Editors' Suggestion

Automated Tuning of Double Quantum Dots into Specific Charge States Using Neural Networks

R. Durrer,[1] B. Kratochwil,[1] J.V. Koski,[1] A.J. Landig,[1] C. Reichl,[1] W. Wegscheider,[1] T. Ihn,[1] and E. Greplova[2,*]

[1] *Department of Physics, ETH Zürich, Otto-Stern-Weg 1, CH-8093 Zürich, Switzerland*

[2] *Department of Physics, ETH Zürich, Wolfgang-Pauli-Str. 27, CH-8093 Zürich, Switzerland*

(Received 9 December 2019; revised manuscript received 5 March 2020; accepted 6 April 2020; published 8 May 2020)

While quantum dots are at the forefront of quantum-device technology, the tuning of multidot systems requires a lengthy experimental process as multiple parameters need to be accurately controlled. This process becomes increasingly time-consuming and difficult to perform manually as the devices become more complex and the number of tuning parameters grows. In this work, we present a crucial step toward automated tuning of quantum-dot qubits. We introduce an algorithm driven by machine learning that uses a small number of coarse-grained measurements as its input and tunes the quantum-dot system into a preselected charge state. We train and test our algorithm on a GaAs double-quantum-dot device and we consistently arrive at the desired state or its immediate neighborhood.

DOI: 10.1103/PhysRevApplied.13.054019

I. INTRODUCTION

Engineered quantum devices are currently at the forefront of scientific research [1–3]. The enormous progress in the precision of the engineering and control of these devices at the level of individual particles or excitations allows researchers to build quantum-coherent devices such as qubits, which are the building blocks of quantum-information-processing architectures. Such devices have successfully been built utilizing different platforms such as N-V centers [4,5], superconducting qubits [6,7], thin nanomaterials [8], and nanowires [9,10], to name just a few.

Extrapolating the success of semiconductor technology and its scaling properties from the past into the future, qubits based on gate-defined semiconductor quantum dots (QDs) are seen as promising candidates for scalable quantum computing. Experiments have successfully demonstrated long coherence times [11], fast gate operations [12,13], and long-distance qubit coupling [14–17]. However, to change the number of electrons and/or holes in the QDs and to tune the coupling to neighboring QDs and reservoirs, an operator has to change the corresponding voltages applied to the gate electrodes. The relation between the applied gate voltages and the physical parameters to be tuned is highly device specific and, thus, requires calibration measurements. Automation of the tuning process is therefore one of the key challenges in making semiconductor architectures scalable.

Progress in characterizing, controlling, and tuning complex quantum systems has been achieved through a variety of algorithmic methods [18–21]. The application of machine-learning techniques for parameter estimation and the tuning of quantum systems has been a promising avenue within this endeavor [22–29]. Machine-learning methods can be used to automate tasks previously done by humans [30] and construct high-quality abstract models interpreting complex measurements. They are fast to evaluate even without implementing a device- or system-specific physical model, which can be complex and lengthy to simulate in the case of QD qubits. In particular, a scheme for the efficient measurement of the charge-stability diagram driven by machine learning has previously been introduced in Ref. [22]. Automated tuning to a specific QD regime [a single or double quantum dot (DQD)] has been recently achieved in Refs. [26], [28], and [31].

In this work, we present a stepping stone toward fully automated QD tuning. Our algorithm automatically tunes a double DQD that is initially in an unknown charge state into a predefined charge state. Thus, our method represents a machine-learning approach that complements the work presented in Refs. [26], [28], and [31] by bringing QDs tuned to a DQD regime into a desired charge configuration. This is relevant as different qubit implementations require a well-defined number of electrons in each QD, such as for the hybrid qubit [32,33], the resonant exchange qubit [34,35], or the quadrupolar exchange-only qubit [36]. We employ convolutional neural networks to recognize transitions between charge states in the charge-stability diagram. We test our method by training and testing it

*geliska@phys.ethz.ch

experimentally with two DQD devices based on a GaAs heterostructure and we evaluate its performance.

II. EXPERIMENTAL SETUP

A scanning electron micrograph of our device, which is capable of forming up to three QDs, is shown in Fig. 1. The device has a two-dimensional electron gas 90 nm below the surface, embedded into a GaAs-(Al,Ga)As heterostructure. We confine electrons by applying negative voltages to gold gates on top of the heterostructure. The design permits the formation of up to three QDs (labeled QD1, QD2, and QD3) in a linear array. This allows us to realize two different DQD devices, formed either by QD1 and QD2 (DQD1) or by QD2 and QD3 (DQD2). Whenever we form a DQD, the remaining QD is not formed and is therefore part of the reservoir. Three finger gates, visible at the bottom of Fig. 1, are used individually to define a quantum point contact (QPC) measuring charge transitions in the nearby QDs. When we measure charge transitions of DQD1 (DQD2), we chose the middle (right) gate as the QPC gate and keep the remaining two gates grounded.

Figure 2 shows a charge-stability diagram of DQD1. We plot the change $\partial I_{\mathrm{QPC}}/\partial V_{\mathrm{PG}}$ of the current I_{QPC} in the quantum point contact as a function of the plunger gate voltages V_{PG1} and V_{PG2}. A change of the electron number in either of the dots causes a sharp peak in $\partial I_{\mathrm{QPC}}/\partial V_{\mathrm{PG}}$, seen as tilted lines in Fig. 2. For a mathematical definition of $\partial I_{\mathrm{QPC}}/\partial V_{\mathrm{PG}}$, we refer to Eq. (A1) in Appendix A. The slopes, intensities, and separations of these lines, called charge-transition lines, are not precisely reproducible with present-day technologies in nominally identical devices and are therefore device specific. Thus, the charge-stability diagrams of distinct devices may look different. Furthermore, the properties of the charge-stability diagrams may vary for different gate voltages using the same device

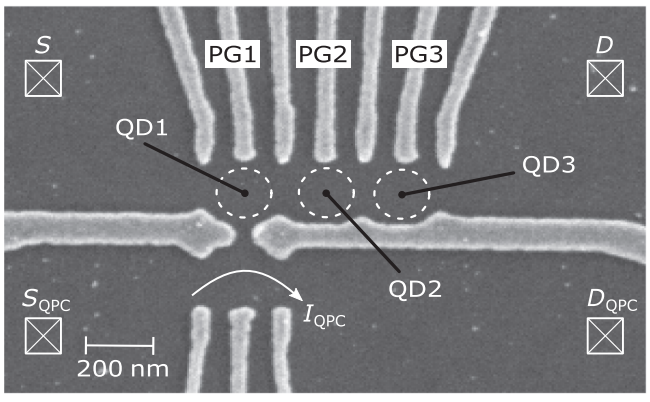

FIG. 1. A scanning electron microscope image of the GaAs-(Al,Ga)As sample. The dashed circles indicate the positions of the three QDs that can be formed. The crossed boxes indicate ohmic contacts. For this work, we form either DQD1 by using QD1 and QD2 or DQD2 using QD2 and QD3.

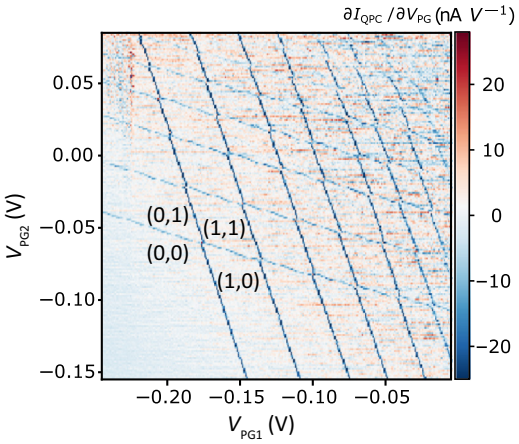

FIG. 2. The charge-stability diagram of the DQD based on QD1 and QD2 from the sample in Fig. 1. The charge occupation is indicated as (n, m), where n denotes the number of electrons on QD1 and m the number of electrons on QD2.

[37,38]. This diversity of charge-stability diagrams makes manual tuning of such quantum devices a matter of human supervision, where the experience of the operator influences the efficiency of the process.

We label the charge states as (n, m), where n and m are the numbers of electrons in QD1 and QD2, respectively. We assign the charge states to plaquettes of the charge-stability diagram by counting the number of charge-transition lines starting from the $(0, 0)$ state, as illustrated in Fig. 2.

The starting point for the procedures developed here is any gate-voltage setting, where a DQD is in the Coulomb-blockade regime, with an unknown and arbitrary charge state. The problem we attempt to solve is then stated as follows: given a DQD in an unknown initial charge configuration (n_i, m_i), find a configuration of plunger gate voltages V_{PG1} and V_{PG2} for which the charge configuration is equal to the preselected charge state (n_f, m_f) required for operating the qubit.

III. ALGORITHM

We split the task of finding the required charge configuration into two stages. Starting from (n_i, m_i), the first task is to find a plunger gate-voltage configuration for which the DQD is completely empty [charge state $(0, 0)$]. In the second stage, the algorithm loads electrons onto the QDs until the desired charge occupation (n_f, m_f) is reached. For each of the two stages, we use a convolutional neural network trained to recognize charge-transition lines in the charge-stability diagram. A detailed discussion of the machine-learning models used is provided in Sec. IV.

As illustrated in Fig. 3(a), we find the $(0, 0)$ charge configuration by measuring small rectangular patches of the charge-stability diagram at low resolution. If the

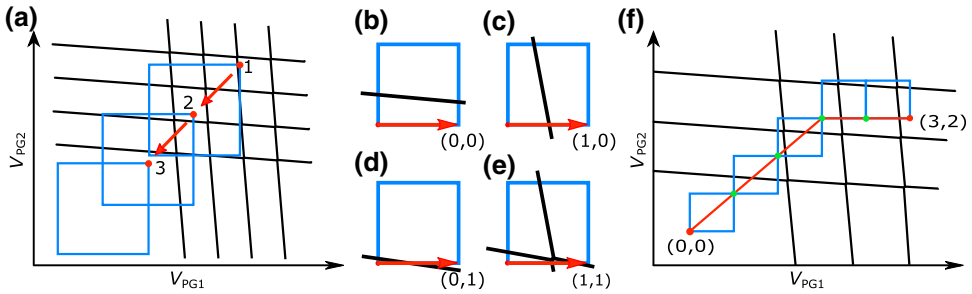

FIG. 3. (a) A schematic illustration of the algorithm finding a charge reference point. The black lines indicate charge-transition lines and the blue boxes are the patches used for the classification with the neural network. The red dot at the top right corner of a patch corresponds to the point in the charge-stability diagram for which the charge occupation is assessed. (b)–(e) The schematics of the patches for the second neural network in blue and the charge-transition lines in black. The red arrows indicate path segments. Along each of the indicated path segments, a different charge transition occurs, indicated as (n, m) at the end of the arrows. (f) An illustration of a path (red line) connecting the (0, 0) charge regime with the (3, 2) charge regime. The path constitutes of five path segments belonging to the patches in blue. The algorithm chooses the diagonal path when electron number in each quantum dot needs to be increased. As soon as the final occupation on the second dot is reached, only horizontal path segments are used to load electrons on QD1 only.

algorithm identifies any charge-transition lines in the measured patch, it decreases both plunger gate voltages. This process is repeated until no charge-transition lines are recognized by the convolutional neural network. Compared to the measurement shown in Fig. 2 (resolution 1 mV, 241 × 241 points), the patches are low resolution (resolution 6–9 mV) and small size (20 × 20 points) and thus fast to measure. However, the size of these patches is chosen to be at least twice as large as the largest line spacing between two consecutive charge-transition lines to ensure that no line is missed. We can determine the patch size experimentally by one-dimensional plunger gate sweeps or estimate it from the geometric capacitance between the gate and the dots [37]. We refer to the plunger gate-voltage configuration, for which the algorithm determines the DQD to be empty, as the "reference point." This reference point is the starting point for the second stage of the tuning algorithm.

In the second stage, we tune from the reference point to the desired charge state (n_f, m_f). We do this by again measuring rectangular patches of the charge-stability diagram and subsequently classifying them using another convolutional neural network. However, the patches are now 28 × 28 measurement points with a finer resolution (of 1 mV). Their size is chosen such that at most one charge transition per QD occurs in between any two corners of the patch. We determine this step size by measuring half the distance between the first and second electron charge-transition lines in the charge-stability diagram. For the desired final occupation numbers that we investigate within the scope of this work, this is no limitation. For higher occupation numbers, however, the distance between successive charge transitions decreases. This would require additional training and adjustment of parameters. The convolutional neural network is trained to recognize charge transitions occurring between the lower left corner of the patch and the remaining corners. The four different classification outcomes of this step (no transition lines, a vertical transition line, a horizontal transition line, or both vertical and horizontal transition lines) are illustrated in Figs. 3(b)–3(e). We place the lower left corner of the first patch at the $(0, 0)$ reference point. We then proceed as depicted in Fig. 3(f). The patches are connected and jointly build a path (depicted with the red line) along which charge transitions are identified. For each corner (green dots) of the patches, the charge occupation of the DQD is known. The path connecting the charge reference point and the desired charge regime (n_f, m_f) can be understood as follows. As long as the electron number in each quantum dot needs to be increased, a diagonal path element is added. As soon as the final occupation on one of the dots is reached, horizontal (vertical) path segments are chosen. This allows us to increase the charge occupation on dot 1 (2), respectively—and if there are too many electrons on a quantum dot, the respective plunger gate voltage is reduced again.

IV. MACHINE-LEARNING MODELS

We use two different neural network-based models for the two stages described above, namely to (a) find the zero-charge reference point $(0, 0)$ and (b) determine the path in the parameter space of plunger gate voltages toward the preselected charge configuration (n_f, m_f). Both models are trained to recognize the charge-transition lines in the measured patches. The plunger gate voltages are then adjusted based on the presence and direction of the charge-transition lines.

The convolutional neural network for the first stage of the algorithm, responsible for finding the $(0, 0)$ state, is trained by supervised learning using coarsely measured patches of 20 × 20 data points in the charge-stability

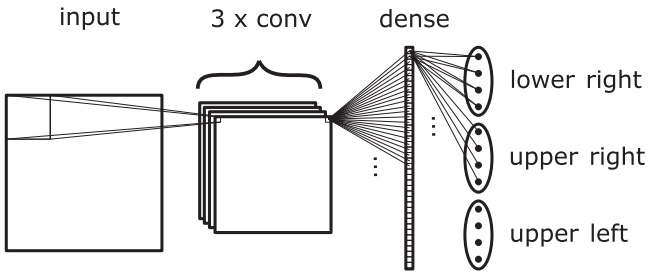

FIG. 4. The architecture of the convolutional neural network that recognizes charge transitions between the corners of a given input frame. The neural network consists of three convolutional layers, one dense layer, and three outputs with four classification outcomes each.

diagram as an input. The output of the network is binary: either both QDs are empty or at least one of them is occupied. In other words, since the patch of the charge-stability diagram extends beyond the maximal spacing of two charge-transition lines, the network effectively detects whether there is a charge-transition line for any of the QDs within the patch. The model we use for the first part of the algorithm consists of two convolutional and one dense layer. For the full technical description, see Appendix B.

The neural network for the second stage of the tuning algorithm is trained as a classifier that recognizes individual charge transitions. The output of the network is then used to reach an arbitrary final charge state (n_f, m_f) from the state $(0, 0)$. The input of this more complex classifier consists of finer 28×28 patches of the charge-stability diagram. The output contains 12 elements that precisely specify which charge transition has been observed in a given patch. With respect to the bottom left corner, we distinguish the following occupation-number changes for each of the three remaining corners: $(n, m) \rightarrow (n, m)$, $(n, m) \rightarrow (n + 1, m)$, $(n, m) \rightarrow (n, m + 1)$, and $(n, m) \rightarrow (n + 1, m + 1)$. The mentioned scenarios are shown in Figs. 3(b)–3(e) and are evaluated in the network shown in Fig. 4. We want to stress at this point that the algorithm is not trained to recognize specific charge transitions, e.g., $(1, 1) \rightarrow (1, 2)$, but general charge-transition lines, as explained above. Apart from this specific structure of the output design, which we choose to fit to the physics of interest, the body of the network has a standard construction with three convolutional layers and one dense layer. For the technical details of the architecture, see Appendix B.

We train both neural networks on experimentally measured data. All data used for the training is measured on DQD1. In particular, we measure 128 (470) complete charge stability diagrams of DQD1 in finer resolution (coarse resolution) and the operator marks each charge-transition line for all data sets. For each run, we choose different plunger gate-voltage ranges, while keeping the other voltages constant. From each charge-stability diagram, we cut out random patches and we use a script to automatically translate the marked transition lines into labels for each patch. For the first stage of the algorithm, the labels are single binary variables, while for the second stage of the algorithm each patch has three labels, i.e., one for each corner (see Fig. 4). To create a richer training set, we augment each charge-stability diagram by a factor of 18. The augmentation is achieved by various combinations of rotations and axis scaling and allows us to include a larger variety of physically plausible measurement scenarios in our training set (for details, see Appendix A). The resulting training sets contain on the order of 10^5 patches and both networks are optimized to minimize the error between the labels assigned by the human operator and those assigned by the network.

V. RESULTS

To analyse the behavior of the algorithm, we use two methods of performance testing. First, we evaluate the performance on a labeled data set acquired on DQD1. In a second step, we test the performance of the algorithm actively on the sample (both on DQD1 and DQD2) by letting it tune the DQD into a desired selected charge configuration. Below, we discuss in detail the results of both types of performance evaluation.

First, we evaluate the performance of the two stages of the algorithm on the labeled evaluation data set acquired on DQD1. After training the first stage of the algorithm on 650 000 labeled patches, we evaluate its accuracy in distinguishing a patch showing a completely empty dot or a patch containing charge-transition lines. We find an accuracy of 98.9%. The second stage of the algorithm is trained on 530 000 labeled patches. The test on the labeled evaluation data set to correctly identify 12 outcomes shown in Fig. 4 results in an accuracy of 96%.

Having reached satisfactory performance on the labeled evaluation data, we move on to test our algorithm actively on the device. We preconfigure either DQD1 or DQD2 manually in two different tunnel-gate configurations. We then initialize the system into arbitrary starting charge configurations (n_i, m_i). This is achieved experimentally by randomly setting V_{PG1} and V_{PG2} within a voltage range that is considered safe for the device. We then randomly choose the considered target charge configuration that we wish to achieve, (n_f, m_f), from the set $\{(1, 1)(1, 2)(2, 1)(2, 2)\}$. For both DQDs and two tunnel-gate configurations, we repeat this test 40 times (such that we test each of the four final states ten times per configuration per dot), resulting in 160 complete tuning runs. By a complete tuning run, we mean starting from an arbitrary charge state and running both stages of our algorithm (reference-point finding and target-charge-state finding). Out of these 160 complete tuning runs, 91 runs arrive at the desired final charge

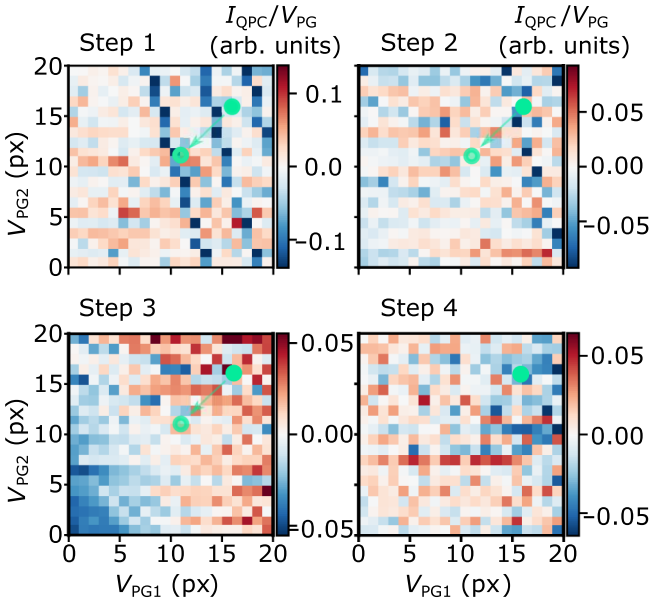

FIG. 5. A successful run of the reference-finding algorithm. The solid green dot indicates the voltage configuration for which the charge occupation of the DQD is assessed. In this way, we hope to minimize the chance of missing a charge-transition line right at the edge of the measurement window. The faint green dot indicates the voltage configuration for which the charge occupation is assessed in the next step. In steps 1–3, the DQD is occupied, whereas in step 4 the DQD is empty and the algorithm terminates. The four plots have individual color bars. The differential current is normalized and plotted for each step individually. We attribute the different amplitudes to noise and the fact that after three steps we recalibrate the QPC voltage to obtain the optimal sensitivity.

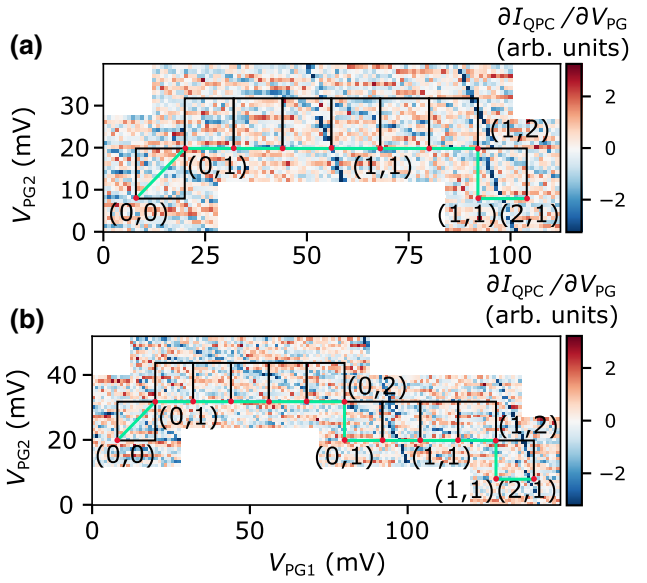

FIG. 6. Two successful runs of the autotuner for which the plunger gate voltages for the desired charge occupation are found. The boxes in black denote the patches that linked together, forming a path shown in red connecting the $(0,0)$ charge occupation with the desired charge occupation.

occupation (n_f, m_f). This results in a success rate of 57%. In the following, we analyze this score in more detail.

Looking at the first-stage algorithm (reference-point finding) only, we find that 144 out of the 160 times, the charge reference point is found successfully. This translates into a success rate of 90% for the first stage of the algorithm. Figure 5 shows a run of the first stage of the algorithm that identifies the charge reference point. To make sure that we do not miss any charge-transition lines at the edge of the patch, the algorithm evaluates whether or not the dot is empty at the position indicated by a green dot. If the DQD is not yet empty, a new patch is measured at the position marked by the arrow. In steps 1–3, there are charge-transition lines within the patch and the algorithm correctly classifies them as occupied. In step 4, no charge-transition lines are present and the patch is correctly classified as empty.

The majority of the errors occur in the second more complex stage of the algorithm. The total success rate of this step is 63% and therefore lower than the individual success rate of the neural network discussed above, due to the repeated application of the neural network in each step. In Fig. 6, we show two paths connecting a reference point to a point in the charge-stability diagram with the target charge state configuration. Both runs are successful and the correct charge regime $(2,1)$ is found. The charge occupation, shown in black, is indicated after each detected charge transition. Both tuning runs are measured on DQD2 (the system that is not used for training).

Furthermore, it is interesting to note that we do not find a difference when comparing the performance of the algorithms of DQD1 and DQD2. As discussed above, the success rate of the complete end-to-end tuning algorithm is 57%. When we evaluate the success rate of the whole tuning run *per dot*, we find end-to-end success rates of 56.3% (57.5%) on DQD1 (DQD2). The difference in the success rates corresponds to one more successful tuning run on DQD2.

We show the confusion matrix of the evaluation of the model in Fig. 7. The entries on the diagonal in the confusion matrix correspond to successful runs, whereas the off-diagonal entries represent unsuccessful runs. In particular, one can use the off-diagonal entries to identify how exactly each of the erroneous outcomes are misclassified. The off-diagonal entries indicate that most errors occur due to the presence of charge transitions that the algorithm does not recognize in the data. The reverse effect, i.e., identifying a transition that is not physically present in the system, almost never occurs. Via analysis of the performance of the algorithm in the different regions of the charge-stability diagram, we conclude that a weak signal-to-noise ratio

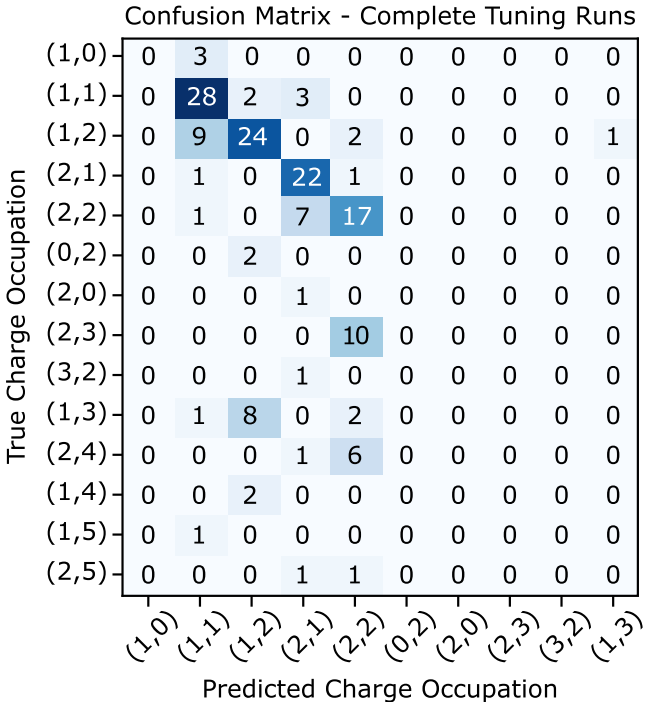

FIG. 7. The confusion matrix indicating the outcomes of the test runs of the autotuner. The x axis shows the charge occupation that is predicted by the neural network. The true charge configuration refers to the actual charge configuration in the dot after the neural network stops tuning. We manually determine this by measuring large-area charge-stability diagrams. Two runs are omitted, since their charge occupation cannot be determined.

in the measurements is the main reason for false classifications (transitions in the regions farthest from the QPC suffer most from this error).

VI. CONCLUSION AND OUTLOOK

In this work, we implement and experimentally test a machine-learning-assisted autotuner that sets the plunger gate voltages to reach any charge configuration of a DQD. We verify that the accuracy of the resulting algorithm persists when testing it on a different DQD than the double quantum dot that was used for training of machine-learning models. In our experiments, we reach a success rate of 57%. The primary error source is identified as a weak signal-to-noise ratio. We believe that this can, in principle, be alleviated by performing more finely grained measurements for the subsets of the charge-stability diagram and, thus, training the convolutional neural networks on higher-resolution images. Another suitable strategy is to implement more targeted sampling of the charge-stability diagram [22]. Currently, we perform homogeneous steps in voltages at all positions of the chosen patch. Measuring more densely around the places on the diagram where the local standard deviation of the data is maximal would lead to better resolution of the transition lines and might increase the accuracy of the algorithm as well. Further improvements might arise from including simulated data in the training set [31].

In general, neural networks are very good at performing fast and efficient classification, but it is extremely difficult to arrive at 100% accuracy on any training set of practical significance. Thus, the accuracy of individual tuning steps will always have a finite success rate when implemented with a neural-network architecture and these finite rates will accumulate if many steps are needed to reach the final state of tuning. These types of autotuners may therefore still require some form of human input. Alternatively, additional algorithms that consider the full problem instance, rather than only individual steps, could serve as a better approach.

ACKNOWLEDGMENTS

We gratefully acknowledge financial support from the Swiss National Science Foundation the National Centre of Competence in Research (NCCR) "Quantum Science & Technology" ("QSIT"). This work has received funding from the European Research Council under Grant Agreement No. 771503. We thank Sebastian D. Huber and Klaus Ensslin for fruitful discussions.

APPENDIX A: MEASUREMENTS AND DATA PROCESSING

All measurements are recorded in a commercially available dilution refrigerator with a base temperature of 12 mK. The electronic temperature is 132 mK $\pm$ 9 mK.

In Fig. 8, we show a charge-stability diagram (a) and three of its random augmentations (b)–(d). The charge-stability diagrams have a resolution of 1 mV and size 241 mV $\times$ 241 mV. Subsequently, approximately 300 fine patches are drawn from each of the augmented charge-stability diagrams (including their original).

The individual augmentation transformations are as follows: (i) rotation of the whole charge-stability diagram; (ii) a rotation of 45° followed by scaling the diagram in the PG1 and/or PG2 direction, followed by a rotation of −45°; (iii) scaling of the PG1 axis; (iv) scaling of the PG2 axis; and (v) flipping of the PG1 and PG2 axes. Flipping the axes simulates measurements for which the plunger gate voltages are swept in reverse order. For all charge stability diagrams, the final augmentation sequence consists of a randomly chosen transformation with random parameters, such as the rotation angle or the scaling factor.

After augmentations of all charge-stability diagrams are performed, the patches are drawn. In order to distribute the patches equally over the charge-stability diagram but still preserve randomness, we draw patches of size 28 $\times$ 28 px,

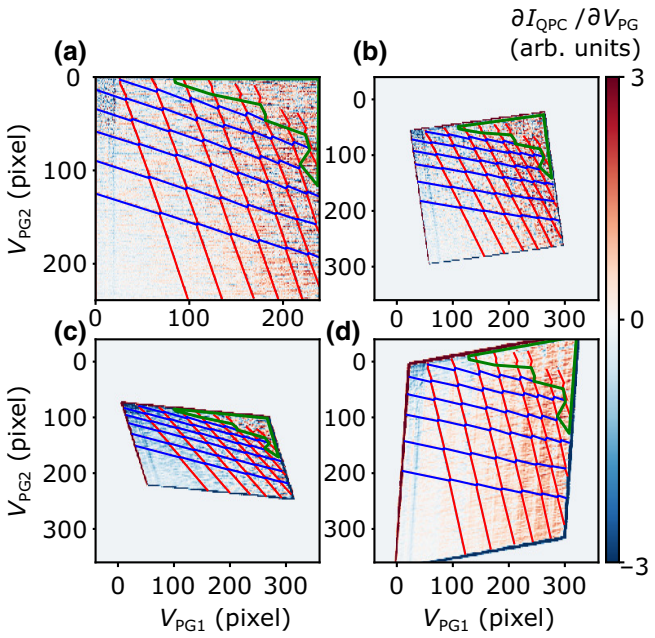

FIG. 8. (a) A charge-stability diagram with indicated charge transitions. Blue lines correspond to charge-transition lines of QD2 and red lines to charge transitions of QD1. In the area enclosed by the green line, no charge-transition lines can be recognized. (b)–(d) Three different augmentations with randomly chosen transformations of (a).

where the grid points are randomly shifted by up to 5 px in each direction.

Each of the patches is processed separately. In the first processing step, the data are rescaled such that the variance equals 1. Second, we construct the charge-stability diagrams by evaluating the derivative of the measured current I_{QPC}, according to the formula

$$\frac{\partial I_{QPC}}{\partial V_{PG}} = \frac{1}{2}\left(\frac{\partial I_{QPC}}{\partial V_{PG1}} + \frac{\partial I_{QPC}}{\partial V_{PG2}}\right). \tag{A1}$$

Subsequently, we remove far-outlying measurement points based on a fixed standard deviation criterion.

APPENDIX B: MACHINE-LEARNING MODELS

Both machine-learning models are constructed in PYTHON using the high-level TensorFlow application programming interface (API) KERAS [39]. All of the hyperparameters that define the architectures are shown in Tables I and II. Tables III–VI show the performance of the two neural networks on the evaluation data set. The scores that we show are as follows: recall, which is calculated as the number of elements that are evaluated correctly by the network divided by the number of true positives; precision, which is the number of elements that are evaluated as positive divided by the total number of samples; and F1, which is the harmonic mean between recall and precision [40]. The accuracy score, which is the fraction of correctly classified data points, is given by the micro average of the above scores.

TABLE I. The relevant parameters used to create the neural network used to determine charge-transition lines in the coarse patches.

Parameter	Value
Dropout 1	0.05
Convolutional layer 1	$(4,4) \times 48$
Activation 1	ReLU
Dropout 2	0.05
Convolutional layer 2	$(3,3) \times 12$
Activation 2	ReLU
Dropout 3	0.4
Dense layer	50
Activation	Sigmoid
Outputs	2
Activation	Softmax
Loss	Categorical crossentropy
Optimizer	Adam
Batch size	128
Learning rate	0.001
Epochs	8
Class weights	True

TABLE II. The relevant parameters used to create the neural network that recognizes individual charge transitions between the corners of the high-resolution patches.

Parameter	Value
Dropout 1	0.2
Convolutional layer 1	$(6,6) \times 72$
Activation 1	ReLU
Dropout 2	0.1
Convolutional layer 2	$(3,3) \times 24$
Activation 2	ReLU
Dropout 3	0.2
Convolutional layer 3	$(2,3) \times 12$
Activation 3	ReLU
Dropout 4	0
Dense layer	50
Activation	Sigmoid
Outputs	3×4
Activation	Softmax
Loss	Categorical crossentropy
Optimizer	Adam
Batch size	128
Learning rate	0.001
Epochs	8
Class weights	False

TABLE III. Classification scores for the neural network used to determine charge-transition lines in the coarse patches. The numbers are obtained from the evaluation data set containing 3152 data points. The micro average corresponds to the overall accuracy, i.e., the fraction of data points that are correctly classified. All accuracy metrics are defined in Appendix B.

Class	Precision	Recall	F1	Support
occupied	0.996	0.986	0.991	2063
empty	0.975	0.993	0.984	1089
Micro average	0.989	0.989	0.989	3152

TABLE IV. Classification scores for the classification of the charge transitions occurring between the lower left and top left corners of the high-resolution patches. The numbers are obtained from the evaluation data set containing 2054 data points. The micro average corresponds to the overall accuracy, i.e., the fraction of data points that are correctly classified. All accuracy metrics are defined in Appendix B.

Class	Precision	Recall	F1	Support
(0, 0)	0.97	0.97	0.98	1202
(0, 1)	0.96	0.95	0.95	643
(1, 0)	0.97	0.93	0.95	157
(1, 1)	0.83	0.83	0.83	52
Micro average	0.96	0.96	0.96	2054

TABLE V. Classification scores for the classification of the charge transitions occurring between the lower left and top right corners of the high-resolution patches. The numbers are obtained from the evaluation data set containing 2054 data points. The micro average corresponds to the overall accuracy, i.e., the fraction of data points that are correctly classified. All accuracy metrics are defined in Appendix B.

Class	Precision	Recall	F1	Support
(0, 0)	0.97	0.98	0.97	632
(0, 1)	0.95	0.96	0.95	517
(1, 0)	0.96	0.96	0.96	500
(1, 1)	0.95	0.93	0.94	405
Micro average	0.96	0.96	0.96	2054

TABLE VI. Classification scores for the classification of the charge transitions occurring between the lower left and lower right corners of the high-resolution patches. The numbers are obtained from the evaluation data set containing 2054 data points. The micro average corresponds to the overall accuracy, i.e., the fraction of data points that are correctly classified. All accuracy metrics are defined in Appendix B.

Class	Precision	Recall	F1	Support
(0, 0)	0.98	0.99	0.98	1193
(0, 1)	0.90	0.87	0.89	151
(1, 0)	0.98	0.97	0.97	643
(1, 1)	0.84	0.78	0.81	63
Micro average	0.97	0.97	0.97	2054

[1] W. I. L. Lawrie, H. G. J Eenink, N. W. Hendrickx, J. M. Boter, L. Petit, S. V. Amitonov, M. Lodari, B. Paquelet Wuetz, C. Volk, and S. Philips, Quantum dot arrays in silicon and germanium, arXiv:1909.06575.

[2] Philip Krantz, Morten Kjaergaard, Fei Yan, Terry P. Orlando, Simon Gustavsson, and William D. Oliver, A quantum engineer's guide to superconducting qubits, Appl. Phys. Rev. **6**, 021318 (2019).

[3] Elsa Prada, Pablo San-Jose, Michiel W. A. de Moor, Attila Geresdi, Eduardo J. H. Lee, Jelena Klinovaja, Daniel Loss, Jesper Nygård, Ramón Aguado, and Leo P. Kouwenhoven, From Andreev to Majorana bound states in hybrid superconductor-semiconductor nanowires, arXiv:1911.04512.

[4] Romana Schirhagl, Kevin Chang, Michael Loretz, and Christian L. Degen, Nitrogen-vacancy centers in diamond: Nanoscale sensors for physics and biology, Annu. Rev. Phys. Chem. **65**, 83 (2014).

[5] C. E. Bradley, J. Randall, M. H. Abobeih, R. C. Berrevoets, M. J. Degen, M. A. Bakker, M. Markham, D. J. Twitchen, and T. H. Taminiau, A 10-qubit solid-state spin register with quantum memory up to one minute, arXiv:1905.02094.

[6] Morten Kjaergaard, Mollie E. Schwartz, Jochen Braumüller, Philip Krantz, Joel I.-Jan Wang, Simon Gustavsson, and William D. Oliver, Superconducting qubits: Current state of play, arXiv:1905.13641.

[7] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, and David A. Buell *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[8] Marius Eich, Riccardo Pisoni, Hiske Overweg, Annika Kurzmann, Yongjin Lee, Peter Rickhaus, Thomas Ihn, Klaus Ensslin, František Herman, and Manfred Sigrist *et al.*, Spin and Valley States in Gate-Defined Bilayer Graphene Quantum Dots, Phys. Rev. X **8**, 031023 (2018).

[9] Fabrizio Nichele, Asbjørn C. C. Drachmann, Alexander M Whiticar, Eoin C. T. O'Farrell, Henri J. Suominen, Antonio Fornieri, Tian Wang, Geoffrey C. Gardner, Candice Thomas, and Anthony T. Hatke *et al.*, Scaling of Majorana Zero-Bias Conductance Peaks, Phys. Rev. Lett. **119**, 136803 (2017).

[10] R. M. Lutchyn, E. P. A. M. Bakkers, L. P. Kouwenhoven, P. Krogstrup, C. M. Marcus, and Y. Oreg, Majorana zero modes in superconductor-semiconductor heterostructures, Nat. Rev. Mater. **3**, 52 (2018).

[11] E. Kawakami, P. Scarlino, D. R. Ward, F. R. Braakman, D. E. Savage, M. G. Lagally, Mark Friesen, S. N. Coppersmith, M. A. Eriksson, and L. M. K. Vandersypen, Electrical control of a long-lived spin qubit in a Si/SiGe quantum dot, Nat. Nanotechnol. **9**, 666 (2014).

[12] K. C. Nowack, F. H. L. Koppens, Yu. V. Nazarov, and L. M. K. Vandersypen, Coherent control of a single electron spin with electric fields, Science **318**, 1430 (2007).

[13] Y. He, S. K. Gorman, D. Keith, L. Kranz, J. G. Keizer, and M. Y. Simmons, A two-qubit gate between phosphorus donor electrons in silicon, Nature **571**, 371 (2019).

[14] D. J. van Woerkom, P. Scarlino, J. H. Ungerer, C. Müller, J. V. Koski, A. J. Landig, C. Reichl, W. Wegscheider, T. Ihn, K. Ensslin, and A. Wallraff, Microwave Photon-Mediated Interactions between Semiconductor Qubits, Phys. Rev. X **8**, 041018 (2018).

[15] P. Scarlino, D. J. van Woerkom, U. C. Mendes, J. V. Koski, A. J. Landig, C. K. Andersen, S. Gasparinetti, C. Reichl, W. Wegscheider, K. Ensslin, T. Ihn, A. Blais, and A. Wallraff, Coherent microwave-photon-mediated coupling between a semiconductor and a superconducting qubit, Nat. Commun. **10**, 3011 (2019).

[16] A. J. Landig, J. V. Koski, P. Scarlino, C. Müller, J. C. Abadillo-Uriel, B. Kratochwil, C. Reichl, W. Wegscheider, S. N. Coppersmith, Mark Friesen, A. Wallraff, T. Ihn, and K. Ensslin, Virtual-photon-mediated spin-qubit-transmon coupling, Nat. Commun. **10**, 5037 (2019).

[17] F. Borjans, X. G. Croot, X. Mi, M. J. Gullans, and J. R. Petta, Long-range microwave mediated interactions between electron spins, arXiv:1905.00776.

[18] Julian Kelly, Peter O'Malley, Matthew Neeley, Hartmut Neven, and John M. Martinis, Physical qubit calibration on a directed acyclic graph, arXiv:1803.03226.

[19] Eliska Greplova, Edward A. Laird, G. Andrew D. Briggs, and Klaus Mølmer, Conditioned spin and charge dynamics of a single-electron quantum dot, Phys. Rev. A **96**, 052104 (2017).

[20] C. J. Van Diepen, Pieter T. Eendebak, Bruno T. Buijtendorp, Uditendu Mukhopadhyay, Takafumi Fujita, Christian Reichl, Werner Wegscheider, and Lieven M. K. Vandersypen, Automated tuning of inter-dot tunnel coupling in double quantum dots, Appl. Phys. Lett. **113**, 033101 (2018).

[21] Steffen J. Glaser, Ugo Boscain, Tommaso Calarco, Christiane P. Koch, Walter Köckenberger, Ronnie Kosloff, Ilya Kuprov, Burkhard Luy, Sophie Schirmer, and Thomas Schulte-Herbrüggen *et al.*, Training Schrödinger's cat: Quantum optimal control, Eur. Phys. J. D **69**, 279 (2015).

[22] D. T. Lennon, H. Moon, L. C. Camenzind, Liuqi Yu, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, E. A. Laird, and N. Ares, Efficiently measuring a quantum device using machine learning, npj Quantum Inf. **5**, 1 (2019).

[23] Eliska Greplova, Christian Kraglund Andersen, and Klaus Mølmer, Quantum parameter estimation with a neural network, arXiv:1711.05238.

[24] Julian D. Teske, Simon Sebastian Humpohl, René Otten, Patrick Bethke, Pascal Cerfontaine, Jonas Dedden, Arne Ludwig, Andreas D. Wieck, and Hendrik Bluhm, A machine learning approach for automated fine-tuning of semiconductor spin qubits, Appl. Phys. Lett. **114**, 133102 (2019).

[25] Easwar Magesan, Jay M. Gambetta, Antonio D. Córcoles, and Jerry M. Chow, Machine Learning for Discriminating Quantum Measurement Trajectories and Improving Readout, Phys. Rev. Lett. **114**, 200501 (2015).

[26] J. Darulová, S. J. Pauka, N. Wiebe, K. W. Chan, M. C. Cassidy, and M. Troyer, Autonomous tuning and charge state detection of gate defined quantum dots, arXiv:1911.10709.

[27] Sandesh S. Kalantre, Justyna P. Zwolak, Stephen Ragole, Xingyao Wu, Neil M. Zimmerman, M. D. Stewart, and Jacob M. Taylor, Machine learning techniques for state recognition and auto-tuning in quantum dots, npj Quantum Inf. **5**, 1 (2019).

[28] H. Moon, D. T. Lennon, J. Kirkpatrick, N. M. van Esbroeck, L. C. Camenzind, Liuqi Yu, F. Vigneau, D. M. Zumbühl, G. A. D. Briggs, and M. A. Osborne *et al.*, Machine learning enables completely automatic tuning of a quantum device faster than human experts, arXiv:2001.02589.

[29] N. M. van Esbroeck, D. T. Lennon, H. Moon, V. Nguyen, F. Vigneau, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. D. Briggs, and D. Sejdinovic *et al.*, Quantum device fine-tuning using unsupervised embedding learning, arXiv:2001.04409.

[30] Eliska Greplova, Carolin Gold, Benedikt Kratochwil, Tim Davatz, Riccardo Pisoni, Annika Kurzmann, Peter Rickhaus, Mark H. Fischer, Thomas Ihn, and Sebastian Huber, Fully automated identification of 2D material samples, arXiv:1911.00066.

[31] Justyna P. Zwolak, Thomas McJunkin, Sandesh S. Kalantre, J. P. Dodson, E. R. MacQuarrie, D. E. Savage, M. G. Lagally, S. N. Coppersmith, Mark A. Eriksson, and Jacob M. Taylor, Auto-tuning of double dot devices *in situ* with machine learning, arXiv:1909.08030.

[32] Gang Cao, Hai-Ou Li, Guo-Dong Yu, Bao-Chuan Wang, Bao-Bao Chen, Xiang-Xiang Song, Ming Xiao, Guang-Can Guo, Hong-Wen Jiang, Xuedong Hu, and Guo-Ping Guo, Tunable Hybrid Qubit in a GaAs Double Quantum Dot, Phys. Rev. Lett. **116**, 086801 (2016).

[33] Zhan Shi, C. B. Simmons, J. R. Prance, John King Gamble, Teck Seng Koh, Yun-Pil Shim, Xuedong Hu, D. E. Savage, M. G. Lagally, M. A. Eriksson, Mark Friesen, and S. N. Coppersmith, Fast Hybrid Silicon Double-Quantum-Dot Qubit, Phys. Rev. Lett. **108**, 140503 (2012).

[34] J. Medford, J. Beil, J. M. Taylor, E. I. Rashba, H. Lu, A. C. Gossard, and C. M. Marcus, Quantum-Dot-Based Resonant Exchange Qubit, Phys. Rev. Lett. **111**, 050501 (2013).

[35] A. J. Landig, J. V. Koski, P. Scarlino, U. C. Mendes, A. Blais, C. Reichl, W. Wegscheider, A. Wallraff, K. Ensslin, and T. Ihn, Coherent spin-photon coupling using a resonant exchange qubit, Nature **560**, 179 (2018).

[36] Maximilian Russ, J. R. Petta, and Guido Burkard, Quadrupolar Exchange-Only Spin Qubit, Phys. Rev. Lett. **121**, 177701 (2018).

[37] 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).

[38] Thomas Ihn, *Semiconductor Nanostructures* (Oxford University Press, New York, USA, 2009).

[39] François Chollet *et al.*, https://github.com/fchollet/keras.

[40] Aurélien Géron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems* (O'Reilly Media, Inc., Sebastopol, California, USA, 2017).