Controlling Quantum Device Measurement using Deep Reinforcement Learning

Vu Nguyen[1]**, Dominic T. Lennon**[1]**, Hyungil Moon**[1]**, Nina M. van Esbroeck**[1]
Dino Sejdinovic[2]**, G. Andrew D. Briggs**[1]**, Michael A. Osborne**[3]**, Natalia Ares**[1]
[1]Department of Materials, University of Oxford
[2]Department of Statistics, University of Oxford
[3]Department of Engineering Science, University of Oxford
Email: vu.nguyen@materials.ox.ac.uk

Qubits based on semiconductor quantum dot devices are promising building blocks for the realisation of quantum computers. However, measuring and characterising these quantum dot devices can be challenging and laborious for the experimentalists. In this paper, we develop an elegant application using deep reinforcement learning for controlling the measurement of quantum dot devices. Specifically, we present a computer-automated algorithm that measures a map of current flowing through a double quantum dot device for different settings of its gate electrodes. The algorithm seeks particular features called bias-triangles indicating the device is in the right operating regime of realising a qubit. Our approach requires no human intervention and significantly reduces the measurement time. This work alleviates the user effort required to measure multiple quantum dot devices, each with multiple gate electrodes.

Introduction

In quantum computing, information is encoded in quantum bit (or qubits). A qubit is a two-state quantum-mechanical system, embodying the superposition that is peculiar to quantum mechanics. Success in quantum computing relies on high fidelity physical qubits which can be realised in many material systems [1–3]. One of the most promising is gate-defined semiconductor quantum dots [4,5]. In these devices, qubits are encoded in electron spins, which are confined and controlled by gate electrodes [6–10].

However, operating such quantum dot devices can be challenging and time-consuming. This is because semiconductor quantum devices operate using individual electrostatic gates with a relatively large gate voltage range. In addition, the quantum dot systems are often under the unavoidable noise dynamics. The correct gate voltages setting for the targeted bias-triangles can vary with time, temperature and environment factors in the device. The voltages applied to these gates must be carefully set to produce an electrostatic confinement potential so that a qubit can be realised [4,11]. The current practice of characterising the gate voltage parameter space is time-consuming, with the decision of how to adjust the gate voltages made by the experimentalists, based on experience. However, in the huge search space of gate voltages, such ideal quantum dots are like finding a needle in a haystack.

Recent progress in using computer-support methods to tune quantum dot devices has been demonstrated [5,12–16]. There have also been first steps towards automating device measurements using machine learning (ML) [17]. However, the potential of reinforcement learning in device measurement is unexplored. Tremendous progress in machine learning (ML) algorithms suggests that such techniques may be used to accelerate the experimental control from a de novo device to a fully controlled device, replacing the gross-scale heuristics, developed by experimentalists to deal with tuning of parameters particular to experiments.

The emerging field of deep reinforcement learning [18] has led to remarkable empirical successes in rich and varied domains like robotics [19], strategy games [20,21], and multi-agent interaction [22,23]. Existing literature in

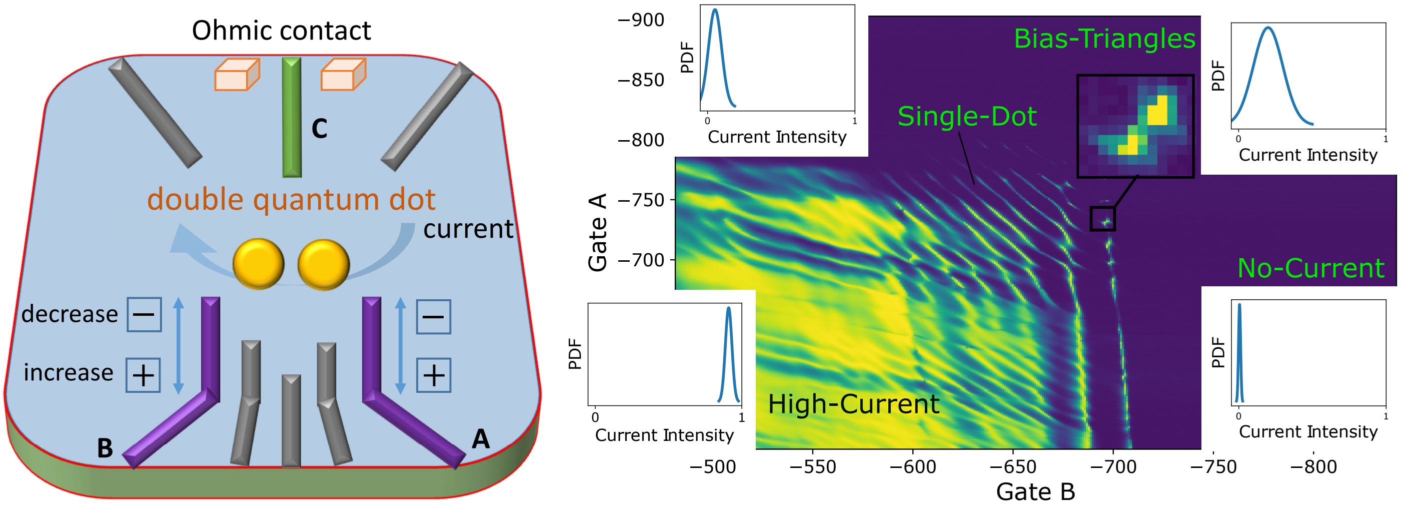

Figure 1: Left: While the device has eight gate electrodes, we use two main gates (pink color) and an additional gate (green color) that control the quantum dot charge states, labelled as Gate *A*, Gate *B* and Gate *C*. Right: An example of the electric current measurement as a function of two gates. The bias-triangles (in a double quantum dot regime) are our target of interest. The unit in each axis is mV. The target bias-triangles (Right) is found after the current going through the double quantum dot region (Left).

quantum technologies has initially used deep reinforcement learning for controlling quantum logical gates [24,25]. To the best of our knowledge, reinforcement learning has never been used to control the real time measurements of a quantum device where the automated decision making in RL can be beneficial for the experimentalists.

We develop a new paradigm using deep reinforcement learning for controlling the measurement process of quantum gate electrode voltages that currently relies on human heuristics. Our algorithm can make the optimal decisions of which regions to measure next to find the bias-triangles using the fewest number of measurements. Thus, our approach is efficient in that it selectively measures a small region of the gate voltage space. This optimal decision establishes a closed-loop system for experimental initialisation without the need for human intervention. Our major contribution can be summarised as twofold. (1) We develop the first deep reinforcement learning for controlling the quantum device measurement in real time. (2) We identify the statistical features representing different states of the quantum device which are robust across different quantum device architectures. (3) We extend and demonstrate the applicability of our system in controlling three gate voltages. Bringing machine learning to automate discovery rather than using a brute-force approach has the potential to substantially accelerate scientific progress in quantum technology.

The quantum dot device

Our quantum dot devices are defined in 2-dimensional electron gas by Ti/Au gates electrodes on top of a GaAs/AlGaAs heterostructures [26,27], as shown in Fig. 1. By applying a negative voltage to the gate electrodes, the electric field depletes the electron gas creating a confinement potential for the double quantum dot. The confinement potential is controlled by the gate voltages to create two regions (one for each quantum dots) of few nanometre size, relatively isolated from the environment by electrostatic barriers. The small size of the quantum dot leads to a significant charging energy that defines discrete energy levels for electrons in both quantum dots. The current flowing thought the double quantum dot depends on the strength of the barriers and the energy levels in the quantum dots [28]. It is maximal when an energy level of both quantum dot aligns with the electrochemical potential of the ohmic contacts.

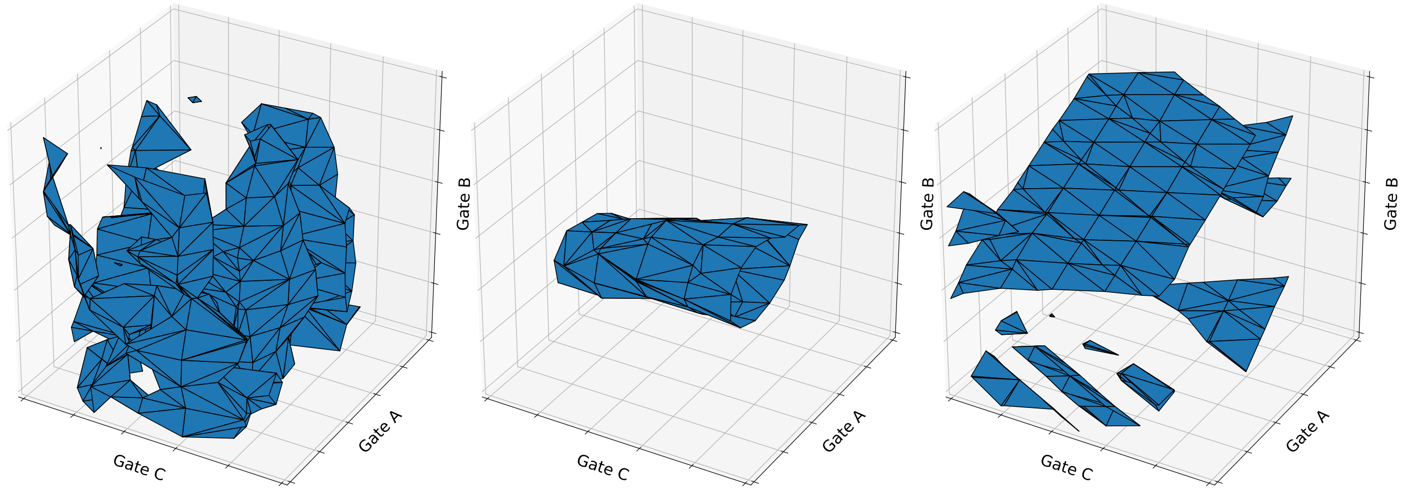

Figure 2: We illustrate the examples of 3D block measurement by varying the three gate voltages. The bias-triangles in 3D can be seen in the middle plot where there is an isolation of the current with respect to Gate A and Gate B. Our approach will be more efficient to decide where to measure next given such 3D object while a human expert can encounter difficulty.

The measurements are performed in real time on a double quantum dot device to obtain the desired property of the bias-triangles, an important feature characterising qubit. We measure the current flowing through the device as a function of gate voltages. We can optionally use two or three gate voltages to control the device. When a potential that defines a bias-triangles is formed, gates A and B shift the electrostatic potential inside quantum dots where gate A handles the right dot and gate B handles the left dot. An example of the bias-triangles behaviour using two gates is shown in Right Fig. 1 and using three gates in Middle Fig. 11. Within the large and complicated surrounding area, this bias-triangles is difficult to control. In addition, this bias-triangles, as well as the current features, is not static in the gate voltage space. Instead, it is shifting and evolving over the course of the experiments on two-electron spin qubits presented in [29–33].

In Right Fig. 1, we depict four different quantum regions (aka regimes [28]) of *no-current*, *high-current*, *single-dot* and *bias-triangles* with distinctive statistical representations. Particularly, we use a univariate Gaussian distribution to represent the density (or histogram) of electric current at each quantum state region. These regions can be represented as follows. No-current is the region where we have zero intensity mean μ (or very low current due to noise) and zero (or very low) standard deviation σ. High-current is the region with high current and low standard deviation. Single-dot is the mixed region with and without current. The bias-triangles constitute our target of interest as shown in Fig. 1.

Due to the property of the gate movement (see Fig. 1), we can not suddenly jump across different places in the gate voltage space. Instead, the measurement process needs to be done step by step sequentially. In addition, measuring the electric current throughout the gate voltage space is time consuming wherein fast and large changes in gate voltage should be avoided. This measurement control problem turns out to be finding target bias-triangles using the fewest number of measurements in the large gate voltage space. Alternatively, our task can be seen as the advanced version of the famous Grid World task in DRL [18].

Building quantum environment for training deep reinforcement learning. In DRL, the agent will interact with the environment to gain experiences. Therefore, we build an environment from the quantum dot device for training our algorithm and name it as *Quantum Environment* (QE). Our designed QE follows the setting of Open Gym AI [34] with functionalities and interfaces. This is ready to be used for benchmarking and training

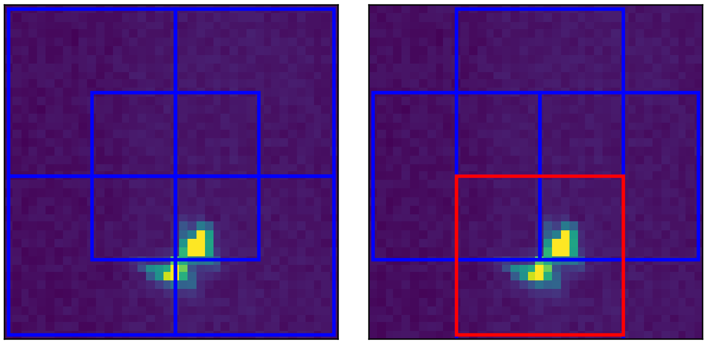

Figure 3: We represent a state of the electric measurement by 9 overlapping blocks represented by the blue squares (5 in the left and 4 in the right). A block with the bias-triangles is in red. A special property of the bias-triangles region is that only one or two blocks have the bias-triangles and the remainers are no-current.

existing DRL algorithms. In addition, this environment is useful for interested DRL researchers to develop their methods for improving quantum technologies.

To construct this environment, we make multiple measurements of electric current maps as shown in Fig. 1. We have considered a gate voltage space of 360mV ×360mV. For each electric current map, the quantum physicists will annotate which location has bias-triangles property. These ground truth locations in gate voltages space (or states) are marked in the environment. Then, we train the DRL algorithm starting at different locations in the environments to reach this marked locations.

State. A state s is an electric current measurement given gate voltages. We note that we do not use the gate voltage values, but the electric measurement. This is to mitigate the effect of switches in the measurement. Instead of using a single image, we construct multiple overlapping blocks to represent each state. This design is to ensure that our desired target of bias-triangles does not lie in the border between blocks and is not effected by the noise from the device in which the electric current is evolving over time. Specifically, we define each block size of 18×18 mV[1] to fully capture the target bias-triangles. In our environment, given different size and position of the blocks, the state feature could be different.

Instead of making densely overlapping blocks by a moving kernel horizontally and vertically, we propose to represent each state by 3 blocks per dimension for simplicity. In other words, the image is represented as 3^d number of the blocks where d is the dimension. The dimension d corresponds to the number of gates used. In our setting, each state in two gates includes 9 blocks as the tensor of $9 \times 18 \times 18$ dimensions and three gates will have the tensor of $27 \times 18 \times 18$. Examples of the state and blocks using two gates can be found in Fig. 3.

After defining the state as an electric current above, we propose to use the statistical feature summarising the electric current magnitude μ and standard deviation σ. The second design of the state will bring two major advantages. The first benefit is from our device property that the raw electric measurement can vary significantly across the devices while such statistical estimation is more robust. The second benefit is for scalability that we can extend to higher number of dimensions using the randomly sampling the block rather than a full scan which scales exponentially with the dimensions. Under this representation, the state includes a feature vector of 9×2 dimensions.

To prevent from shifting effect and from the situation where the bias-triangles can locate in the boundary, we set 50% overlapping between neighboring states. Since the state can also be defined using more than two

[1] We empirically found this is the good parameter.

gates, we illustrate the examples of the 3D states using three gates in Fig. 2 in which we depict the bias-triangles in the Middle as the isolated feature.

Actions. Our action space includes increasing $(+)$ or decreasing $(-)$ each gate voltage. We have specially designed two actions to modify both gates simultaneously as shown in Fig. 5. This is inspired from the physical property in the our device. In higher dimensional setting, such as controlling $d > 2$ gates, this action space can be generalized wherein the number of actions is $2 \times d + 2$.

Reward. We assign high reward to our target state of bias-triangles and vice versa. We encourage the algorithm to find the bias-triangles using the fewest number of measurement by designing the reward score as follows. We assign the highest reward $r = 10$ to the bias-triangles location. This reward score for the target state is provided by the domain expert at a few selected places during training. Then, other states will take $r = -1$. In addition, to prevent from repetition in measurement which is wasteful, any action which leads to the location (i, j) revisited will take the penalised reward $r = -10 \times n_{i,j}$ where $n_{i,j}$ indicates the number of time this location (i, j) has been visited. The maximum number of steps per episode during training is set as 100. Beyond this threshold, if the algorithm can not find the bias-triangles, it will terminate and assign $r = -10$. The maximum number of steps controls how far away from a starting point the device-measurement can go.

Results

We present a deep reinforcement learning algorithm for controlling the measurement process in the quantum dot device. We aim to minimise the number of required measurement to find the bias-triangles, a desirable feature to realise a spin qubit. In addition, our algorithm can efficiently operate without human intervention.

We summarise an overview of the algorithm in Fig. 4. The proposed framework consists of three steps: (1) starting measurement at the quantum device at some gate voltages (either random or pre-specified), (2) determining the next measurement and changing the gate voltages accordingly, and (3) performing the measurement. The steps (2) and (3) are repeated until we find the desired bias-triangles. We aim to keep the number of measurement as the fewest as possible.

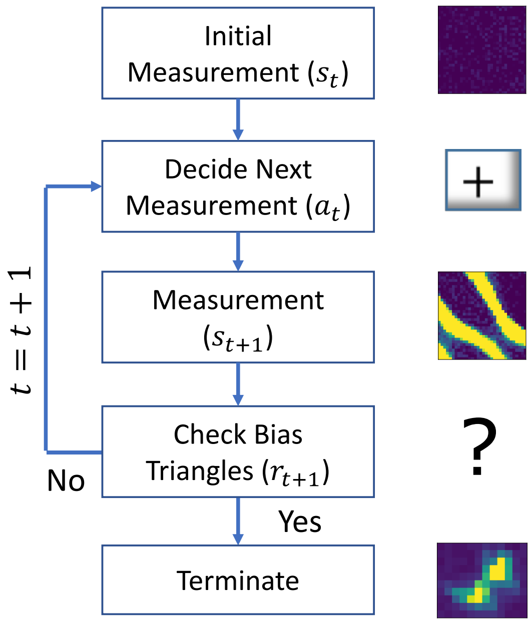
Initial Measurement (s_t)
Decide Next Measurement (a_t)
+
Measurement (s_{t+1})
Check Bias Triangles (r_{t+1})
?
$t = t + 1$
No
Yes
Terminate

Figure 4: Algorithm summary. The image is the electric current scanned given two gate voltages in the device. The algorithm aims to use the fewest number of measurement to find the target bias-triangles.

Deep Reinforcement Learning. We consider a sequential decision making setting, where an agent (e.g., computer algorithm) interacts with a Quantum Environment over discrete time steps. We refer the interested readers to [18,35] for an elegant introduction of reinforcement learning. In our measurement control problem, the agent perceives an electric measurement as a state s_t at time step t. The agent then chooses a next measurement by selecting an action from a discrete set a_t from 6 possible actions (using two gates) or from 8 possible actions (using three gates) and observes a reward signal r_t indicated how good or bad the decision a_t (given the state s_t) is. We aim to learn an optimal policy $\pi(a \mid s)$ which fully defines the behavior of an agent over actions given states.

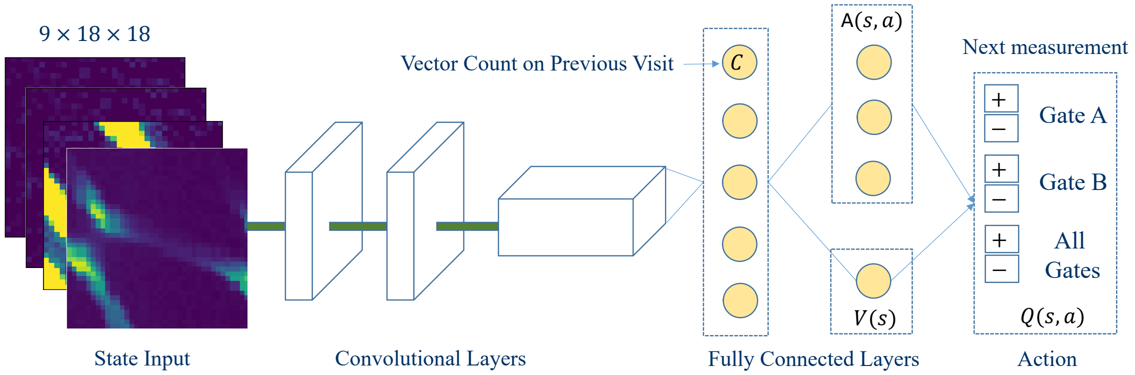

Figure 5: Our deep reinforcement learning framework. The vector count C representing the 6 dimensional adjacent matrix is included in one of the last layer to discourage from revisiting the previous locations.

In deep reinforcement learning, we construct a neural network to approximate the input state s and produce the Q-values for every action in the action space [19,36,37]. The neural network is used to learn the parameters over multiple episodes (iterations) so that when the training is done, we get this trained network to predict the next best action to take in the environment [18].

Duelling deep Q network. We consider a particular attractive form of model-free strategy [36] which means that no explicit model of the state-transition dynamics is estimated during computation of the policy. Among different version of DRL algorithms, we consider the duelling architecture [38] to train the algorithm. The key insight behind duelling architecture is that for many states, it is unnecessary to estimate the value of each action choice. For example, knowing whether to move left or right only matters when the target bias-triangles is nearby. In some states, it is of significant importance to know which action to take, but in many other states the choice of action has less impact on what happens – the case in our quantum device when the measurement is at the empty current region (no-current) or full current region (high-current).

The module that combines the two streams of fully connected layers to output a Q estimate requires thoughtful design [38]. From the expressions for advantage $Q^{\pi}(s,a) = V^{\pi}(s) + A^{\pi}(s,a)$ and state-value $V^{\pi}(s) = \mathbb{E}_{a\sim\pi(s)}[Q^{\pi}(s,a)]$, it follows that $\mathbb{E}_{a\sim\pi(s)}[A(s,a)] = 0$. To address the issue of identifiability in the sense that given Q we cannot recover V and A uniquely, we can force the advantage function estimator to have zero advantage at the chosen action. That is, the duelling DQN [38] defines the Q function as $Q(s,a) = V(s) + A(s,a) - \max_{a'} A(s,a')$to increase the stability of the optimisation.

To discourage from reselecting the visited locations, we have defined the count statistics over the number of previous visit (denoted as C) around neighboring locations and concatenate it into the last fully connected layer as shown in Fig. 5. For this purpose, the dimension of C is equal to the number of actions.

Optimal decision given the input measurement. Given the estimated policy π, we make the next measurement as the optimal decision given the electric current s_t of the block (step 3 in Fig. 4) is as $a_t = \arg\max_{a'} Q^{\pi}(a', s_t)$. This estimation is the forward computation in the DRL framework. This decision can be made either in the same quantum device used for training or in a different quantum device. For stable training, we have made use of the prioritised experience replay technique [39].

Deciding when to stop measurement. We stop the measurement when the maximum number of measurement is reached or when the target of bias-triangles is detected. This becomes a binary classification problem described in the appendix.

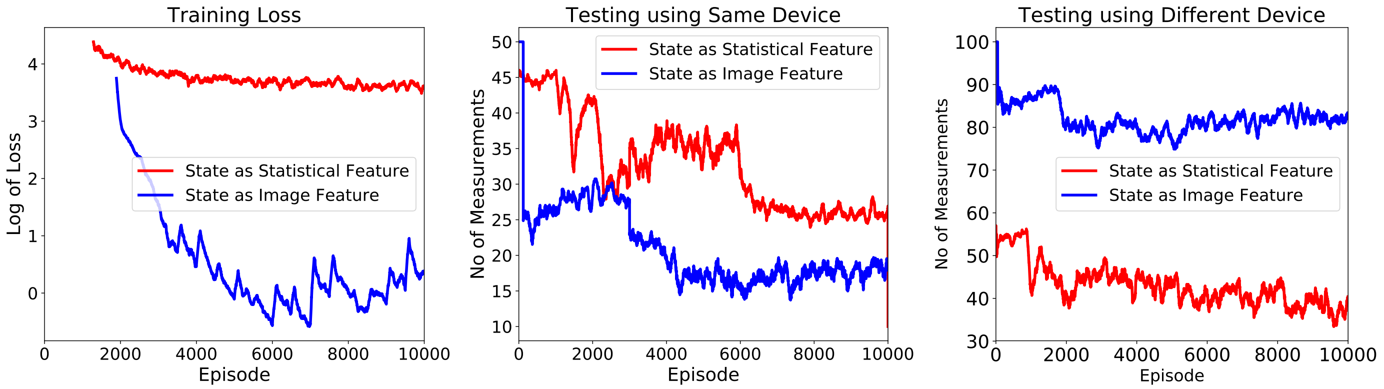

Figure 6: The training loss and number of measurements comparison on the two versions of our framework. Our CNN version for image state is overfitted in the training device (Left) and performs worse in the test device (Right) comparing to the case of statistical feature state.

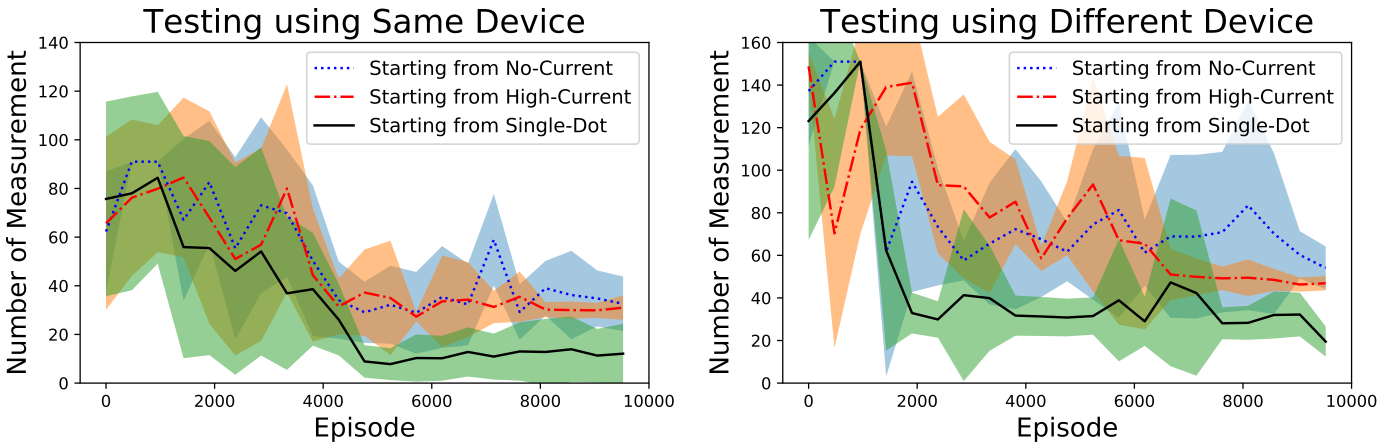

Figure 7: We evaluate the performance of our algorithm using statistical feature with different starting locations. The performance is recorded on the same quantum device in which training is performed (Left) and a different device (Right). This demonstrates that our algorithm is flexible and robust against device variability.

We present our network architectures in Fig. 5 and Fig. 13 in the appendix where we have two versions: the convolutional neural network (CNN) version for state as *image features* or fully connected (FC) version for state as *statistical features*.

Our primary focus in this measurement control problem is to find the bias-triangles in the gate voltage space. For efficiency, we aim to use the fewest number of measurement, i.e. spend the minimum measurement time, instead of measuring the electric current in the entire gate voltage space. Therefore, we use the number of measurement as the main criteria for evaluation. Here, each measurement refers to scan a small block (or a state) by varying the gate voltages. We first present the experimental results using two gates and then three gates. Without explicitly stated, by defaults the experiments are considered using two gate voltages.

Experiment setting. We implement our system in Python with Tensorflow. The algorithm is trained on a computer with GPU Titan V 32GB of RAM. The training process takes approximately $3-4$ hours for two gates and $8-9$ hours for three gates. We summarise the deep learning architecture and hyperparameters in Table 1 in the appendix where we have two separate DRL models using CNN [40] for state as image feature and

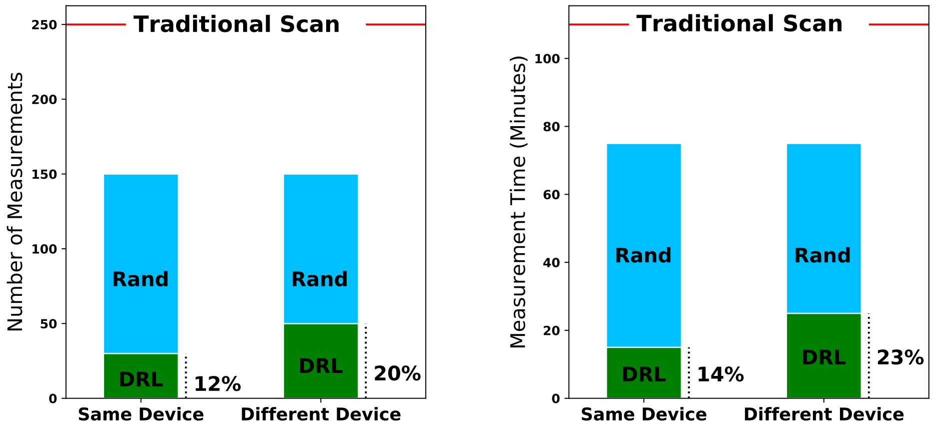

Figure 8: DRL significantly reduces the number of required measurements. Comparing to the traditional scan, DRL takes only 20% of the measurements on the test device, which implies a reduction on measurement time of 23%. Moreover, our DRL does not require a human intervention as opposed to the traditional scan.

fully connected layers for state as statistical features.

Training convergence. Each episode length is defined up to a maximum of 100 steps unless it is terminated earlier after the bias-triangles is found. The algorithm is converged after training over $10,000$ episodes (iterations). We illustrate the convergence of our algorithm by showing the training loss reducing over time in Left Fig. 6. In addition, we estimate the number of required measurement in the training device which is converging to 20 steps in Middle Fig. 6.

Comparison of image features vs statistical features. Fig. 6 shows that our DRL model using CNN version with state as image feature is overfitted to the training environment. Although it produces lower training loss than the fully connected (FC) version, the performance of CNN version is not robust in the testing device.

Testing from different regions. We next consider the measurement efficiency from different starting locations including no-current, high-current and single-dot regions defined previously. In Fig. 7, the results are averaging using 20 different locations from the above regions. Due to the property of the single-dot region which is located closer to the bias-triangles, the required number of steps is the fewest (black line). The number of measurement for starting locations from no-current (blue) and high-current (red) are somewhat similar, around 50.

Testing from different devices. To demonstrate the generality and flexibility of our approach, we consider testing the performance of our trained DRL agent in a new quantum device in Right Fig. 7. Although the performance slightly drops, the number of required measurements is still low comparing to the traditional approach of grid scan, as shown in Fig. 8. Our DRL agent takes only 12% number of measurements comparing to the traditional baseline (of whole scanning) when testing in the same device and 20% when testing in the different quantum device. Comparing to a random policy where the action is randomly taken, we achieve approximately 25% improvement on the same device and 33% improvement on the testing device. We further highlight that the traditional grid scan requires a human expert to find where is the bias-triangles, while using our approach does not require human intervention.

Visualising the measurement trajectories. We illustrate the measurement process by plotting the trajectory starting from different locations in the gate voltage space. Although these three trajectories are different, they finally find the bias-triangles without human intervention and without measuring the entire gate voltage

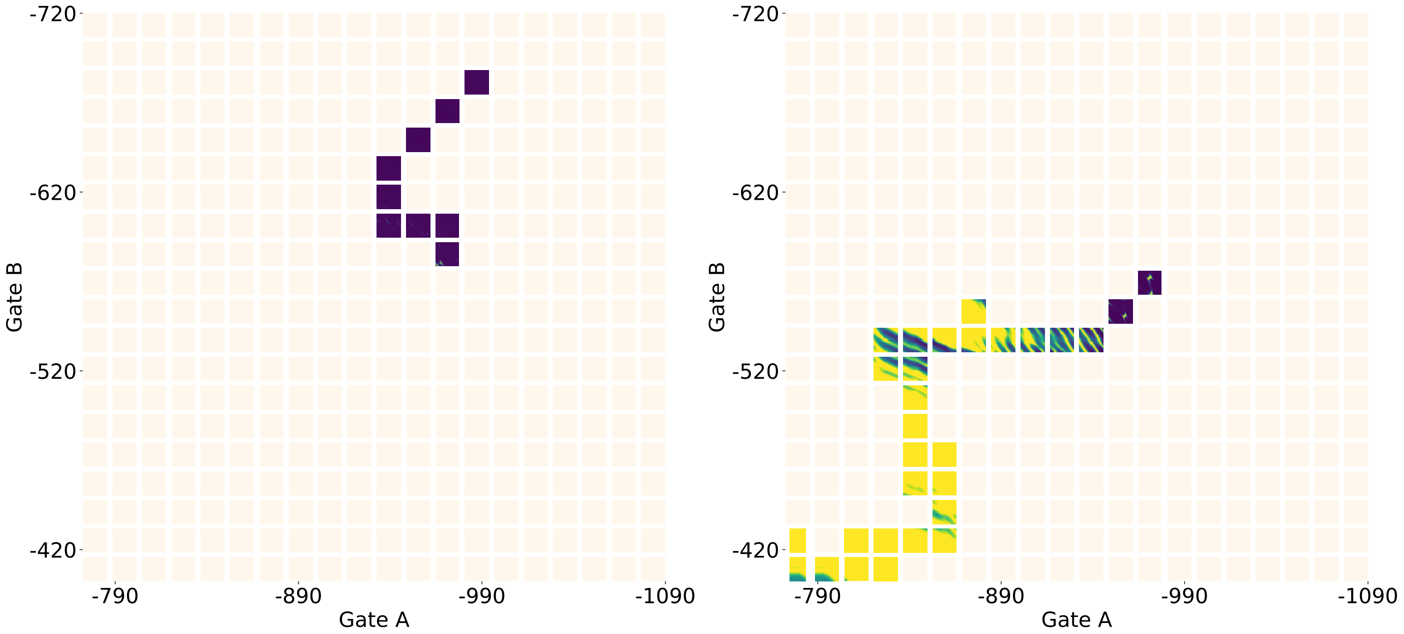

Figure 9: Example of two measurement trajectories starting at different locations in the gate voltage space. Based on the state feature, the algorithm will decide the next measurement until reaching the bias-triangles.

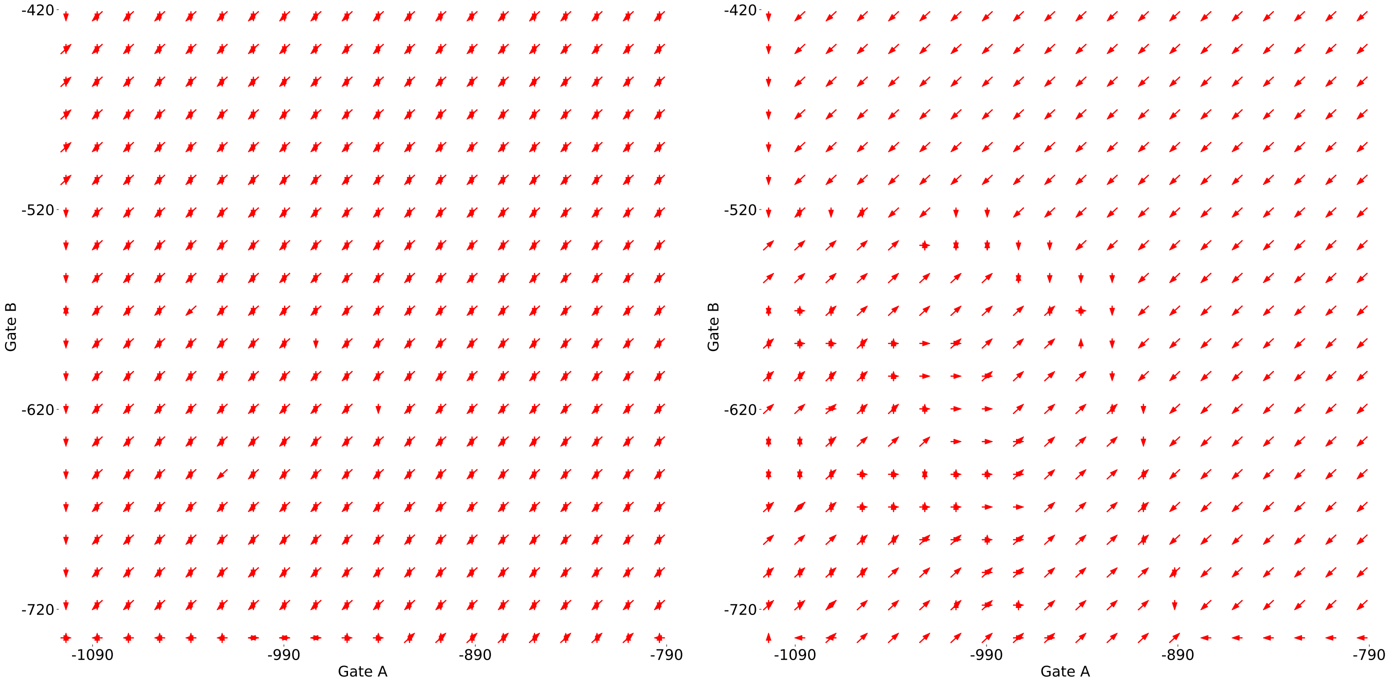

Figure 10: We plot the optimal policies learned at early stage and later stage of the training process. The arrows indicates, at the given location, the optimal direction to move. Two arrows represent two probable actions which both with high chances of being optimal.

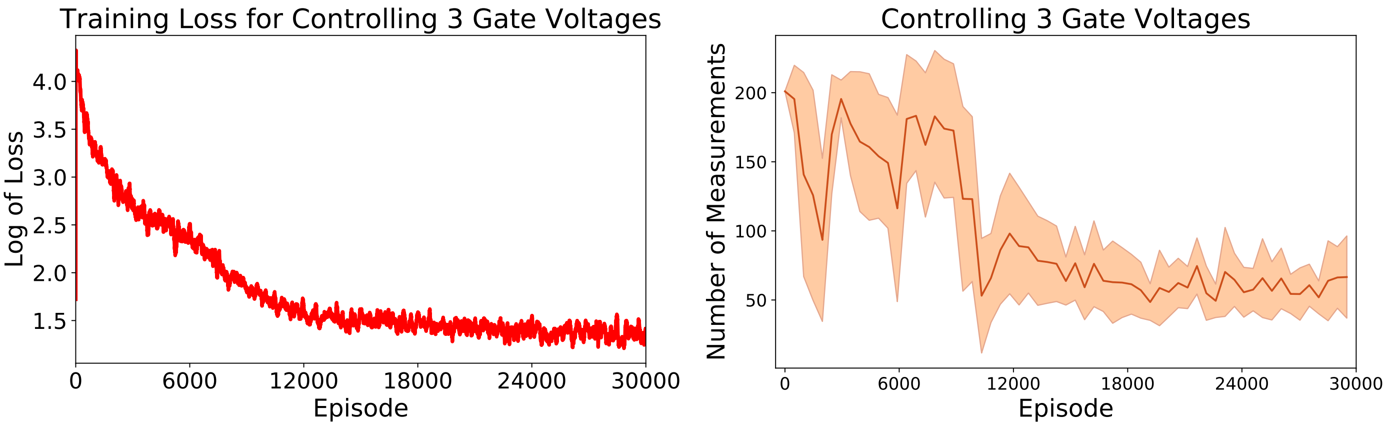

Figure 11: The experiments on controlling the measurement through 3 gate voltages over 30,000 episodes (iterations). Left: the convergence in the training loss. Right: the average number of measurement to reach the bias-triangles target over 20 independent runs.

space. By choosing appropriate voltage values for the gate electrodes, the device could operate properly to form the bias-triangles, as shown in Fig. 1.

Illustrating the optimal policy. In the reinforcement learning context, a policy defines what an agent does to accomplish a task. We present the optimal policies at different training stages in Fig. 10 wherein we use arrows to indicate the action, i.e., the direction to move in the gate voltage space to perform the next measurement. The algorithm learns that it should move into bottom left (more positive gate voltages) if the state is no-current or go into top right (more negative gate voltages) if the quantum state is high-current. In Fig. 10, we have placed the bias-triangles at the center of the plot for convenience in visualisation. In practice, this location of interest is unknown.

Controlling more gate voltages. In the previous experiments, we have demonstrated the capacity of our model on controlling two gates. We now further extend our model to handle three gate voltages as marked in Left Fig. 1 where the additional Gate C is used. We note that training our algorithm on three gate voltages requires more number of iterations (episodes) than on the two gate case. That is, our algorithm in three gates takes 30,000 episodes for convergence comparing to 10,000 episodes in two gates, as shown in Fig. 11. At the convergence, the average number of required measurement is 70 for reaching the bias-triangles location.

This experiment demonstrates the applicability of the proposed approach in controlling the measurement using more number of gates. Due to the complex shape and feature in high dimensional setting, it poses challenges for human expert to measure and decide the next measurement. On the other hand, the machine learning can learn and generalise well in high dimension. This step highlights the potential toward automatic control in higher number of gate voltages beyond the human's ability.

Since the gate voltage space increases exponentially with the number of used gates, measuring in three gates space is always harder and time consuming than in two gates. We present the measurement comparison of different approaches in using two and three gate voltages in Fig. 12. The experimental results show that our DRL approach can scale well in controlling more number of gates. That is, the number of required measurements using our approach does not significantly go up while the Random policy and the traditional baseline will seriously fail through. In other words, our DRL for 3 gates can be 40 times more efficient than the traditional scan and 10 times than the Random policy. We note that each block measurement in 3D will take approximately 12 minutes. Due to the expensiveness of measurement, this comparison on three gate voltages

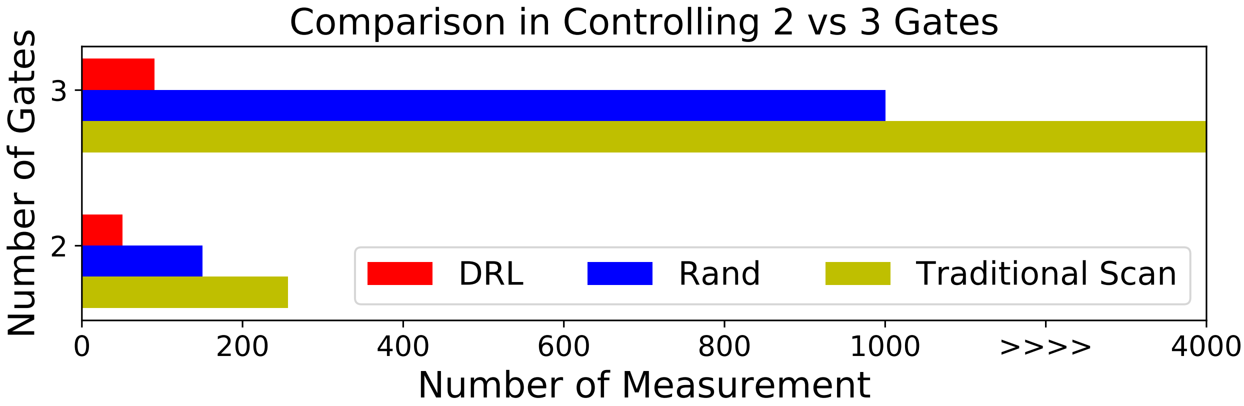

Figure 12: The relative comparison of measuring the device using 2 vs 3 gate voltages. The gate voltage space increases exponentially with the number of used gates. We show that our DRL can scale well to higher number of gates while the Random policy and traditional scan will significantly suffer the curse of dimensionality.

is done using our built Quantum Environment that reflects well our real device.

Discussion

We have presented novel techniques towards controlling the measurement of double quantum dot devices. Given that building scalable quantum computing devices is now on the horizon, we hope that such methods will present themselves as natural prerequisite for construction of real devices against the rely on human heuristics. Looking further, we shall control multiple gate voltages and multiple dots will present new challenges as a result of the higher dimensional space of gate voltages. This curse of dimensionality will seriously pose challenges for the experimentalists and be in need of the automatic discovery and control by machine learning.

Concluding Remarks

We have developed a deep reinforcement learning approach for controlling the measurement of the double quantum dot device. Our algorithm can identify the desired bias-triangles using the fewest number of measurements. This is a significant step toward enabling fully automated procedure for characterising robust qubit - a building block in quantum computer. Our algorithm is a key contribution to the development of scalable quantum technologies substantially. Moreover, we have contributed the Quantum Environment to facilitate interested researchers in quantum technologies.

References

1. Brown, K. R., Kim, J. & Monroe, C. Co-designing a scalable quantum computer with trapped atomic ions. *npj Quantum Information* **2**, 16034 (2016).
2. Neill, C. *et al.* A blueprint for demonstrating quantum supremacy with superconducting qubits. *Science* **360**, 195–199 (2018).
3. Li, R. *et al.* A crossbar network for silicon quantum dot qubits. *Science advances* **4**, eaar3960 (2018).
4. Awschalom, D. D., Bassett, L. C., Dzurak, A. S., Hu, E. L. & Petta, J. R. Quantum spintronics: engineering and manipulating atom-like spins in semiconductors. *Science* **339**, 1174–1179 (2013).
5. Baart, T., Eendebak, P., Reichl, C., Wegscheider, W. & Vandersypen, L. Computer-automated tuning of semiconductor double quantum dots into the single-electron regime. *Applied Physics Letters* **108**, 213104

(2016).

6. Loss, D. & DiVincenzo, D. P. Quantum computation with quantum dots. *Physical Review A* **57**, 120 (1998).
7. Maune, B. M. *et al.* Coherent singlet-triplet oscillations in a silicon-based double quantum dot. *Nature* **481**, 344 (2012).
8. Veldhorst, M. *et al.* An addressable quantum dot qubit with fault-tolerant control-fidelity. *Nature nanotechnology* **9**, 981 (2014).
9. Kawakami, E. *et al.* Electrical control of a long-lived spin qubit in a si/sige quantum dot. *Nature nanotechnology* **9**, 666 (2014).
10. Veldhorst, M. *et al.* A two-qubit logic gate in silicon. *Nature* **526**, 410 (2015).
11. Nichol, J. M. *et al.* High-fidelity entangling gate for double-quantum-dot spin qubits. *npj Quantum Information* **3**, 3 (2017).
12. Van Diepen, C. *et al.* Automated tuning of inter-dot tunnel coupling in double quantum dots. *Applied Physics Letters* **113**, 033101 (2018).
13. Botzem, T. *et al.* Tuning methods for semiconductor spin qubits. *Physical Review Applied* **10**, 054026 (2018).
14. Kalantre, S. S. *et al.* Machine learning techniques for state recognition and auto-tuning in quantum dots. *npj Quantum Information* **5**, 6 (2019).
15. Teske, J. D. *et al.* A machine learning approach for automated fine-tuning of semiconductor spin qubits. *Applied Physics Letters* **114**, 133102 (2019).
16. Volk, C. *et al.* Loading a quantum-dot based qubyte register. *npj Quantum Information* **5**, 29 (2019).
17. Lennon, D. *et al.* Efficiently measuring a quantum device using machine learning. *npj Quantum Information* **5**, 1–8 (2019).
18. Sutton, R. S. & Barto, A. G. *Reinforcement learning: An introduction*, vol. 1 (MIT press Cambridge, 1998).
19. Lillicrap, T. P. *et al.* Continuous control with deep reinforcement learning. *International Conference on Learning Representations (ICLR)* (2015).
20. Mnih, V. *et al.* Playing atari with deep reinforcement learning. *NIPS Deep Learning Workshop* (2013).
21. Mnih, V. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529 (2015).
22. Foerster, J., Assael, I. A., de Freitas, N. & Whiteson, S. Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*, 2137–2145 (2016).
23. Palmer, G., Tuyls, K., Bloembergen, D. & Savani, R. Lenient multi-agent deep reinforcement learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 443–451 (2018).
24. An, Z. & Zhou, D. Deep reinforcement learning for quantum gate control. *arXiv preprint arXiv:1902.08418* (2019).
25. Niu, M. Y., Boixo, S., Smelyanskiy, V. N. & Neven, H. Universal quantum control through deep reinforcement learning. *npj Quantum Information* **5**, 33 (2019).
26. Amasha, S. *et al.* Electrical control of spin relaxation in a quantum dot. *Physical review letters* **100**, 046803 (2008).
27. Camenzind, L. C. *et al.* Spectroscopy of quantum dot orbitals with in-plane magnetic fields. *Physical review letters* **122**, 207701 (2019).
28. Hanson, R., Kouwenhoven, L. P., Petta, J. R., Tarucha, S. & Vandersypen, L. M. Spins in few-electron quantum dots. *Reviews of modern physics* **79**, 1217 (2007).
29. Bluhm, H. *et al.* Dephasing time of gaas electron-spin qubits coupled to a nuclear bath exceeding 200 μs. *Nature Physics* **7**, 109 (2011).
30. Shulman, M. D. *et al.* Demonstration of entanglement of electrostatically coupled singlet-triplet qubits. *Science* **336**, 202–205 (2012).
31. Shulman, M. D. *et al.* Suppressing qubit dephasing using real-time hamiltonian estimation. *Nature communications* **5**, 5156 (2014).

32. Nichol, J. M. *et al.* Quenching of dynamic nuclear polarization by spin–orbit coupling in gaas quantum dots. *Nature communications* **6**, 7682 (2015).
33. Botzem, T. *et al.* Quadrupolar and anisotropy effects on dephasing in two-electron spin qubits in gaas. *Nature communications* **7**, 11170 (2016).
34. Brockman, G. *et al.* Openai gym. *arXiv preprint arXiv:1606.01540* (2016).
35. Kaelbling, L. P., Littman, M. L. & Moore, A. W. Reinforcement learning: A survey. *Journal of artificial intelligence research* **4**, 237–285 (1996).
36. Arulkumaran, K., Deisenroth, M. P., Brundage, M. & Bharath, A. A. Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine* **34**, 26–38 (2017).
37. Henderson, P. *et al.* Deep reinforcement learning that matters. In *Thirty-Second AAAI Conference on Artificial Intelligence* (2018).
38. Wang, Z. *et al.* Dueling network architectures for deep reinforcement learning. In *International Conference on Machine Learning*, 1995–2003 (2016).
39. Schaul, T., Quan, J., Antonoglou, I. & Silver, D. Prioritized experience replay. *International Conference on Learning Representations* (2016).
40. Krizhevsky, A., Sutskever, I. & Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 1097–1105 (2012).

Author Contributions

V.N. developed the algorithm and performed the experiment. D.T.L, H.M. and N.M.v.E. supported V.N. for the experiments. The project was conceived by D.S., G.A.D.B., M.A.O. and N.A. All authors discussed results and contributed to the manuscript .

Competing Financial Interests

The authors declare no competing financial interests.

Data Availability

The data that support the findings of this study are available from the corresponding author.

Acknowledgments

The quantum dot devices were provided by L.C. Camenzind, Liuqi Yu and D.M. Zumbühl at the Department of Physics, University of Basel, Switzerland. We acknowledge discussions with E. A. Laird and F. Vigneau. We acknowledge J. Zimmerman and A. C. Gossard for the growth of the AlGaAs/GaAs heterostructure. This work was supported by the EPSRC National Quantum Technology Hub in Networked Quantum Information Technology (EP/M013243/1), Quantum Technology Capital (EP/N014995/1), Nokia, Lockheed Martin, the Swiss NSF Project 179024, the Swiss Nanoscience Institute and the NCCR QSIT. 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. We gratefully acknowledge the support of the NVIDIA Corporation with the donation of one NVIDIA TITAN V GPU for our research.

Table 1: Deep Reinforcement Learning Architecture

CNN Version	
Learning Rate	$2.5e-6$
Conv Layers 1	$32,8,4$
Conv Layers 2	$32,4,2$
Conv Layers 3	$32,4,2$
FC Layers	$64,32$
FC Layers (Dueling)	$64,1$

Common Parameters	
Discount Factor	0.5
Optimizer	Adam
Number of Episodes for 2 gates	$10,000$
Number of Episodes for 3 gates	$30,000$
Mini batch-size	32
Decay rate in ε greedy	$1e^{-4}$
Replay buffer	20000
PER-β (start, final, no steps)	$(1.0, 0.6, 1000)$

Fully Connected Version for 2 gates	
Learning Rate	$2.5e-6$
FC Layers	$128,64,32$
FC Layers (Dueling)	$64,1$

Fully Connected Version for 3 gates	
Learning Rate	$2.4e-6$
FC Layers	$128,64,32,32$
FC Layers (Dueling)	$64,1$

Code Availability

All code required to replicate the results presented in this paper will be made publicly available on Github.

Supplementary Materials

In the appendix, we first summarise the network architecture and hyperparameters used in Table 1. We further illustrate the model architecture in Fig. 13. We then summarise all steps for training our DRL agent in Algorithm 1. Finally, we present the classification step used to decide when to stop the algorithm. We stop the measurement when the bias-triangles is detected or when the maximum number of measurement is reached. This becomes a binary classification problem of bias-triangles found or not. Since we have built two models for image feature and statistical feature, we develop two classifiers to detect the bias-triangles as follows.

A convolutional neural network for classifying image feature. Given image feature of raw measurement, we train a CNN to recognise images. In our quantum experiment, we have a limited set of bias-triangles observations while we have a bunch of negative examples. Moreover, our bias-triangles may exist in a variety of conditions, such as different locations, scales, brightness etc. We account for these cases by training our network with additional synthetically modified data. That is, we make minor alternations to our existing bias-triangles observations, each of which is a block of image. Minor changes include scalings, translations and rotations. This essentially is the premise of data augmentation. Our augmentation parameters are as follows. We scale the image inward 0.9 and outward 1.1 of the original images. We translate the image in four directions within 20% of size. We rotate the images of $90^{\circ}, 180^{\circ}$ and 270°.

After obtaining the training data by image augmentation, we build a CNN model with the parameters and architecture presented in Table 1.

A Kullback–Leibler divergence for classifying statistical feature. We can use the statistical features to represent each state. That is, we estimate the univariate Gaussian distribution using the electric current value.

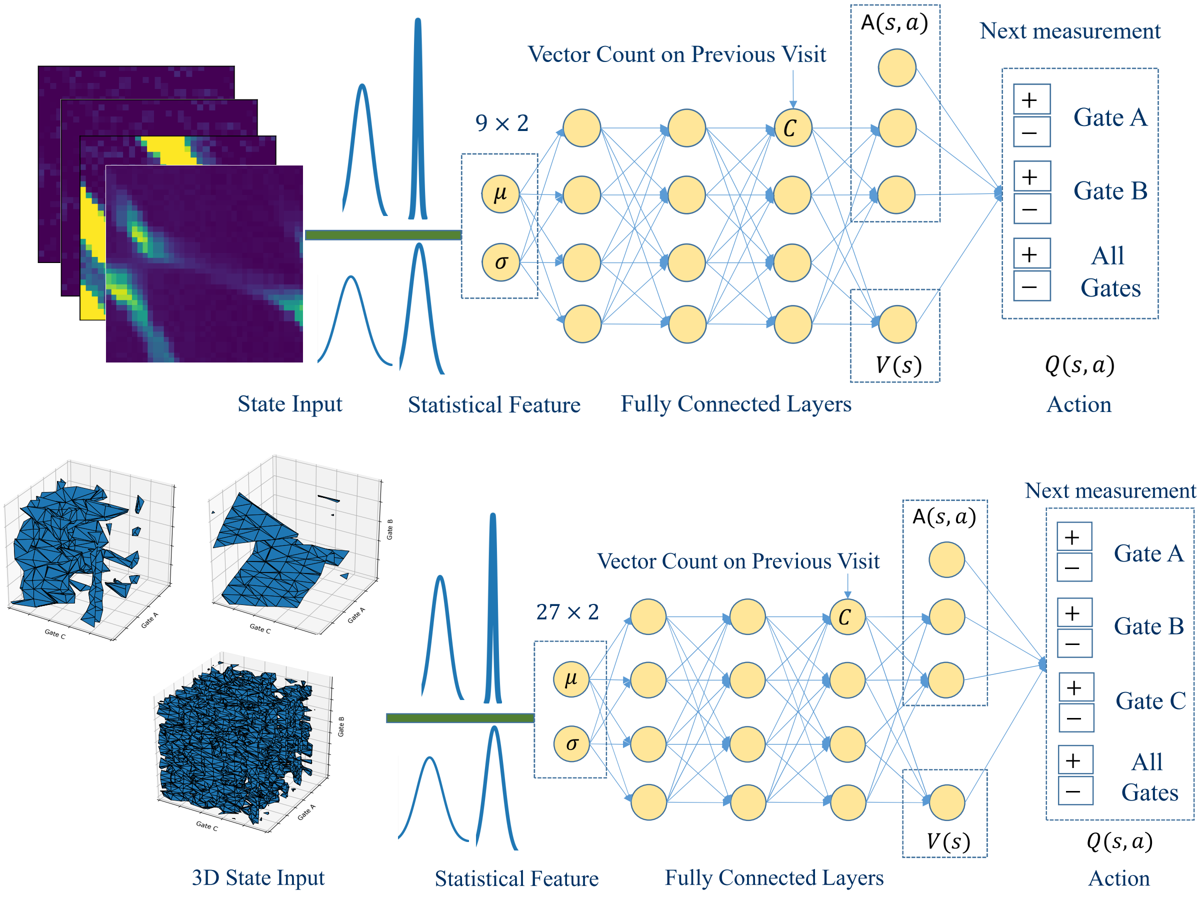

Figure 13: Our deep reinforcement learning framework using state as statistical feature of 2D image from 2 gates (Top) and 3D blocks from 3 gates (Bottom). The vector count C representing the 6 dimensional adjacent matrix is included in one of the last layer to discourage from revisiting the previous locations.

Algorithm 1 Training duelling deep Q network with prioritised experience replay.

```
Input: Replay buffer $B$, neural network model weight $\theta$, minibatch size $k$, $\Delta = 0, p_1 = 1$
for episode $m \leq M$ do
    Observe $s_0$ and make the action $a_0 = \pi_\theta(s_0)$
    for step $t = 1$ to $T$ do
        Observe $s_t, r_t$ and store transition $(s_{t-1}, a_{t-1}, s_t, r_t)$ in buffer $B$
        for $j \leq k$ # prioritised experience replay do
            Sample transition $j \sim P_j = p_j / \sum_i p_i$
            Compute importance-sampling weight $w_j = (N \times P_j)^{-\beta} / \max_i w_i$
            Compute TD-error $\delta_j = r_j + \gamma_j \max_{a'} Q(s_{j+1}, a') - Q(s_j, a_j)$ and update $p_j \leftarrow |\delta_j|$
            Accumulate weight-change $\Delta \leftarrow \Delta + w_j \times \delta_j \times \nabla_\theta Q(S_{j-1}, A_{j-1})$
        end for
        Update the network parameter using gradient descent: $\theta \leftarrow \theta + \alpha\Delta$, reset $\Delta = 0$
    end for
end for
```

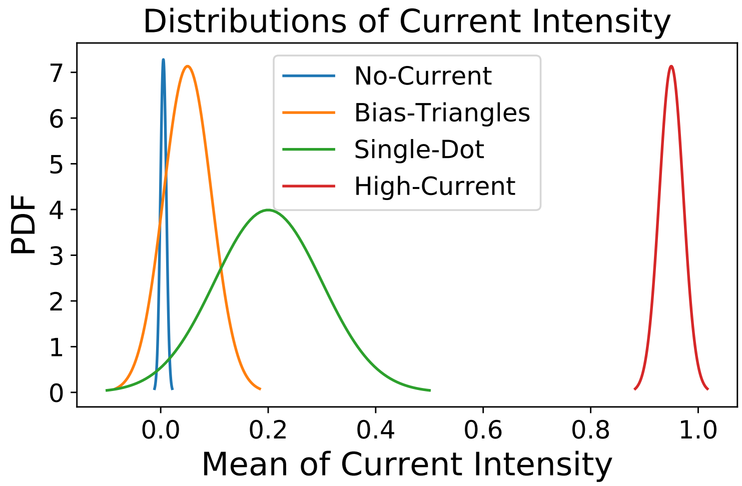

Figure 14: Distribution of current intensity at different regions in our gate voltage space.

For robustness in the estimation, we have normalised the current value between 0 to 1. Bias-triangles appears at $1-2$ blocks and in all remaining blocks there is no-current as shown in Fig. 3. We first estimate the true distribution for each state in Fig. 14. From these true distributions, each state has a distinctive representation that will be useful for classification.

We can assign each block into a corresponding state by using a Kullback–Leibler divergence of two univariate Gaussian distribution (one is estimated from the empirical distribution given the block and one is from the true distribution). Then, we define if the state is bias-triangles if it contains blocks assigned into bias-triangles and the remaining ones are assigned into no-current.

We have the closed-form formula for the KL divergence between two univariate Gaussian distribution $p \sim \mathcal{N}(\mu_a, \sigma_a)$ and $q \sim \mathcal{N}(\mu_b, \sigma_b)$ is as below

$$KL(p||q) = \log \frac{\sigma_b}{\sigma_b} + \frac{\sigma_a^2 + (\mu_a - \mu_b)^2}{2\sigma_b^2} - \frac{1}{2}.$$

Finally, classifying a block using the statistical representation $p \sim \mathcal{N}(\mu, \sigma)$ to one of the four states is as $\arg\min_{i \in \{1,2,3,4\}} KL(p||q_i)$ where q_i is the golden distribution shown in Fig. 14.

Classifying the state in 3D. Since the 3D block can be considered as multiple scan of the 2D grids. Therefore, the 3D block is assigned as the target bias-triangles is when its 2D slice image is estimated as bias-triangles. That is, we shall use the statistical feature approach described above to predict in 2D feature, then we take the output of multiple 2D feature for identifying 3D.