RESEARCH ARTICLE | MARCH 14 2025

# Environment model construction toward auto-tuning of quantum dot devices based on model-based reinforcement learning

Chihiro Kondo ; Raisei Mizokuchi ; Jun Yoneda ; Tetsuo Kodera



*APL Mach. Learn.* 3, 016114 (2025)
https://doi.org/10.1063/5.0251336

# Environment model construction toward auto-tuning of quantum dot devices based on model-based reinforcement learning

Cite as: APL Mach. Learn. 3, 016114 (2025); doi: 10.1063/5.0251336
Submitted: 2 December 2024 • Accepted: 28 February 2025 •
Published Online: 14 March 2025



Chihiro Kondo,[1] Raisei Mizokuchi,[1] Jun Yoneda,[2,3] and Tetsuo Kodera[1,a)]

## AFFILIATIONS

[1] Department of Electrical and Electronic Engineering, Institute of Science Tokyo, Meguro-ku, Tokyo 152-8552, Japan
[2] Academy of Super Smart Society, Institute of Science Tokyo, Meguro-ku, Tokyo 152-8552, Japan
[3] Department of Advanced Materials Science, University of Tokyo, Kashiwa, Chiba 277-8561, Japan

[a)] **Author to whom correspondence should be addressed:** kodera.t.ac@m.titech.ac.jp

## ABSTRACT

Semiconductor quantum dots (QDs) are promising hosts for quantum computers because of their scalability. In order to expedite the development process, there is a strong need for fully automated tuning of QDs that allows for en masse characterization of newly fabricated devices and control over large-scale systems with appreciable variability. Machine learning has been actively explored as a means to this end; however, challenges remain in terms of versatility for different tasks and device types. In this study, we explore a model-based reinforcement learning (MBRL) approach: unlike traditional reinforcement learning techniques, the learning process of MBRL progresses by constructing a model for the environment, which is to be diverted for other tasks and/or devices, thereby minimizing time-consuming learning processes. Using pre-measured data, we construct an environment model and, despite the intrinsic sparse reward distribution of the QD system, demonstrate its suitability for MBRL by emulating the process of auto-tuning to a single QD region. Our results highlight the potential of MBRL for more generic QD auto-tuning techniques, providing a promising step toward fully automated QD tuning.





## INTRODUCTION

Semiconductor quantum dots (QDs) are promising for dense qubit integration because of their small footprint and compatibility with semiconductor technologies.[1] One of the severe impediments to exploring novel QD structures or materials is the labor-intensive tuning process required to operate devices as qubits. This process involves repeated manual tunings of gate voltages by human experts over several days. Human experts conduct the tuning process based on charge stability diagrams (visual representation of QD characteristics as a function of control voltages). As the number of QDs increases, this approach will become impractical. To partially automate the tuning process, machine learning techniques have been studied;[2–8] however, their applicability is limited to specific tuning tasks or single device structures.

To overcome this limitation, we propose employing a model-based reinforcement learning (MBRL) system for QD tuning.[9] Reinforcement learning (RL) is an area of machine learning concerned with behavior within a certain environment.[10] In general RL, an agent directly interacts with the environment to learn behaviors that lead it to the desired goals. In contrast, MBRL constructs a model for the environment and uses it for learning. Since this model can be diverted for other tasks and/or devices with similar environments, the MBRL approach is promising in yielding tuning protocols with greater generality.

We note that the applicability of MBRL in QD experiments should not be taken for granted. Auto-tuning of QDs usually relies on finding the target patterns, which represent specific charge transport characteristics in charge stability diagrams obtained by sweeping two gate voltages,[2–4,8] or by performing one-dimensional

characterization, using, e.g., a ray-based method.[5–7] Unfortunately, such target patterns are only sparsely distributed, which is recognized to undermine the ability of the environment model to accurately represent the real environment.[11,12] In addition, an agent trained in the environment model, which is constructed under a sparse reward distribution, may struggle to perform QD tuning, with well-known RL challenges such as the credit assignment problem.[13] These problems arise because sparse rewards make it difficult for the agent to determine the contributions of individual actions to the final outcome. Therefore, verifying the applicability of MBRL in systems with sparse reward distributions is an essential first step toward the practical realization of the MBRL approach in such systems.

In this work, we construct an environment model for a QD system with a sparse reward distribution and assess its suitability for MBRL purposes. We evaluate the agent's behavior after the learning process and check the consistency of its reward prediction using a pre-measured charge stability diagram. Our work focuses on the tuning task toward a single QD condition (used, e.g., for charge sensor operation[14]) as a first key step toward MBRL-based QD tuning.

## LEARNING SYSTEM

Figure 1 shows the proposed MBRL system. Similar to the general RL framework, the "agent" (i.e., the learning system) interacts with the "environment" (i.e., a QD device in our case) and learns the "action" that maximizes the "reward" obtained from the environment. It consists of the following four major steps for learning: (i) the agent measures a small charge stability diagram that partially characterizes the QD and obtains its corresponding reward; (ii) the agent updates the construction of the environment model based on the measurement results and rewards; (iii) the agent learns the relation between actions and rewards in the constructed environment model many times; (iv) the agent determines which area will be measured in the next action to obtain a higher reward based on the current learning situation. One of the advantages of MBRL is that step (iii) is faster than in non-MBRL systems because the MBRL agent does not interact with the environment via time-consuming measurements. In the following, we refer to this cycle of four steps as iteration. We repeat $5 \times 10^6$ iterations in our work, and it takes a couple of days to complete the entire learning process on a computer with a single GPU (RTX 2070 SUPER, NVIDIA). We employ the neural-network RL framework, DreamerV2, as the algorithm for our agent.[15] It is a MBRL framework developed by DeepMind with the potential to outperform the top single-GPU agents such as Rainbow[16] and IQN.[17] This improvement stems from utilization of predictions in the compact latent space of a powerful world model. The detailed parameters of DreamerV2 used in this study are provided in the supplementary material, Note 1.

For the proof-of-concept demonstration of MBRL auto-tuning, we use as the environment a pre-measured wide-range charge stability diagram. The use of pre-measured data provides several benefits, such as accelerated interaction speed and avoided risk of potential device damage by eliminating the need for real measurements. The data were obtained by sweeping the voltages of the plunger gates, $V_{G1}$ and $V_{G2}$, across a wide enough range. This ensures that the dataset captures key conditions relevant to QD formation, including pinch-off, single- and double-QD configurations. Pre-measured data obtained from an n-type multiple QD device were employed that was electrostatically defined by gate electrodes fabricated on a silicon-on-insulator substrate.[18,19] The data store the QD current flowing between the drain and source measured at 4.2 K as a function of two gate-electrode voltages $V_{G1}$ and $V_{G2}$.

## REWARD DETERMINATION BY IMAGE CLASSIFICATION

In RL, reward plays an important role because the agent decides its action based on reward predictions. In contrast to video games such as Atari games,[20] where the environment outputs a score that plays the role of reward, in QD measurements, rewards must be provided separately. We use a convolutional neural network (CNN) technique to evaluate the base rewards for the measured results based on the similarity to computer-generated "target patterns," as shown in Fig. 2(a).[21] These patterns (stripes with a negative slope) represent the charge stability diagrams expected for the single QD region[22] and help us eliminate the need for human intervention in the labeling process. This is in stark contrast to the usual image

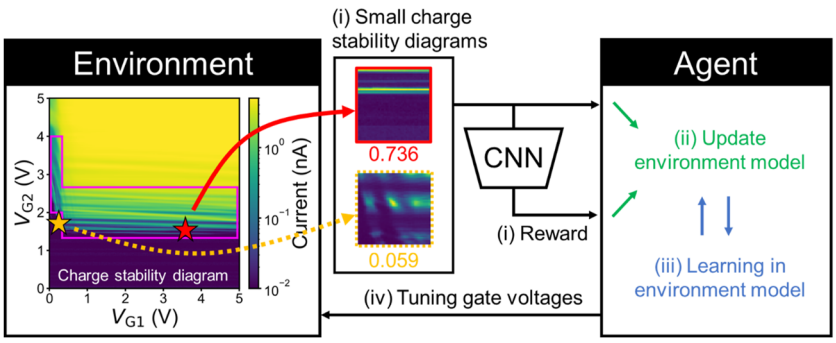


**FIG. 1.** Learning system. A sequence of steps (i)–(iv) constitutes an iteration cycle. The magenta frame on the charge stability diagram represents the single QD region set by human experts, which will be used in the section titled The MBRL training and the result for the numerical evaluation of the environment model. Examples of small charge stability diagrams (cropped from the overall plot) are shown along with the confidence scores received from the CNN model (see the main text).

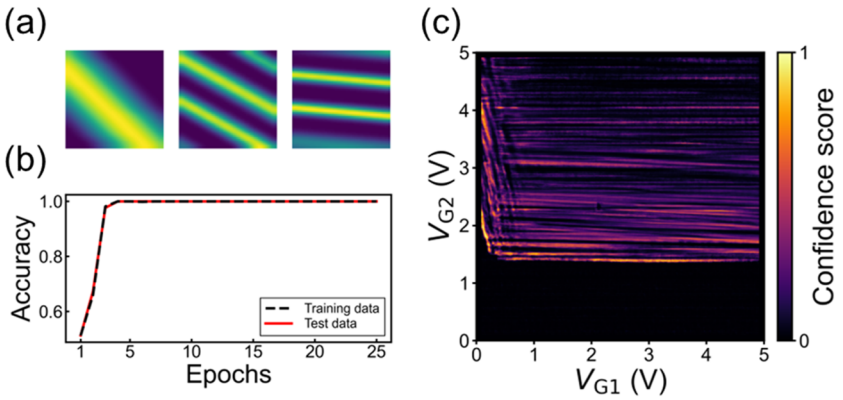


**FIG. 2.** (a) Example of computer-generated "target patterns" used for training the CNN. (b) Learning result of CNN with supervised learning for the classification task of whether or not a device is in a single QD region. The training and test data include 7000 and 3000 images, respectively. All images in the datasets are used in each epoch. (c) The distribution of confidence scores obtained by analyzing pre-measured data with CNN.

classification with CNN, where it is necessary for humans to manually label data obtained through real measurements in order to perform supervised learning. We train the CNN with supervised learning for the classification task with the target pattern dataset (5000 images) and CIFAR-10 as the dummy dataset (5000 images).[23] These datasets are divided into a training dataset (7000 images) used to train the CNN and a test dataset (3000 images) used exclusively for evaluation. We tune the hyperparameters of the CNN model structure, such as the number of hidden layers, nodes in the layers, and learning rate, using Optuna, an open-source hyperparameter tuning framework.[24] The details of the CNN are presented in the supplementary material, Note 1. The CNN model outputs a scalar value between 0 and 1, hereinafter referred to as the confidence score, for an input image. We regard a classification as correct when the confidence score is higher than 0.5. Figure 2(b) shows the evolution of classification accuracy as a function of epoch (An epoch elapses when every image in the training dataset is used once for training). The trained CNN achieves >99% accuracy on the test dataset.

We assess the CNN's performance based on the distribution of confidence scores for the pre-measured data. The distribution is visualized from the calculations of the confidence score on each data point in the pre-measured data [Fig. 2(c)]; the CNN calculates the score based on the small charge stability diagram centered on the point. The distribution shows large scores in regions where human experts recognize the target patterns; see also the small charge stability diagrams in Fig. 1. As will be discussed later, the MBRL agent using the learning system incorporating the trained CNN succeeds in QD auto-tuning, which ultimately demonstrates that the CNN possesses sufficient performance to achieve our objective.

Using the sum of confidence scores (between 0 and 1) received in each iteration as the base reward, we design the reward to be employed in the MBRL process of QD auto-tuning as follows: a +100 reward is given upon reaching the goal with the confidence score above 0.5 and a penalty of −100 reward is issued when one of the gate voltages exceeds the pre-determined limits (0 and 5 V in the present case). Since we terminate each episode of the auto-tuning process after at most 100 iterations and the total base reward is between 0 and 100, the extra rewards strongly motivate the MBRL agent to move to the desired goals.

## THE MBRL TRAINING AND THE RESULT

We now train our MBRL agent on the pre-measured data to perform QD auto-tuning. During the MBRL training, the agent explores the desired goal by repeating the iterations described in the section titled Learning system. In each episode, the starting voltage condition of the agent is randomly selected within the range of 0.5–1.0 V for both gate voltages. The MBRL actions in this work are restricted to four types of movements: 0.1 V shift in the up, down, left, and right directions on the map shown in Fig. 1. After taking an action, the agent performs a measurement in the square range of ±0.15 V around the new location. The measurement result is evaluated by the CNN described in the section titled Reward determination by image classification.

Here, we discuss how the agent works toward achieving the QD auto-tuning task by analyzing the training results from three perspectives. Figure 3(a) presents the episode reward, i.e., the cumulative reward per episode as a function of the learning iteration. In the figure, the MBRL agent's episode reward decreases and then increases at the early learning stages (~$1.3 \times 10^5$ iterations), implying that our agent learns to avoid penalized areas. As learning progresses, the episode reward acquired by the MBRL agent promptly increases and mostly remains around 100, with occasional dips that soon recover, indicating consistent achievement of the goal. These dips are possibly a result of the agent temporarily employing inefficient policies in an effort to explore better ones in the long term. On the other hand, the control experiment, where the agent selects the action randomly (random agent), has much lower rewards (~−70) patently due to the penalty. The rewards do not depend on the learning iteration as expected for its randomness in action selection. These results suggest that the reward system is well designed to guide the agent toward successful QD auto-tuning. After the training, we perform QD auto-tuning with the MBRL agent and monitor its action histories to validate its decision-making process. Figure 3(b) shows 50 trajectories randomly selected from 1000 episode runs conducted on the pre-measured data. We see that the agent first moves upward on the pre-measured data to reach the area where current flows and then explores for the target pattern in the vicinity, similar to the way human experts tune devices toward a single QD condition. In this connection, the agent successfully completed tuning within the single QD region

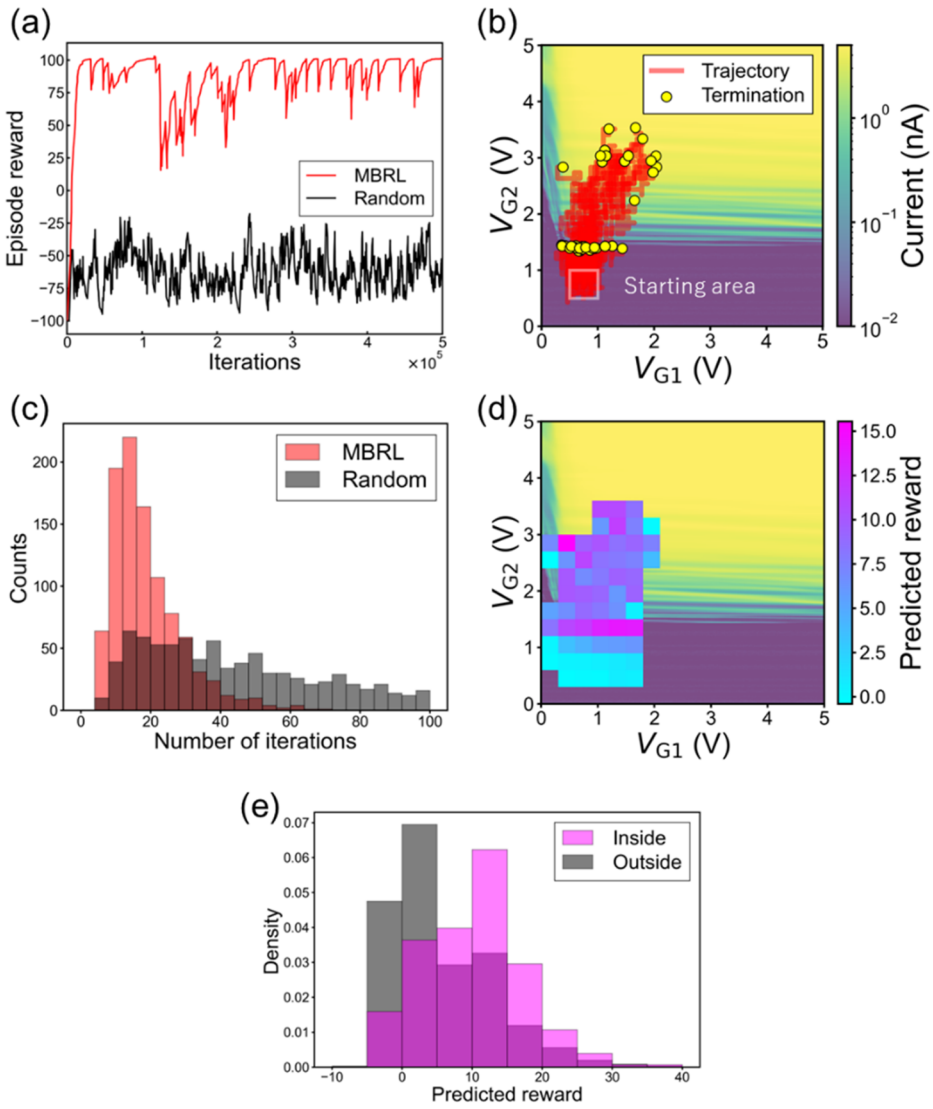


**FIG. 3.** (a) Evolution of episode reward. The red and black traces are for the MBRL agent and random agent, respectively. (b) MBRL agent's measurement trajectories on the pre-measured charge stability diagram. The randomly selected trajectories are shown. (c) Histograms of the results of 1000 episode runs each for the MBRL and random agents. (d) Predicted rewards of the environment model plotted on the stability diagram. Reward prediction is output only in areas where the agent has passed. (e) Histograms of the predicted rewards inside and outside the single QD region, as defined in Fig. 1.

with a probability of 98.0%. As a baseline for comparison, we also apply the Deep Q-Network (DQN) framework under the same environment and tasks. DQN is a widely adopted model-free RL framework, and its variant has been employed in previous studies on QD auto-tuning.[8] The DQN successfully completed tuning with a probability of 87.1%, which is lower than that of MBRL and underscores the capability of MBRL to outperform a well-established model-free approach, further highlighting its effectiveness in the QD system. We note that in some trajectories where the MBRL agent first deviates slightly to the left, it turns back to the right to avoid penalties. We also find that the agent mostly ends up under the G2 dot condition (the lower-right region), although high scores are also output under the G1 dot condition (the upper-left region) by the CNN; see Fig. 2(c). This indicates that the agent learns to move toward a high-reward region closer to the starting point; indeed, the training algorithm employed in this study is designed to maximize the expected value of the cumulative discounted reward (the expected sum of the rewards in the next 15 iterations with the reward obtained in each iteration discounted by a factor of 0.1% after each iteration[15]). This discount encourages the agent to learn to find the goal with fewer iterations. To clarify the performance of the MBRL agent, we compare the distributions in the number of iterations required to complete the tuning task for the MBRL agent and the random agent [Fig. 3(c)]. We apply the Wilcoxon rank-sum test[8,25] to the two distributions and obtain a p-value (<0.001) well below the commonly used significance level of 0.05. Thus, we reject the null hypothesis that there is no difference in the median number of iterations between the MBRL and the random agents, indicating that the MBRL agent makes meaningful action selections. These findings collectively demonstrate that the agent trained within the environment model achieves the auto-tuning toward single QD condition.

To confirm the impact of the sparse reward distribution on the environment model's ability to represent the original system, we perform evaluation from the perspective of the rewards predicted by the environment model. The MBRL algorithm we evaluate in this study, DreamerV2, incorporates a reward prediction mechanism in its environment model, allowing for the prediction of rewards for subsequent iterations based on previous ones. This mechanism enables us to obtain a property of the environment model, leading to an explainability of its internal states—a feature distinct from typical systems with black-boxed internal states. Figure 3(d) shows the predicted reward distribution averaged over 1000 episodes within a grid of 0.3 × 0.3 V sub-regions, which is the same size as a small charge stability diagram. We see that the predicted rewards remain close to zero in regions without current and start to increase around $V_{G2}$ = 1.4 V, where the target pattern begins to emerge; this behavior is consistent with what is expected from the confidence score distribution. To give more numerical evaluation, we present in Fig. 3(e) the comparison between the distributions of the predicted rewards inside and outside the single QD region set by human experts (as illustrated in Fig. 1). By applying the test again to the two distributions, we reveal significantly higher predicted rewards inside the single QD region compared to outside, with a p-value (<0.001), as with the analysis conducted in Fig. 3(c). These results suggest that the QD device behaviors are adequately modeled in MBRL algorithms for auto-tuning purposes even though the QD systems have inherently sparse reward distributions.

Overall, we have confirmed the applicability of MBRL to QD systems through the effective auto-tuning by the MBRL agent trained within the environment model and the successful construction of the environment model that represents the sparse rewards. This is a first key step toward more general QD auto-tuning techniques overcoming challenges in terms of versatility for different tasks and device types.

To gain more insight into the performance of the MBRL agent, we now present a preliminary analysis of failure cases. Among 20 failure cases out of 1000 episode runs, we have 15 cases of "drifting" outcome and five cases of "left-exit" outcome. In the "drifting" cases, the agent remains within the given boundary of the charge stability diagram but fails to reach the target pattern even after 100 iterations. We attribute such cases to the inherent stochasticity of DreamerV2,[15] which, while advantageous for escaping local optima, introduces randomness in the agent's decision making. In the "left-exit" cases, on the other hand, the agent moves beyond the left boundary of the diagram ($V_{G1} < 0$ V) and incurs a penalty. A possible scenario is that they result from the agent aiming to tune to the G1 dot condition, which happens to be in close proximity to the left boundary. The stochastic nature of the agent's decision making in DreamerV2 may lead to trespassing. Optimizing the stochasticity of the agent's behavior is an important topic of future studies.

## CONCLUSIONS

In this work, we explored a MBRL approach toward QD tuning and confirmed a construction of a model of the QD environment, which leads, in the future, to the versatility in qubit tuning. A warranted concern is that the sparsely distributed QD characteristics impede the construction of a model functional in QD tuning and therefore the application of MBRL to QD tuning. To demonstrate the applicability of MBRL to QD systems, we first trained the MBRL agent on the pre-measured data and constructed an environment model. In the training, we utilized the automated reward determination via image classification using a CNN. We evaluated the performance of the MBRL agent by analyzing the training results; it turned out that the agent trained within the environment model achieved the auto-tuning toward single QD condition. Then, we also evaluated the environment model from the perspective of the predicted reward, which demonstrates its ability to represent the original system, including the inherent sparse reward distribution. Our results suggest the applicability of MBRL to QD measurements and represent the first key step toward MBRL-based QD tuning. The generalization of auto-tuning for QD devices is essential for promoting the advancement of QD-based quantum computing because we need to perform tuning on non-standardized[26] and diverse characteristics of current QD devices. A next step toward auto-tuning of integrated silicon qubits involves the reduction in learning time by leveraging the MBRL's ability to adapt an environment model to a different system without requiring complete re-training. As a preliminary investigation of this potential, we conduct additional experiments under a different starting condition designed to mimic a device with distinct threshold voltages (see the supplementary material, Note 2). This result provides preliminary evidence for the robustness of the MBRL procedure across different device configurations. We believe that the MBRL approach investigated in this work offers more versatility than other auto-tuning methods and will accelerate the realization of the general-purpose auto-tuning technique.

## SUPPLEMENTARY MATERIAL

See the supplementary material for details regarding the CNN model, the DreamerV2 model, and the additional experiment on the robustness of the MBRL procedure in QD tuning.

## ACKNOWLEDGMENTS

This work was financially supported by JST Moonshot R&D Grant No. JPMJMS2065, MEXT Quantum Leap Flagship Program (MEXT QLEAP) Grant No. JPMXS0118069228, JST PRESTO Grant No. JPMJPR21BA, JST CREST Grant No. JPMJCR24A1, and JSPS KAKENHI Grant Nos. JP23H05455, JP23H01790, and JP23K17327.

## AUTHOR DECLARATIONS

### Conflict of Interest

The authors have no conflicts to disclose.

### Author Contributions

**Chihiro Kondo**: Conceptualization (equal); Data curation (equal); Formal analysis (equal); Investigation (equal); Methodology (equal); Software (equal); Validation (equal); Visualization (equal); Writing – original draft (equal). **Raisei Mizokuchi**: Conceptualization

(equal); Methodology (supporting); Project administration (equal); Supervision (supporting); Validation (supporting); Writing – original draft (supporting); Writing – review & editing (lead). **Jun Yoneda**: Methodology (supporting); Project administration (supporting); Supervision (supporting); Writing – review & editing (supporting). **Tetsuo Kodera**: Conceptualization (equal); Data curation (equal); Funding acquisition (equal); Project administration (equal); Resources (equal); Supervision (equal); Writing – review & editing (equal).

## DATA AVAILABILITY

The data that support the findings of this study are available within the article and at https://10.5281/zenodo.14241888.

## REFERENCES


[1] L. M. K. Vandersypen, H. Bluhm, J. S. Clarke, A. S. Dzurak, R. Ishihara, A. Morello, D. J. Reilly, L. R. Schreiber, and M. Veldhorst, "Interfacing spin qubits in quantum dots and donors—Hot, dense, and coherent," npj Quantum Inf. **3**, 34 (2017).

[2] S. S. Kalantre, J. P. Zwolak, S. Ragole, X. Wu, N. M. Zimmerman, M. D. Stewart, Jr., and J. M. Taylor, "Machine learning techniques for state recognition and auto-tuning in quantum dots," npj Quantum Inf. **5**, 6 (2019).

[3] H. Moon, D. T. Lennon, J. Kirkpatrick, N. M. van Esbroeck, L. C. Camenzind, L. Yu, F. Vigneau, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, D. Sejdinovic, E. A. Laird, and N. Ares, "Machine learning enables completely automatic tuning of a quantum device faster than human experts," Nat. Commun. **11**, 4161 (2020).

[4] J. P. Zwolak, T. McJunkin, S. S. Kalantre, J. P. Dodson, E. R. MacQuarrie, D. E. Savage, M. G. Lagally, S. N. Coppersmith, M. A. Eriksson, and J. M. Taylor, "Autotuning of double-dot devices *in situ* with machine learning," Phys. Rev. Appl. **13**, 034075 (2020).

[5] J. P. Zwolak, T. McJunkin, S. S. Kalantre, S. F. Neyens, E. R. MacQuarrie, M. A. Eriksson, and J. M. Taylor, "Ray-based framework for state identification in quantum dot devices," PRX Quantum **2**, 020335 (2021).

[6] J. P. Zwolak, S. S. Kalantre, T. McJunkin, B. J. Weber, and J. M. Taylor, "Ray-based classification framework for high-dimensional data," arXiv:2010.00500 (2020).

[7] J. Ziegler, F. Luthi, M. Ramsey, F. Borjans, G. Zheng, and J. P. Zwolak, "Tuning arrays with rays: Physics-informed tuning of quantum dot charge states," Phys. Rev. Appl. **20**, 034067 (2023).

[8] V. Nguyen, S. B. Orbell, D. T. Lennon, H. Moon, F. Vigneau, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, D. Sejdinovic, and N. Ares, "Deep reinforcement learning for efficient measurement of quantum devices," npj Quantum Inf. **7**, 100 (2021).

[9] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed. (MIT Press, Cambridge, 2018), p. 195.

[10] T. M. Mitchell, *Machine Learning* (McGraw Hill, New York, 1997), p. 367.

[11] H. Charlesworth and G. Montana, "PlanGAN: Model-based planning with sparse rewards and multiple goals," arXiv:2006.00900 (2020).

[12] L. Antonyshyn and S. Givigi, "Deep model-based reinforcement learning for predictive control of robotic systems with dense and sparse rewards," J. Intell. Rob. Syst. **110**, 100 (2024).

[13] M. Minsky, "Steps toward artificial intelligence," Proc. IRE **49**, 8 (1961).

[14] D. J. Reilly, C. M. Marcus, M. P. Hanson, and A. C. Gossard, "Fast single-charge sensing with a rf quantum point contact," Appl. Phys. Lett. **91**, 162101 (2007).

[15] D. Hafner, T. Lillicrap, M. Norouzi, and J. Ba, "Mastering Atari with discrete world models," arXiv:2010.02193 (2020).

[16] M. Hessel, J. Modayil, H. van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, *Proceedings of the AAAI Conference on Artificial Intelligence* (Association for the Advancement of Artificial Intelligence, 2018), Vol. 32, p. 3215.

[17] W. Dabney, G. Ostrovski, D. Silver, and R. Munos, "Implicit quantile networks for distributional reinforcement learning," Proc. Mach. Learn. Res. **80**, 1096 (2018).

[18] T. Utsugi, N. Lee, R. Tsuchiya, T. Mine, R. Mizokuchi, J. Yoneda, T. Kodera, S. Saito, D. Hisamoto, and H. Mizuno, "Single-electron pump in a quantum dot array for silicon quantum computers," Jpn. J. Appl. Phys. **62**, SC1020 (2023).

[19] D. Hisamoto, N. Lee, R. Tsuchiya, T. Mine, T. Utsugi, S. Saito, and H. Mizuno, "Electron charge sensor with hole current operating at cryogenic temperature," Appl. Phys. Express **16**, 036504 (2023).

[20] M. G. Bellemare, Y. Naddaf, J. Veness, and M. Bowling, "The arcade learning environment: An evaluation platform for general agents," J. Artif. Intell. Res. **47**, 253 (2013).

[21] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," Neural Comput. **1**, 541 (1989).

[22] W. G. van der Wiel, S. De Franceschi, J. M. Elzerman, T. Fujisawa, S. Tarucha, and L. P. Kouwenhoven, "Electron transport through double quantum dots," Rev. Mod. Phys. **75**, 1 (2002).

[23] A. Krizhevsky, "Learning multiple layers of features from tiny images," www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf (2009).

[24] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A next-generation hyperparameter optimization framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining* (Association for Computing Machinery, 2019), p. 2623.

[25] F. Wilcoxon, "Individual comparisons by ranking methods," Biom. Bull. **1**, 80 (1945).

[26] G. Burkard, T. D. Ladd, A. Pan, J. M. Nichol, and J. R. Petta, "Semiconductor spin qubits," Rev. Mod. Phys. **95**, 025003 (2023).