Evaluation of synthetic and experimental training data in supervised machine learning applied to charge-state detection of quantum dots

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 045023 https://doi.org/10.1088/2632-2153/ac104c

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
19 March 2021

REVISED
17 June 2021

ACCEPTED FOR PUBLICATION
30 June 2021

PUBLISHED
13 September 2021

PAPER

Evaluation of synthetic and experimental training data in supervised machine learning applied to charge-state detection of quantum dots

J Darulová[1], M Troyer[2] and M C Cassidy[3,*]

[1] Theoretische Physik, ETH Zurich, 8093 Zurich, Switzerland
[2] Microsoft Quantum, Redmond, WA 98052, United States of America
[3] Microsoft Quantum, The University of Sydney, Sydney, NSW 2006, Australia
[*] Author to whom any correspondence should be addressed.

E-mail: maja.cassidy@microsoft.com

Keywords: quantum dots, machine learning, automated tuning, semiconductor qubits

Abstract

Automated tuning of gate-defined quantum dots is a requirement for large-scale semiconductor-based qubit initialisation. An essential step of these tuning procedures is charge-state detection based on charge stability diagrams. Using supervised machine learning to perform this task requires a large dataset for models to train on. In order to avoid hand labelling experimental data, synthetic data has been explored as an alternative. While providing a significant increase in the size of the training dataset compared to using experimental data, using synthetic data means that classifiers are trained on data sourced from a different distribution than the experimental data that is part of the tuning process. Here we evaluate the prediction accuracy of a range of machine learning models trained on simulated and experimental data, and their ability to generalise to experimental charge stability diagrams in two-dimensional electron gas and nanowire devices. We find that classifiers perform best on either purely experimental or a combination of synthetic and experimental training data, and that adding common experimental noise signatures to the synthetic data does not dramatically improve the classification accuracy. These results suggest that experimental training data as well as realistic quantum dot simulations and noise models are essential in charge-state detection using supervised machine learning.

A commercially viable quantum computer will require the manufacture of a large number of qubit devices, as well as autonomous procedures for qubit initialisation and tuning. For semiconductor-based qubits such as charge [1–3], spin [4–7] and topological qubits [8–10], this implies the initialisation of quantum dots of a known charge state. These quantum dots are formed by choosing appropriate voltages applied to electrostatic gates, depleting the electron gas underneath and thus isolating electrons from surrounding charge carriers. Although the manual formation of these dots is now routine in a wide range of systems, challenges still exist in automating this procedure. Material defects and fabrication variances result in non-uniform device performance, resulting in unique operating gate voltages between nominally identical devices and even their tune-ups.

With machine learning and artificial intelligence succeeding in an increasing range of sophisticated tasks, it is worth investigating their capabilities in automating the task of defining quantum dots. Several efforts in automating double quantum dot tune-up have been made using either supervised deep learning [11–14], unsupervised statistical methods [15, 16] or deterministic algorithms [17–19]. For these approaches to be useful for large-scale tune-up, the tuning outcome needs to be determined reliably despite noise and without human intervention. This is achieved by verifying the charge state of the qubit device based on a measured charge stability diagram. Using supervised learning to perform this task requires a significant amount of labelled data, where each dataset carries an attributed label indicating its charge state. The process of

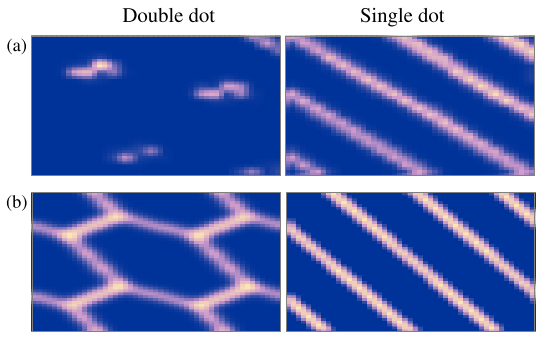

Figure 1. Examples of noiseless synthetic charge stability diagrams. Left and right column show double and single dot regimes respectively. (a) Typical diagrams of the QFlow-lite dataset, simulated using the Thomas–Fermi approximation. (b) Typical diagrams generated using a capacitance model.

measuring and labelling experimental data is slow, making synthetic data a way to increase the efficiency of this training process. The success of synthetic training data has been quantified mainly by classifying further synthetic data or curated experimental data [11–13]. However, classification results of realistic data during tuning suggests this performance to be lower [12].

Here we evaluate the ability of supervised machine learning models trained on synthetic data to determine the charge state of experimental charge stability diagrams, and compare to ones trained on data from real devices. Two convolutional neural network architectures and six parametric binary classifiers are trained to distinguish single versus double quantum dots when trained on purely synthetic data, a combination of synthetic and experimental data or experimental data only. We also investigate how adding noise to noiseless synthetic data affects classification accuracy, a technique showing promising success in the identification of impurities in scanning tunnelling microscope images [20].

Quantum dots are formed by applying voltages to electrostatic gates fabricated on top of a semiconductor structure, which creates potential wells isolating charges in regions with length scales on the order of the Fermi wavelength. One or two regions of charges can be formed, resulting in a single or double quantum dot. To determine the regime, i.e. single versus double, two gate voltages are stepped over while the current through the device is measured, resulting in a so-called charge stability diagram. A single dot features sharp diagonal lines, while a double dot shows either triple points with no charge transition lines between them, or a honey comb pattern with transition lines connecting bright spots corresponding to triple points. It has been shown that voltage combinations not resulting in the formation of any dots can be excluded through simple gate characterisation steps [17]. Machine learning techniques are therefore only required to distinguish between single and double dots of different qualities to complete the tuning process.

In this work, we assess the accuracy of convolutional neural networks and binary classifiers trained on synthetic data, experimental data or a combination of both. Convolutional neural networks trained on synthetic data benchmarked on either synthetic or curated experimental data have previously shown high classification accuracy [11–13]. We use the same neural network architecture with two convolutional layers [11, 12], summarised in table A.1. The binary classifiers we compare are logistic regression, multi-layer perceptron, decision tree, random forest, K nearest neighbours and support vector machine, and were selected based on the accuracy comparison in [17].

All classifiers are trained and tested on the same data combinations. Binary classifiers are trained and tested on both transport measurements and their frequency spectrum, extracted using a Fourier transform [17]. Neural networks are trained and tested on transport measurements only. While binary classifiers are trained on original experimental data, the neural network is trained on an augmented experimental dataset, created using standard augmentation techniques. The approaches presented also apply to other types of measurements such as charge sensing and radio-frequency reflectometry.

Our synthetic dataset of simulated single and double dot charge stability diagrams is based on a capacitance model [21] and the Qflow-lite dataset [13], i.e. the Thomas–Fermi approximation. Examples of both dot regimes generated by these models are shown in figure 1. Details about the data generation and post-processing steps can be found in appendix A.

We implement five noise models typically encountered in experiments which are added to the noiseless synthetic data. We refer to this dataset as a noisy synthetic dataset. The added noise types are white noise, random telegraph noise, $1/f$ noise, charge fluctuations on gates, low-frequency current modulations and

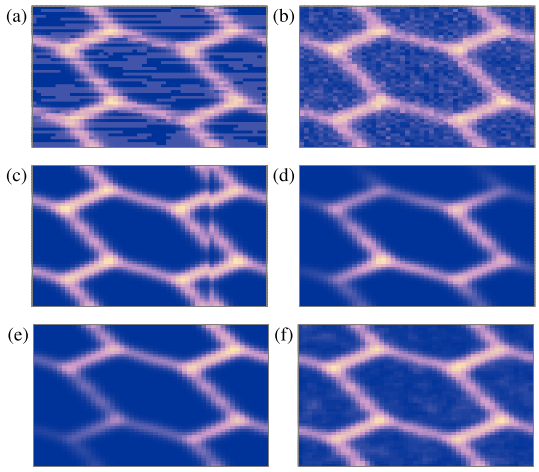

Figure 2. Examples of noisy synthetic data. Random current maps were generated for each noise type independently, rescaled with varying amplitudes and then added to noiseless synthetic data. (a) Random telegraph noise. (b) White noise. (c) Charge jumps. (d) Random current modulation. (e) Pinchoff current modulation. (f) $1/f$ noise.

pinch-off current modulation. An example of each is shown in figure 2. White noise typically arises due to thermal fluctuations, while $1/f$ noise and charge fluctuations on gates are two types of random fluctuations due to defects in the semiconductor. Random telegraph noise on the other hand is a low-frequency modulation of current caused by the spontaneous capture and emission of charge carriers. Low-frequency current modulations and pinch-off current modulation are consequences of the electron gas being depleted at decreasing gate voltages. Details of their implementation as well as additional examples can be found in appendix B.

Our experimental data originates from quantum dots formed in InSb nanowires [22] as well as GaAs two-dimensional electron gases [17, 23]. Each charge stability diagram is hand-labelled by two labels indicating the charge state, i.e single or double, and quality, i.e. sufficient or insufficient for subsequent tuning steps. As an example, double-dot diagrams are labelled as sufficient if they feature clear triple points suitable for qubit parameter fine tuning procedures discussed in [24–26], and insufficient otherwise.

Sufficient experimental data is further divided into ideal and good measurements, and we assess the classification accuracy on the subsets 'ideal', 'good', and 'all' measurements. Ideal measurement outcomes show features similar those found in synthetic data. Good measurements diagrams show some types of noise, but are suitable for further fine tuning. Using standard augmentation techniques, we augment our original experimental dataset of 221 ideal, 1681 good and 4613 bad charge stability diagrams to 10 000 ideal, 13 000 good and 25 000 bad diagrams. Examples of non-augmented measurements are shown in figure 3, showing common noises. Specifically, figure 3(e) shows a gradual current drop towards negative gate voltages, and the broadening of transitions and charge jumps, while figure 3(f) shows white noise, $1/f$ noise and random current modulation. These noise types have been added with varying, randomly sampled strengths in order to simulate the variation seen in experimental data.

The data described above is used to assess the ability of convolutional neural networks and parametric binary classifiers to generalise from synthetic data to a variety of experimental data. Each classifier is trained on the following dataset combinations: noiseless synthetic data, noisy synthetic data, good experimental data, all (i.e. ideal, good and bad) experimental data, noiseless synthetic data and good experimental data, noiseless synthetic data and all experimental data. Classification accuracies are evaluated on ideal, good and all experimental data. Where applicable, these datasets are split into 80% train and 20% test set and none of the data used in training is used in testing. We perform ten random train and test splits and report the average accuracy and standard deviation. These datasets are balanced, meaning they contain the same number of single and double quantum dots. This produces different sizes of train and test data for each dataset combination.

The results illustrated in figure 4 and detailed in table A.2 show that training the neural network on only synthetic data allows to predict ideal experimental data with an average accuracy of 76.54%. Broadening the scope to predict good and all experimental data sees the accuracies decrease to 64.61% and 60.28% respectively. The accuracies when predicting ideal data improve to 79.49% when good experimental data is

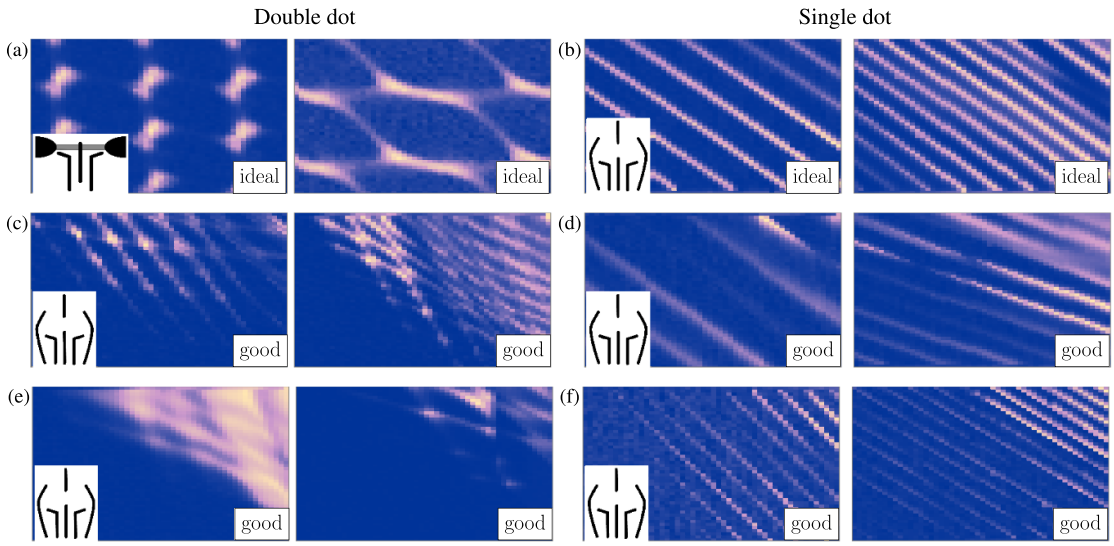

Figure 3. Examples of ideal and good experimental data, as well as noise types typically seen in measurements. Insets indicate the device type, based on either a nanowire or two-dimensional electron gas, the quantum dots were formed in. (a) Ideal double dot charge stability diagram formed in a InSb nanowire. Both diagrams show similar features as our synthetic data, i.e. isolated triple points or a honey comb structure of weak and strong coupling regimes. (b) Ideal single-dot charge stability diagrams formed in a GaAs two-dimensional heterostructure featuring sharp and regular charge transitions. (c) Good double-dot charge stability diagrams showing easily identifiable triple points suitable for further fine tuning. (d) Good single dot charge stability diagrams showing broadened and weak transition lines. (e) Good double dot regimes with pinch-off current modulation and broadened triple points (left image) and charge jumps (right image). (f) Good single-dot regimes illustrating typical noise types, including white noise, $1/f$ noise and random current modulation (left image) as well as pinch-off current modulation (right image).

added to the training set, and are highest when synthetic and experimental data is combined to a single dataset, reaching 89.50%.

Overall, adding synthetic data to an experimental training set improves accuracy by up to 10%. Confusion matrices for each classification, detailing which subclasses tend to be misclassified and listed in table A.3, show that single dots tend to be misclassified as double dots more often than double dots as single dots.

Adding noise to the synthetic training data results in lower accuracies than noiseless synthetic data. A detailed study of the effect of individual noise models can be found in table A.5, where only one noise type was added at a time with various amplitudes. All accuracies decrease when noise is added.

We further investigate potential benefits of transfer learning, during which the network is pre-trained on a synthetic dataset and then re-trained using an experimental dataset while keeping weights of all but the last layer fixed. This technique could potentially reduce training time or increase accuracies when not enough training data is available [27]. Results of transfer learning using either synthetic or noisy synthetic data for pre-training and good or all experimental data for re-training are illustrated in figure 4 and detailed in table A.4. We see little improvement when predicting good and all experimental data, but networks predicting ideal experimental data benefit from transfer learning by up to 5% compared to the pre-trained network. Overall, these accuracies are significantly lower than when both datasets are used together in a single training step.

Classification accuracies of binary classifiers are summarised in figure 5 and detailed in table A.6. Here, classification accuracies are highest when only experimental data is used for training. Exceptions are the decision tree classifier predicting ideal experimental data, and k-nearest neighbour and multi-layer perceptron predicting all experimental data, which benefit from added synthetic training data. Unlike the neural network, training with ideal experimental data does not show higher accuracies than good or all experimental data combined. The multi-layer perceptron performs best, followed by K-nearest neighbour, logistic regression and support vector machines. Adding noise to synthetic training data increases accuracies for the multi-layer perceptron, k-nearest neighbour and logistic regression, while it decreases accuracies for the decision tree classifier and support vector machine. These classifiers also show high training accuracies, suggesting that overfitting occurs. However, when trained with the augmented experimental data, we saw a significant overall decrease in accuracies.

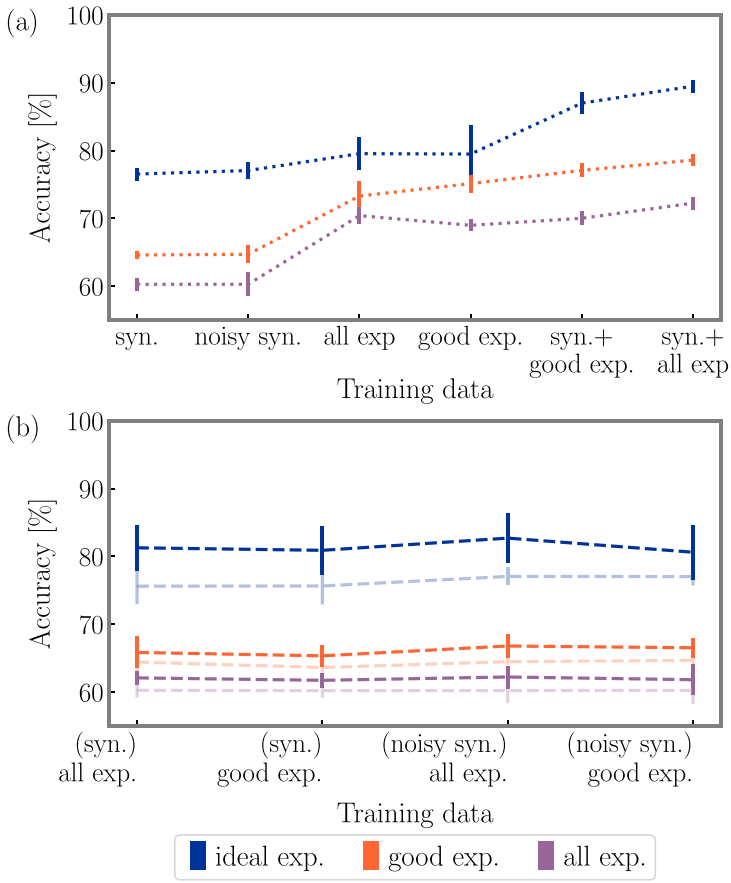

Figure 4. Neural network classification results showing average accuracy and standard deviation over ten train and test splits. Augmented experimental (exp.) data is used. Dashed lines are a guide for the eye and connect average values, vertical bars indicate standard deviations. Classification is performed on ideal, good and all experimental data. (a) Classification accuracies when trained on either synthetic (syn.), noisy synthetic (noisy syn.) or a combination of synthetic and experimental data. (b) Classification accuracies achieved using transfer learning. Pre-training data is indicated in parenthesis. Pre-trained and post-transfer accuracies are plotted in light and dark coloured lines respectively.

To summarise, we find the highest prediction accuracies are achieved by training classifiers on either experimental or a combination of synthetic and experimental training data. Adding only a small experimental dataset to a large synthetic dataset allows the convolutional neural network to learn the specific type of noise present in real measurements and hence improve classification accuracy. Adding more variety to training data from either improved device models or more experimental data is necessary to achieve higher success rates. Even small improvements will go a long way as inaccurate classification within a tuning algorithm multiply, resulting in a negative cascading effect which significantly reduces the overall tuning performance [14]. In particular neural networks with additional convolutional layers could reach higher accuracy and learn a larger variety of charge stability diagrams originating from different materials and device architectures. But a deeper architecture is more prone to overfitting and requires an even larger training dataset. Segmenting experimental data into regions with fewer regime variations may also increase accuracies.

The noise added to synthetic data does not improve classification accuracies, showing that it does not match the noise found in experimental data. More realistic noise models and quantum dot simulations taking into account impurities and fabrication defects are expected to improve the accuracy of classifiers trained on synthetic data. Realistic semiconductor quantum dot simulations are complex and noise encountered in today's state-of the-art devices, which have been used in this work, are not well understood. Future devices with less fabrication variances and impurities may reduce noise and facilitate charge-state detection based on supervised machine learning using synthetic data. But until these devices are reliable and simulations sophisticated enough to reproduce their behaviour, investing time into labelling experimental data is required.

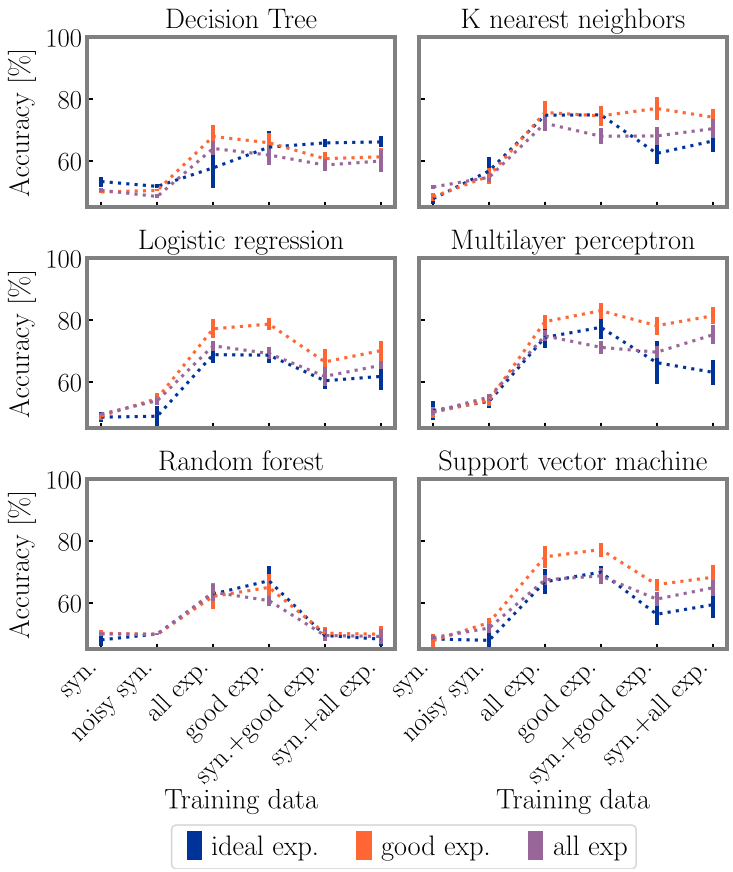

Figure 5. Binary classifier accuracies showing average accuracy and standard deviation over ten train and test splits. Original, i.e. non-augmented, experimental data is used. Dashed lines are a guide for the eye and connect average values, vertical bars indicate standard deviations. Synthetic data or combinations of synthetic and experimental data was used for training and ideal, good or all, i.e. ideal, good and poor, experimental data for testing.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgment

We thank Rachpon Kalra and John M Hornibrook for helpful discussions and critical feedback.

Appendix A. Noiseless synthetic data

Our synthetic dataset of simulated single and double dot charge stability diagrams is based on data generated by a capacitance model [21] and the Qflow-lite dataset [13], which uses the Thomas–Fermi approximation. Examples of both dot regimes generated by these models are shown in figure 1. The capacitance model replicates a device made of six gates coupled to two dots, similar to device architectures used to define charge and spin qubits [23, 28–33]. A set of 2000 diagrams was generated by randomly sampling capacitances from a Gaussian distribution centred around one of several capacitance combinations generating diagrams encountered in experiments.

We use segments of the original Qflow-lite dataset made available online, divided into 15 subregions of 30×30 pixels per original charge stability diagram. We use python's scikit-image [34] resize method to resize these segments to 50×50 pixels, the size of our data. We first normalise each diagram individually to a range between 0 and 1 and then multiply them by an overall factor of 0.3 to ensure charge transitions are of a similar strength as in our experimental and synthetic data. The original Qflow-lite labels, which are vectors of four components indicating the probability of the diagram showing a fully open, fully closed, single-dot and double-dot regime, were transformed into binary labels differentiating only between single and double

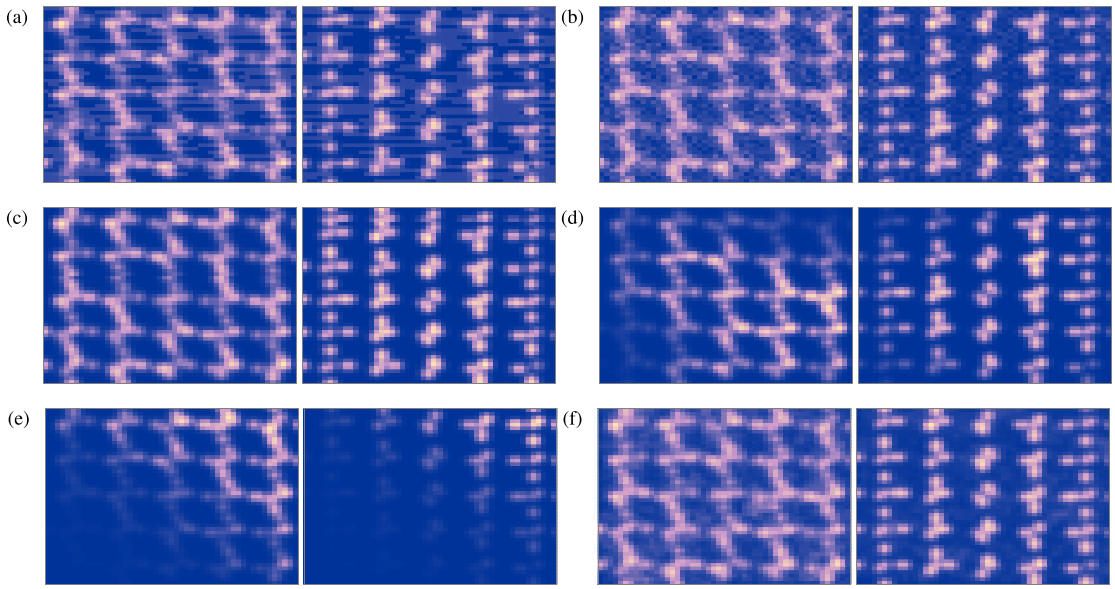

Figure A.1. More examples of noise models added to synthetic data. Random current maps were generated for each noise type independently, rescaled with varying amplitudes and then added to the noiseless synthetic dataset. (a) Random telegraph noise. (b) White noise. (c) Charge jumps. (d) Random current modulation. (e) Pinchoff current modulation. (f) $1/f$ noise.

Table A.1. The neural network architecture used, with layer names in the left column and their respective parameters on the right. It was implemented in TensorFlow 2, with two convolutional layers (Conv2D) and four dense layers with a dropout rate of 0.5. The dense layers consists of 1024, 512, 128 and 2 neutrons (units) respectively. The RELU activation function is used in all but the last layer, which uses the softmax activation function. We use the Adam optimizer and a learning rate of 0.001.

Layer	Details	
Conv2D	Filter	32
	Kernel size	(3, 3)
	Activation	RELU
Conv2D	Filter	64
	Kernel size	(3, 3)
	Activation	RELU
MaxPooling2D	Pool size	(2, 2)
	Strides	2
Dense	Units	1024
	Activation	RELU
Dropout	Rate	0.5
Dense	Units	512
	Activation	RELU
Dropout	Rate	0.5
Dense	Units	128
	Activation	RELU
Dropout	Rate	0.5
Dense	Units	2
	Activation	Softmax
	Optimizer	Adam
	Learning rate	0.001

dot. For this, only diagrams with a fidelity of being in either single- or double-dot regime of more than 60% are retained and relabelled as either a single or double dot.

Appendix B. Noisy synthetic data

We implement five noise models typically encountered in experiments that are added to the noiseless synthetic data, which are referred to as noisy synthetic data sets. These noise types are white noise, random telegraph noise, $1/f$ noise, charge fluctuations on gates, low-frequency current modulations and pinch-off current modulation. White noise typically arises due to thermal fluctuations, while $1/f$ noise and charge fluctuations on gates are two types of random fluctuations due to defects in the semiconductor. Random

Table A.2. Accuracies of the convolutional neural network summarised in table A.1, using augmented experimental data. Average accuracies and standard deviation are taken over ten train and test splits, which were selected randomly with equal numbers of single and double diagrams.

	Average accuracy (std)				
Training data	Training	Synthetic	Clean experimental	Good experimental	All experimental
Synthetic	98.42 (0.55)	93.82 (0.26)	76.54 (0.89)	64.61 (0.52)	60.28 (0.93)
Good experimental	95.39 (1.07)	– (–)	79.49 (4.22)	75.13 (1.27)	68.98 (0.86)
All experimental	91.99 (2.73)	– (–)	79.55 (2.40)	73.30 (2.16)	70.41 (1.22)
Synthetic and good experimental	97.44 (0.57)	– (–)	87.00 (1.59)	77.08 (0.96)	70.02 (1.01)
Synthetic and all experimental	95.93 (0.39)	– (–)	89.50 (0.91)	78.60 (0.86)	72.25 (0.89)
Synthetic with noise	97.19 (1.24)	– (–)	77.06 (1.18)	64.70 (1.26)	60.29 (1.74)

Table A.3. Confusion matrices of convolutional neural network classification results, using augmented experimental data. Diagonal elements indicate correct classification of single (SD) and double (DD) dots, while off diagonals correspond to false single dot (FSD) and false double dot (FDD): $\begin{bmatrix} \text{SD} & \text{FDD} \\ \text{FSD} & \text{DD} \end{bmatrix}$.

	Average accuracy				
Training data	Training	Synthetic	Clean experimental	Good experimental	All experimental
Synthetic	98.42 (0.55)	$\begin{bmatrix} 4765 & 363 \\ 272 & 4830 \end{bmatrix}$	$\begin{bmatrix} 6316 & 2863 \\ 1444 & 7735 \end{bmatrix}$	$\begin{bmatrix} 3993 & 2795 \\ 2009 & 4779 \end{bmatrix}$	$\begin{bmatrix} 8456 & 5121 \\ 5663 & 7914 \end{bmatrix}$
Good experimental	95.39 (1.07)	– (–)	$\begin{bmatrix} 7040 & 2139 \\ 1621 & 7558 \end{bmatrix}$	$\begin{bmatrix} 1044 & 316 \\ 359 & 996 \end{bmatrix}$	$\begin{bmatrix} 2159 & 550 \\ 1134 & 1587 \end{bmatrix}$
All experimental	91.99 (2.73)	– (–)	$\begin{bmatrix} 6744 & 2435 \\ 1318 & 7861 \end{bmatrix}$	$\begin{bmatrix} 1011 & 345 \\ 379 & 979 \end{bmatrix}$	$\begin{bmatrix} 2070 & 665 \\ 942 & 1754 \end{bmatrix}$
Synthetic and good experimental	97.44 (0.57)	– (–)	$\begin{bmatrix} 7878 & 1301 \\ 1085 & 8094 \end{bmatrix}$	$\begin{bmatrix} 1085 & 270 \\ 351 & 1009 \end{bmatrix}$	$\begin{bmatrix} 2184 & 533 \\ 1095 & 1619 \end{bmatrix}$
Synthetic and all experimental	95.93 (0.39)	– (–)	$\begin{bmatrix} 8277 & 902 \\ 1025 & 8154 \end{bmatrix}$	$\begin{bmatrix} 1106 & 262 \\ 320 & 1026 \end{bmatrix}$	$\begin{bmatrix} 2201 & 509 \\ 999 & 1721 \end{bmatrix}$
Synthetic with noise	97.19 (1.24)	– (–)	$\begin{bmatrix} 6477 & 2702 \\ 1508 & 7671 \end{bmatrix}$	$\begin{bmatrix} 4178 & 2611 \\ 2181 & 4607 \end{bmatrix}$	$\begin{bmatrix} 9064 & 4513 \\ 6269 & 7308 \end{bmatrix}$

Table A.4. Transfer learning results of convolutional neural network summarised in table A.1, using augmented experimental data. Average accuracies and standard deviation are taken over ten train and test splits, which were selected randomly with equal numbers of single and double diagrams.

	Average accuracy (std)						
Training and transfer data	Transfer training	Clean experimental		Good experimental		All experimental	
		Pre-train	Transfer	Pre-train	Transfer	Pre-train	Transfer
Synthetic and good experimental	64.68 (2.07)	75.65 (2.72)	80.91 (3.57)	63.63 (1.83)	65.35 (1.57)	60.22 (1.05)	61.74 (1.07)
Synthetic and all experimental	61.16 (1.35)	75.62 (2.66)	81.28 (3.32)	64.43 (2.35)	65.85 (2.32)	60.26 (1.05)	62.10 (0.97)
Noisy synthetic and good experimental	65.36 (1.80)	77.04 (1.32)	80.63 (4.00)	64.69 (1.31)	66.54 (1.43)	60.26 (1.99)	61.83 (2.23)
Noisy synthetic and all experimental	61.51 (1.92)	77.06 (1.28)	82.71 (3.64)	64.49 (1.35)	66.80 (1.75)	60.23 (1.82)	62.23 (1.69)

Table A.5. Noise model study showing classification accuracies of the convolutional neural network summarised in table A.1, using augmented experimental data. Each noise type is added individually and at varying amplitudes. Average accuracies and standard deviation are taken over ten train and test splits, which were selected randomly with equal numbers of single and double diagrams.

	Average accuracy (std)			
Max noise amplitude	Training	Clean experimental	Good experimental	All experimental
No noise				
0	98.42 (0.55)	76.54 (0.89)	64.61 (0.52)	60.28 (0.93)
1/f				
0.05	96.64 (2.56)	73.21 (3.11)	62.26 (2.18)	58.18 (1.57)
0.1	95.37 (2.90)	71.90 (4.42)	61.88 (2.33)	59.34 (1.04)
0.2	93.55 (2.32)	71.17 (5.39)	61.01 (2.51)	59.41 (1.16)
0.3	96.07 (2.11)	74.68 (1.45)	63.10 (0.62)	59.14 (1.38)
White				
0.05	96.52 (2.82)	73.01 (4.13)	61.89 (1.89)	58.29 (1.30)
0.1	95.46 (2.78)	72.72 (3.16)	62.51 (1.36)	58.96 (1.08)
0.2	96.01 (2.28)	73.79 (2.89)	62.59 (1.50)	58.42 (1.77)
0.3	96.88 (1.75)	73.87 (1.32)	62.36 (0.97)	57.73 (1.36)
Random current modulation				
0.05	96.20 (1.78)	73.99 (1.85)	63.05 (0.81)	59.37 (1.01)
0.1	94.39 (5.62)	71.55 (7.04)	61.62 (3.31)	57.98 (1.45)
0.2	95.95 (2.90)	72.35 (4.45)	61.74 (2.29)	58.04 (0.98)
0.3	94.47 (2.33)	71.29 (4.40)	60.94 (2.16)	57.94 (1.48)
Pinchoff modulation				
0.05	94.31 (2.60)	71.18 (4.47)	61.50 (1.87)	58.66 (1.40)
0.1	95.55 (2.19)	73.71 (2.50)	62.15 (1.76)	58.61 (1.62)
0.2	95.98 (2.18)	72.93 (3.43)	62.66 (2.38)	58.97 (1.70)
0.3	97.78 (1.16)	75.39 (0.79)	63.35 (0.62)	58.68 (1.23)
Random telegraph				
0.05	96.38 (2.69)	73.68 (3.16)	62.17 (2.02)	58.63 (1.71)
0.1	95.96 (1.91)	74.63 (2.13)	62.90 (1.39)	59.44 (1.30)
0.2	94.76 (2.58)	71.10 (3.89)	60.94 (2.28)	58.01 (1.59)
0.3	97.81 (1.29)	75.71 (1.29)	63.55 (0.90)	58.51 (1.33)
Charge jumps				
0.05	95.21 (2.72)	73.01 (2.58)	62.26 (1.36)	59.15 (1.55)
0.1	94.95 (3.08)	71.17 (4.53)	61.44 (2.43)	58.31 (1.78)
0.2	96.78 (1.61)	74.79 (1.18)	63.49 (1.00)	59.05 (1.30)
0.3	96.34 (1.80)	74.84 (1.41)	63.49 (1.30)	59.21 (1.24)

telegraph noise on the other hand is a low-frequency modulation of the current caused by the spontaneous capture and emission of charge carriers. Low-frequency current modulations and pinch-off current modulation are consequences of the electron gas being depleted for decreasing gate voltages. Additional examples of each noise type are shown in figure A.1. Noise is generated as follows:

The $1/f$ noise is generated in frequency domain by creating 2D frequencies mesh and taking the inverse of their norm to calculate the magnitude of spectral coefficients:

$$C_{k,l} = \begin{cases} \frac{1}{\sqrt{f_k^2+f_l^2}}, & \text{if } f_k^2+f_l^2 > 0. \\ 0, & \text{otherwise.} \end{cases} \tag{B1}$$

We set the phases of these coefficients to random values:

$$C_{k,l} = C_{k,l} e^{i\phi_{k,l}}, \tag{B2}$$

where $\phi_{k,l}$ is chosen randomly from a uniform distribution over $[0, 2\pi)$. The inverse Fourier transform is then added to the images. White noise is generated as a 2D map of random, normally distributed coefficients with zero mean and a variance of 1. Pinch-off current modulation is achieved by convoluting the image with

$$W_{i,j} = \tanh(\alpha * x_{i,j} + \beta), \tag{B3}$$

where α and β are drawn from a uniform distribution between over $[0, 10)$ and $[-5, 5)$ respectively, and $x_{i,j}$ is a pixel coordinate matrix. Random current modulation is realized by convolving an image with a 2D map of Gaussian blobs of random mean and standard deviation, drawn uniformly between $[-1, 1)$ and $[0.3, 0.8)$ respectively. Random telegraph noise is simulated by adding charge jumps following a Poisson distribution

Table A.6. Parametric binary classification results using original experimental data. Average accuracies and standard deviation are taken over ten train and test splits, which were selected randomly with equal numbers of single and double diagrams.

	Average accuracy (std)				
	Logistic regression				
Training data	Training	Synthetic	Clean experimental	Good experimental	All experimental
Synthetic	99.61 (0.10)	87.43 (0.66)	46.57 (3.69)	47.53 (2.65)	49.07 (1.13)
Good experimental	99.94 (0.06)	– (–)	68.48 (2.85)	78.86 (1.81)	69.39 (2.48)
All experimental	99.63 (0.16)	– (–)	69.04 (3.50)	75.83 (2.84)	71.52 (2.28)
Synthetic and good experimental	98.12 (0.12)	– (–)	59.57 (3.10)	67.70 (3.54)	62.58 (2.71)
Synthetic and all experimental	97.27 (0.11)	– (–)	61.79 (4.77)	68.71 (3.33)	65.23 (2.11)
Synthetic with noise	94.82 (0.19)	– (–)	48.30 (3.39)	54.21 (2.02)	53.53 (1.50)
	MLP classifier				
	Training	Synthetic	Clean experimental	Good experimental	All experimental
Synthetic	99.57 (0.44)	94.09 (0.57)	50.33 (2.75)	49.53 (1.47)	49.98 (1.14)
Good experimental	96.92 (1.94)	– (–)	77.47 (4.28)	81.13 (3.43)	70.67 (3.09)
All experimental	95.41 (2.26)	– (–)	75.44 (6.67)	80.25 (3.01)	75.51 (2.51)
Synthetic and good experimental	99.46 (0.81)	– (–)	65.43 (6.15)	80.12 (3.18)	71.09 (2.39)
Synthetic and all experimental	99.02 (0.46)	– (–)	65.90 (5.70)	81.15 (2.64)	75.49 (2.86)
Synthetic with noise	100.00 (0.00)	– (–)	54.83 (2.95)	54.61 (1.62)	55.03 (0.92)
	Decision tree classifier				
	Training	Synthetic	Clean experimental	Good Experimental	All experimental
Synthetic	92.52 (0.33)	90.47 (0.82)	54.70 (4.79)	50.77 (2.84)	51.18 (1.93)
Good experimental	83.39 (2.75)	– (–)	60.08 (7.25)	67.10 (3.43)	62.91 (2.97)
All experimental	77.92 (2.01)	– (–)	57.54 (6.74)	68.91 (3.86)	64.55 (2.65)
Synthetic and good experimental	90.09 (0.41)	– (–)	66.12 (1.36)	61.61 (4.20)	59.79 (2.52)
Synthetic and all experimental	88.12 (0.44)	– (–)	65.66 (2.68)	62.66 (3.09)	60.31 (2.84)
Synthetic with noise	90.74 (0.35)	– (–)	51.23 (0.84)	50.43 (0.33)	48.54 (0.62)
	Random forest classifier				
	Training	Synthetic	Clean experimental	Good experimental	All experimental
Synthetic	86.54 (0.76)	85.53 (0.70)	48.73 (2.67)	50.43 (0.98)	50.13 (0.64)
Good experimental	79.87 (1.99)	– (–)	65.81 (4.56)	65.48 (3.84)	61.49 (2.00)
All experimental	72.61 (1.56)	– (–)	63.47 (3.69)	61.34 (4.06)	63.29 (3.04)
Synthetic and good experimental	83.28 (0.70)	– (–)	48.33 (2.26)	49.24 (3.02)	49.88 (2.05)
Synthetic and all experimental	80.80 (0.63)	– (–)	48.91 (2.00)	49.72 (2.95)	49.93 (2.35)
Synthetic with noise	83.91 (0.54)	– (–)	50.00 (0.40)	49.82 (0.18)	49.84 (0.08)
	K neighbors classifier				
	Training	Synthetic	Clean experimental	Good experimental	All experimental
Synthetic	91.46 (0.34)	84.76 (0.76)	47.02 (2.56)	48.75 (1.49)	51.82 (0.95)
Good experimental	88.74 (0.89)	– (–)	75.14 (3.08)	75.16 (3.02)	67.37 (2.78)
All experimental	87.39 (0.90)	– (–)	73.44 (3.02)	76.56 (3.55)	72.10 (2.64)
Synthetic and good experimental	91.42 (0.19)	– (–)	60.97 (3.32)	75.87 (3.24)	67.19 (2.43)
Synthetic and all experimental	90.72 (0.19)	– (–)	66.44 (3.41)	74.37 (2.85)	69.80 (2.78)
Synthetic with noise	90.06 (0.71)	– (–)	57.67 (4.41)	54.29 (2.53)	54.31 (1.63)
	SVC				
	Training	Synthetic	Clean experimental	Good experimental	All experimental
Synthetic	99.95 (0.02)	85.57 (0.71)	48.37 (1.22)	48.13 (2.09)	49.02 (0.79)
Good experimental	100.00 (0.00)	– (–)	69.67 (3.48)	77.92 (3.29)	68.90 (2.42)
All experimental	99.91 (0.06)	– (–)	67.91 (4.58)	74.62 (4.25)	68.17 (2.23)
Synthetic and good experimental	99.03 (0.15)	– (–)	57.32 (3.77)	66.31 (2.01)	61.63 (2.33)
Synthetic and all experimental	98.40 (0.08)	– (–)	59.80 (4.19)	68.35 (3.85)	65.14 (2.41)
Synthetic with noise	96.10 (0.13)	– (–)	47.81 (3.05)	53.49 (1.60)	51.48 (1.65)

with an expected number of occurrences drawn from a uniform distribution between $[0, 0.2)$. Charge jumps, which appear as voltage jumps on gates, are achieved by randomly choosing a location in gate voltage space and a step size, defining the subregion of the current map which will be removed. The new image is then resized to its original size using python's scikit-image resize method. A 2D Gaussian convolution is applied to all image to simulate thermal broadening.

We generate 10 000 random noise maps for each noise type, which are normalised to a range between 0 and 1. These are then added to noiseless synthetic data by choosing a random sub-selection of maps and random amplitudes distributed uniformly between 0 and 0.2, an amplitude range over which we saw the highest accuracy variation. Varying the noise strength ensures to cover different noise levels found in experimental data. Random telegraph, white and $1/f$ noise are added to noiseless current maps, while random current and pinch-off current modulation maps are convolved. The resulting diagrams are scaled by the ratio of old and new maximum current values to ensure previous normalisations are preserved. Charge jumps are added to a number of charge stability diagrams determined by the random amplitude times the total number of diagrams in the target dataset.

Appendix C. Classification

We compare accuracies of a convolutional neural network and six binary classifiers trained on synthetic and experimental charge stability diagrams. The convolutional neural network was implemented in python's TensorFlow 2 and is summarized in table A.1. It consists of two convolutional layers and four dense layers with a drop out rate of 0.5. The convolutional layer have 32 and 64 kernels respectively, all 3×3 in size. The dense layers consist of 1024, 512, 128 and 2 neurons respectively. The RELU activation function is used in all but the last layer, which uses the softmax activation function. We use the Adam optimiser and a learning rate of 0.001. When trained with experimental data, the augmented dataset is used. We use standard augmentation techniques including shear, rotation, flip and crop transformations.

Additional examples of noise types added to synthetic data are pictured in figure A.1. Neural network classification results are listed in tables A.2 and A.3, showing accuracies and confusion matrices respectively. Neural network transfer learning accuracies are summarized in table A.4 and accuracies achieved by adding one noise type add a time to synthetic data are displayed in table A.5. Accuracies of simple binary classifiers are shown in table A.6. We use the original experimental dataset for training as accuracies are lower when augmented data is used.

ORCID iD

J Darulová https://orcid.org/0000-0002-9886-5007

References

[1] Petersson K D, Petta J R, Lu H and Gossard A C 2010 Quantum coherence in a one-electron semiconductor charge qubit *Phys. Rev. Lett.* **105** 246804
[2] Gorman J, Hasko D G and Williams D A 2005 Charge-qubit operation of an isolated double quantum dot *Phys. Rev. Lett.* **95** 090502
[3] Yang Y C, Coppersmith S N and Friesen M 2019 Achieving high-fidelity single-qubit gates in a strongly driven charge qubit with 1/f charge noise *npj Quantum Inf.* **5** 12
[4] Loss D and DiVincenzo D P 1998 Quantum computation with quantum dots *Phys. Rev.* A **57** 120–6
[5] Hanson R, Kouwenhoven L P, Petta J R, Tarucha S and Vandersypen L M K 2007 Spins in few-electron quantum dots *Rev. Mod. Phys.* **79** 1217–65
[6] Petta J R, Johnson A C, Taylor J M, Laird E A, Yacoby A, Lukin M D, Marcus C M, Hanson M P and Gossard A C 2005 Coherent manipulation of coupled electron spins in semiconductor quantum dots *Science* **309** 2180–4
[7] Veldhorst M *et al* 2015 A two-qubit logic gate in silicon *Nature* **526** 410–14
[8] Kitaev A Y 2001 Unpaired Majorana fermions in quantum wires *Phys.-Usp.* **44** 131–6
[9] Karzig T *et al* 2017 Scalable designs for quasiparticle-poisoning-protected topological quantum computation with Majorana zero modes *Phys. Rev.* B **95** 235305
[10] Alicea J, Oreg Y, Refael G, von Oppen F and Fisher M P A 2011 Non-Abelian statistics and topological quantum information processing in 1D wire networks *Nat. Phys.* **7** 412–7
[11] 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
[12] Zwolak J P *et al* 2020 Autotuning of double-dot devices *in situ* with machine learning *Phys. Rev. Appl.* **13** 034075
[13] 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** 1–17
[14] Durrer R, Kratochwil B, Koski J, Landig A, 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
[15] Lennon D T, Moon H, Camenzind L C, Yu L, Zumbühl D M, Briggs G A D, Osborne M A, Laird E A and Ares N 2019 Efficiently measuring a quantum device using machine learning *npj Quantum Inf.* **5** 79
[16] Moon H *et al* 2020 Machine learning enables completely automatic tuning of a quantum device faster than human experts *Nat. Commun.* **11** 4161

[17] Darulová J, Pauka S, Wiebe N, Chan K, Gardener G, Manfra M, Cassidy M and Troyer M 2020 Autonomous tuning and charge-state detection of gate-defined quantum dots *Phys. Rev. Appl.* **13** 054005
[18] 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
[19] Lapointe-Major M, Germain O, Camirand Lemyre J, Lachance-Quirion D, Rochette S, Camirand Lemyre F and Pioro-Ladrière M 2020 Algorithm for automated tuning of a quantum dot into the single-electron regime *Phys. Rev.* B **102** 085301
[20] Wang C, Li H, Hao Z, Li X, Zou C, Cai P, Wang Y, You Y Z and Zhai H 2020 Machine learning identification of impurities in the STM images *Chin. Phys. B* **29** 116805
[21] van der Wiel W G, De Franceschi S, Elzerman J M, Fujisawa T, Tarucha S and Kouwenhoven L P 2002 Electron transport through double quantum dots *Rev. Mod. Phys.* **75** 1–22
[22] Kroll J G *et al* 2019 Magnetic-field-resilient superconducting coplanar-waveguide resonators for hybrid circuit quantum electrodynamics experiments *Phys. Rev. Appl.* **11** 064053
[23] Croot X, Pauka S, Watson J, Gardner G, Fallahi S, Manfra M and Reilly D 2018 Device architecture for coupling spin qubits via an intermediate quantum state *Phys. Rev. Appl.* **10** 044058
[24] Teske J D, Humpohl S S, Otten R, Bethke P, Cerfontaine P, Dedden J, Ludwig A, Wieck A D and Bluhm J H 2019 A machine learning approach for automated fine-tuning of semiconductor spin qubits *Appl. Phys. Lett.* **114** 133102
[25] Botzem T *et al* 2018 Tuning methods for semiconductor spin qubits *Phys. Rev. Appl.* **10** 054026
[26] van Diepen C J, Eendebak P T, Buijtendorp B T, Mukhopadhyay U, Fujita T, Reichl C, Wegscheider W and Vandersypen L M K 2018 Automated tuning of inter-dot tunnel coupling in double quantum dots *Appl. Phys. Lett.* **113** 033101
[27] Weiss K, Khoshgoftaar T M and Wang D D 2016 A survey of transfer learning *J. Big Data* **3** 9
[28] Yoneda J, Takeda K, Noiri A, Nakajima T, Li S, Kamioka J, Kodera T and Tarucha S 2020 Quantum non-demolition readout of an electron spin in silicon *Nat. Commun.* **11** 1144
[29] Harvey S P, Bøttcher C G, Orona L A, Bartlett S D, Doherty A C and Yacoby A 2018 Coupling two spin qubits with a high-impedance resonator *Phys. Rev.* B **97** 235409
[30] Xue X *et al* 2019 Benchmarking gate fidelities in a Si/SiGe two-qubit device *Phys. Rev.* X **9** 021011
[31] Watson T F *et al* 2018 A programmable two-qubit quantum processor in silicon *Nature* **555** 633–7
[32] Cerfontaine P, Otten R, Wolfe M A, Bethke P and Bluhm H 2020 High-fidelity gate set for exchange-coupled singlet-triplet qubits *Phys. Rev.* B **101** 155311
[33] Hendrickx N W, Franke D P, Sammak A, Scappucci G and Veldhorst M 2020 Fast two-qubit logic with holes in germanium *Nature* **577** 487–91
[34] van der Walt S, Schönberger J L, Nunez-Iglesias J, Boulogne F, Warner J D, Yager N, Gouillart E and Yu T 2014 the scikit-image contributors, scikit-image: image processing in Python *PeerJ* **2** e453