox.ac.uk

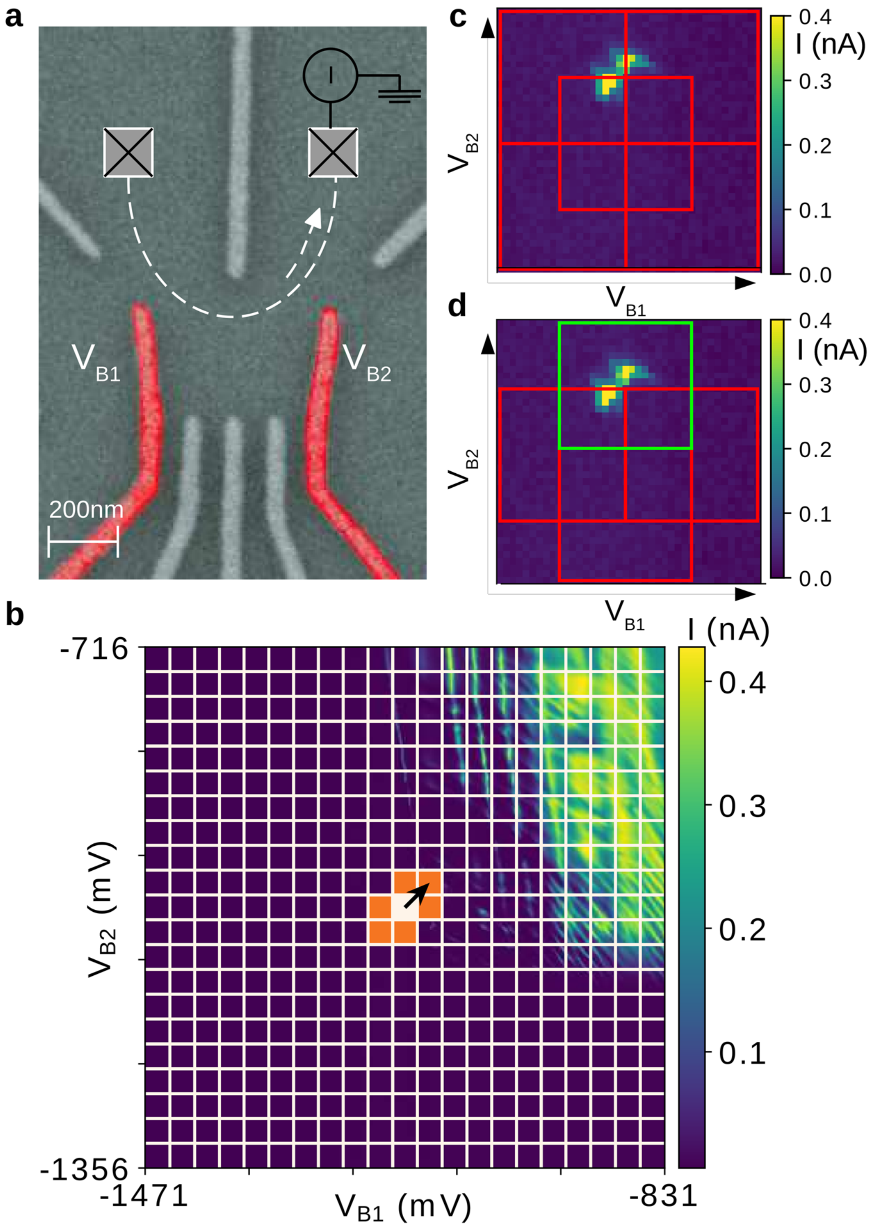

Fig. 1 Overview of device architecture and quantum dot environment. a False-colour SEM image of a GaAs double quantum dot device. Barrier gates, labelled V_{B1} and V_{B2}, are highlighted in red. The arrow represents the flow of current through the device between source and drain contacts. **b** A current map. The white grid represents the blocks available for investigation by the DRL agent. The DRL agent is initiated in a random block (state) indicated by a filled white square. The filled orange blocks show the available action space for the DRL agent and the arrow shows a possible policy decision. **c** and **d** The nine sub-blocks defined within each block, a 32 × 32 mV window in gate voltage space, to calculate a statistical state vector. These sub-blocks are equal in gate voltage size, five of them are shown in (**c**) and four in (**d**). The green sub-block in (**d**) contains bias triangles.

transport' regime, the current is maximal when the electrochemical potentials of each quantum dot are within the bias voltage window V_{bias} between source and drain contacts.

Our algorithm interacts with a quantum dot environment within which our DRL decision agent operates to efficiently find the target transport features. The environment consists of states, defined by sets of measurements in gate voltage space, and a set of actions and rewards to navigate that space. This quantum dot environment has been developed based upon the OpenAI Gym interface[48] (see Supplementary Note A. for further details of the quantum dot environment's state, action and reward). Manual identification and characterisation of transport features requires a high-resolution measurement of a current map defined by, for example, barrier gate voltages V_{B1} and V_{B2} while keeping other gate voltages fixed, an example of which is shown in Fig. 1b. A super coarse tuning algorithm allows us to choose a set of promising gate voltages and focus on exploring the current map as a function of two gates, for example the two barrier gates[36]. This is the gate voltage space our DRL agent will navigate.

Our DRL algorithm takes the gate voltage coordinates found by our previous super coarse tuning algorithm[36], and divides the gate voltage space corresponding to the unmeasured current map into blocks. The size of the blocks is chosen such that they can fully contain bias triangles (blocks are shown as a white grid in Fig. 1b). Devices with similar gate architectures often show bias triangles of similar sizes for a given V_{bias}. The DRL agent is initiated in a random block. The agent acquires a reduced number of current measurements from this block and makes a decision on whether a

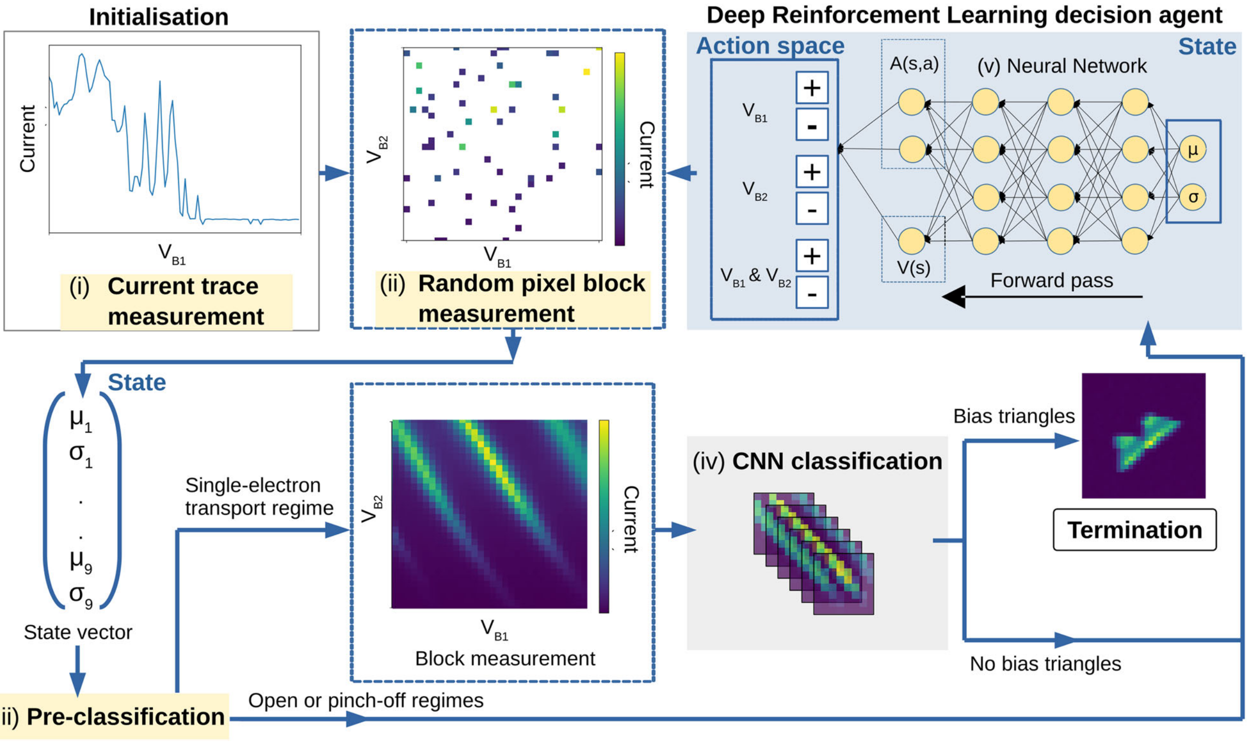

Fig. 2 Schematic depicting the algorithmic workflow (see main text for a full description). In the initialisation stage, starting from the gate voltages coordinates proposed by a coarse tuning algorithm, the algorithm measures low-resolution current traces as a function of V_{B1} (V_{B2}) with V_{B2} (V_{B1}) set to the maximum voltage given by the gate voltage window of interest (**i**). The algorithm then performs a random pixel measurement in the block corresponding to the proposed starting gate voltages (**ii**). In this measurement, mean current values and standard deviation are calculated for nine sub-blocks within the block until convergence. The statistical state representation vector (state vector) obtained is then assessed by the pre-classification stage (**iii**). If the mean current value corresponding to any of the sub-blocks falls within threshold values given by the initialisation stage, then the block is pre-classified as corresponding to a possible single-electron transport regime. In this case, the block is explored further by performing a high-resolution scan. This block measurement is normalised and input into a CNN binary classification algorithm (**iv**). If the CNN identifies bias triangles, then the algorithm terminates. If either the pre-classifier or the CNN classifier rejects a block, then the state vector is input into the DRL decision agent (**v**). The decision agent subsequently selects an action on the gate voltages, which determines the next block to measure via the random pixel method.

high-resolution measurement is required and on which block to explore next if bias triangles are not observed. The agent has a possible action space represented by a vector of length six; this means the agent can decide to acquire measurements in any of the four contiguous blocks (‘up’, ‘down’, ‘left’, or ‘right’) or in the two diagonal blocks that permit the agent to efficiently move between the ‘open’ and the ‘pinch-off’ transport regimes. These blocks correspond to an increase or decrease of both gate voltages, which strongly modulates the current through the device. The remaining two diagonal blocks, which correspond to a decrease of one gate voltage and an increase of the other, do not often lead to such significant changes in the transport regime and are thus not included in the agent’s action space to maximise the efficiency of the algorithm. The DRL agent can be efficiently trained using current maps already recorded from many other devices. This is because their transport features are sufficiently similar, even though the gate voltage values at which they are observed vary for different devices.

The decision of which block to explore next is based on the current measurements acquired by the DRL agent in a given block. The block is divided into nine sub-blocks (Fig. 1c, d) and the mean μ and standard deviation σ of the current measurements corresponding to each sub-block are calculated. These statistical values, constituting an 18-element vector, provide the agent with information of its probable location in the current map. The statistical state vector or state representation vector enables the DRL decision agent to abstract knowledge about the transport regime, distinguishing between ‘open’, ‘pinch-off’, and ‘single-electron transport’ regimes, with a reduced number of measurements. In this way, the state vector defines a state in the quantum dot environment.

This statistical approach, compared to the alternative of using CNNs to evaluate acquired measurements, makes the agent less prone to over-fitting during training and more robust to experimental noise. To decide whether the agent has found bias triangles in a given block, the algorithm uses a CNN as a binary classification tool. Combining a state representation based on measurement statistics and CNNs in a reinforcement-learning framework, which makes use of the experience of the agent navigating similar environments during training, our algorithm provides a decision process for efficient measurement without human intervention.

RESULTS

Description of the algorithm

The algorithm is comprises different modules for classification and decision making (Fig. 2). In the initialisation stage, two low-resolution current traces are acquired by the algorithm as a function of V_{B1} (V_{B2}) with V_{B2} (V_{B1}) set to the maximum voltage given by the gate voltage window to be explored. The algorithm extracts from these measurements the maximum and minimum current values and its standard deviation, which will be used in a later stage by the classification modules. The gate voltage regions we explore are delimited by a 640 × 640 mV window centred in the gate voltage coordinates proposed by a super coarse tuning algorithm, as mentioned in the Introduction, and the current traces in this stage have a resolution of 6.4 mV.

The gate voltage region is divided in 32 × 32 mV blocks and the agent is initialised in a randomly selected block. The algorithm takes random pixel measurements of current within this block. Each pixel is 1 × 1 mV. As these measurements are performed, the algorithm estimates the 18-dimensional state vector given by μ and σ for each of the 9 sub-blocks in which the block is divided. Pixels are sampled randomly from the block until the statistics from the state representation have converged. Convergence is generally achieved after sampling fewer than 100 pixels, significantly less than the 1024 pixels in a block (see Supplementary Note B. for the convergence curves and the convergence criterion).

The state vector is first evaluated by a pre-classification module. A block is classified as corresponding to the single-electron transport regime if any of the nine μ values, corresponding to the nine sub-blocks, falls between two predefined values. These values are set to be 0.01 to 0.3 times the maximum current range detected in the initialisation stage (see Supplementary Note B. for further details about the design of the pre-classifier). We have found that the choice of such hyperparameters does not have a significant impact in the performance of the algorithm. If the pre-classifier identifies the block as corresponding to the single-electron transport regime, a high-resolution current measurement (1024 pixels, 1 × 1 mV resolution) of the block is acquired. This block measurement is normalised and evaluated by a CNN binary classifier. For any output value >0.5, the block is identified as containing bias triangles. If bias triangles are identified within the block, the algorithm is terminated. Figure 3a shows the blocks in a current map that would be identified by the pre-classifier as corresponding to the single-electron transport regime, while Fig. 3b shows the blocks that would be evaluated by the CNN binary classification to determine if bias triangles are observed (see Supplementary Note C. for a summary of the CNN's architecture and its training).

If the pre-classifier considers the block to correspond to the 'open' or 'pinch-off' regimes, or if the CNN does not identify bias triangles within the block, the DRL agent has to decide which block to explore next. With this objective, the state vector is normalised using the variance and mean current values obtained in the initialisation stage, and fed into a deep neural network, which controls the DRL decision agent. The agent will then propose an action, which it expects will lead to the highest long-term reward. This action a_t, given by $a_t = \arg\max_{a'} Q^{\pi}(s_t, a')$, is the action, which maximises the Q-function for the agent's stochastic policy π in the state-action pair (s_t, a') at time t. The Q-function measures the value of choosing an action a' when in state s_t and therefore the action a_t represents the agent's prediction for the most efficient route to bias triangles. In our quantum dot environment setting, the action determines the next block to explore and the algorithm begins a new iteration.

The deep reinforcement-learning agent

Our algorithm makes use of the deep Q-learning framework, which uses deep neural networks to approximate the Q-function[1]. The Q-function is defined by $Q^{\pi}(s_t, a_t) = \mathbb{E}[R_t | s = s_t, a = a_t, \pi]$, which gives an expected reward R_t for a chosen action a_t taken by an agent with a policy π in the state s_t. This expected reward is defined as $R_t = \sum_{\tau=t}^{\infty} \gamma^{\tau - t} r_\tau$, where $\gamma \in [0, 1]$ is a discount factor that trades-off the importance of immediate rewards r_t, and future rewards $r_\tau > t$. The agent aims to maximise R_t via the $Q^{\pi}(s_t, a_t)$ learnt by the neural network. In particular, we chose to implement the dueling deep Q-network (dueling DRL decision agent (DQN))[49] architecture for our DRL decision agent. This architecture factors the neural network into two entirely separate estimators for the state-value function and the state-dependent action advantage function[49]. The state-value function, $V^{\pi}(s_t) = \mathbb{E}_{a_t \sim \pi(s_t)}[Q^{\pi}(s_t, a_t)]$ gives a measure for how valuable it is, for an agent with a stochastic policy π in the search for a promising reward, to be in a given state s_t. The state-dependent action advantage function[49] gives a relative measure of the importance of each action, given by $A^{\pi}(s_t, a_t) = Q^{\pi}(s_t, a_t) - V^{\pi}(s_t)$. In dueling DQN, when combining the state-value function and the state-dependent action advantage function, it is crucial to ensure that given Q we can recover $V^{\pi}(s_t)$ and $A^{\pi}(s_t, a_t)$ uniquely. For this purpose, the advantage function estimator is forced to be zero at the chosen action a_t[49]. Our dueling DQN consists of three fully connected layers with 128, 64, and 32 units respectively. The dueling component is defined by a further fully connected layer with 64 units for the action advantage function estimator and one unit for the state-value function estimator.

This approach allows the agent, through the estimation of $V^{\pi}(s_t)$, to learn the value of certain states in terms of their potential to guide the agent to a promising reward. This is particularly beneficial in our case, since different state vectors can correspond to the same transport regime and thus be equally valuable in the search of bias triangles. Consequently the most beneficial action in these states would often coincide. For example, in most states corresponding to the 'pinch-off' regime, the most beneficial action is often to increase both gate voltages.

To train the DRL agent, we designed a reward function to ensure that the agent would learn to efficiently locate bias

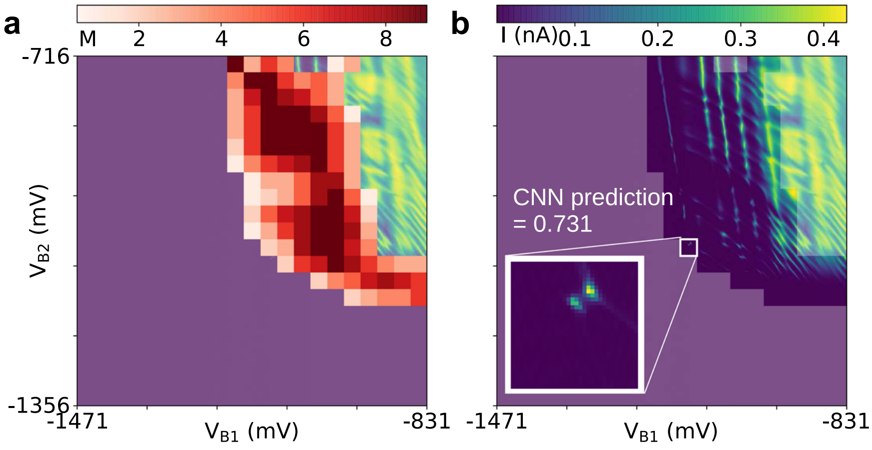

Fig. 3 Classification tools. a Example of blocks considered by the pre-classifier as corresponding to the 'single-electron transport' regime overlaid on the corresponding current map. The colour-bar represents the number (M), out of nine, of sub-blocks, which were not rejected by the pre-classification stage. **b** Blocks in (**a**), displaying features corresponding to the 'single-electron transport' regime, overlaid on the corresponding current map. Inset: A block displaying bias triangles and the corresponding output value of the CNN binary classifier.

triangles. To this end, during training, the agent is rewarded for the detection of bias triangles and penalised for the number of blocks explored or measured in a single algorithm run, N. The reward $r = +10$ is assigned to the blocks exhibiting bias triangles. Other blocks are assigned $r = -1$. During training, the maximum number of blocks that could be measured in a given run, N_{max}, is set to 300. If after N_{max} block measurements the agent had not found bias triangles, the algorithm is terminated and the agent is punished with $r = -10$ (see Supplementary Note A. for further details regarding the design of the reward function). In other words, N_{max} determines how far from the starting block the agent can reach in gate voltage space, as it can only explore contiguous blocks.

We trained the dueling DQN using the prioritised experience replay method[50] from a memory buffer. This method ensures that successful policy decisions are replayed more frequently in the DRL agent's-learning process. The agent does not benefit from an ordered sequence of episodes during learning, yet it is able to learn from rare but highly successful policy decisions and it is less likely to settle in local minima of the decision policy. We trained the agent over 10,000 episodes (algorithm runs) using the Adam optimiser[51], each time initialised in a random block for four different current maps, which were previously recorded. The training takes less than an hour on a single CPU (Intel(R) Core(TM) i5-8500 CPU @ 3.00GHz).

Experimental results

We demonstrate the real-time ('online') performance of our algorithm in a double quantum dot device. The algorithm performance is evaluated according to the number of blocks explored in an algorithm run, N, which is equal to the number of blocks explored to successfully identify bias triangles unless $N = N_{\mathrm{max}}$, and according to the laboratory time spent in this task. For training and testing the algorithm's performance we use different devices, both similar to the device shown in Fig. 1a. We ran the DRL algorithm in two different regions of gate voltage space, I and II, which are centred in the coordinates from our super coarse tuning algorithm[36]. We ran the algorithm ten times in each region. The DRL agent was initiated in a different block for every run, sampled uniformly at random. From these repeated runs, we can estimate the median $\overline{N}$ of the distribution of values of N obtained for a given region. We can also estimate (L, U), where L and U are the lower and upper deciles of the distribution. To identify bias triangles, the DRL agent required $\overline{N} = 40(9, 104)$ for region I and $\overline{N} = 32(10, 94)$ for region II. In both regions considered, our algorithm efficiently located bias triangles in a mean time of 30 min and, on one occasion, in <1 min. This is an order of magnitude improvement in measurement efficiency compared to the laboratory time required to acquire a current map with the grid scan method, i.e., measuring the current while sweeping V_{B2} and stepping V_{B1}, which is ~5.5 hours with pixel resolution (1×1 mV resolution). This time corresponds to the measurement of the whole gate voltage window to be explored with no automatic bias triangle identification or any other computational overheads. This is the most time-consuming measurement strategy, but the most common approach until this work. The agent cannot move outside of the gate voltage window, nor can it measure the same block twice. In the worst case scenario, the algorithm will thus measure as many blocks as the traditional grid scan method with a small computational overhead. A grid scan of each 32×32 mV block with 1×1 mV pixel resolution takes 50 s. For the online runs of our algorithm, each block was measured and assessed for a median time of 23 s in region I and 26 s in region II. This demonstrates the low-computational overhead required for the pre-classification, CNN classification, and forward pass of the DRL agent, as well as the efficacy of our random pixel measurements. Using a single CPU of a standard desktop computer, the algorithm is not limited by computation time. It can thus be run with the computing resources available in most laboratories.

Example trajectories of the agent within the gate voltage space give an insight into the transport properties that the agent has implicitly learnt from its environment. When initiated in a transport regime corresponding to pinch-off (low current), the agent reduces the magnitude of the negative voltage applied to the gate electrodes, as humans experts would do (Fig. 4a). Conversely, when initiated in a transport regime corresponding to higher currents, the agent increases the magnitude of the negative voltage applied to the gate electrodes (Fig. 4b). The policy thus leads to block measurements in the areas of gate voltage space where bias triangles are usually located. These areas can be anywhere within the defined parameter space.

We have performed an ablation study. Ablation studies are used to identify the relative contribution of different algorithm components. In this case, our aim is to determine the benefit of using a DRL agent. We thus produced an algorithm in which the DRL decision agent was replaced with a random decision agent. We compared its performance with the DRL algorithm. The random agent selects an action, sampled uniformly and randomly. The quantum dot environment's (QDE's) action space is six-dimensional except in instances where the agent is in a state (block) along the edges (five-dimensional action space) and in the corners (four-dimensional action space) of the gate voltage window considered. This measurement strategy is similar to a random walk within the gate voltage space, but unlike a pure random walk strategy, it will not measure the same block twice. The random decision agent's measurement run will be terminated when the CNN classifies a block measurement as containing bias triangles. The random agent was initialised in the same random positions as the DRL agent so that a fair comparison could be made between their performances. We performed ten runs of each algorithm in each of the two different regions of parameter space considered in this work, I and II (Fig. 4c, d). The DRL agent outperforms the random decision agent in the value of $\overline{N}$, and thus in the laboratory time required to successfully identify bias triangles. Note that the relation between $\overline{N}$ and the laboratory time is not linear, as high-resolution block measurements are only performed for each block classified as corresponding to the single-electron transport regime by the pre-classification stage.

In region II, the random agent requires $\overline{N}$ equal to 85 (50, 143), which is ~2.6 times larger than the $\overline{N}$ corresponding to the DRL agent (see Supplementary Note D. for the value of $\overline{N}$ in region I and corresponding lab times). The good performance of the random decision agent can be explained by its use of the pre-classifier, which makes the random search efficient. The random decision agent is an order of magnitude quicker than the grid scan method.

To test the statistical significance of the DRL agent's advantage, we have tested the performance of both algorithms in a much larger number of runs. To perform this statistical convergence test would have been too costly in laboratory time, so we used previously recorded current maps, which were measured by the grid scan method. We will call this performance test 'offline', as opposed to 'online' in the case of real-time measurements. By initiating both agents 1024 times in each of the blocks in I and II, we obtained a histogram of the N blocks measured to successfully terminate the algorithm (see Fig. 4e, f for I and II, respectively). We observe a higher number of runs for which the DRL algorithm performed fewer block measurements for successful termination. In region II, the DRL agent requires $\overline{N}$ of 17 (2, 31), while the $\overline{N}$ for the random agent is 30 (3, 101) (see Supplementary Note D. for the value of $\overline{N}$ for region I). Our results suggest that the DRL advantage is statistically significant. The two-tailed Wilcoxon-signed rank test[52] allows us to make a statistical comparison of the two distributions corresponding to the DRL and the random

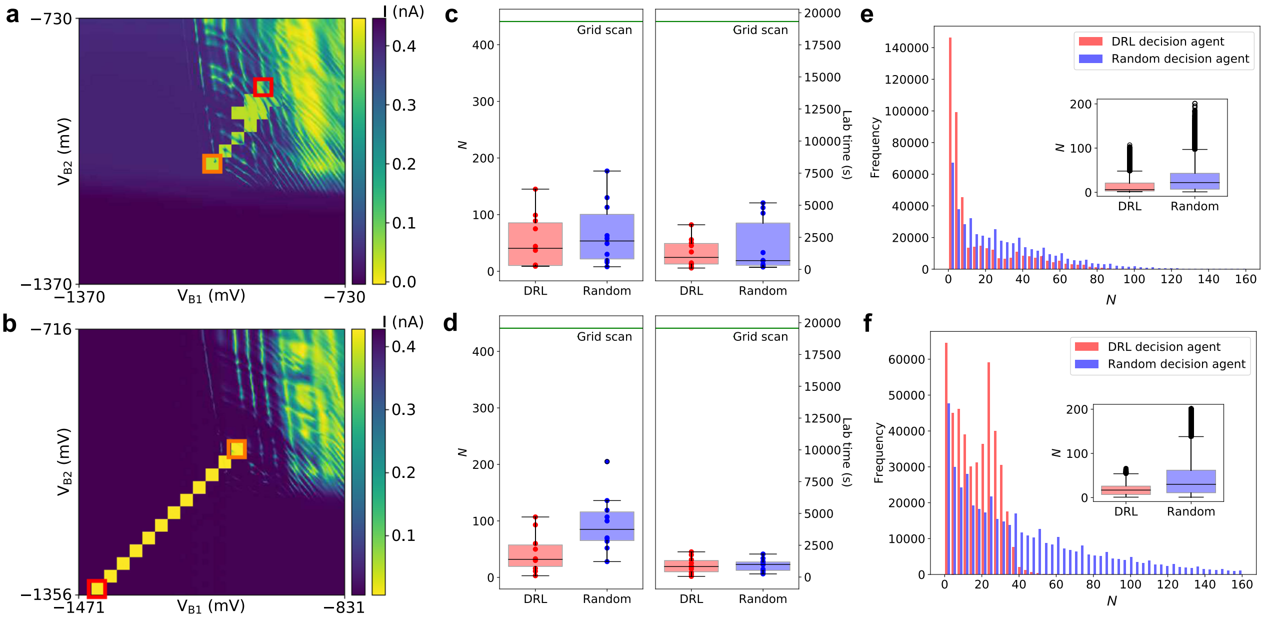

Fig. 4 Performance benchmark. **a**, **b** Example trajectories of the DRL agent in gate voltage space. **a** (**b**) Corresponds to region I (II). The trajectories are indicated by inverting the colour scale of the current map for the blocks measured by the algorithm. The full current map measured by the grid scan method is displayed for illustrative purposes and it is not seen by the DRL agent. The red and orange squares indicate the start and end of the trajectory, respectively. **c**, **d** Real-time performance corresponding to the grid scan method (green line), the algorithm with a random decision agent (blue) and the algorithm with a DRL decision agent (red). The box plots indicate the laboratory time (right) and the corresponding number of blocks explored, N, (left) for regions of the gate voltage space I and II in **c** and **d**, respectively. The laboratory time and N are not proportional, as each block requires a different measurement and computational time depending on whether a block measurement is performed and processed by the CNN. The results of all ten runs for both agents in each regime are plotted as points. The central line of the box plot corresponds to $\overline{N}$, while the upper and lower boundaries of the box display the upper ($Q3$) and lower ($Q1$) quartiles. The minimum and maximum whisker bars display ($Q1 - 1.5 \times IQR$) and ($Q3 + 1.5 \times IQR$) respectively, where IQR is the interquartile range. **e**, **f)** Histograms of values of N for the random and DRL decision agents over 10 algorithm runs for each region, I (**e**) and II (**f**). This performance test was performed offline for better statistical convergence. The insets show the box plots, indicating the quartiles and $\overline{N}$ values for the DRL and random agents. In the inset only the outlier points are plotted.

agent. We have applied this test to the offline performance for regions I and II (see Supplementary Note D. for the results of the Wilcoxon-signed rank test applied to the online performance test, for which critical values for the test threshold are used instead of assuming a normal approximation, given the number of algorithm runs is below 20). The two-tailed Wilcoxon-signed rank test yields a p-value < 0.001 for both regions. This means that the null hypothesis, stating there is no difference in the median performance between the two agents, can be rejected. In addition, the median of the differences $(\overline{N}_{\mathrm{DRL}} - \overline{N}_{\mathrm{Random}})$, estimated using the one-tailed Wilcoxon-signed rank test, is less than zero. We can therefore confirm that the DRL agent offers a statistically significant advantage over the random agent.

To further illustrate the advantages of our algorithm, we compared its performance with a Nelder–Mead numerical optimisation method applied to achieve automatic tuning of quantum dots[38,43]. To ensure a fair comparison with our reinforcement-learning method, our implementation of the Nelder–Mead optimisation (see Supplementary Note E. for further details) was terminated when the CNN classified a block as exhibiting bias triangles in the same way as our DRL algorithm, i.e., when the output value of the CNN classifier was >0.5. In the original implementation, stricter numerical stopping conditions must be met, thereby increasing the number of measurements performed before termination.

The Nelder–Mead, random decision, and DRL decision algorithms were compared offline. We have initiated the algorithms in each block within each gate voltage region and estimated $\overline{N}$, creating a performance distribution or heat-map (Fig. 5). We observe that large areas of gate voltage space that do not exhibit transport features correspond to large flat areas in the optimisation landscape and thus severely limit the Nelder–Mead method. Often the simplex was initiated in these areas and in those cases, the Nelder–Mead algorithm just repeatedly measured the area around the initial simplex. On other occasions, the algorithm moved away from the initial simplex but then became trapped in other areas of the parameter space in which transport features are not present. The method only succeeded in locating bias triangles when it was initiated in the double dot regime. The DRL decision agent's performance is non-uniform as the 'pinch-off' regime is less effectively characterised by the agent than the 'open' and 'single-electron transport' regimes. The performance of the random decision agent is also non-uniform, as it completes the tuning procedure more efficiently when initiated close to the target transport features.

The Nelder–Mead algorithm was also tested online under the same conditions as the DRL and random decision agents. None of 20 runs succeeded before reaching the predefined maximum number of measurements, $N_{\max}$, and thus the results are not presented alongside the online results of grid scan, random decision, and DRL decision algorithms in Fig. 4. Other numerical optimisation methods, better suited to the task, could offer significantly better performances.

DISCUSSION

We have demonstrated efficient measurement of a quantum dot device using reinforcement learning. We are able to locate bias triangles fully automatically, from a set of gate voltages defined by a super coarse tuning algorithm[36], in a mean time of <30 min and

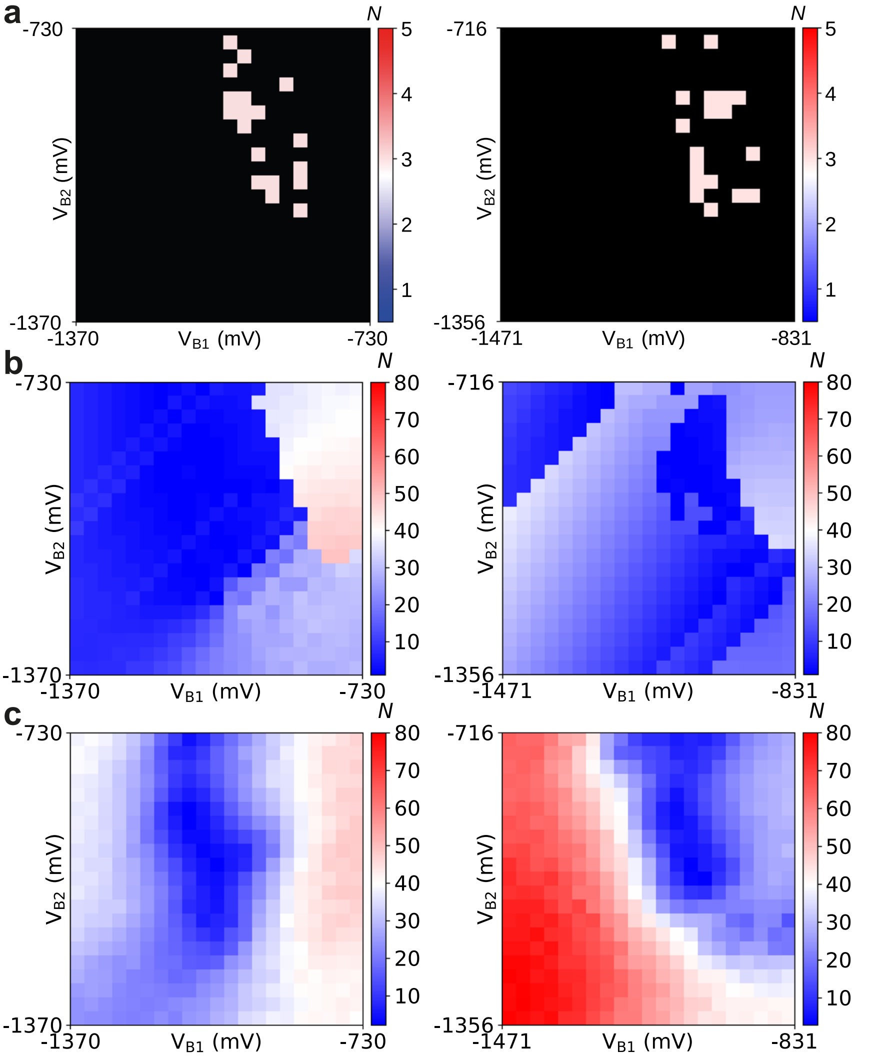

Fig. 5 Offline performance distribution. The performance of different algorithms is evaluated by initiating an algorithm run in each block and estimating N for regions I and II. Black areas indicate that the algorithm failed when initiated at those blocks. **a** Performance distribution (heat-map) for the Nelder–Mead method, **b** the DRL decision agent, and (**c**) the algorithm with a random decision agent.

in as little as 1 min. Our approach gives a ten times speed up in the median time required to locate bias triangles compared with grid scan methods. The approach is also less dependant on the transport regime in which the algorithm is initiated, compared to an algorithm based on a random agent and to a Nelder–Mead numerical optimisation method. We have also demonstrated the statistical advantage of a DRL decision agent over a random decision agent. Our DRL approach is also robust against featureless areas in the parameter space, which limit other approaches. Our algorithm uses statistics calculated via pixel sampling to explore the transport landscape. This statistical state representation allows us to efficiently measure the transport regime (or the state of the environment in DRL terms) and avoid over-fitting during agent training. Other options for state representation that go beyond a statistical summary of current values could be considered. The measurement time remains, however, the dominant contribution in the time required to identify transport features. Fast readout techniques such as radio-frequency reflectometry can be used to reduce measurement times[53–58]. However, these techniques are better suited to the measurement of small gate voltage windows. A full tuning procedure could consist of a super coarse tuning algorithm, followed by this algorithm to locate bias triangles, and a fine-tuning algorithm such as the one described in ref. [39]. Different pairs of bias triangles could be explored. Once a pair of bias triangles is chosen, now within a reduced gate voltage window, the fast readout can be optimised.

Our method is inherently flexible and modular such that it could be generalised to automate a variety of efficient measurement tasks. For example, the reward function could be modified so that the agent could learn to locate and score multiple bias triangles within the current map. Furthermore, by retraining the CNN classifier and the DRL agent, the method would be able to locate different types of transport features. Our algorithm could also incorporate other gate electrodes by increasing the action space and retraining. This approach would significantly speed up super coarse, coarse and fine-tuning algorithms. We also expect DRL approaches to scale better than random searches as the dimensionality of the problem increases.

We envisage possible extensions of our approach using probabilistic reinforcement-learning methods including: Bayesian deep reinforcement learning[59–61] and model-based reinforcement learning[62,63], where the goal is to estimate the uncertainty when making a decision and incorporate domain knowledge into the reinforcement-learning model. The resulting reinforcement-learning model may be more efficient, especially when a limited amount of data is available.

An additional benefit of reinforcement learning is the capacity of the network's policy to be continuously updated. Thereby, the agent's policy can be updated in real-time as the algorithm becomes familiar with a new device. This not only improves the general policy but also means that, over time, the pre-trained agent could learn the particularities of a specific device. To tune large quantum device arrays, due to the increasing dimensionality of the parameter space, DRL could offer a large advantage over conventional heuristic methods. Our algorithm can be implemented in arrays by considering double quantum dots independently[64], and compensating for the cross talk. Our quantum dot environment and algorithmic framework offer a valuable resource to develop and test other algorithms and decision agents for quantum device measurement and tuning. Additionally, our dueling deep Q-network methods can be translated to further applications in experimental research.

DATA AVAILABILITY

The data sets used for the training of the model are available from the corresponding author upon reasonable request.

CODE AVAILABILITY

A documented implementation of the algorithm is available at https://github.com/oxquantum-repo/drl_for_quantum_measurement.

Received: 29 October 2020; Accepted: 24 May 2021;
Published online: 18 June 2021

REFERENCES

1. Mnih, V. et al. Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015).
2. Silver, D. et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **362**, 1140–1144 (2018).
3. Silver, D. et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **529**, 484–489 (2016).
4. August, M. & Hernández-Lobato, J. M. Taking gradients through experiments: LSTMs and memory proximal policy optimization for black-box quantum control. *Lect. Notes Comput. Sci.* **11203 LNCS**, 591–613 (2018).
5. Fösel, T., Tighineanu, P., Weiss, T. & Marquardt, F. Reinforcement learning with neural networks for quantum feedback. *Phys. Rev. X* **8**, 31084 (2018).
6. Bukov, M. et al. Reinforcement learning in different phases of quantum control. *Phys. Rev. X* **8**, 31086 (2018).
7. Niu, M. Y., Boixo, S., Smelyanskiy, V. & Neven, H. Universal quantum control through deep reinforcement learning. *NPJ Quantum Inf.* **5**, 33 (2019).
8. Xu, H. et al. Generalizable control for quantum parameter estimation through reinforcement learning. *NPJ Quant. Inf.* **5**, 82 (2019).
9. Daraeizadeh, S., Premaratne, S. -P. & Matsuura, A. -Y. Designing high-fidelity multi-qubit gates for semiconductor quantum dots through deep reinforcement learning. *IEEE QCE* **1**, 30–36 (2020).
10. Herbert, S. & Sengupta, A. Using reinforcement learning to find efficient qubit routing policies for deployment in near-term quantum computers. Preprint at http://arxiv.org/abs/1812.11619 (2018).
11. Palittapongarnpim, P., Wittek, P., Zahedinejad, E., Vedaie, S. & Sanders, B. C. Learning in quantum control: High-dimensional global optimization for noisy quantum dynamics. *Neurocomputing* **268**, 116–126 (2017).
12. An, Z. & Zhou, D. L. Deep reinforcement learning for quantum gate control. *EPL-EUROPHYS LETT* **126**, https://arxiv.org/abs/1902.08418 (2019).
13. Porotti, R., Tamascelli, D., Restelli, M. & Prati, E. Coherent transport of quantum states by deep reinforcement learning. *Commun. Phys.* **2**, https://arxiv.org/abs/1901.06603 (2019).
14. Schuff, J., Fiderer, L. J. & Braun, D. Improving the dynamics of quantum sensors with reinforcement learning. *N. J. Phys* **22**, 035001 (2020).
15. Wang, T. et al. Benchmarking model-based reinforcement learning. Preprint at http://arxiv.org/abs/1907.02057 (2019).
16. Wei, P., Li, N. & Xi, Z. Open quantum system control based on reinforcement learning. *Chin. Control Conf.* **38**, 6911–6916 (2019).
17. Gao, X. & Duan, L. M. Efficient representation of quantum many-body states with deep neural networks. *Nat. Commun.* **8**, 662 (2017).
18. Barr, A., Gispen, W. & Lamacraft, A. Quantum ground states from reinforcement learning. *Proc. Mach. Learn Res* **107**, 635–653 (2020).
19. Deng, D. L. Machine learning detection of bell nonlocality in quantum many-body systems. *Phys. Rev. Lett.* **120**, 240402 (2018).
20. Carleo, G. & Troyer, M. Solving the quantum many-body problem with artificial neural networks. *Science* **355**, 602–606 (2017).
21. Sørdal, V. B. & Bergli, J. Deep reinforcement learning for quantum Szilard engine optimization. *Phys. Rev. A* **100**, 042314 (2019).
22. Loss, D., DiVincenzo, D. P. & DiVincenzo, P. Quantum computation with quantum dots. *Phys. Rev. A* **57**, 120–126 (1997).
23. Malinowski, F. K. et al. Notch filtering the nuclear environment of a spin qubit. *Nat. Nanotechnol.* **12**, 16–20 (2017).
24. Jirove, D. et al. A singlet-triplet hole spin qubit in planar Ge. *Nature Materials* https://doi.org/10.1038/s41563-021-01022-2 (2021).
25. Cerfontaine, P. et al. Closed-loop control of a GaAs-based singlet-triplet spin qubit with 99.5% gate fidelity and low leakage. *Nat. Commun.* **11**, 5–10 (2020).
26. Veldhorst, M. et al. A two-qubit logic gate in silicon. *Nature* **526**, 410–414 (2015).
27. Huang, W. et al. Fidelity benchmarks for two-qubit gates in silicon. *Nature* **569**, 532–536 (2019).
28. Zajac, D. M. et al. Resonantly driven CNOT gate for electron spins. *Science* **359**, 6374 (2018).
29. Nowack, K. C., Koppens, F. H. L., Nazarov, Y. V. & Vandersypen, L. M. K. Coherent control of a single electron spin with electric fields. *Science* **318**, 1430–1433 (2007).
30. Veldhorst, M. et al. An addressable quantum dot qubit with fault-tolerant control-fidelity. *Nat. Nanotechnol.* **9**, 981–985 (2014).
31. Tarucha, S. et al. A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%. *Nat. Nanotechnol.* **13**, 2 (2018).
32. Laird, E. A. et al. Coherent spin manipulation in an exchange-only qubit. *Phys. Rev. B* **82**, 7 (2010).
33. Medford, J. et al. Self-consistent measurement and state tomography of an exchange-only spin qubit. *Nat. Nanotechnol.* **8**, 9 (2013).
34. Baart, T. A., Eendebak, P. T., Reichl, C., Wegscheider, W. & Vandersypen, L. M. Computer-automated tuning of semiconductor double quantum dots into the single-electron regime. *Appl. Phys. Lett.* **108**, 213104 (2016).
35. Darulová, J. et al. Autonomous tuning and charge state detection of gate defined quantum dots. *Phys. Rev. Appl* **13**, 054005 (2020).
36. Moon, H. et al. Machine learning enables completely automatic tuning of a quantum device faster than human experts. *Nat. Commun.* **11**, 4161 (2020).
37. Zwolak, J. P., Kalantre, S. S., Wu, X., Ragole, S. & Taylor, J. M. QFlow lite dataset: a machine-learning approach to the charge states in quantum dot experiments. *PLoS ONE* **13**, 10 (2018).
38. Zwolak, J. P. et al. Autotuning of double-dot devices in situ with machine learning. *Phys. Rev. Appl* **13**, 034075 (2020).
39. van Esbroeck, N. M. et al. Quantum device fine-tuning using unsupervised embedding learning. *N. J. Phys.* **22**, 095003 (2020).
40. Teske, J. D. et al. A machine learning approach for automated fine-tuning of semiconductor spin qubits. *Appl. Phys. Lett.* **114**, 133102 (2019).
41. Durrer, R. et al. Automated tuning of double quantum dots into specific charge states using neural networks. *Phys. Rev. Applied* **13**, 054019 (2020).
42. Lennon, D. T. et al. Efficiently measuring a quantum device using machine learning. *NPJ Quant. Inf.* **5**, 79 (2019).
43. Kalantre, S. S. et al. Machine learning techniques for state recognition and auto-tuning in quantum dots. *NPJ Quant. Inf.* **5**, 6 (2019).
44. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. *NeurIPS* **25**, 1097–1105 (2012).
45. Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
46. Camenzind, L. C. et al. Hyperfine-phonon spin relaxation in a single-electron GaAs quantum dot. *Nat. Commun.* **9**, 3454 (2018).
47. Camenzind, L. C. et al. Spectroscopy of quantum dot orbitals with in-plane magnetic fields. *Phys. Rev. Lett.* **122**, 207701 (2019).
48. Brockman, G. et al. OpenAI Gym. Preprint at http://arxiv.org/abs/1606.01540 (2016).
49. Wang, Z. et al. Dueling network architectures for deep reinforcement learning. In *Proc. 33rd International Conference on Machine Learning*. Vol. 48 1995–2003 (JMLR: W&CP, New York, NY, USA, 2016).
50. Schaul, T., Quan, J., Antonoglou, I. & Silver, D. Prioritized experience replay. Preprint at https://arxiv.org/abs/1511.05952 (2016).
51. Kingma, D. P. & Ba, J. L. Adam: A method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980 (2015).

52. Wilcoxon, F. Individual comparisons by ranking methods. *Biometrics Bull.* **1**, 80 (1945).
53. Crippa, A. et al. Level spectrum and charge relaxation in a silicon double quantum dot probed by dual-gate reflectometry. *Nano Lett.* **17**, 1001–1006 (2017).
54. Schupp, F. J. et al. Sensitive radiofrequency readout of quantum dots using an ultra-low-noise SQUID amplifier. *Int. J. Appl. Phys.* **127**, 244503 (2020).
55. Volk, C. et al. Loading a quantum-dot based Qubyte register. *NPJ Quant. Inf.* **5**, 29 (2019).
56. Ares, N. et al. Sensitive radio-frequency measurements of a quantum dot by tuning to perfect impedance matching. *Phys. Rev. Appl.* **5**, 034011 (2016).
57. De Jong, D. et al. Rapid detection of coherent tunneling in an InAs nanowire quantum dot through dispersive gate sensing. *Phys. Rev. Appl.* **11**, 1 (2019).
58. Jung, M., Schroer, M. D., Petersson, K. D. & Petta, J. R. Radio frequency charge sensing in InAs nanowire double quantum dots. *Appl. Phys. Lett.* **100**, 253508 (2012).
59. Vlassis, N. et al. Bayesian Reinforcement Learning. In: Wiering M. & van Otterlo M. (eds) Reinforcement Learning. Adaptation, Learning, and Optimization **13**, 359–386 (Springer, 2012).
60. Azizzadenesheli, K. et al. Efficient exploration through bayesian deep q-networks. *2018 Workshop (ITA) IEEE*. 1–9 (IEEE, 2018).
61. Katt, S., Oliehoek, F. A. & Amato, C. AAMAS '19: Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems. 7–15 (2019).
62. Deisenroth, M. & Rasmussen, C. E. PILCO: A model-based and data-efficient approach to policy search. In *Proc. 28th International Conference on Machine Learning*. 465–472 (Bellevue, WA, USA, 2011).
63. Ayoub, A. et al. Model-based reinforcement learning with value-targeted regression. In *Proc. 37th International Conference on Machine Learning*. 463–474 (PMLR, 2020).
64. Oakes, G. A. et al. Automatic virtual voltage extraction of a 2xN array of quantum dots with machine learning. Preprint at https://arxiv.org/abs/2012.03685 (2020).

ACKNOWLEDGEMENTS

We acknowledge J. Zimmerman and A. C. Gossard for the growth of the AlGaAs/GaAs heterostructure. This work was supported by the Royal Society, the EPSRC National Quantum Technology Hub in Networked Quantum Information Technology (EP/M013243/1), Quantum Technology Capital (EP/N014995/1), EPSRC Platform Grant (EP/R029229/1), the European Research Council (grant agreement 818751), Graphcore, the Swiss NSF Project 179024, the Swiss Nanoscience Institute, the NCCR QSIT, the NCCR SPIN, and the EU H2020 European Microkelvin Platform EMP grant No. 824109. This publication was also made possible through support from Templeton World Charity Foundation and John Templeton Foundation. The opinions expressed in this publication are those of the authors and do not necessarily reflect the views of the Templeton Foundations.

AUTHOR CONTRIBUTIONS

S.B.O., D.T.L., N.A. and the machine performed the experiments. F.V. contributed to the experiment. V.N. and S.B.O. developed the algorithm in collaboration with M.A.O and D.S. The sample was fabricated by L.C.C., L.Y., and D.M.Z. The project was conceived by V.N. and N.A.. G.A.D.B., V.N., S.B.O., and N.A. wrote the manuscript. All authors commented and discussed the results.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41534-021-00434-x.

Correspondence and requests for materials should be addressed to N.A.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.