PAPER • OPEN ACCESS

Robust quantum dots charge autotuning using neural network uncertainty

To cite this article: Victor Yon *et al* 2024 *Mach. Learn.: Sci. Technol.* **5** 045034

IOP Publishing Mach. Learn.: Sci. Technol. 5 (2024) 045034 https://doi.org/10.1088/2632-2153/ad88d5

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
29 June 2024

REVISED
11 September 2024

ACCEPTED FOR PUBLICATION
18 October 2024

PUBLISHED
7 November 2024

PAPER

Robust quantum dots charge autotuning using neural network uncertainty

Victor Yon[1,2,3,*], **Bastien Galaup**[1,2,3], **Claude Rohrbacher**[2,3,4], **Joffrey Rivard**[2,3,4], **Clément Godfrin**[5], **Ruoyu Li**[5], **Stefan Kubicek**[5], **Kristiaan De Greve**[5], **Louis Gaudreau**[6], **Eva Dupont-Ferrier**[2,3,4], **Yann Beilliard**[1,2,3], **Roger G Melko**[7,8] and **Dominique Drouin**[1,2,3]

[1] Institut Interdisciplinaire d'Innovation Technologique (3iT), Université de Sherbrooke, Sherbrooke, QC J1K 0A5, Canada
[2] Laboratoire Nanotechnologies Nanosystémes (LN2) - CNRS 3463, Université de Sherbrooke, Sherbrooke, QC J1K 0A5, Canada
[3] Institut quantique (IQ), Université de Sherbrooke, Sherbrooke, QC J1K 2R1, Canada
[4] Département de physique, Université de Sherbrooke, Sherbrooke, QC J1K 2R1, Canada
[5] IMEC, Kapeldreef 75, 3001 Leuven, Belgium
[6] National Research Council Canada, Quantum and Nanotechnologies Research Center, Ottawa, ON K1A 0R6, Canada
[7] Department of Physics and Astronomy, University of Waterloo, Waterloo, ON N2L 3G1, Canada
[8] Perimeter Institute for Theoretical Physics, Waterloo, ON N2L 2Y5, Canada
[*] Author to whom any correspondence should be addressed.

E-mail: victor.yon@usherbrooke.ca

Keywords: artificial neural network, Bayesian neural network, quantum dot, charge autotuning, uncertainty estimation

Supplementary material for this article is available online

Abstract

This study presents a machine learning-based procedure to automate the charge tuning of semiconductor spin qubits with minimal human intervention, addressing one of the significant challenges in scaling up quantum dot technologies. This method exploits artificial neural networks to identify noisy transition lines in stability diagrams, guiding a robust exploration strategy leveraging neural network uncertainty estimations. Tested across three distinct offline experimental datasets representing different single-quantum-dot technologies, this approach achieves a tuning success rate of over 99% in optimal cases, where more than 10% of the success is directly attributable to uncertainty exploitation. The challenging constraints of small training sets containing high diagram-to-diagram variability allowed us to evaluate the capabilities and limits of the proposed procedure.

1. Introduction

Multiple quantum computing technologies compete to outperform classical computers for practical applications, yet the journey toward achieving this objective is fraught with considerable technical challenges. This article specifically studies quantum systems based on semiconductor spin qubits [1–4] formed by electrostatically confining electrons in quantum dots (QDs). This technology presents several advantages, such as high gate fidelities [5–9], long coherence times [10, 11], thermal robustness at temperatures greater than one kelvin [12] and compatibility with well-established metal-oxide-semiconductor (CMOS) technologies [13–15]. This compatibility further amplifies scalability potential and enables co-integration with industrial electronics [16, 17]. However, QD technology is still in an early stage of development, and the best materials [18, 19], fabrication techniques, use cases, and control procedures are yet to be discovered.

One bottleneck in the large-scale implementation of quantum computers based on QDs is the complexity of charge tuning: one of the preliminary tasks required to set up a qubit. This tuning task confines a specific number of charge carriers in one or multiple QDs using the information provided by indirect measurements, which is typically obtained by tuning the voltages at two control gates (G1 and G2 in figure 1). The results of voltage sweeping can be visualized as two-dimensional images termed stability diagrams (examples in figures 2(a)–(c), where pixels encode the current measured using single-electron transistor (SET) [20] or quantum point contact (QPC) [21]. One could then identify the transition lines (highlighted with green lines

in figures 2(d)–(f) that represent a change in the number of charge carriers inside the QD. Given the knowledge that electron-based QDs are empty at low gate voltages, it is possible to deduce the number of charges for each area in the voltage space (blue areas in figures 2(d)–(f)).

However, the task is challenging to automate because the stability diagram parameter space is vast, noisy, and device-dependent. For this reason, the number of charges inside the QD is usually tuned manually by experimentalists based on informed guesses and human heuristics. While this approach is sufficient for small proof-of-concept studies in research laboratories, it is incompatible with large-scale industrial applications. Moreover, measuring a large area of a stability diagram could take hours, which significantly slows down QD initialization, especially since it often requires iterative measurements. One could attempt to automate the tuning procedure using classical algorithms and human expertise [22], but this approach still requires human involvement and is tailored to specific hardware. Thus, it does not satisfy the requirements for scaling up the technology or accelerating the development of new designs.

Recent studies [23, 24] have suggested automating the tuning procedure using machine learning (ML) approach with artificial deep neural networks (NNs) [25]. These methods have shown promising results but are not yet reliable enough to consider fully autonomous large-scale QD control. The relatively high failure rate of these ML approaches (25% [23] for single-QD and 43% [24] for double-QD charge tuning) could be attributed to *(i)* the overconfidence of the model while facing ambiguous, out-of-distribution, too small, or noisy measurements and *(ii)* the low error tolerance of the exploration strategy. For example, one transition line misdetection will often lead to the failure of the entire charge tuning procedure. In the context of coarse tuning, [26] addressed a similar issue by first training a NN to evaluate the measurement quality before sending it to a second NN to infer the class. [27] combined this method with custom peak-finding techniques to tune offline double QDs into a specific charge regime with an accuracy of 89.7%. Here, we chose a more versatile approach by training a single NN to detect the transition lines and providing a classification confidence score at the same time. One could also improve the performance of ML models by using more parameters and a larger dataset, but a very large model could be challenging to integrate near the tuned devices, and the high cost of collecting more diagrams is prohibitive. Even so, ML models never provide the guarantee of perfect accuracy.

This study introduces a hardware-agnostic autotuning procedure that leverages NN uncertainty to significantly enhance the robustness of semiconductor spin qubit charge tuning with minimal human intervention. This is achieved by training NN to identify transition lines in a stability diagram using supervised learning, coupled with an exploration strategy incorporating the NN's predictions and confidence score (measure of uncertainty). We considered two methods to estimate the confidence score of the NNs: one with a heuristic from classical NNs and one using the Bayesian framework applied to NNs [28, 29]. These two methods have been evaluated based on their concrete ability to reduce the critical failure rate during the autotuning procedure across three different experimental offline datasets measured using different single-QD hardware.

2. Problem definition

2.1. Project scope

Before using QDs as qubits to perform operations, each QD must be tuned to a specific charge state by applying appropriate gate voltages. The complete calibration procedure can be divided into five distinct steps: *(i) bootstrapping*: cooling the device and bringing its regime into the appropriate parameter range; *(ii) coarse tuning* [26, 27, 30–35]: tuning the QDs to a specific topology (e.g. single QD, double QD); *(iii) establishing controllability* [27, 30, 36, 37]: setting up virtual gates that compensate for capacitive cross-talk; *(iv) charge state tuning* [23, 24, 27, 38]: tuning the QDs to a specific charge configuration (number of electrons in our case); *(v) fine-tuning* [30, 39]: adjusting the inter-dot tunnel coupling. For a more detailed description of these steps, refer to [40]. Performing these tasks quickly and reliably is one of the technical challenges preventing the scale-up of the number of qubits in QD-based quantum computers.

This work focuses on automating the task of charge state tuning *(iv)*. We assume the system is bootstrapped *(i)* and the voltage range contains a single QD *(ii)*. However, our method does not require setting up virtual gates *(iii)*, since we opted for a flexible approach by detecting the slope of the transition lines during the procedure.

2.2. Problem frame and constraints

We framed the problem of charge state tuning as an exploration task, where each step consists of detecting charge transition lines in the measurement of a voltage space subsection. The ultimate goal is to reach a specific charge regime while minimizing the exploration cost. In this case, the cost is the measurement time, which is directly affected by the number of steps and the size of the subsection measured. This approach

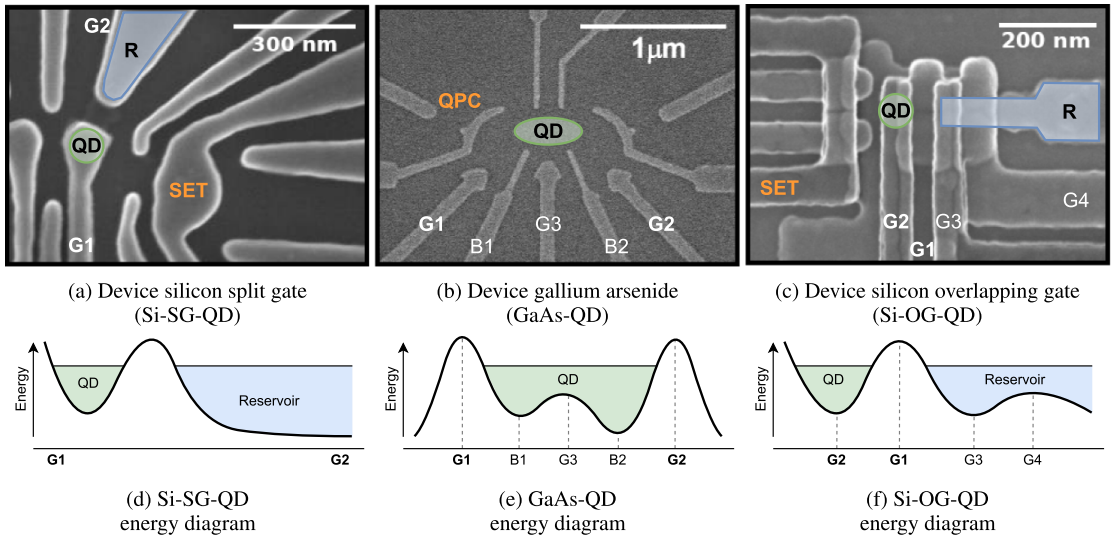

Figure 1. QD devices used to measure the stability diagrams. (a)–(c) Scanning electron microscope (SEM) images of experimental devices. A single QD can be created by tuning electrostatic gates 1 (G1) and 2 (G2) to appropriate voltages. (a) G2 controls the electronic density of an electron reservoir (R), and the single-electron transistor (SET) is used as a charge sensor. See [41] for experimental details. (b) The tunnel barriers labeled B1 and B2 are set to a fixed voltage, as well as gate 3 (G3), which is required for more than one QD only. The quantum point contact (QPC) is used as a charge sensor. See [42] for experimental details. (c) G1 and G4 are tunnel barriers. The voltages at G3, G4, and R are fixed to allow for single-QD formation under G2 and an extended reservoir. The SET is used as a charge sensor. See [15] for experimental details. (d)–(f) Energy diagrams representing the formation of a single QD for each device.

allows us to break down the problem into two distinct tasks: transition line detection (section 3) and the diagram exploration strategy (section 4).

To uphold the robustness and adaptability of our methodology, we consciously minimized the utilization of preprocessing techniques while maintaining a relatively simple NN model. This decision was grounded in the ultimate objective of implementing this solution in custom electronics inside the cryogenic environment of a dilution refrigerator to perform *in situ* online tuning. The computational constraints and the need for a streamlined process inherent to this environment necessitate a careful balance between model complexity and operational energy efficiency. To evaluate the efficacy of our approach across various QD designs, we tested our method on three distinct datasets, each obtained from a different hardware and research group. This ensures that our solution is not tailored specifically to one quantum device but can be applied to a wide range of QD-based spin qubit hardware.

2.3. Datasets

We tested our approach on offline experimental measurements obtained from three different QD device architectures across distinct experimental setups. The first dataset, referred to as silicon split gate (Si-SG-QD) [41], contains 17 stability diagrams (figure 2(a)) measured from devices fabricated with metal-oxide-semiconductor (MOS) single-layer architecture (figure 1(a)). The second dataset, referred to as gallium arsenide(GaAs-QD) [42], is composed of 9 stability diagrams (figure 2(b)) obtained by measuring the current of a device fabricated with a GaAs/AlGaAs heterostructure (figure 1(b)). The third and last dataset, referred to as silicon overlapping gate (Si-OG-QD) [15], includes 12 stability diagrams (figure 2(c)) measured from devices fabricated with MOS stacked-layer architecture (figure 1(c)). The diagrams for the first and third datasets are measured using SET, while the ones for the second dataset are measured using a QPC. The measurements are two-dimensional stability diagrams represented as images, where the x- and y-axes correspond to the voltages applied to gates G1 and G2, respectively. If more than two gates are available in the experimental setup, the other ones are fixed to a voltage compatible with a single-QD regime.

Using simulated stability diagrams could increase the size of the training set [34, 43], but at the cost of a distribution shift between the training and testing sets, which could harm the quality of uncertainty quantification. We therefore chose to rely exclusively on experimental data. We also minimized measurement preprocessing to ensure compatibility with future *in situ* online implementation of this autotuning method. Only input normalization was applied to the measured patch before being sent to the classification model. See supplementary section S1 for detailed information regarding dataset preparation and diagram selection.

Each dataset features different defects, noise, and transition line patterns, creating a large variety of challenges for the autotuning procedure. The diagrams from Si-SG-QD present strong oscillating

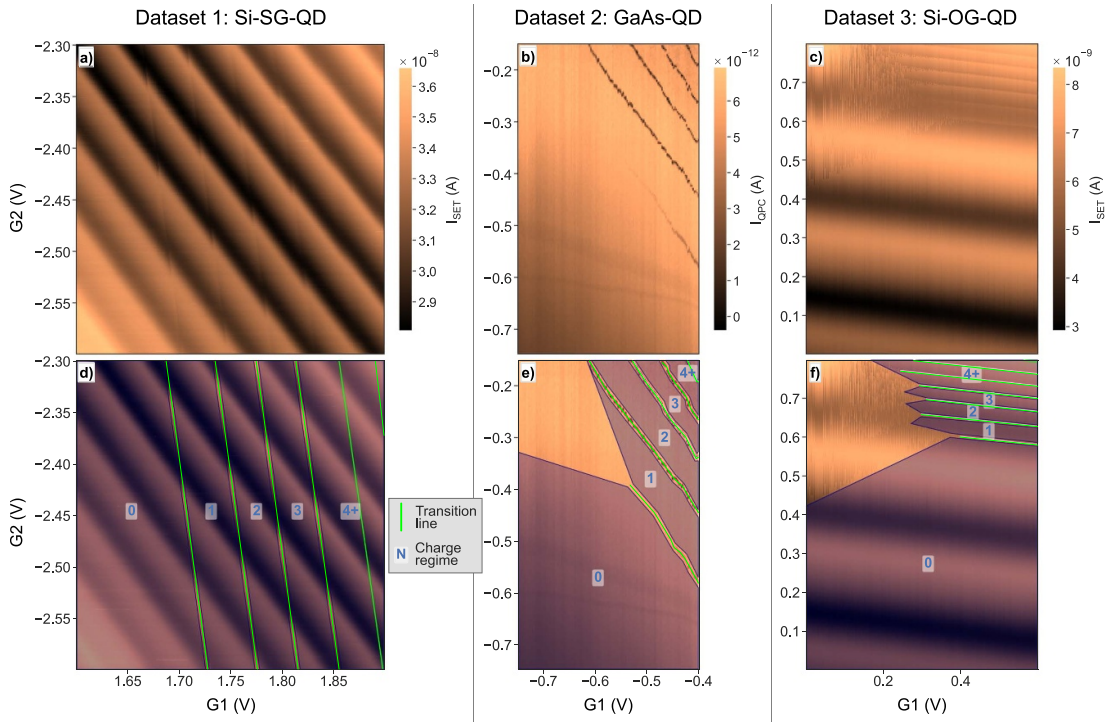

Figure 2. Example of one stability diagram from each dataset. (a)–(c) Representation of the stability diagrams as images, where pixels encode the measured current for given voltage values at the gates (G1 and G2). (d)–(f) Same diagrams with manual annotations of transition lines in green and charge regime areas in blue. The regions with more than 3 charges are grouped under the annotation '4+'. A voltage area not annotated with a charge regime due to fading lines or ambiguous boundaries is considered an 'unknown charge regime'. More examples can be found in supplementary section S1.6.

background noise caused by the cross-capacitance effects of G1 and G2 acting on the SET, which can be hard to differentiate from the transition lines (see diagonal parasitic oscillations in figure 2(a). Therefore, the most challenging part of tuning such devices is reliably detecting the lines, especially in low- and high-current areas. The challenge is entirely different for diagrams from the GaAs-QD dataset, where the transition lines are usually easy to identify but can fade (see first line in figure 2(b) and curve unexpectedly. This specificity moves the challenge to the exploration task, where missing a fading line can lead to a wrong number of charges in the QD. Finally, the diagrams from Si-OG-QD present a peculiar hysteresis noise when the reservoir tunneling rate is slower than the voltage sweeping speed (see top left in figure 2(c), which must be correctly classified by the model and taken into account by the exploration strategy. supplementary section S1.6 presents a diagram sample of each dataset. All measurement files used to generate the diagrams in the present study can be downloaded from [44].

3. Line detection

3.1. Methods

The first part of the procedure consists of automatically detecting charge transition lines in subsections of offline experimental stability diagrams (figure 3). An image segmentation model [45, 46] could identify lines, but it would require scanning large diagram sections, which is expected to slow down the autotuning process. To relax the input size requirement and reduce the model's complexity, we address this problem as a supervised binary classification task ('*line*' or '*no-line*'). The choice of a NN rather than alternative statistical methods [47–50] is firstly driven by the goal of designing a hardware-agnostic method that can adapt to the rapidly evolving QD research field. ML algorithms can automatically adjust to new data features, such as a new artifact caused by a novel measurement setup or new transition line patterns induced by a change in the QD hardware design. NN methods also have the potential to employ transfer learning methods [51] to facilitate swift adaptation of the model to newly introduced devices. Furthermore, NNs have demonstrated outstanding performance in detecting patterns within images and filtering various types of noise, especially convolutional neural networks (CNNs) [52–55]. Therefore, they are well suited for detecting lines within images, such as transition lines in two-dimensional stability diagrams. Finally, NNs are known to maintain good scalability [56] with respect to problem size and complexity; a crucial characteristic as we aim to develop a solution that is compatible with large arrays of QDs.

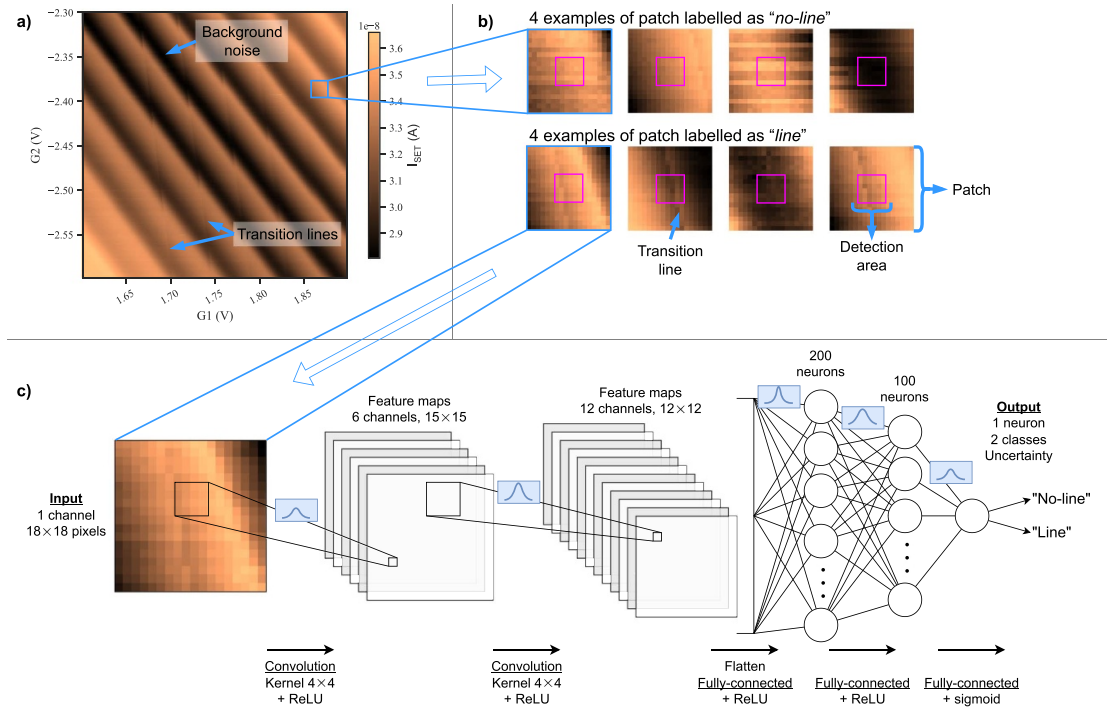

Figure 3. Automatic transition line detection using a Bayesian convolutional neural network (BCNN) in a stability diagram. (a) A subsection (patch) of the voltage space is measured. This diagram presents strong oscillating background noise that should not be misinterpreted as a transition line. (b) To train the model, each patch is categorized as '*line*' if a transition line annotation intersects with the detection area (in pink) in the center of the patch. Otherwise, the patch is categorized as '*no-line*'. (c) The patch is then sent as input to a model, where a forward pass propagates the information through convolutional and fully connected layers. In this example, a BCNN is represented, where each parameter is encoded as a Gaussian distribution.

For this classification problem, we evaluated the performance of three NN architectures: CNNs, Bayesian convolutional neural networks (BCNNs), and feed-forward neural networks (FFs). CNNs are known for their effectiveness in image classification tasks [54] and they provide a good tradeoff between complexity and performance. BCNNs are expected to offer robust uncertainty estimations [57, 58], which can be beneficial for improving the reliability of automatic charge-tuning procedures. BCNNs implement the variational inference [59, 60] method with the Bayes-by-Backprop [61] learning rule. Meanwhile, the simpler FF is a reference for comparison with the more complex CNN and BCNN models.

All NNs presented in this article are fed with the same inputs and have comparable binary outputs. The model input is a small subsection of the voltage space (also referred to as a patch), where pixels represent the measured current values. The patch size is fixed to 18×18 data points as an empirical tradeoff between measurement speed and model accuracy. The patches are generated by splitting the diagrams into evenly spaced squares and then distributing them into training, validation, and test sets, where the test patches are extracted from a unique diagram excluded from the training set, and the other patches are randomly distributed between the training (90%) and validation (10%) sets. Each patch is automatically classified as '*line*' if an annotation of the transition line intersects with the detection area in its center (figure 3(b)). This labeling approach provides more context to the NN while keeping the classification window narrow enough to fit between two transition lines.

The most basic NN used here is the FF, which propagates the input through two fully connected layers. The CNN extends this model by adding two convolution layers before the fully connected layers. The BCNN architecture is identical to the CNN, with each free parameter (weights and biases) encoded as a Gaussian distribution, defined by a mean and a variance (figure 3(c)). Each model is trained to infer the patch class as a binary output and express the uncertainty of this classification as a confidence percentage. More technical details about data processing, manual diagram annotation, automatic patch labeling, and NN meta-parameters can be found in supplementary sections S1 and S2.

Introducing a confidence score into our model provides an additional layer of information about the model's predictions. We leverage this information at the exploration level to significantly reduce the impact of misclassifications, which are usually responsible for disrupting the autotuning procedure. We distinguish between three types of uncertainties [62, 63]:

1. Model uncertainty (also known as epistemic or systemic uncertainty) arises from a lack of knowledge due to imperfect training. In the context of transition line detection, this could be due to non-optimal meta-parameters or a low number of training diagrams. This type of uncertainty can be reduced as the model is exposed to more data, highlighting the importance of a comprehensive training process. In practice, the size of the training set is restrained by the acquisition cost of the experimental stability diagrams.
2. Data uncertainty (also known as aleatoric or statistical uncertainty), on the other hand, is associated with the inherently stochastic nature of the tested data. For the line classification task, this could result from variability in measurement tools, fabrication imperfections, or cross-talk, among other factors. These factors are often out of the experimentalist's control.
3. Finally, distributional uncertainty arises from an inadequate representation of the test set within the training set. In the context of line classification, this can occur when the stability diagram used for testing presents noise or features that are not present in the training set. Not using synthetic data helps us mitigate this type of uncertainty, but diagram-to-diagram variability is a source of distributional uncertainty that can never be entirely avoided.

The computation of the confidence score is handled differently between standard (CNN or FF) and Bayesian NNs. For a standard NN, we employ a simple heuristic score we employ a simple heuristic score formula (1) to estimate uncertainty based on the distance between the output y of the NN and the closest class [64, 65]. This estimation is based on the probabilistic interpretation [66] of the cross-entropy function [67] used to optimize the model. In the case of BCNNs, we use the variational inference [60, 62] approach by computing the normalized standard deviation σ based on N repeated inferences with newly sampled parameters (Formula 2). Both formulas are trivial in the case of binary classification with one output neuron, but they can be generalized to multi-class problems [64]

$$\text{Heuristic confidence score} = |0.5 - y| \times 2 \tag{1}$$

$$\text{Bayesian confidence score} = 1 - \sigma\left(y_1, y_2, \ldots, y_N\right) \times 2. \tag{2}$$

In addition, we performed a calibration step to optimize the exploration–exploitation tradeoff [68, 69]. NNs are typically calibrated using regularization methods during training or by empirically scaling individual bins of the reliability diagram after training [62, 70–72] to ensure that the confidence score is a good approximation of the actual probability of correctness. For example, among all classifications rated with 80% confidence, 20% of them are expected to be incorrect. In our case, the validation set does not consistently cover the range of confidence scores due to the low number of classification errors in some cases (e.g. the middle panel of supplementary figure S6(b)). This constraint makes the computation of calibration metrics unreliable, specifically for bin-based approaches. We worked around this problem by calibrating the confidence thresholds instead of the model confidence scores. The threshold defines the confidence value under which the classification will be considered too uncertain to be trusted during stability diagram exploration. After each training, the threshold value is optimized by minimizing the score defined in Formula 3 through a grid search on the validation set. The goal is to find an optimal tradeoff between the number of errors above the threshold (Err) and the total number of samples under the threshold (UT), where τ is a meta-parameter that defines the target ratio (fixed to 0.2 in this study). This approach relaxes the requirement regarding the number of calibration samples while maintaining the practical functionality of the confidence score. More details about the calibration process can be found in supplementary section S2.2

$$\text{Threshold score} = \text{Err} + \text{UT} \times \tau. \tag{3}$$

3.2. Results

With appropriate meta-parameters (supplementary table S2), the line classification task reached more than 90% accuracy on every dataset and model combination (table 1). The most significant performance difference between the NN architectures is visible for the Si-SG-QD dataset, where convolution layers are necessary to filter the oscillating parasitic background. For the GaAs-QD and Si-OG-QD datasets, the benefits of the convolution are less evident, since the measurement noise is lower and the transition lines are often clearly visible. For all the datasets, the classification performance of the Bayesian CNN is close to that of their classical counterparts, but the additional complexity of the Bayesian layers slows down the training by a factor of 4 on average. No overfitting was observed during training (supplementary figure S5), suggesting that the dropout of the classical layers and the Bayesian layers both act as efficient regularization methods.

A qualitative analysis of the misclassified patch samples (figure 4 and supplementary section S2.3) suggests that we approach an optimal classification rate for each dataset, even if the model accuracy never

Table 1. Line detection results for each dataset and model. The performances are averaged over 10 runs using different random seeds. The standard deviation of the run performances represents the variability of the methods. Each run is a cross-validation across every diagram of the dataset. The best test accuracy scores of each dataset are highlighted in bold. The equivalent table without cross-validation is available in supplementary table S4.

Dataset	Model	Accuracy	Accuracy above	Error reduction using threshold	Rate below threshold
Si-SG-QD	BCNN	96.9% ± 0.1	99.3% ± 0.1	78.8% ± 3.0	4.8% ± 0.4
	CNN	96.9% ± 0.1	**99.4%** ± 0.1	82.4% ± 2.7	4.9% ± 0.5
	FF	90.4% ± 1.1	99.3% ± 0.1	90.8% ± 1.3	22.3% ± 1.3
GaAs-QD	BCNN	93.3% ± 0.5	96.2% ± 0.4	48.5% ± 6.8	9.5% ± 1.1
	CNN	94.6% ± 0.2	**97.4%** ± 0.2	59.6% ± 4.3	8.4% ± 0.8
	FF	93.1% ± 0.2	96.5% ± 0.4	57.5% ± 5.9	9.4% ± 1.6
Si-OG-QD	BCNN	92.5% ± 0.4	96.9% ± 0.3	56.8% ± 3.2	10.6% ± 0.7
	CNN	90.7% ± 0.3	**98.1%** ± 0.2	77.8% ± 1.9	16.2% ± 0.5
	FF	90.8% ± 0.2	97.9% ± 0.1	77.8% ± 1.0	17.0% ± 0.5

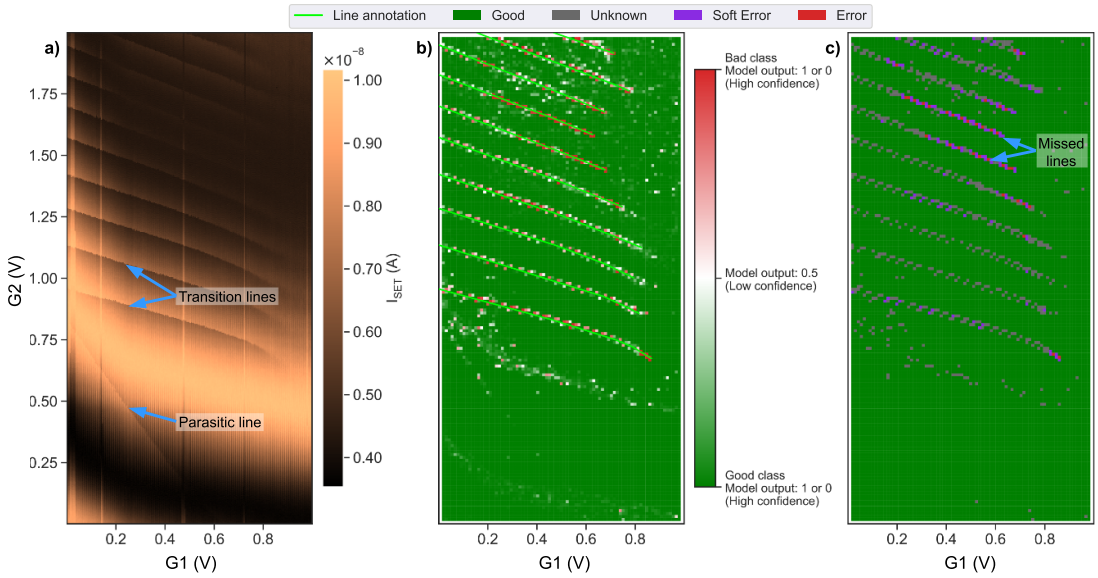

Figure 4. (a) Example of a stability diagram from the Si-OG-QD dataset. This specific diagram presents a parasitic line in the low-voltage area. (b) The same stability diagram is divided into patches classified using CNN trained using cross-validation [73] (supplementary figure S4). The color gradient of the patches represents the model uncertainty, where 0.5 is the lowest confidence score according to formula 1. The confidence score is lower in low-contrast areas, around the parasitic line, and near transition lines (due to possible intersection ambiguity between the line labels and the detection area of the patch). (c) The same stability diagram after the application of the confidence threshold. Most errors are avoided, except for the end of a few fading lines and some line sections in the low-current area at the top of the diagram. The 'soft errors' represent misclassifications near a line, which are not expected to induce tuning errors.

reaches 100%. The remaining errors are often caused by ambiguous inputs that would be hard to classify at the patch scale, even by a human expert. The most common causes of misclassification are as follows: *(i)* ambiguous labeling (a fading line or a line annotation near the detection area), *(ii)* annotation errors (position inaccuracy or human error), and *(iii)* strong noise at the patch location. The proportion of ambiguous patches varies between the datasets, which explains why the average line classification accuracy differs. The transition lines from GaAs-QD and Si-OG-QD can be irregular and fading (94.4% and 92.5% accuracy with the best models, respectively), while the Si-SG-QD lines are straight and more predictable, simplifying patch classification (96.9% accuracy with the CNN). The most challenging errors to address are the ones related to out-of-distribution issues caused by features in the test diagram that were missing from the training set. In an online scenario, this issue could occur if the device we are trying to tune is too different or if there are physical defects that did not exist in the previous experiments. This problem could be mitigated by a more diversified training set, a higher QD fabrication quality with less variability, or a reliable confidence score that correctly expresses the distributional uncertainty.

One can improve the effective model accuracy by excluding model predictions that are below the confidence threshold. The model outputs can then be interpreted as three-class inferences: '*line*' (output = 1

and confidence $\geqslant$ threshold), '*no-line*' (output = 0 and confidence $\geqslant$ threshold), and '*unknown*' (confidence $<$ threshold). On average, across all the datasets and models, this approach decreased the number of errors by 70%, at the price of 11% of the patches being classified as '*unknown*' (table 1 and supplementary figure S7). This method pushes the CNN classification accuracy above 97% for all the datasets, which allows for robust autotuning procedures. However, we did not observe any benefits of using the Bayesian confidence score compared to the classical heuristic.

4. Autotuning

4.1. Methods

Now that we have established a high-accuracy line detection method using NNs, we need to define an exploration strategy that uses model classification and uncertainty to efficiently explore the stability diagram space until the region of interest can be located. One step of this exploration is a cycle of the following: *(i)* patch measurement from a charge sensor on the device (simulated by extracting data from offline diagrams in this study), *(ii)* line detection using the trained NN, and *(iii)* deciding the following patch coordinates based on the exploration policy. The borders of the diagrams constrain the exploration to safe gate voltage ranges. The number of steps can be seen as a proxy for the relative tuning time, since the measurement is the longest part of the tuning process by orders of magnitude. The actual duration depends on the number of data points, the sensor type, and the electronic performance. For example, obtaining an 18×18 patch of Si-SG-QD takes approximately 2 min by measuring the SET current for the 324 points over a range of 36×36 mV using a *Keysight 34465A* multimeter. Although the measurement time can be shortened using more advanced and integrated equipment, it is likely to remain the limiting factor. Therefore, our exploration strategy aims to minimize the number of steps while maximizing the tuning success rate.

The typical strategy [23, 24, 27, 38] is to first search for the zero-electron regime, which is characterized by the absence of transition lines in a large area (bottom-left corners in the diagrams shown in figure 2). Then, we count the number of transition lines until we reach the desired regime. In the case of one-electron tuning, the target location will be between the first and second lines. However, this simple exploration strategy is very susceptible to misclassification or hardware changes.

To make the exploration robust and compatible with most QD technologies, we adapt this exploration strategy, as illustrated in figure 5 and a video[9]. This approach can adapt on the fly to different line slopes (step 2) and spacings (step 3) while checking for any fading lines (step 4). When a patch is classified with a confidence score below the threshold, we explore the space surrounding it to reduce the risk of critical failure due to NN misclassification. We validate or refute the presence of a line by taking additional steps in its supposed direction (see the purple arrow in figure 5(b), step 4) until a patch is classified with high confidence.

We evaluated this exploration strategy for each diagram that contained the one-electron regime within the measured voltage range (nine for each dataset). We used a k-fold cross-validation method [73] (supplementary figure S4) to test our approach on every valid diagram while preventing the inclusion of testing patches in the training set. This testing method should be close to an online autotuning situation, where we want to tune a new device based on previous experiments. The tuning success represents the proportion of final voltage coordinates inside the one-electron regime area out of 50 random starting points for every diagram in the dataset. The entire experiment is reproduced 10 times using different random seeds that affect the NN parameter initialization, the training process, and the random starting points. In total, 1.1×7 steps were evaluated in 81 000 offline autotuning simulations using 810 independently trained NNs.

4.2. Results

The results for each possible combination of dataset and NN are summarized in table 2, showing the autotuning success rate with and without leveraging the NN's confidence score (uncertainty-based). Using the model uncertainty information consistently reduces the number of tuning failures (-53% on average) at the price of a few additional steps ($+22\%$ on average). The best improvement is observed for the CNN on Si-SG-QD, where the tuning success rate increased from 88.8% to 99.5% when the autotuning procedure exploited the confidence score (-95% tuning failures and $+8.5\%$ steps).

Surprisingly, the confidence score provided by the Bayesian version of the CNN does not provide a significant improvement over the simple heuristic obtained using the standard CNN. The average line detection accuracy is similar for the CNN and BCNN on the Si-SG-QD dataset, and the line detection performance is slightly better for the BCNN on Si-OG-QD. However, on every dataset, the CNN reaches a higher tuning success by using the confidence heuristic score (highlighted in bold in table 2). Even in the case

[9] Method presentation and animated autotuning examples: https://youtu.be/9pPrgrIx9O0.

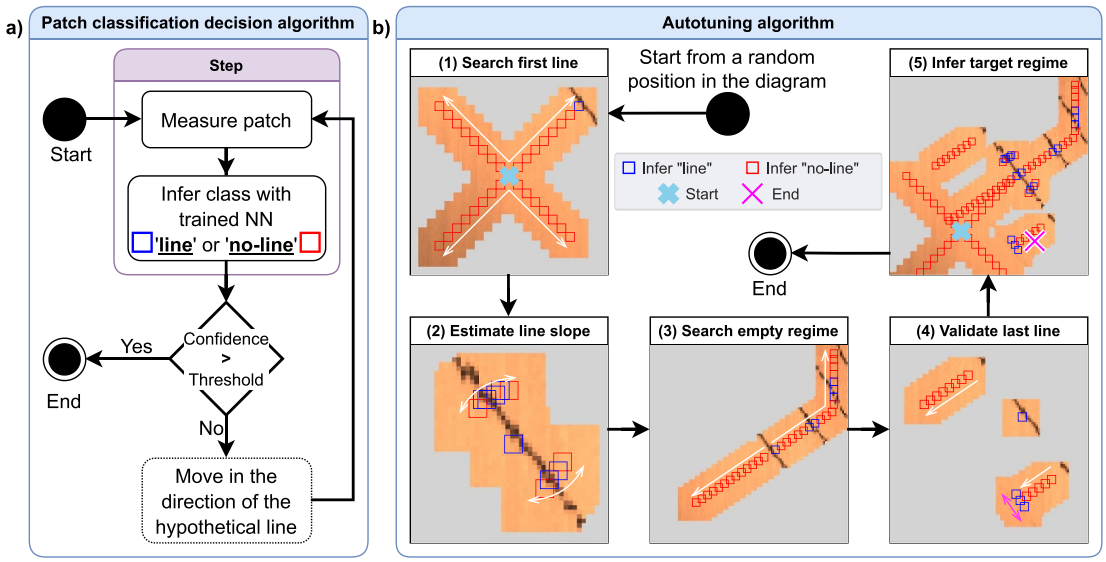

Figure 5. (a) Schematic representation of the patch classification algorithm. The purple box encloses the logic of one exploration step. (b) Step-by-step example of the autotuning algorithm using the GaAs-QD dataset (complete diagram in figure 2(b). The arrows represent the direction of the exploration, and the gray area is unmeasured space during the current step. *(1)* Search the first line by exploring the voltage space in 4 directions. *(2)* Estimate the line slope by scanning two sequences of patches in circular arcs around the first patch with a line. *(3)* Explore the space perpendicularly to the first line to gather information about the distance between lines. Stop after reaching 3 times the average distance without detecting a new line at lower voltages. *(4)* Search for possible missed lines under the first line by scanning multiple sections where we would expect to find a new line according to the average line distance and slope estimation. In this example, a fading line is correctly detected at the bottom of the image, but the low confidence of the model triggers a validation procedure. More scans are then processed on the hypothetical line direction (purple arrow) until a higher confidence inference validates or invalidates the line's existence. *(5)* Deduce the one-electron regime location based on the leftmost line position, the slope, and the average space between lines (all scans from previous steps are represented here). The algorithmic details of each step are available in supplementary section S3.1.

Table 2. Autotuning results for each dataset and model, with and without using the model uncertainty information provided by the confidence score. The line detection accuracy and tuning success variability are computed over 10 runs using different random seeds. Each run is a cross-validation over every diagram in the dataset. The best tuning success rates for each dataset are highlighted in bold. The tuning success rates can be compared to the baselines presented in supplementary table S3. The equivalent table without cross-validation is available in supplementary table S5.

Dataset	Model	Line detection accuracy	Uncertainty-based tuning	Average step number	Tuning success
Si-SG-QD	BCNN	96.9% ± 0.1	Yes	164	99.2% ± 0.7
			No	148	88.2% ± 2.3
	CNN	96.9% ± 0.1	Yes	165	**99.5%** ± 0.7
			No	152	88.8% ± 2.4
	FF	90.4% ± 1.1	Yes	194	72.4% ± 6.3
			No	122	23.9% ± 5.4
GaAs-QD	BCNN	93.3% ± 0.5	Yes	104	75.3% ± 4.2
			No	92	55.9% ± 3.8
	CNN	94.6% ± 0.2	Yes	103	**80.6%** ± 3.9
			No	93	72.4% ± 2.9
	FF	93.1% ± 0.2	Yes	105	72.8% ± 4.5
			No	92	58.4% ± 3.4
Si-OG-QD	BCNN	92.5% ± 0.4	Yes	185	75.2% ± 2.5
			No	155	66.3% ± 3.4
	CNN	90.7% ± 0.3	Yes	193	**78.1%** ± 1.7
			No	150	61.7% ± 3.0
	FF	90.8% ± 0.2	Yes	200	78.0% ± 2.5
			No	151	59.3% ± 2.4

where the BCNN line detection accuracy is higher (Si-OG-QD), the tuning success benefits more from the CNN confidence score (+16.4% tuning success) than the one from the BCNN (+8.9% tuning success). This result suggests that the number of critical misclassifications above the confidence threshold is more frequent with the Bayesian NN's uncertainty compared to the classical confidence heuristic score.

The line detection accuracy can provide valuable indications regarding the NN's performance, but it is not entirely correlated to the tuning success rate, since some misclassifications are much more harmful than others during diagram exploration. If a classification error occurs near a line (soft errors in figure 4(c), it will likely be corrected at the next step. However, an error on a line, or far from it, will likely affect the final tuning result. For example, if a noisy patch is wrongly classified as a line in the empty regime area, the number of charge carriers inside the QD will be overestimated by one.

The uncertainty-based autotuning method with convolution models allows for a nearly perfect tuning success rate (>99%) on the Si-SG-QD dataset. This high performance is attributable to *(i)* the convolution layers' ability to efficiently filter the oscillating background noise, *(ii)* the exploitation of the model confidence score to prevent most critical errors, and *(iii)* the straight transition lines that simplify the exploration. The line detection task also benefits from the large training set (72 000 patches) and the low variability between diagram features.

The GaAs-QD and Si-OG-QD datasets present more feature diversity in their diagrams with a smaller training set, which makes the classification task more challenging. The shape of the transition lines is also less predictable than the Si-SG-QD dataset, which increases the risk of missing a line or failing to detect its slope and spacing. The significantly higher tuning performance without cross-validation (supplementary table S5) suggests that the training sets do not cover the diversity of the stability diagrams. This is well illustrated by the CNNs on GaAs-QD, where the tuning success rate is improved by 11.6% without cross-validation (92.2% versus 80.6%). This hypothesis is confirmed by the high variability of the tuning success rates between the diagrams in these two datasets. The results are very polarized between the diagrams for which the one-electron regime is found nearly 100% of the time, while some are more complex to tune and exhibit a very low success rate. A larger or more consistent dataset could reduce the tuning success loss observed with cross-validation.

Overall, noise and hardware imperfections are relatively easy to manage for the line detection task but more challenging to deal with at the diagram scale during exploration. Therefore, the reliability of the autotuning procedure is mainly affected by the shape and consistency of the transition lines.

5. Discussion

The proposed method can be used to automate the charge tuning of any spin-based semiconductor single QD using less than 10 annotated stability diagrams for training and some basic prior knowledge of the device characteristics (expected working range voltage, approximate transition lines slope, and spacing). The autotuning procedure is resilient to noise and physical imperfections but sensitive to high device-to-device variability and training dataset quality (number of diagrams and feature coverage). This is well illustrated by the Si-SG-QD dataset, which features strong noise but consistent diagrams, allowing for a robust autotuning procedure with only 23 tuning failures over 4500 CNN uncertainty-based tuning simulations. Therefore, this method, associated with a good fabrication yield, has the potential to enable parallel charge tuning of large QD arrays, which is currently not practical using a manual approach.

This study demonstrates the benefits of using NN uncertainty to automate the tuning of QDs based on only partial measurements of the stability diagrams. The confidence score provides valuable information regarding the model prediction, which can be used to design robust autotuning procedures. However, we did not see a clear benefit of using a Bayesian NN over a standard one, even though Bayesian models are specifically designed to provide uncertainty measurements. This behavior could be explained by the distributional uncertainty being poorly captured by the Bayesian NNs [74], which could be the predominant source of uncertainty when using the cross-validation testing method. Moreover, many approximations [28, 60] are necessary to keep the Bayesian NN training and inference tractable, which could be detrimental to the overall quality of the model and its coherence with the initial Bayesian framework. Thus, the additional complexity and computing cost induced by the Bayesian NN is hard to justify when a simpler NN can provide a more reliable confidence score. In future work, other methods of uncertainty quantification [75] (e.g. dropout as Bayesian approximation [29] or ensemble learning [76]) and prior selection [77, 78] could be evaluated to improve the reliability of the confidence score. Bayesian optimization associated with classical machine learning [79, 80] also appears to be a promising avenue to harness the complexity of quantum engineering.

While applying quantum computing gates requires more than one QD, we developed and tested our approach on single-QD datasets as a first step toward robust double- and triple-QD autotuning. Since NNs are known for their good scalability, we expect this method to be expandable to larger input dimensions [35, 81] and greater numbers of classes. When the problem's complexity increases, the benefits of the confidence score could become even more important.

To avoid any wiring bottleneck [82] between the fridge and external electronics, this calibration method could be integrated near the QDs in the 4 K stage of the cryogenic environment. One way to meet this requirement would be to transfer the trained NN to low-power and cryo-compatible hardware, such as circuits based on arrays of memristive devices [83–85]. This specialized hardware takes advantage of in-memory computing [86, 87] to efficiently perform the multiply–accumulate operations [88] required for NN inference. Therefore, memristor-based systems represent promising candidates to realize scalable autotuning from inside the fridge with close-loop measurements and minimal disturbance over the QDs. Future studies should be performed to simulate these hardware implementations and demonstrate real-time online autotuning using the proposed method.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://doi.org/10.5281/zenodo.11 402 792.

Acknowledgments

V Y acknowledges Stefanie Czischek, who inspired this study by working on a first version of the autotuning procedure. The authors also acknowledge the experimentalists who provided the stability diagram measurements used in this paper (Michel Pioro-Ladriére, Marc-Antoine Roux, Marc-Antoine Genest, Julien Camirand-Lemire, and Sophie Rochette). V Y, B G, Y B, and D D acknowledge support from the National Science Engineering Research Council of Canada, Grant ALLRP 580722-22, and the Fonds de Recherche du Québec—Nature et Technologies, Grant 300253. C R, J R, A M, D L, and E D F acknowledge support from the FRQNT établissement de la relève professorale, Grant 2020–NC–268397, and the CRSNG, Grant RGPIN–2020–0573. R G M acknowledges support from NSERC and the Perimeter Institute for Theoretical Physics. Research at the Perimeter Institute is supported in part by the Government of Canada through the Department of Innovation, Science and Economic Development Canada and by the Province of Ontario through the Ministry of Economic Development, Job Creation and Trade.

Conflicts of interest

The authors declare no conflicts of interest.

Author contributions

All authors contributed to this article and approved of the submitted version.

Victor Yon: methodology, datasets aggregation and processing, software implementation, running experiments, results analysis and visualization, writing—original draft preparation.

Bastien Galaup: contributed to software implementation, running experiments, review and editing.

Claude Rohrbacher and Joffrey Rivard: provided data for the Si-OG-QD dataset, provided expertise and technical assistance with quantum dot technology, review and editing.

Clément Godfrin, Ruoyu Li, Stefan Kubicek, and Kristiaan De Greve: manufactured the devices used to obtain the Si-OG-QD dataset.

Louis Gaudreau: provided data for the GaAs-QD dataset, provided expertise and technical assistance on quantum dot technology, review and editing.

Yann Beilliard: methodology, supervision, funding acquisition, review and editing.

Eva Dupont-Ferrier and Roger Melko: supervision, review and editing.

Dominique Drouin: methodology, supervision, funding acquisition, review and editing.

Code and data availability

The Python source code used to aggregate and build the quantum dot stability diagram dataset is publicly accessible on *GitHub*: https://github.com/3it-inpaqt/qdsd-dataset.

The Python source code used to run all the experiments presented in this article is publicly accessible on *GitHub*: https://github.com/3it-inpaqt/dot-calibration-v2/tree/offline-article.

All simulation outputs and results used to generate the tables and figures presented in this article are publicly available for download from [89].

ORCID iDs

Victor Yon https://orcid.org/0000-0003-4517-5042
Bastien Galaup https://orcid.org/0009-0005-0384-1109
Claude Rohrbacher https://orcid.org/0009-0003-5789-3807
Clément Godfrin https://orcid.org/0000-0002-5244-3474
Ruoyu Li https://orcid.org/0000-0002-2145-7590
Kristiaan De Greve https://orcid.org/0000-0002-1314-9715
Louis Gaudreau https://orcid.org/0000-0002-1929-2715
Yann Beilliard https://orcid.org/0000-0003-0311-8840
Roger G Melko https://orcid.org/0000-0002-5505-8176
Dominique Drouin https://orcid.org/0000-0003-2156-967X

References

[1] Loss D and DiVincenzo D P 1998 Quantum computation with quantum dots *Phys. Rev.* A **57** 120–6
[2] Veldhorst M *et al* 2015 A two-qubit logic gate in silicon *Nature* **526** 410–4
[3] Watson T F *et al* 2018 A programmable two-qubit quantum processor in silicon *Nature* **555** 633–7
[4] Burkard G, Ladd T D, Pan A, Nichol J M and Petta J R 2023 Semiconductor spin qubits *Rev. Mod. Phys.* **95** 025003
[5] Takeda K *et al* 2016 A fault-tolerant addressable spin qubit in a natural silicon quantum dot *Sci. Adv.* **2** e1600694
[6] Yoneda J *et al* 2017 A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9 *Nat. Nanotechnol.* **13** 102–6
[7] Mills A R, Guinn C R, Gullans M J, Sigillito A J, Feldman M M, Nielsen E and Petta J R 2022 Two-qubit silicon quantum processor with operation fidelity exceeding *Sci. Adv.* **8** 99
[8] Noiri A, Takeda K, Nakajima T, Kobayashi T, Sammak A, Scappucci G and Tarucha S 2022 A shuttling-based two-qubit logic gate for linking distant silicon quantum processors *Nat. Commun.* **13** 5740
[9] Xue X, Russ M, Samkharadze N, Undseth B, Sammak A, Scappucci G and Vandersypen L M K 2022 Quantum logic with spin qubits crossing the surface code threshold *Nature* **601** 343–7
[10] Tyryshkin A M *et al* 2011 Electron spin coherence exceeding seconds in high-purity silicon *Nat. Mater.* **11** 143–7
[11] Veldhorst M *et al* 2014 An addressable quantum dot qubit with fault-tolerant control-fidelity *Nat. Nanotechnol.* **9** 981–5
[12] Petit L, Russ M, Eenink G H G J, Lawrie W I L, Clarke J S, Vandersypen L M K and Veldhorst M Design and integration of single-qubit rotations and two-qubit gates in silicon above one kelvin *Commun. Mater.* **3** 2022
[13] Maurand R *et al* 2016 A CMOS silicon spin qubit *Nat. Commun.* **7** 13575
[14] Dumoulin Stuyck N I *et al* 2021 Uniform spin qubit devices with tunable coupling in an all-silicon 300 mm integrated process *2021 Symp. on VLSI Circuits* (IEEE)
[15] Elsayed A *et al* 2024 Low charge noise quantum dots with industrial CMOS manufacturing *npj Quantum Inf.* **10** 2024
[16] Gonzalez-Zalba M F, de Franceschi S, Charbon E, Meunier T, Vinet M and Dzurak A S 2021 Scaling silicon-based quantum computing using CMOS technology *Nat. Electron.s* **4** 872–84
[17] Rohrbacher C, Rivard J, Ritzenthaler R, Bureau B, Lupien C, H, Horiguchi N and Dupont-Ferrier E 2023 Dual operation of gate-all-around silicon nanowires at cryogenic temperatures: FET and quantum dot (arXiv: 2312.00903)
[18] Liu X and Hersam M C 2019 2D materials for quantum information science *Nat. Rev. Mater.* **4** 669–84
[19] Saraiva A, Han Lim W, Hwan Yang C, Escott C C, Laucht A and Dzurak A S 2021 Materials for silicon quantum dots and their impact on electron spin qubits *Adv. Funct. Mater.* **32** 2105488
[20] Patel R, Agrawal Y and Parekh R 2020 Single-electron transistor: review in perspective of theory, modelling, design and fabrication *Microsyst. Technol.* **27** 1863–75
[21] Simmons C B, Thalakulam M, Shaji N, Klein L J, Qin H, Blick R H, Savage D E, Lagally M G, Coppersmith S N and Eriksson M A 2007 Single-electron quantum dot in Si/SiGe with integrated charge sensing *Appl. Phys. Lett.* **91** 213103
[22] Baart T A, Eendebak P T, Reichl C, Wegscheider W and Vandersypen L M K 2016 Computer-automated tuning of semiconductor double quantum dots into the single-electron regime *Appl. Phys. Lett.* **108** 213104
[23] Czischek S *et al* 2021 Miniaturizing neural networks for charge state autotuning in quantum dots *Mach. Learn.: Sci. Technol.* **3** 015001
[24] Durrer R, Kratochwil B, Koski J V, Landig A J, Reichl C, Wegscheider W, Ihn T and Greplova E 2020 Automated tuning of double quantum dots into specific charge states using neural networks *Phys. Rev. Appl.* **13** 054019
[25] LeCun Y, Bengio Y and Hinton G 2015 Deep learning *Nature* **521** 436–44
[26] Ziegler J, McJunkin T, Joseph E S, Kalantre S S, Harpt B, Savage D E, Lagally M G, Eriksson M A, Taylor J M and Zwolak J P 2022 Toward robust autotuning of noisy quantum dot devices *Phys. Rev. Appl.* **17** 024069
[27] Ziegler J, Luthi F, Ramsey M, Borjans F, Zheng G and Zwolak J P 2023 Tuning arrays with rays: physics-informed tuning of quantum dot charge states *Phys. Rev. Appl.* **20** 034067
[28] Goan E and Fookes C 2020 *Bayesian Neural Networks: An Introduction and Survey* (Springer) pp 45–87
[29] Gal Y 2016 Uncertainty in deep learning *PhD Thesis* University of Cambridge
[30] Liu H, Wang B, Wang N, Sun Z, Yin H, Li H, Cao G and Guo G 2022 An automated approach for consecutive tuning of quantum dot arrays *Appl. Phys. Lett.* **121** 084002
[31] Zwolak J P, McJunkin T, Kalantre S S, Dodson J P, MacQuarrie E R, Savage D E, Lagally M G, Coppersmith S N, Eriksson M A and Taylor J M 2020 Autotuning of double-dot devices in situ with machine learning *Phys. Rev. Appl.* **13** 034075
[32] Severin B *et al* 2024 Cross-architecture tuning of silicon and SiGe-based quantum devices using machine learning *Sci. Rep.* **14**
[33] Kalantre S S, Zwolak J P, Ragole S, Wu X, Zimmerman N M, Stewart M D and Taylor J M 2019 Machine learning techniques for state recognition and auto-tuning in quantum dots *npj Quantum Inf.* **5** 6
[34] Darulová J, Troyer M and Cassidy M C 2021 Evaluation of synthetic and experimental training data in supervised machine learning applied to charge-state detection of quantum dots *Mach. Learn.: Sci. Technol.* **2** 045023

[35] Moon H *et al* 2020 Machine learning enables completely automatic tuning of a quantum device faster than human experts *Nat. Commun.* **11** 4161
[36] Perron J K, Stewart M D and Zimmerman N M 2015 A quantitative study of bias triangles presented in chemical potential space *J. Phys.: Condens. Matter* **27** 235302
[37] Hensgens T 2018 Emulating Fermi-Hubbard physics with quantum dots *Phd Thesis* Delft University of Technology
[38] Lapointe-Major M, Germain O, Camirand Lemyre J, Lachance-Quirion D, Rochette S, Camirand Lemyre F and Pioro-Ladriére M 2020 Algorithm for automated tuning of a quantum dot into the single-electron regime *Phys. Rev.* B **102** 085301
[39] Teske J D, Sebastian Humpohl S, Otten R, Bethke P, Cerfontaine P, Dedden J, Ludwig A, Wieck A D and Bluhm H 2019 A machine learning approach for automated fine-tuning of semiconductor spin qubits *Appl. Phys. Lett.* **114** 133102
[40] Zwolak J P and Taylor J M 2023 Colloquium : advances in automation of quantum dot devices control *Rev. Mod. Phys.* 193101 **95**
[41] Rochette S *et al* Quantum dots with split enhancement gate tunnel barrier control *Appl. Phys. Lett.* **114** 083101
[42] Gaudreau L, Kam A, Granger G, Studenikin S A, Zawadzki P and Sachrajda A S 2009 A tunable few electron triple quantum dot *Appl. Phys. Lett.* **95** 193101
[43] Zwolak J P, Kalantre S S, Wu X, Ragole S and Taylor J M 2018 QFlow lite dataset: a machine-learning approach to the charge states in quantum dot experiments *PLoS One* **13** e0205844
[44] Yon V et al 2024 Quantum dots stability diagrams dataset *Zenodo* (https://doi.org/10.5281/zenodo.11402792)
[45] Minaee S, Boykov Y Y, Porikli F, Plaza A J, Kehtarnavaz N and Terzopoulos D 2021 Image segmentation using deep learning: a survey *IEEE Trans. Pattern Anal. Mach. Intell.* **44** 3523–42
[46] Zhang Y, Shen Z and Jiao R 2024 Segment anything model for medical image segmentation: current applications and future directions *Comput. Biol. Med.* **171** 108238
[47] Ding L and Goshtasby A 2001 On the Canny edge detector *Pattern Recognit.* **34** 721–5
[48] Bishop C M 2006 *Pattern Recognition and Machine Learning* (*Information Science and Statistics*) 1st edn (Springer)
[49] Mukhopadhyay P and Chaudhuri B B 2015 A survey of Hough transform *Pattern Recognit.* **48** 993–1010
[50] Sun R, Lei T, Chen Q, Wang Z, X, Zhao W and Nandi A K 2022 Survey of image edge detection *Front. Signal Process.* **2** 826967
[51] Bengio Y 2012 Deep learning of representations for unsupervised and transfer learning *Proc. ICML Workshop on Unsupervised and Transfer Learning* (*Proc. Machine Learning Research Bellevue, Washington, USA*) vol 27, ed I Guyon, G Dror, V Lemaire, G Taylor and D Silver (PMLR) pp 17–36
[52] Krizhevsky A, Sutskever I and Hinton G E 2017 ImageNet classification with deep convolutional neural networks *Commun. ACM* **60** 84–90
[53] Patil A and Rane M 2020 *Convolutional Neural Networks: An Overview and Its Applications in Pattern Recognition* (Springer) pp 21–30
[54] Chen L, S, Bai Q, Yang J, Jiang S and Miao Y 2021 Review of image classification algorithms based on convolutional neural networks *Remote Sens.* **13** 4712
[55] Li Z, Liu F, Yang W, Peng S and Zhou J 2022 A survey of convolutional neural networks: analysis, applications and prospects *IEEE Trans. Neural Netw. Learn. Syst.* **33** 6999–7019
[56] Lienhard B *et al* 2022 Deep-neural-network discrimination of multiplexed superconducting-qubit states *Phys. Rev. Appl.* **17** 014024
[57] Kwon Y, Won J-H, Joon Kim B J and Cho Paik M C 2020 Uncertainty quantification using Bayesian neural networks in classification: application to biomedical image segmentation *Comput. Stat. Data Anal.* **142** 106816
[58] Valentin Jospin L, Laga H, Boussaid F, Buntine W and Bennamoun M 2022 Hands-on Bayesian neural networks-a tutorial for deep learning users *IEEE Comput. Intell. Mag.* **17** 29–48
[59] Hinton G E and van Camp D 1993 Keeping the neural networks simple by minimizing the description length of the weights *Proc. 6th Annual Conf. on Computational Learning Theory, COLT '93* (ACM Press)
[60] Graves A 2011 Practical variational inference for neural networks *Advances in Neural Information Processing Systems* vol 24, ed J Shawe-Taylor, R Zemel, P Bartlett, F Pereira and K Q Weinberger (Curran Associates, Inc)
[61] Blundell C, Cornebise J, Kavukcuoglu K and Wierstra D 2015 Weight uncertainty in neural networks (arXiv: 1505.05424)
[62] Gawlikowski J, Rovile Njieutcheu Tassi C, Ali M, Lee J, Humt M, Feng J, Kruspe A, Triebel R, Jung P, Roscher R, Shahzad M, Yang W, Bamler R and Xiang Zhu X 2021 A survey of uncertainty in deep neural networks (arXiv: 2107.03342)
[63] Smith L and Gal Y 2018 Understanding measures of uncertainty for adversarial example detection (arXiv: 1803.08533)
[64] Zaragoza H and d'Alché Buc F 1998 Confidence measures for neural network classifiers *Proc. 7th Int. Conf. Information Processing and Management of Uncertainty in Knowlegde Based Systems* (*Citeseer*) vol 9
[65] Mandelbaum A and Weinshall D 2017 Distance-based confidence score for neural network classifiers (arXiv: 1709.09844)
[66] Ramos D, Franco-Pedroso J, Lozano-Diez A and Gonzalez-Rodriguez J 2018 Deconstructing cross-entropy for probabilistic binary classifiers *Entropy* **20** 208
[67] Mao A, Mohri M and Zhong Y 2023 Cross-entropy loss functions: theoretical analysis and applications *Proc. 40th Int. Conf. on Machine Learning* (*Proc. Machine Learning Research*) vol 202 (PMLR) (available at: https://proceedings.mlr.press/v202/mao23b.html) pp 23803–28
[68] Auer P 2002 Using confidence bounds for exploitation-exploration trade-offs *J. Mach. Learn. Res.* **3** 397–422
[69] Simpkins A, de Callafon R and Todorov E 2008 Optimal trade-off between exploration and exploitation *2008 American Control Conf.* (IEEE)
[70] Guo C, Pleiss G, Sun Y and Weinberger K Q 2017 On calibration of modern neural networks (arXiv: 1706.04599)
[71] Vaicenavicius J, Widmann D, Andersson C, Lindsten F, Roll J and Schön T B 2019 Evaluating model calibration in classification (arXiv: 1902.06977)
[72] Silva Filho T, Song H, Perello-Nieto M, Santos-Rodriguez R, Kull M and Flach P 2023 Classifier calibration: a survey on how to assess and improve predicted class probabilities *Mach. Learn.* **112** 3211–60
[73] Raschka S 2018 Model evaluation, model selection, and algorithm selection in machine learning (arXiv: 1811.12808)
[74] Izmailov P, Vikram S, Hoffman M D and Gordon Gordon Wilson A 2021 What are Bayesian neural network posteriors really like? *Proc. 38th Int. Conf. on Machine Learning* (*Proc. Machine Learning Research*) vol 139, ed M Meila and T Zhang (PMLR) (available at: https://proceedings.mlr.press/v139/izmailov21a.html) pp 4629–40
[75] Abdar M *et al* 2021 A review of uncertainty quantification in deep learning: techniques, applications and challenges *Inf. Fusion* **76** 243–97
[76] Lakshminarayanan B, Pritzel A and Blundell C 2016 Simple and scalable predictive uncertainty estimation using deep ensembles (arXiv: 1612.01474)

[77] Fortuin V, Garriga-Alonso A, Ober S W, Wenzel F, Rätsch G, Turner R E, van der Wilk M and Aitchison L 2021 Bayesian neural network priors revisited (arXiv: 2102.06571)
[78] Silvestro D and Andermann T 2020 Prior choice affects ability of Bayesian neural networks to identify unknowns (arXiv: 2005.04987)
[79] Gebhart V, Santagati R, Andrea Gentile A, Gauger E M, Craig D, Ares N, Banchi L, Marquardt F, Pezzé L and Bonato C 2023 Learning quantum systems *Nat. Rev. Phys.* **5** 141–56
[80] Szulakowska L and Dai J 2022 Bayesian autotuning of Hubbard model quantum simulators (arXiv: 2210.03077)
[81] Krause O, Brovang B, Rasmussen T, Chatterjee A and Kuemmeth F 2022 Estimation of convex polytopes for automatic discovery of charge state transitions in quantum dot arrays *Electronics* **11** 2327
[82] Reilly D J 2019 Challenges in scaling-up the control interface of a quantum computer *2019 IEEE Int. Electron Devices Meeting (IEDM)* (IEEE)
[83] Mouny P-A, Dawant R, Galaup B, Ecoffey S, Pioro-Ladrière M, Beilliard Y and Drouin D 2023 Analog programming of CMOS-compatible Al_2O_3/TiO_{2-x} memristor at 4.2K after metal-insulator transition suppression by cryogenic reforming *Appl. Phys. Lett.* **123** 163505
[84] Marcotte F'eric, Mouny P-A, Yon V, Dagnew G A, Kulchytskyy B, Rochette S, Beilliard Y, Drouin D and Ronagh P 2023 A cryogenic memristive neural decoder for fault-tolerant quantum error correction (arXiv: 2307.09463)
[85] Dawant R *et al* 2024 Damascene versus subtractive line CMP process for resistive memory crossbars BEOL integration *Micro Nano Eng.* **23** 100251
[86] Christensen D V *et al* 2022 2022 roadmap on neuromorphic computing and engineering *Neuromorph. Comput. Eng.* **2** 022501
[87] Wang W, Song W, Yao P, Li Y, Van Nostrand J, Qiu Q, Ielmini D and Joshua Yang J 2020 Integration and co-design of memristive devices and algorithms for artificial intelligence *iScience* **23** 101809
[88] Amirsoleimani A, Alibart F, Yon V, Xu J, Pazhouhandeh M R, Ecoffey S, Beilliard Y, Genov R and Drouin D 2020 In memory vector matrix multiplication in monolithic complementary metal-oxide-semiconductor memristor integrated circuits: design choices, challenges and perspectives *Adv. Intell. Syst.* **2** 11
[89] Yon V, Galaup B, Melko R G, Beilliard Y and Drouin D 2024 Robust quantum dots charge autotuning using neural network uncertainty - output data *Zenodo* (https://doi.org/10.5281/zenodo.11403192)