

Identifying Pauli spin blockade using deep learning

Jonas Schuff¹, Dominic T. Lennon¹, Simon Geyer², David L. Craig¹, Federico Fedele¹, Florian Vigneau¹, Leon C. Camenzind², Andreas V. Kuhlmann², G. Andrew D. Briggs¹, Dominik M. Zumbühl², Dino Sejdinovic³, and Natalia Ares⁴

¹Department of Materials, University of Oxford, Oxford OX1 3PH, United Kingdom

²Department of Physics, University of Basel, 4056 Basel, Switzerland

³School of Computer and Mathematical Sciences & AIML, University of Adelaide, SA 5005, Australia

⁴Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, United Kingdom

Pauli spin blockade (PSB) can be employed as a great resource for spin qubit initialisation and readout even at elevated temperatures but it can be difficult to identify. We present a machine learning algorithm capable of automatically identifying PSB using charge transport measurements. The scarcity of PSB data is circumvented by training the algorithm with simulated data and by using cross-device validation. We demonstrate our approach on a silicon field-effect transistor device and report an accuracy of 96% on different test devices, giving evidence that the approach is robust to device variability. Our algorithm, an essential step for realising fully automatic qubit tuning, is expected to be employable across all types of quantum dot devices.

Electrostatically defined quantum dots are promising candidates for scalable quantum computation and simulation [1–3]. They can achieve universal quantum computation [4] with gates reaching high fidelity [5, 6]. Their properties are attractive for large-scale quantum processors, namely all-electrical control, compact size [2, 7, 8], and potential operating temperatures of above 1K [9–11].

Pauli spin blockade (PSB) is often a crucial requirement for spin qubit initialisation and readout. It allows for spin-to-charge conversion, as spin-conserved tunneling leads to current rectification [12]. We can rely on PSB even at elevated temperatures [9–11, 13]. It is thus essential to reliably and efficiently detect PSB. But PSB is elu-

sive; in the few-charges regime it can be found in unexpected gate voltage locations or it might be absent, and in the multi-charge regime it has to be found like the proverbial needle in a haystack. Its detection is challenging even for experienced human experimenters since evidence for PSB is subtle and it relies on several factors, including the details of the confinement potential. Those details are affected by fluctuations in the disorder potential due to fabrication variances and defects within the material.

To achieve true scalability, we need an automatic method for detecting PSB that can be incorporated as a fundamental building block into a fully automatic qubit tuning algorithm. The scarcity of available data makes reliable automation tough. In addition, PSB data tends to be unbalanced, meaning that there are many more examples of measurements in which PSB is not present than examples evidencing PSB. Measurements exhibiting PSB are therefore rare in an already scarce body of data. An automatic approach would also allow us to gather sufficient data and insight to reveal the factors that determine the presence of PSB, which can be even more difficult to identify in material systems with strong spin orbit coupling.

We demonstrate how to detect PSB using deep neural networks. Deep neural networks were used in the context of charge state identification, coarse and fine tuning and readout [14–30], with some approaches using simulated data to train their algorithms [18, 19, 31]. Unlike these cases, automatic PSB detection required us to make use of extremely scarce and unbalanced quantum device data. We developed a physics-inspired simulator and introduced cross-device validation to address this challenge.

Jonas Schuff: jonas.schuff@materials.ox.ac.uk

Natalia Ares: natalia.ares@eng.ox.ac.uk

We demonstrate our algorithm in a silicon fin field-effect transistor (FinFET) confining holes [11]. We show that we can achieve an accuracy of over 96% on identifying signs of PSB on unseen devices. The data stems from four silicon FinFET devices with different gate dimensions. We designate training devices, from which we extract training data, and testing devices, from which we extract data to test our algorithm. We discuss the performance of the algorithm for different types of training data, using simulated training data, measured training data, and a combination of both.

1 Experiment

A schematic representation of a silicon FinFET device similar to the ones used in this work and a cross-sectional transmission electron microscope (TEM) image are shown in Figs. 1a, b. The devices are fabricated using a CMOS-compatible fabrication process, where a self-alignment technique allows for ultra-small gate length and intrinsically perfect layer-to-layer alignment [32]. The fin provides a quasi 1D confinement for holes and a double quantum dot (DQD) can be defined using gate electrodes. Source and drain reservoirs are formed by lead gates L1 and L2, which accumulate p-type carriers. The plunger gates P1 and P2 allow for control of the hole occupancy. The inter-dot coupling is controlled by gate B. We perform transport measurements by applying a bias voltage V_{SD} between source and drain drawing a current I through the device. Measurements of current as a function of the plunger gate voltages V_{P1} and V_{P2} are called stability diagrams. Energetically allowed charge transitions appear as two bias triangles in stability diagrams, see Fig. 1e. Bias triangles indicate that the device is tuned into the DQD regime.

Four devices with different dimensions (for details see Appendix A) were measured at different bias voltages and at temperatures ranging from 20 mK to 1.5 K. In all measurements the magnetic field was applied in-plane and perpendicular to the fin as indicated in Fig. 1a.

1.1 Pauli spin blockade

We look at the $(1, 1) - (0, 2)$ charge transition, where (m, n) denotes the effective hole occupancy

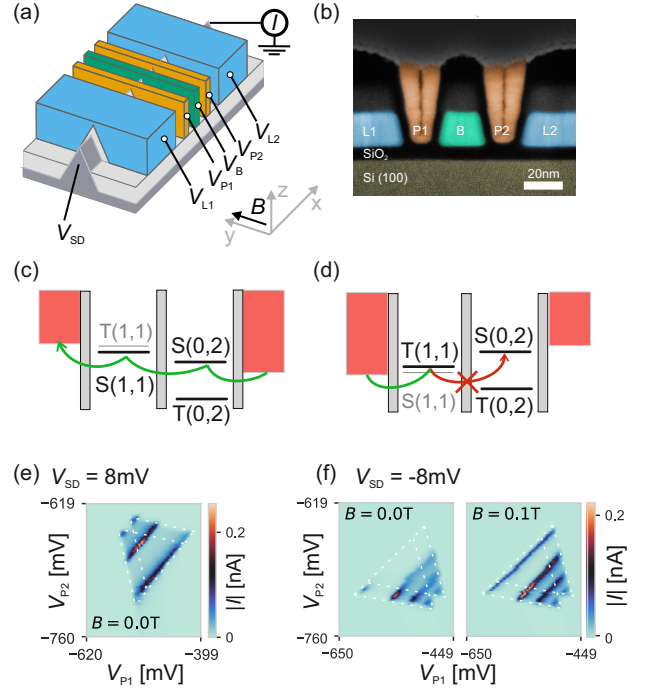


Figure 1: **Pauli spin blockade in silicon FinFET devices.** **a** Schematic and **b** cross-sectional TEM image of a silicon FinFET device. The plunger gates (P1 and P2) accumulate holes in a DQD, the inter-dot barrier is controlled by B and source and drain reservoirs are accumulated using lead gates (L1 and L2). **c**, **d** Schematic of a transport cycle in unblocked/blocked configurations respectively. While in the unblocked configuration holes can easily tunnel through the device, spin-conservation blocks the transport through the ground state transition when inverting V_{SD} due to the forbidden $T(1, 1) - S(0, 2)$ transition. **e** Bias triangles with positive V_{SD} . **f** Bias triangles with negative V_{SD} . The current at the base line of the triangles, i.e. current due to the $T(1, 1) - S(0, 2)$ transition, is blocked (left). A finite magnetic field $B = 0.1$ T lifts the blockade (right). We show the absolute value of current for ease of comparison between figures. We outline the bias triangles with white dashed lines to guide the eye.

of the left and right dot omitting filled shells. For positive V_{SD} , we expect a hole to tunnel onto the right dot and form a singlet state $S(0, 2)$, since the triplet state $T(0, 2)$ is energetically unavailable (see Fig. 1c). The large energy splitting comes from the fact that the symmetric triplet spin state requires an anti-symmetric orbital state with higher energy. The hole can now tunnel via the $S(1, 1)$ state to the left reservoir completing the transport cycle. When applying a negative V_{SD} , we expect a hole to enter the left dot from the reservoir (see Fig. 1d). Due to the weak inter-dot coupling $S(1, 1)$ and $T(1, 1)$ are nearly deg-

erate, such that both states are accessible. If $T(1,1)$ is occupied, transport through the DQD is blocked since the $T(1,1)$ – $S(0,2)$ transition is forbidden by spin conservation and the $T(0,2)$ state is energetically unavailable. The effect is also possible in the opposite bias direction, i.e. a blockade can occur with positive bias voltage for a $(1,1) - (2,0)$ charge transition.

The blockade can be lifted by processes that allow transitions out of the $T(1,1)$ state. In systems dominated by hyperfine interaction [33] or spin-flip cotunneling [34], these transitions are enabled at zero magnetic field giving rise to a finite current. For holes in silicon FinFETs, however, spin orbit coupling is the dominant interaction [11, 32], which lifts the PSB at finite magnetic field due to spin-flip tunneling coupling the $T(1,1)$ states with the $S(0,2)$ state [35–37]. This mechanism for holes and strong spin orbit coupling in a magnetic field is described in detail in Ref. [38]. Furthermore, independent of the magnetic field, the blockade is lifted when the interdot energy level detuning reaches or exceeds the singlet-triplet splitting of the $S(0,2)$ and $T(0,2)$ states.

Signatures of PSB can be observed in Figs. 1e, f. Two bias triangles are visible for $V_{SD} = 8$ mV. For opposite polarity $V_{SD} = -8$ mV the current at the common base line of the triangles, i.e. current due to ground state transitions, is strongly suppressed at zero magnetic field. The excited state transitions are visible as parallel stripes away from the common base line. They appear at a detuning exceeding the singlet-triplet splitting and are visible in both bias directions, although the magnitude of the corresponding current might differ due to device asymmetries. The blockade at the base line of the bias triangles is lifted for a magnetic field of $B = 0.1$ T. PSB can be detected by comparing stability diagrams displaying bias triangles at $B = 0$ and $B \neq 0$ and looking for changes in the base line current. Two stability diagrams showing the same bias triangles at $B = 0$ and $B \neq 0$ will be called a *pair*. These pairs are the type of data required by our algorithm to identify PSB. A comparison between stability diagrams corresponding to opposite signs of bias voltage could also be used to identify PSB. We expect that this comparison is difficult to use in PSB detection since differences in the transport features might arise from device asymmetries.

1.2 Simulator

A simulator allows us to generate large and diverse data sets needed to train the deep learning algorithm. Our goal is to simulate pairs of stability diagrams, as introduced in Section 1.1. To achieve this, we first calculate the steady state current [39] with one energy level in each quantum dot. Next, we consider multiple energy levels in each dot and sum the contribution of every possible combination of energy levels to determine the total current.

A simple approach to account for PSB in the simulator involves suppressing the tunneling rate between ground states in each quantum dot. By suppressing those tunneling rates, we identify the ground states with the $T(1,1)$ and $S(0,2)$ configurations. The suppression of the tunnel rates between these two configurations has an analogous effect to that of PSB. While this can lead to simulated measurements that may be unphysical, it enables efficient training of the algorithm. We add various sources of noise to the current simulation. We sample each parameter of the simulator, e.g. lever arms, from a given range of possible values. We did not perform an optimisation over those sampling ranges to avoid introducing a greater bias to the output of the simulator. A comprehensive description of the simulator is available in Appendix B.

Fig. 2 shows examples of simulated bias triangles. PSB is introduced for $B_{sim} = 0$ while it is not considered for $B_{sim} \neq 0$ (Fig. 2a). Each pair of simulated stability diagrams is marked by a grey box in Fig. 2. Examples of bias triangles where PSB does not occur are shown in Fig. 2b. In this case the difference between measurements in a pair is due to the added noise.

1.3 Deep learning

The input of the neural network, a deep residual network [40] with 18 layers, is a pair consisting of two stability diagrams. Such pairs are defined in Section 1.1. The current input values are jointly normalized between 0 and 1 for each pair. The neural network outputs a score between 0 and 1. A score of 1 (0) corresponds to maximum (minimum) confidence in the occurrence of PSB. The threshold of classification is set at 0.5.

Due to the random sampling of parameters in the simulations and the randomness in the data

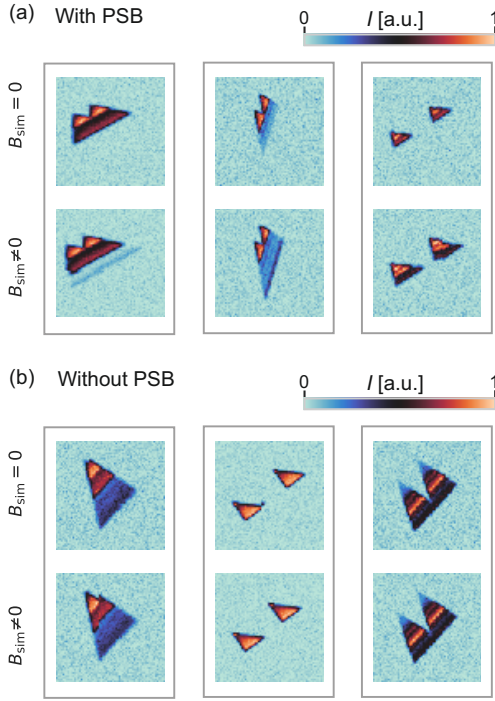


Figure 2: **Examples of simulated training data.** Each pair in a and b (grey boxes) represents a different set of parameters. The simulated current values are normalized for each of these pairs. **a** Examples displaying PSB. **b** Examples in which PSB is not introduced in the simulation. Running the simulator twice with the same parameters results in two similar measurements that only differ due to the noise added to the simulator.

augmentation process, we expect a high variance in the results of the classification if we train the neural network more than once. We thus train the neural network ten times to obtain ten individual classifiers that we combine into an ensemble of classifiers. By ensemble classifier we mean a classifier that uses the average score of all individual classifiers as the score of the ensemble. This approach is expected to produce more robust results than individual classifiers. Details on the choice of architecture and the training of the neural networks can be found in Appendices C and D.

2 Results

To test our algorithm, we consider a device that is tuned to a double quantum dot regime. Often, the next step is to identify bias triangles which might exhibit PSB. We select gate voltage windows enclosing bias triangles as indicated by rectangles labelled A-C in Fig. 3a. All other

bias triangles that could be observed in this device are displayed in Appendix H. To create the pairs introduced in Section 1.1 we combine the bias triangles delimited by the chosen rectangles with an equivalent version of these measurements at low magnetic field.

Magnetic field hysteresis can shift the PSB signature away from $B=0$. This can sometimes occur when using superconducting coils. We thus do not set $B = 0$ for these measurements but we choose 9 different equidistant magnetic fields between $B = -0.08$ T and $B = 0.08$ T. In this way, we create nine pairs of stability diagrams for each gate voltage window A to C. Example pairs are displayed in Fig. 3b.

The predictions obtained by the ensemble of classifiers trained on simulated data as described in Section 1.3 can be seen in Fig. 3c. All predictions for charge transitions A and B are negative, i.e. no signs of PSB are detected. For charge transitions in C PSB is detected for pairs with low magnetic field values $B = -0.06, -0.04, -0.02$ T even though the base line is very faint when PSB is lifted. Our algorithm classifies pairs, not transitions. For charge transition C, some pairs are classified as not exhibiting PSB for those magnetic fields for which the signature of PSB is not apparent. Also, for this transition, the blocked current occurs at $B \neq 0$. We assume that the effective magnetic field is 0 for the blocked case and the offset is due to hysteresis.

To confirm the predictions, we measure the current at the base of the bias triangles as a function of the magnetic field and detuning, see Fig. 3d. The detuning axes, i.e. the sweep direction of gate voltages V_{P1} and V_{P2} , are indicated as white dotted lines in Fig. 3b. For charge transitions in C, the current suppression is evident at low magnetic field values. This verifies that the corresponding pairs are correctly identified as displaying PSB. Conversely, no current reduction is observed for bias triangles A and B, confirming the absence of PSB. An automated tuning pipeline could use this information to perform further experiments, such as measurements of Rabi oscillations. Additional results can be found in Appendix H.

Benchmarking

We now benchmark the performance of the algorithm for different types of training data, and considering both individual and ensemble classi-

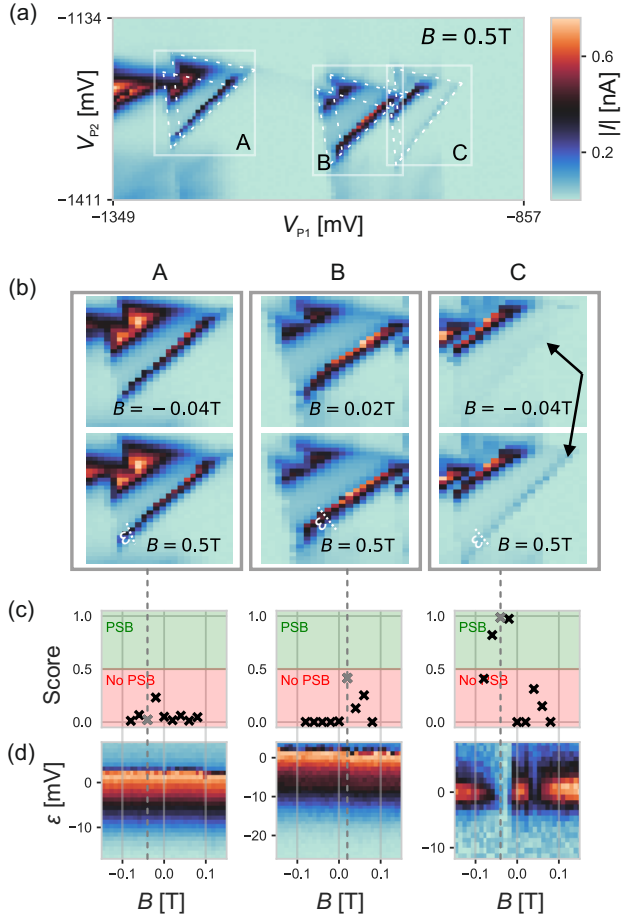


Figure 3: **Using the classifier to find bias triangles exhibiting PSB.** **a** Stability diagram taken at $B = 0.5$ T. White rectangles are drawn by a human wishing to classify bias triangles A-C. As in Fig. 1, we display the absolute value of current and outline the bias triangles with dashed white lines. **b** Examples of pairs that are inputs to the classifier. The bottom row shows the bias triangles from Fig. 3a, which serve as a reference, since we expect PSB to be lifted at this magnetic field value. The same bias triangles are shown in the top row at low magnetic field. Black arrows point at a vanishing common base line indicating the presence of PSB. **c** Classification of bias triangles A-C. The ensemble of classifiers produces a score for each of the nine pairs corresponding to charge transitions A-C, which are composed by paired measurements at $B = 0.5$ T and at values of B close to zero ($B = -0.08$ to 0.08 T). A score of over 0.5 predicts the occurrence of PSB. **d** Magnetic field dependence of current measurements along the detuning axis, indicated by white dotted lines in Fig. 3b. A reduction in current confirms the predicted PSB for three pairs of C. We draw a dashed line from the pairs in Fig. 3b through Fig. 3c and Fig. 3d to indicate the magnetic field values corresponding to those pairs. More results can be found in Appendix H.

fiers.

We built a data set consisting of 53 pairs, as

defined in Section 1.1, originating from measurements of four different devices. In this data set, we only included examples that exhibit well shaped bias triangles and measurements that show either clear signatures of PSB or no signatures of PSB at all, so that human experts can verify the correctness of the label. Table 1 shows how many pairs are associated with each device. All pairs used are shown in Fig. 12 in Appendix H.

	Device			
	i	ii	iii	iv
Positive	1	15	2	1
Negative	0	16	14	4

Table 1: **Structure of experimental data used to benchmark the algorithm.** For each of the devices considered, number of pairs displaying PSB (positive) and not displaying PSB (negative) as assessed by a human judge. Data from device ii was collected over multiple cool-downs. A few individual pairs might show the same charge transitions in different locations in gate voltage space, e.g. for different tunnel coupling strengths. The data set includes measurements corresponding to $B = -0.04$ T from Fig. 3 and Fig. 11.

To study the effect of different training data on the performance of the algorithm, we investigate three cases, which we refer to as Simulated data (**Sim**), Experimental data (**Exp**) and Mixed data (**Mix**). **Sim** corresponds to the case of training the classifiers with only simulated data. These classifiers are the same as those used for the predictions in Fig. 3. For **Exp**, training is performed only with experimental data from the devices listed in Table 1. **Mix** is a mix of training with experimental and simulated data; half of the training data is experimental and the other half is simulated.

We augment the training data by random shearing, stretches, crops, contrast and brightness such that there are 50,000 pairs. Details of training and augmentation can be found in Appendix D.

Since our data set of measurements is small we employ a form of cross-validation in the cases **Exp** and **Mix** which we call cross-device validation. This means that each classifier is tested on data from a device that is different from the devices it was trained on. Each of these groups of training and testing data forms a *fold*. The process is repeated until all devices have served as a testing device once. Because we don't have

both positive and negative pairs from device i we don't use that device as a testing device as computing some specific performance metrics is not possible. In computation of the cross-validation performance metrics, we weight each fold according to the number of pairs it holds. This can be seen as a form of inverse-variance weighting. In the case **Sim** we do not need cross-validation since we use only experimental data for testing.

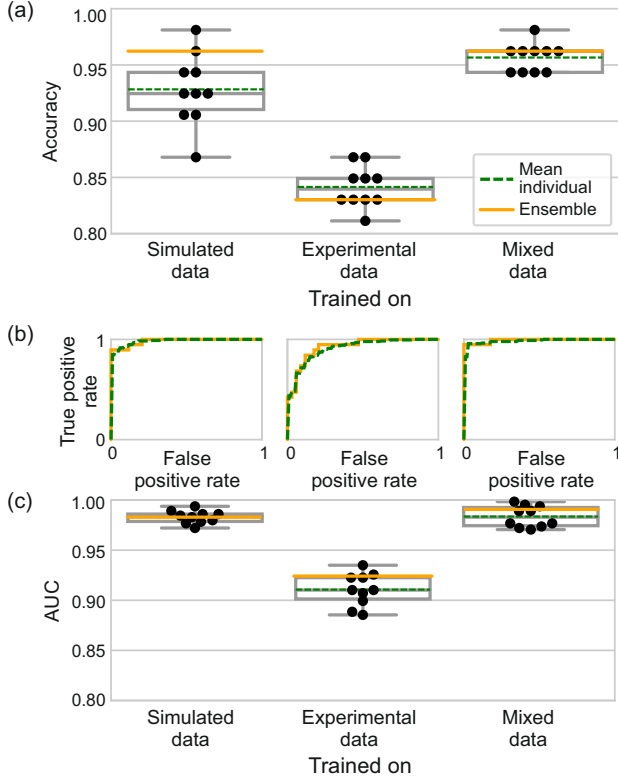


Figure 4: **Benchmarking results.** **a** Accuracy. Box plots with results from individual classifiers plotted as dots, the mean of those is plotted as a dashed green line, and the performance of the ensemble of classifiers as a solid orange line. **b** Receiver operating characteristic (ROC). For each case, the ensemble ROC is plotted as a solid orange line and the mean individual ROC as a dashed green line. **c** Area under the curve (AUC). Legend as in Fig. 4a.

We define accuracy as the proportion of correctly classified data in the complete data set. The accuracies of single and ensemble classifiers are shown in Fig. 4a. Training the classifier with only simulated data (case **Sim**) leads to an ensemble accuracy of 96.2%. This high accuracy indicates that our simulator produces data that are similar to experimental measurements. The ensemble classifier outperforms the mean of individual classifiers (92.8%), justifying the use of ensemble classifiers. Previous classifiers that clas-

sify data from quantum devices show accuracies below 90% [31]. The mean accuracy of individual classifiers decreases to 84.2% for classifiers trained purely on experimental data (case **Exp**) and the ensemble of classifiers achieves an accuracy of only 83.0%, showing the advantage of a simulator. These lower accuracies indicate overfitting as a result of small training data sets. The simulated data provides a more representative and diverse data set that prevents this problem. Mixing the two data types (case **Mix**) leads to the same ensemble classifier accuracy as using only simulated data (96.2%). The mean of the individual classifiers improves to 95.7% and a lower variance of the individual classifiers is observed.

We find that neural networks trained only on experimental data strongly under-perform those that were trained with only simulated data or a mix of both types of data. Thus, the simulated data seems to be the main driver of performance in contrast to the findings in [31]. The superiority of classifiers trained on simulated data may be due to the scarcity of experimental data and specific to the problem of detecting PSB. We train with between 31 and 48 experimental measurements before augmentation in the cases **Exp** and **Mix** depending on the fold. In the case **Sim** (**Exp**) we use 25,000 (12,500) simulated pairs before augmentation. The influence of the number of pairs used in training is discussed in Appendix E.

The accuracy can be affected by the choice of the score threshold so we use other metrics to further analyse our results. Choosing a score threshold means navigating a trade-off between true positive rate (TPR) and false positive rate (FPR). The receiver operating characteristic (ROC) curve, a plot of TPR against FPR, illustrates this trade-off, see Fig. 4b. The area under the ROC curve (AUC) is independent of the score threshold and is 1 for a perfect classifier. An arbitrary classifier would produce a ROC that is a diagonal with an AUC of 0.5. In the case **Sim** we obtain an AUC of 0.983 for the ensemble classifier, see Fig. 4c. This can be slightly improved by mixing in experimental data (case **Mix**), leading to an AUC of 0.991. In comparison, only using experimental data (case **Exp**) gives an ensemble AUC of 0.924. The mean individual classifiers achieve an AUC of 0.983, 0.910, and 0.984 in cases **Sim**, **Exp** and **Mix**, respectively. The re-

sults obtained by estimating the AUC are similar to those obtained by calculating the accuracy of the classifiers; training only on experimental data results in under-performing classifiers. A comparison with the results obtained using a smaller network can be found in Appendix F. An evaluation of performance on simulated data is presented in Appendix G. Classification results for all experimental pairs used here can be found in Fig. 12 in Appendix H.

3 Discussion

We train deep neural networks with simulated and experimental data to detect bias triangles that show signs of PSB. We demonstrate that even in the case of extremely limited data, a neural network can be successfully trained to solve this intricate task. Cross-device validation allows us to show that the method performs well on unseen devices.

We find a higher variance of accuracy of individual classifiers when trained on simulated data compared to classifiers trained on real data. This might be due to the limitations of the simulator and could be mitigated by increasing the number of simulations used in training at a larger computational cost. Forming an ensemble prediction leads to a high accuracy. In contrast to previous work [31], simulated data seems to be more important for training than experimental data. This might be due to the scarcity of experimental data available for the classification of PSB. The small experimental data set potentially lacks comprehensive information, which could hinder the neural network’s ability to learn.

In other types of devices, the signature of PSB might be reversed, i.e. a maximum leakage current is observed for $B = 0$. Our methodology remains applicable to these cases by merely reversing the sequence of the pairs employed in the classification. Competing effects might lead to partial lifts when PSB is present. We can see that a few partial lifts present in the experimental data were correctly classified by our algorithm.

The hurdles in the scarcity of data can be overcome through careful training and high quality simulated data. Scarcity of data could be addressed by the community through open access to data. We expect our algorithm to identify PSB in different types of devices given consistent

datasets.

Alternative approaches to the automatic identification of PSB could include both data-driven and non-data-driven methods, such as computer vision techniques or feature engineering. However, due to the limited availability of data and the elusive nature of signatures associated with PSB, these approaches could prove particularly challenging.

The simulator developed in this study holds the potential for various applications, including the development of an algorithm capable of recognizing energy splittings or defining an energy detuning axis. The integration of simple simulators to train machine learning models can be applied to tackle a wide range of quantum device challenges, including feature extraction and efficient measurement of stability diagrams, Rabi chevrons, and EDSR spectroscopy.

In light of the subtlety of the problem of identifying PSB we expect this method to have promising applications in the automation of tuning procedures for spin qubit devices. The classifier could be embedded in a larger tuning algorithm to reliably determine whether a charge transition is promising for PSB and reveal hidden patterns.

Acknowledgments

This work was supported by the Royal Society, EPSRC Platform Grant (EP/R029229/1), the European Research Council (grant agreement 948932), the Swiss NSF Projects 179024 and NCCR SPIN, and the Swiss Nanoscience Institute. The research leading to these results has also received funding from the European Union’s Horizon 2020 Research and Innovation Programme *European Microkelvin Platform (EMP)* under Grant Agreement no 824109. J.S. acknowledges financial support from the EPSRC, Grant Number R72976/CN001. We thank the Cleanroom Operations Team of the Binning and Rohrer Nanotechnology Center (BRNC) for their help and support in device fabrication.

We want to thank Edward Laird and Christian Schroeder de Witt for helpful discussions, Brandon Severin for useful discussions about the devices, and Floris Zwanenburg, Matthias Brauns, Rafael Egli, Mathieu de Kruijf, Sergey Frolov and Azarin Zarassi for sharing experimental data that was used in early stages of this work.

Data availability

The data is available at doi.org/10.5281/zenodo.7948852

Code availability

The code used in this publication, including the simulator and the training procedure, is available at github.com/oxquantum-repo/identifying-psb.

Author contributions

J.S., S.G. and D.T.L. performed the experiments at the University of Basel under the supervision of L.C.C., A.V.K and D.M.Z. J.S. and D.C. wrote the simulator with guidance from D.T.L., F.F. and F.V. J.S. performed the training and evaluation of the neural networks with help from D.T.L. and under the supervision of D.S., G.A.D.B. and N.A. The samples were fabricated by S.G. and A.V.K. The project was conceived by N.A. All authors contributed to the manuscript, and commented and discussed results.

References

- [1] Daniel Loss and David P DiVincenzo. Quantum computation with quantum dots. *Physical Review A*, 57(1):120, 1998. DOI: [10.1103/PhysRevA.57.120](https://doi.org/10.1103/PhysRevA.57.120).
- [2] LMK Vandersypen, H Bluhm, JS Clarke, AS Dzurak, R Ishihara, A Morello, DJ Reilly, LR Schreiber, and M Veldhorst. Interfacing spin qubits in quantum dots and donors - hot, dense, and coherent. *npj Quantum Information*, 3(1):1–10, 2017. DOI: [10.1038/s41534-017-0038-y](https://doi.org/10.1038/s41534-017-0038-y).
- [3] Toivo Hensgens, Takafumi Fujita, Laurens Janssen, Xiao Li, CJ Van Diepen, Christian Reichl, Werner Wegscheider, S Das Sarma, and Lieven MK Vandersypen. Quantum simulation of a Fermi–Hubbard model using a semiconductor quantum dot array. *Nature*, 548(7665):70–73, 2017. DOI: [10.1038/nature23022](https://doi.org/10.1038/nature23022).
- [4] Menno Veldhorst, CH Yang, JCC Hwang, W Huang, JP Dehollain, JT Muhonen, S Simmons, A Laucht, FE Hudson, Kohei M Itoh, et al. A two-qubit logic gate in silicon. *Nature*, 526(7573):410–414, 2015. DOI: [10.1038/nature15263](https://doi.org/10.1038/nature15263).
- [5] Pascal Cerfontaine, Tim Botzem, Julian Ritzmann, Simon Sebastian Humpohl, Arne Ludwig, Dieter Schuh, Dominique Bougeard, Andreas D Wieck, and Hendrik Bluhm. Closed-loop control of a GaAs-based singlet-triplet spin qubit with 99.5% gate fidelity and low leakage. *Nature Communications*, 11(1):1–6, 2020. DOI: [10.1038/s41467-020-17865-3](https://doi.org/10.1038/s41467-020-17865-3).
- [6] Akito Noiri, Kenta Takeda, Takashi Nakajima, Takashi Kobayashi, Amir Sammak, Giordano Scappucci, and Seigo Tarucha. Fast universal quantum gate above the fault-tolerance threshold in silicon. *Nature*, 601(7893):338–342, 2022. DOI: [10.1038/s41586-021-04182-y](https://doi.org/10.1038/s41586-021-04182-y).
- [7] Stephan GJ Philips, Mateusz T Mądzik, Sergey V Amitonov, Sander L de Snoo, Maximilian Russ, Nima Kalhor, Christian Volk, William IL Lawrie, Delphine Brousse, Larysa Tryputen, et al. Universal control of a six-qubit quantum processor in silicon. *Nature*, 609(7929):919–924, 2022. DOI: [10.1038/s41586-022-05117-x](https://doi.org/10.1038/s41586-022-05117-x).
- [8] Federico Fedele, Anasua Chatterjee, Saeed Fallahi, Geoffrey C Gardner, Michael J Manfra, and Ferdinand Kuemmeth. Simultaneous operations in a two-dimensional array of singlet-triplet qubits. *PRX Quantum*, 2(4):040306, 2021. DOI: [10.1103/PRXQuantum.2.040306](https://doi.org/10.1103/PRXQuantum.2.040306).
- [9] Luca Petit, HGJ Eenink, M Russ, WIL Lawrie, NW Hendrickx, SGJ Philips, JS Clarke, LMK Vandersypen, and M Veldhorst. Universal quantum logic in hot silicon qubits. *Nature*, 580(7803):355–359, 2020. DOI: [10.1038/s41586-020-2170-7](https://doi.org/10.1038/s41586-020-2170-7).
- [10] Chih Heng Yang, RCC Leon, JCC Hwang, Andre Saraiva, Tuomo Tanttu, Wister Huang, J Camirand Lemyre, Kok Wai Chan, KY Tan, Fay E Hudson, et al. Operation of a silicon quantum processor unit cell above one kelvin. *Nature*, 580(7803):350–354, 2020. DOI: [10.1038/s41586-020-2171-6](https://doi.org/10.1038/s41586-020-2171-6).
- [11] Leon C Camenzind, Simon Geyer, Andreas Fuhrer, Richard J Warburton, Dominik M Zumbühl, and Andreas V Kuhlmann. A hole spin qubit in a fin field-effect transistor

- above 4 kelvin. *Nature Electronics*, 5(3):178–183, 2022. DOI: [10.1038/s41928-022-00722-0](https://doi.org/10.1038/s41928-022-00722-0).
- [12] Ronald Hanson, Leo P Kouwenhoven, Jason R Petta, Seigo Tarucha, and Lieven MK Vandersypen. Spins in few-electron quantum dots. *Reviews of Modern Physics*, 79(4):1217, 2007. DOI: [10.1103/RevModPhys.79.1217](https://doi.org/10.1103/RevModPhys.79.1217).
- [13] Luca Petit, Maximilian Russ, Gertjan HGJ Eenink, William IL Lawrie, James S Clarke, Lieven MK Vandersypen, and Menno Veldhorst. Design and integration of single-qubit rotations and two-qubit gates in silicon above one kelvin. *Communications Materials*, 3(1):82, 2022. DOI: [10.1038/s43246-022-00304-9](https://doi.org/10.1038/s43246-022-00304-9).
- [14] J Darulová, SJ Pauka, N Wiebe, KW Chan, GC Gardener, MJ Manfra, MC Cassidy, and Matthias Troyer. Autonomous tuning and charge-state detection of gate-defined quantum dots. *Physical Review Applied*, 13(5):054005, 2020. DOI: [10.1103/PhysRevApplied.13.054005](https://doi.org/10.1103/PhysRevApplied.13.054005).
- [15] H Moon, DT Lennon, J Kirkpatrick, NM van Esbroeck, LC Camenzind, Liuqi Yu, F Vigneau, DM Zumbühl, G Andrew D Briggs, MA Osborne, et al. Machine learning enables completely automatic tuning of a quantum device faster than human experts. *Nature Communications*, 11(1):1–10, 2020. DOI: [10.1038/s41467-020-17835-9](https://doi.org/10.1038/s41467-020-17835-9).
- [16] Brandon Severin, Dominic T Lennon, Leon C Camenzind, Florian Vigneau, F Fedele, D Jirovec, A Ballabio, D Chrastina, G Isella, M de Kruijff, et al. Cross-architecture Tuning of Silicon and SiGe-based Quantum Devices Using Machine Learning. *arXiv preprint arXiv:2107.12975*, 2021. DOI: [10.48550/arXiv.2107.12975](https://doi.org/10.48550/arXiv.2107.12975).
- [17] Timothy A Baart, Pieter T Eendebak, Christian Reichl, Werner Wegscheider, and Lieven MK Vandersypen. Computer-automated tuning of semiconductor double quantum dots into the single-electron regime. *Applied Physics Letters*, 108(21):213104, 2016. DOI: [10.1063/1.4952624](https://doi.org/10.1063/1.4952624).
- [18] Sandesh S Kalantre, Justyna P Zwolak, Stephen Ragole, Xingyao Wu, Neil M Zimmerman, Michael D Stewart, and Jacob M Taylor. Machine learning techniques for state recognition and auto-tuning in quantum dots. *npj Quantum Information*, 5(1):1–10, 2019. DOI: [10.1038/s41534-018-0118-7](https://doi.org/10.1038/s41534-018-0118-7).
- [19] Justyna P Zwolak, Thomas McJunkin, Sandesh S Kalantre, JP Dodson, ER MacQuarrie, DE Savage, MG Lagally, SN Coppersmith, Mark A Eriksson, and Jacob M Taylor. Autotuning of double-dot devices in situ with machine learning. *Physical Review Applied*, 13(3):034075, 2020. DOI: [10.1103/PhysRevApplied.13.034075](https://doi.org/10.1103/PhysRevApplied.13.034075).
- [20] V Nguyen, SB Orbell, Dominic T Lennon, Hyungil Moon, Florian Vigneau, Leon C Camenzind, Liuqi Yu, Dominik M Zumbühl, G Andrew D Briggs, Michael A Osborne, et al. Deep reinforcement learning for efficient measurement of quantum devices. *npj Quantum Information*, 7(1):1–9, 2021. DOI: [10.1038/s41534-021-00434-x](https://doi.org/10.1038/s41534-021-00434-x).
- [21] Justyna P Zwolak, Thomas McJunkin, Sandesh S Kalantre, Samuel F Neyens, ER MacQuarrie, Mark A Eriksson, and Jacob M Taylor. Ray-based framework for state identification in quantum dot devices. *PRX Quantum*, 2(2):020335, 2021. DOI: [10.1103/PRXQuantum.2.020335](https://doi.org/10.1103/PRXQuantum.2.020335).
- [22] NM van Esbroeck, DT Lennon, H Moon, V Nguyen, F Vigneau, LC Camenzind, L Yu, DM Zumbühl, GAD Briggs, Dino Sejdinovic, et al. Quantum device fine-tuning using unsupervised embedding learning. *New Journal of Physics*, 22(9):095003, 2020. DOI: [10.1088/1367-2630/abb64c](https://doi.org/10.1088/1367-2630/abb64c).
- [23] 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. *Applied Physics Letters*, 114(13):133102, 2019. DOI: [10.1063/1.5088412](https://doi.org/10.1063/1.5088412).
- [24] CJ Van Diepen, Pieter T Eendebak, Bruno T Buijtendorp, Uditendu Mukhopadhyay, Takafumi Fujita, Christian Reichl, Werner Wegscheider, and Lieven MK Vandersypen. Automated tuning of inter-dot tunnel coupling in double quantum dots. *Applied Physics Letters*, 113(3):033101, 2018. DOI: [10.1063/1.5031034](https://doi.org/10.1063/1.5031034).

- [25] Tim Botzem, Michael D Shulman, Sandra Foletti, Shannon P Harvey, Oliver E Dial, Patrick Bethke, Pascal Cerfontaine, Robert PG McNeil, Diana Mahalu, Vladimir Umansky, et al. Tuning methods for semiconductor spin qubits. *Physical Review Applied*, 10(5):054026, 2018. DOI: [10.1103/PhysRevApplied.10.054026](https://doi.org/10.1103/PhysRevApplied.10.054026).
- [26] David L Craig, Hyungil Moon, Federico Fedele, Dominic T Lennon, Barnaby van Straaten, Florian Vigneau, Leon C Camenzind, Dominik M Zumbühl, G Andrew D Briggs, Michael A Osborne, Dino Sejdicinovic, and Natalia Ares. Bridging the reality gap in quantum devices with physics-aware machine learning. *arXiv preprint arXiv:2111.11285*, 2021. DOI: [10.48550/arXiv.2111.11285](https://doi.org/10.48550/arXiv.2111.11285).
- [27] Stefanie Czischek, Victor Yon, Marc-Antoine Genest, Marc-Antoine Roux, Sophie Rochette, Julien Camirand Lemyre, Mathieu Moras, Michel Pioro-Ladrière, Dominique Drouin, Yann Beilliard, et al. Miniaturizing neural networks for charge state auto-tuning in quantum dots. *Machine Learning: Science and Technology*, 3(1):015001, 2021. DOI: [10.1088/2632-2153/ac34db](https://doi.org/10.1088/2632-2153/ac34db).
- [28] Renato Durrer, Benedikt Kratochwil, Jonne V Koski, Andreas J Landig, Christian Reichl, Werner Wegscheider, Thomas Ihn, and Eliska Greplova. Automated tuning of double quantum dots into specific charge states using neural networks. *Physical Review Applied*, 13(5):054019, 2020. DOI: [10.1103/PhysRevApplied.13.054019](https://doi.org/10.1103/PhysRevApplied.13.054019).
- [29] Maxime Lapointe-Major, Olivier Germain, J Camirand Lemyre, Dany Lachance-Quirion, Sophie Rochette, F Camirand Lemyre, and Michel Pioro-Ladrière. Algorithm for automated tuning of a quantum dot into the single-electron regime. *Physical Review B*, 102(8):085301, 2020. DOI: [10.1103/PhysRevB.102.085301](https://doi.org/10.1103/PhysRevB.102.085301).
- [30] Yuta Matsumoto, Takafumi Fujita, Arne Ludwig, Andreas D Wieck, Kazunori Komatani, and Akira Oiwa. Noise-robust classification of single-shot electron spin readouts using a deep neural network. *npj Quantum Information*, 7(1):1–7, 2021. DOI: [10.1038/s41534-021-00470-7](https://doi.org/10.1038/s41534-021-00470-7).
- [31] Jana Darulová, Matthias Troyer, and Maja C Cassidy. Evaluation of synthetic and experimental training data in supervised machine learning applied to charge-state detection of quantum dots. *Machine Learning: Science and Technology*, 2021. DOI: [10.1088/2632-2153/ac104c](https://doi.org/10.1088/2632-2153/ac104c).
- [32] Simon Geyer, Leon C Camenzind, Lukas Czornomaz, Veeresh Deshpande, Andreas Fuhrer, Richard J Warburton, Dominik M Zumbühl, and Andreas V Kuhlmann. Self-aligned gates for scalable silicon quantum computing. *Applied Physics Letters*, 118(10):104004, 2021. DOI: [10.1063/5.0036520](https://doi.org/10.1063/5.0036520).
- [33] Franck HL Koppens, Joshua A Folk, Jeroen M Elzerman, Ronald Hanson, LH Willems Van Beveren, Ivo T Vink, Hans-Peter Tranitz, Werner Wegscheider, Leo P Kouwenhoven, and Lieven MK Vandersypen. Control and detection of singlet-triplet mixing in a random nuclear field. *Science*, 309(5739):1346–1350, 2005. DOI: [10.1126/science.1113719](https://doi.org/10.1126/science.1113719).
- [34] Matthias Brauns, Joost Ridderbos, Ang Li, Erik PAM Bakkers, Wilfred G Van Der Wiel, and Floris A Zwanenburg. Anisotropic pauli spin blockade in hole quantum dots. *Physical Review B*, 94(4):041411, 2016. DOI: [10.1103/PhysRevB.94.041411](https://doi.org/10.1103/PhysRevB.94.041411).
- [35] J Danon and Yu V Nazarov. Pauli spin blockade in the presence of strong spin-orbit coupling. *Physical Review B*, 80(4):041301, 2009. DOI: [10.1103/PhysRevB.80.041301](https://doi.org/10.1103/PhysRevB.80.041301).
- [36] S Nadj-Perge, SM Frolov, JWW Van Tilburg, J Danon, Yu V Nazarov, R Algra, EPAM Bakkers, and LP Kouwenhoven. Disentangling the effects of spin-orbit and hyperfine interactions on spin blockade. *Physical Review B*, 81(20):201305, 2010. DOI: [10.1103/PhysRevB.81.201305](https://doi.org/10.1103/PhysRevB.81.201305).
- [37] Ruoyu Li, Fay E Hudson, Andrew S Dzurak, and Alexander R Hamilton. Pauli spin blockade of heavy holes in a silicon double quantum dot. *Nano Letters*, 15(11):7314–7318, 2015. DOI: [10.1021/acs.nanolett.5b02561](https://doi.org/10.1021/acs.nanolett.5b02561).
- [38] FNM Froning, MJ Rančić, B Hetényi, S Bosco, MK Rehmann, Ang Li, Erik PAM Bakkers, Floris Arnoud Zwanenburg, Daniel Loss, DM Zumbühl, et al. Strong spin-orbit interaction and g-factor renormalization of hole spins in Ge/Si nanowire quantum dots. *Physical Review Research*, 3(1):

- 013081, 2021. DOI: [10.1103/PhysRevResearch.3.013081](https://doi.org/10.1103/PhysRevResearch.3.013081).
- [39] TH Stoof and Yu V Nazarov. Time-dependent resonant tunneling via two discrete states. *Physical Review B*, 53(3):1050, 1996. DOI: [10.1103/PhysRevB.53.1050](https://doi.org/10.1103/PhysRevB.53.1050).
- [40] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. DOI: [10.1109/CVPR.2016.90](https://doi.org/10.1109/CVPR.2016.90).
- [41] TorchVision maintainers and contributors. Torchvision: Pytorch’s computer vision library. <https://github.com/pytorch/vision>, 2016.
- [42] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. DOI: [10.1109/5.726791](https://doi.org/10.1109/5.726791).
- [43] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. DOI: [10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980).
- [44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. DOI: [10.48550/arXiv.1912.01703](https://doi.org/10.48550/arXiv.1912.01703).

A Device dimensions

We use four different silicon FinFET devices with varying dimensions in this work. The dimensions that are varied are the length of the plunger gates L_P , the length of the barrier gate L_B , and the width of the fin W . We show a schematic of the top view of the device with the lengths and widths that are varied in Fig. 5. A top view SEM image of similar devices can be found in [32]. Table 2 shows the estimated dimensions for each device.

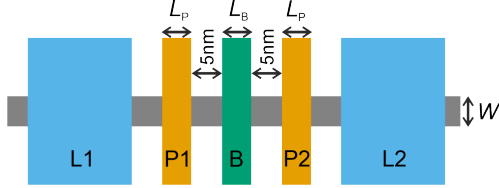


Figure 5: **Top view schematic of device layout.** Colors and labels are the same as in Figs. 1a, b.

	Device			
	i	ii	iii	iv
L_P [nm]	20	15	15	20
L_B [nm]	35	25	35	20
W [nm]	25	20	25	10

Table 2: **Estimated device dimensions for the different devices.** Illustration of the layout is given in Fig. 5. L_B gives the length of the gap between the plunger gates for device i as it has no barrier gate.

B Simulator

We consider the steady state current through a double quantum dot coupled to fermionic reservoirs. For an energy level E_A in the left dot and an energy level E_B in the right dot we define the

$$I = \sum_{E_A} \sum_{E_B} I_{\text{partial}} = \sum_i \sum_k \frac{(\Gamma_T^{(i,k)})^2 \Gamma_R^{(k)}}{(\Gamma_T^{(i,k)})^2 (2 + \Gamma_R^{(k)}/\Gamma_L^{(i)}) + (\Gamma_R^{(k)})^2/4 + \epsilon_{(i,k)}^2}. \quad (2)$$

Each level in the left dot $E_A^{(i)}$ is associated with a source tunneling rate $\Gamma_L^{(i)}$. Accordingly, each level in the right dot $E_B^{(k)}$ is associated with a drain tunneling rate $\Gamma_R^{(k)}$ and each pair of energy levels $(E_A^{(i)}, E_B^{(k)})$ is associated with an inter-dot

detuning as $\epsilon = E_A - E_B$. These energy levels are seen as the combined charge and spin states of an excess electron. We describe the simulator in terms of electrons but it holds true for holes as in the silicon FinFET used in the main part. We consider an electron to already occupy the right dot. The tunneling rates from the left reservoir (the source) and to the right reservoir (the drain) are Γ_L and Γ_R , and the tunneling rate between the dots is Γ_T . This situation is shown in Fig. 6a.

$$I_{\text{partial}} = \frac{\Gamma_T^2 \Gamma_R}{\Gamma_T^2 (2 + \Gamma_R/\Gamma_L) + \Gamma_R^2/4 + \epsilon^2}. \quad (1)$$

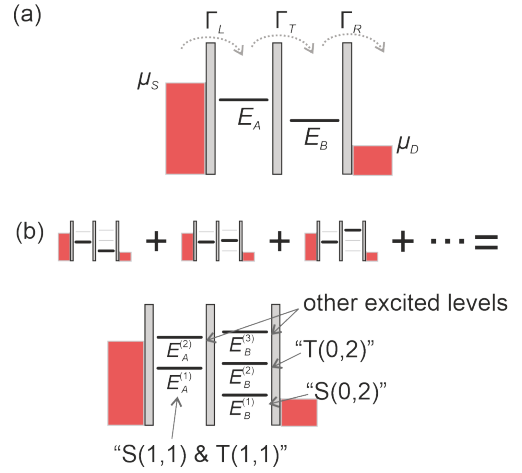


Figure 6: **Explanation of the simulator.** **a** Fundamental building block. If only one energy level is available in each dot, we can compute the current through a double quantum dot. **b** Simulation with several levels in each dot. We simply add the contribution of each pair according to 6a. If the DQD is in Pauli spin blockade, lower levels can be identified with singlet and triplet levels, higher levels are excited levels.

The total current consists of contributions due to different energy levels in the left and right dot, as illustrated in Fig. 6b. It is given by

tunneling rate $\Gamma_T^{(i,k)}$. This equation is only used if the lowest energy levels in each dot are in the bias window. Otherwise the DQD is in Coulomb blockade and current will be suppressed by setting I to zero.

Energy levels are computed by considering the gate voltage of each assigned plunger gate V_A, V_B and the associated lever arm L_A, L_B that translates voltage to an energy. Additionally, we introduce cross talk terms C_A, C_B which describe the influence of a plunger gate on the other quantum dot, i.e. the dot that the gate was not intended to be influenced. This way, the ground energy levels $E_A^{(0)}$ and $E_B^{(0)}$ can be computed as

$$E_A^{(0)} = L_A \cdot V_A + C_B \cdot V_B, \quad (3)$$

$$E_B^{(0)} = C_A \cdot V_A + L_B \cdot V_B. \quad (4)$$

Each energy level $E_A^{(i)}$ ($E_B^{(k)}$) for $i > 0$ ($k > 0$) is split from the previous energy level by an energy $E_{\text{split}}^{(i)}$ ($E_{\text{split}}^{(k)}$):

$$E_A^{(i)} = E_A^{(i-1)} + E_{\text{split}}^{(i)}, \quad (5)$$

$$E_B^{(k)} = E_B^{(k-1)} + E_{\text{split}}^{(k)}. \quad (6)$$

We apply white noise κ to each energy level with $\kappa \sim \mathcal{N}(1, \sigma^2)$, where σ is a sampled parameter. The indices $(i), (k)$ are omitted from now on for readability.

Thermal broadening of the triangles is taken into account by adding the thermal energy $k_B \mathcal{T}$ to source potential $\tilde{\mu}_S$ and drain potential $\tilde{\mu}_D$ with the Boltzmann constant k_B and temperature \mathcal{T} , $\mu_S = \tilde{\mu}_S + k_B \mathcal{T}$, and $\mu_D = \tilde{\mu}_D + k_B \mathcal{T}$. The tunnel rates are modified due to the effects of temperature as,

$$\Gamma_L = \tilde{\Gamma}_L f(E_A, \mu_S), \quad (7)$$

$$\Gamma_R = \tilde{\Gamma}_R [1 - f(E_B, \mu_D)], \quad (8)$$

where $\tilde{\Gamma}_L$ and $\tilde{\Gamma}_R$ are sampled parameters, and $f(\epsilon, \nu) = (1 + \exp(\frac{\epsilon - \nu}{k_B \mathcal{T}}))^{-1}$ is the Fermi-Dirac distribution. Equally, $\tilde{\Gamma}_T$ is sampled and then rectified to only allow for physically possible transitions in the bias direction:

$$\Gamma_T = \begin{cases} \tilde{\Gamma}_T, & \text{if } E_A - E_B \geq 0 \\ 0, & \text{else.} \end{cases} \quad (9)$$

Equations 1 to 9 produce one triangle of the pair we need. We can think of this as the cycle $(0, 1) \rightarrow (1, 1) \rightarrow (0, 2) \rightarrow (0, 1)$ with (m, n) indicating m electrons (holes) in the left dot and n electrons (holes) in the right dot. There is another cycle possible, namely $(1, 2) \rightarrow (1, 1) \rightarrow (0, 2) \rightarrow (1, 2)$. The bias triangle associated with

this cycle is shifted due to the electrostatic coupling energy between the dots E_{C_m} . This second triangle is simulated by shifting all energy levels by E_{C_m} and repeating the current simulation discussed above. If the bias triangles overlap, only the maximum current is used.

To mimic experimental observations and to create a diverse data set, we get a set of two bias triangles by randomly sampling the scalar parameters $\{L_A, L_B, C_A, C_B, \sigma, \tilde{\mu}_S, \tilde{\mu}_D, \mathcal{T}, E_{C_m}\}$, the vector parameters $\{\tilde{\Gamma}_L, \tilde{\Gamma}_R, \mathbf{E}_{\text{split}}\}$ and the elements of the matrix $\tilde{\Gamma}_T$. The dimensions of the vectors and the matrix depend on how many energy levels we consider in each dot, which is also sampled randomly. Overall, the simulator takes between 20 and 50 sampled parameters as input and generates a two dimensional charge stability diagram.

Finally, we add different types of noise to the measurements further to the noise already described. Gaussian blurring is used to smooth the edges of the triangles. The triangles can also move in voltage space due to charge switches or other drift effects. Charge switch noise at random points simulates both effects, see Figure 2b bottom left for an example. We also add white noise to the final current values at each simulated point.

To simulate the effect of PSB we add simple rules about where and how much current is allowed. The right lowest level represents the S(0,2) level and we neglect the small splitting of the left lowest energy level into S(1,1) and T(1,1), see Figure 6b, making it only one level. With PSB, tunneling between these two states is prohibited because an electron will eventually occupy the T(1,1) state and will not be able to tunnel to S(0,2). This leads to the rule:

For bias triangles in PSB, the tunneling rate between the two lowest energy levels is set to 0.

An electron that is stuck in T(1,1) also blocks all other paths through the double dot. Therefore, a second rule is introduced:

We suppress all current at a given point in voltage space if the only available energy levels in the bias window are the lowest ones.

For more details, the code for the simulator is available on our GitHub repository.

C Neural Network Architectures

We experimented with two neural network architectures: a slightly modified version of ResNet18 and a custom LeNet5-like model. Both models were adapted to accept input images with two channels and to produce outputs for two classes. In this section, we briefly describe the architectures and modifications. For details see our GitHub repository.

C.1 Modified ResNet18

The modified ResNet18 architecture is based on the original ResNet18 model [40], with changes to the input and output layers. The first convolutional layer is altered to accept two input channels instead of three, while the final fully connected layer is changed to output two classes instead of 1000. With these changes, the model has 11,174,402 learnable parameters. We use the TorchVision [41] implementation of this model. All results in this paper are based on models with this architecture unless otherwise explicitly stated.

C.2 Modified LeNet5

The ResNet18 architectures could be considered too large compared to the size of the data set. We therefore consider a much smaller neural network. Our modified LeNet5 model [42] consists of two convolutional layers, each followed by batch normalization, a ReLU activation, and max-pooling. The feature maps are then flattened and passed through three fully connected layers, with ReLU activations after the first and second layers. This leads to 942,500 learnable parameters, which is about an order of magnitude smaller compared to the ResNet18 architecture.

D Details of the training procedure

Each training run consists of 50,000 pairs and 100 epochs. We use mini-batches of size 128.

We use the Adam optimizer [43] with a regularisation factor of 0.001. The optimiser is initialised with a learning rate of 0.001. The learning rate is then decayed by a factor of 0.1 with a scheduler once a plateau in the training loss is reached, with a patience of 5 epochs. We used cross-entropy loss as the objective function, with

balanced class weights to account for class imbalance in the training data.

To sample the 50,000 examples for the case where we only use simulated data, we sample 25,000 and augment each image twice.

When using only experimental data, we augment the available training data (see Table 1) until we have 50,000 examples. In the case of mixed data, we sample 12,500 examples from the simulator, augment them twice, and then augment the available experimental training data until we have 25,000 examples which gives us 50,000 examples in total. To counteract class imbalance, i.e. examples with and without PSB, we also weight the classes according to their prevalence in the loss function.

We add random contrast and brightness to all training data and then crop them randomly. Experimental data is additionally randomly sheared and stretched along both axes. Every pair is normalised between 0 and 1.

Testing data, i.e. the corresponding fold in the cross-validation procedure, is not augmented but only normalised. The neural network and its training is implemented in PyTorch [44].

E Influence of the number of simulated pairs

We investigate the role of the number of simulated pairs when training with only simulated data. We simulate 10, 100, 1,000, or 10,000 pairs and augment them until we have 50,000 pairs. The accuracy for classifiers trained on those data sets in comparison with the sample size used in the main text (25,000 sampled pairs) is shown in Fig. 7.

The results show that the main driver of accuracy is the size of the sample set. The number of pairs when training with real data is between 31 and 48, depending on the fold, reaching about 85% mean accuracy. In comparison, 100 simulated pairs lead to a similar mean accuracy but the spread of accuracies is much larger.

F Results from the LeNet5 architectures

We repeat the benchmarking as in Section 2 with the smaller LeNet architecture described in Ap-

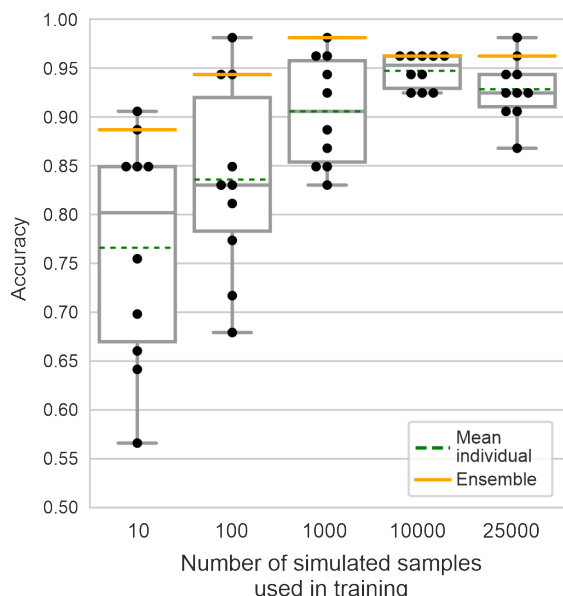


Figure 7: **Varying simulated sample size.** We show the accuracy for varying sizes of the simulated data set. In all cases the simulated pairs were augmented until 50,000 total pairs are created. 25,000 pairs in the training set corresponds to the analysis shown in the main text in Fig 4.

pendix C.2. The results are shown in Fig. 8. The performance in the case **Exp** is improved significantly, reaching an ensemble accuracy of 90.6%. This indicates that the size of the ResNet18 architecture leads to over-fitting. However, given the size of our dataset, we are limited in the creation of a holdout dataset to report generalization capability and to tune the hyperparameters of the training without risking overfitting (see Appendix G). Still, the ensemble trained in the **Mix** setting is outperforming the **Sim** and **Exp** case, showcasing the usefulness of the simulator. The ensemble classifier of **Mix** reaches an accuracy of 98.1% and an AUC of 1.0.

We refrained from optimizing any hyperparameters during the training stage, in an effort to minimize the risk of overfitting. This includes the choice of architecture.

G Testing on simulated data

To further judge the generalisation performance of the models, we test them on new simulated data. We simulate 1000 pairs of stability diagrams. For **Exp** and **Mix** we use models that were trained on all available data, i.e. we do not separate data into folds.

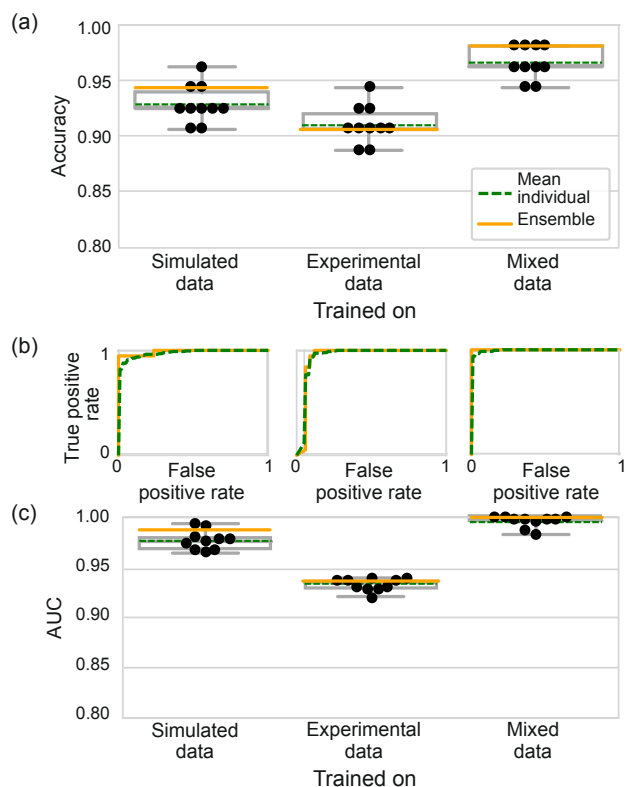


Figure 8: **Benchmarking results with LeNet architecture.** This is the same analysis as in Fig. 4 but with a different neural network architecture. **a** Accuracy. Box plots with results from individual classifiers plotted as dots, the mean of those is plotted as a dashed green line, and the performance of the ensemble of classifiers as a solid orange line. **b** Receiver operating characteristic (ROC). For each case, the ensemble ROC is plotted as a solid orange line and the mean individual ROC as a dashed green line. **c** Area under the curve (AUC). Legend as in Fig. 8a.

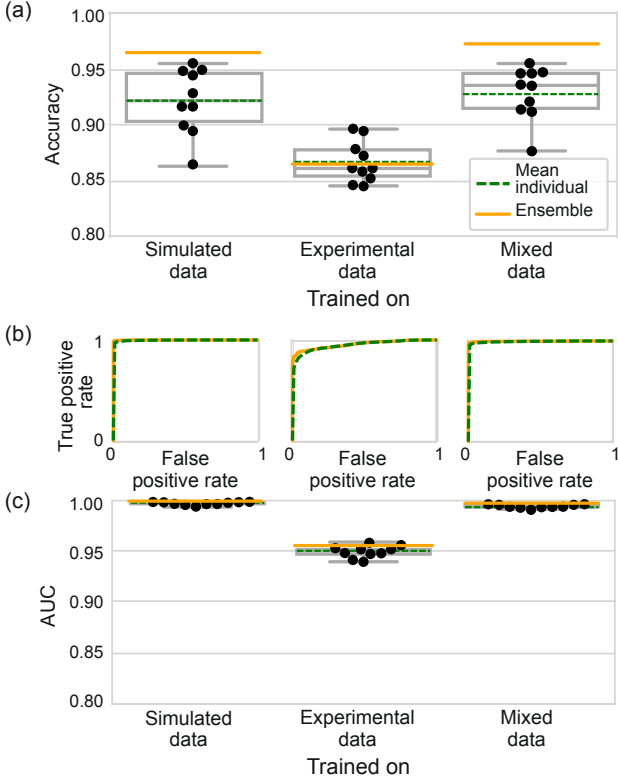


Figure 9: **Benchmarking results on simulated data with LeNet architecture.** Description as in Fig. 8 but for testing on simulated data.

The results are similar to those found testing on experimental data but with notable changes when comparing the models with different sizes. Fig. 9 shows the performance of the smaller LeNet models and Fig. 10 of the larger ResNet models. With an ensemble accuracy of 86.5% and an AUC of 0.956, the smaller LeNet model is performing very similarly to the larger ResNet model in the **Exp** case with an ensemble accuracy of 87.5% and an AUC of 0.943. For the other cases, the larger ResNet (**Sim**: ensemble accuracy 98.4% and AUC 1.0; **Mix**: ensemble accuracy 98.5% and AUC 1.0) performs better than the LeNet (**Sim**: ensemble accuracy 97.3% and AUC 0.997; **Mix**: ensemble accuracy 97.3% and AUC 0.997) which hints at a greater generalisation capability of the larger network when trained with sufficiently diverse data.

H More classification results

Fig. 11 shows more classification results that correspond to section 2 in the main text. Fig. 11a shows the stability diagrams with the identified bias triangles as white boxes and Fig. 11b shows the predictions of the classifier, which predicted

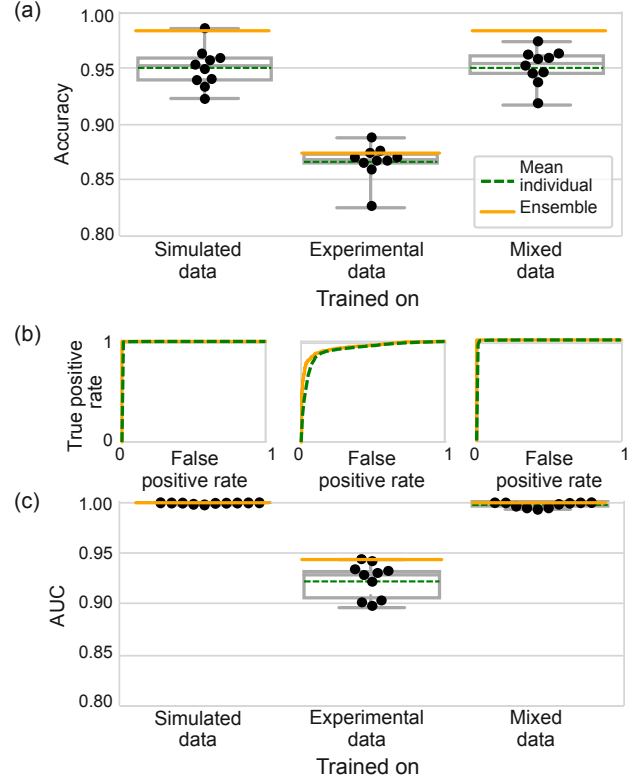


Figure 10: **Benchmarking results on simulated data with ResNet architecture.** Description as in Fig. 8 but for testing on simulated data.

no PSB for charge transitions D-F.

We repeat the same experiment for reversed bias. Fig. 11c shows the stability diagram with identified bias triangles. We call them G-L to distinguish them from the measurements in the main text even though they correspond to charge transitions A-F.

Here, all bias triangles were classified as not having PSB, shown in Fig. 11d.

We show all experimental data used for the training and testing of the classifier in Fig. 12. We split them according to device source (row) and whether they show signs of PSB or not (column). Data from device iv corresponds to charge transitions A, B, C, D, and H with a low magnetic field of $B = -0.04$ T (see Fig. 3 and Fig. 11). We show three ensemble classifier scores (one for each type of training data) for each example. In the left column, we expect a perfect classifier to predict PSB (which would mean a full bar), in the right column it would predict no PSB (which would mean a missing bar).

This shows which examples were hard for the classifiers and it also shows the diversity of examples the classifier needed to deal with.

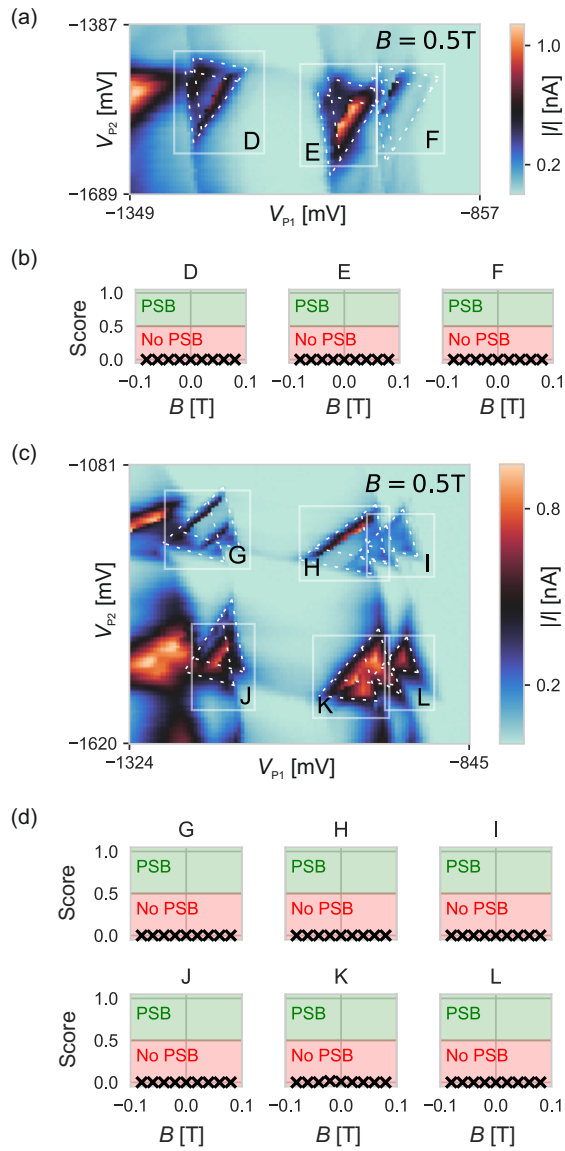


Figure 11: **More classification results.** **a** Stability diagram for more bias triangles. **b** Classification results corresponding to charge transitions in Fig. 11a. **c** Stability diagram for reversed bias. **d** Classification results in the reversed bias case. As in Figs. 1 and 3, the absolute value of current is shown and dashed white lines outline the bias triangles in Fig. 11a and Fig. 11c.

Some bias triangles show only weak signs of PSB, such as the ones from device iii. Even though we can not be sure from those measurements that this is indeed due to PSB and not due to other effects, e.g. orbital effects, we still label those bias triangles as showing PSB. The point is that the signature could be due to PSB and should therefore be caught by a classifier. Confirmation measurements such as the ones done in Fig. 3d need to be performed after that to verify the prediction.

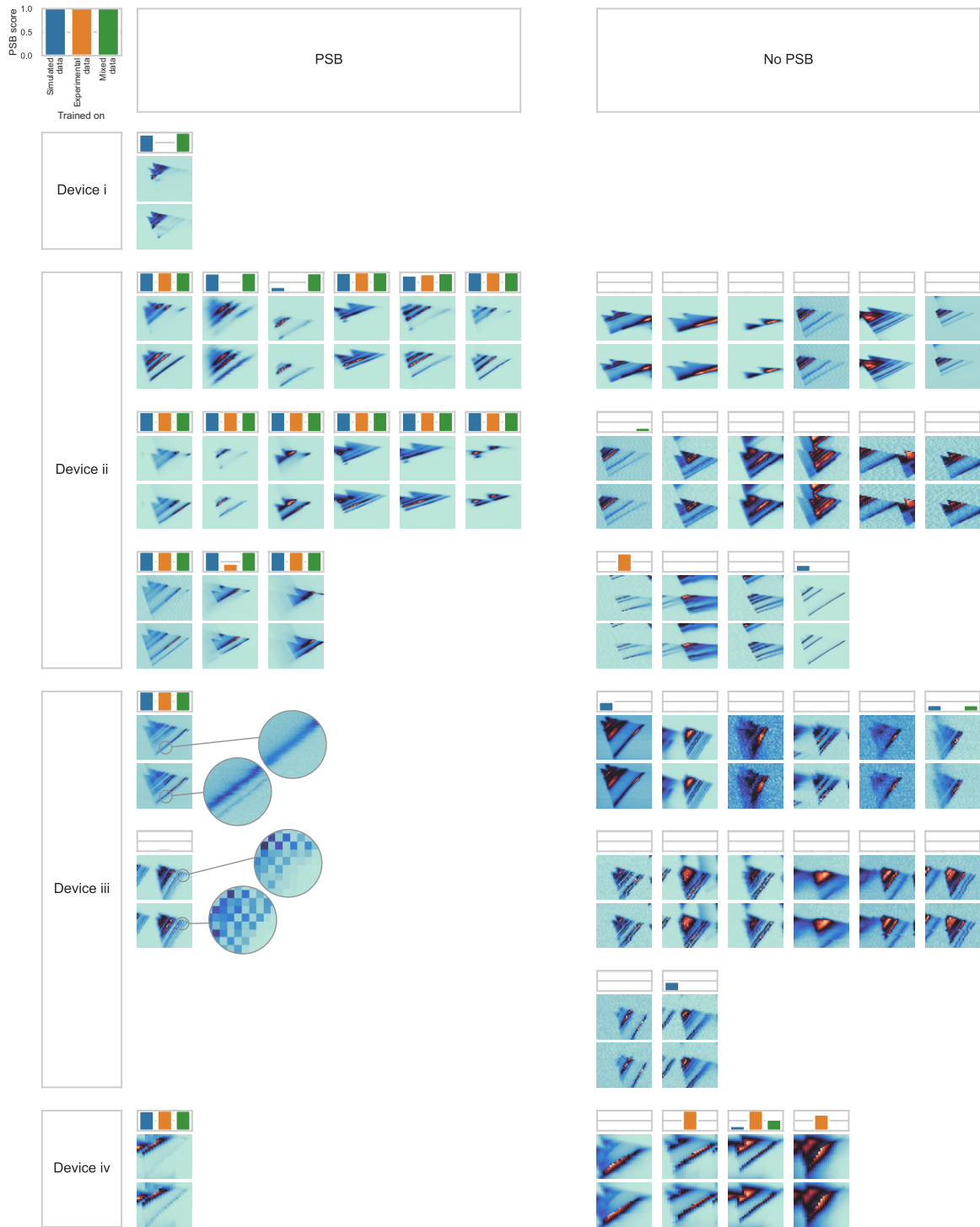


Figure 12: **All experimental data used for benchmarking and the prediction of the ensembles of classifiers.** There are two main columns (sets with and sets without PSB) and four main rows (devices i to iv). The measurements shown in the left column (marked “PSB”) are not definitive proof that PSB is present but show the signs we expect and want to detect with a classifier. In the top left corner there is an example prediction plot with all labels: We show ensemble scores for the case of training on simulated data (blue), experimental data (orange), or a mix of both (green). The prediction threshold is plotted in the background as a horizontal line at a score of 0.5. Each set consists of a current measurement with low or zero magnetic field (top) and with a large magnetic field (bottom) and is jointly normalised between 0 and 1. Above each set the score plot is shown for that set without labels. The sets with PSB of device iii additionally show a magnification of a part of the base of the triangles due to visibility.