Fully autonomous tuning of a spin qubit

Jonas Schuff,[1,][*] Miguel J. Carballido,[2] Madeleine Kotzagiannidis,[3] Juan Carlos Calvo,[3] Marco Caselli,[3] Jacob Rawling,[3] David L. Craig,[1] Barnaby van Straaten,[1] Brandon Severin,[1] Federico Fedele,[4] Simon Svab,[2] Pierre Chevalier Kwon,[2] Rafael S. Eggli,[2] Taras Patlatiuk,[2] Nathan Korda,[3] Dominik Zumbühl,[2] and Natalia Ares[4,][**]

[1]*Department of Materials, University of Oxford, Oxford OX1 3PH, United Kingdom*
[2]*Department of Physics, University of Basel, 4056 Basel, Switzerland*
[3]*Mind Foundry Ltd, Summertown, Oxford OX2 7DD, United Kingdom*
[4]*Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, United Kingdom*
(Dated: February 7, 2024)

Spanning over two decades, the study of qubits in semiconductors for quantum computing has yielded significant breakthroughs [1–6]. However, the development of large-scale semiconductor quantum circuits is still limited by challenges in efficiently tuning and operating these circuits. Identifying optimal operating conditions for these qubits is complex, involving the exploration of vast parameter spaces [7]. This presents a real 'needle in the haystack' problem, which, until now, has resisted complete automation due to device variability and fabrication imperfections [8, 9]. In this study, we present the first fully autonomous tuning of a semiconductor qubit, from a grounded device to Rabi oscillations, a clear indication of successful qubit operation. We demonstrate this automation, achieved without human intervention, in a Ge/Si core/shell nanowire device. Our approach integrates deep learning, Bayesian optimization, and computer vision techniques. We expect this automation algorithm to apply to a wide range of semiconductor qubit devices, allowing for statistical studies of qubit quality metrics. As a demonstration of the potential of full automation, we characterise how the Rabi frequency and *g*-factor depend on barrier gate voltages for one of the qubits found by the algorithm. Twenty years after the initial demonstrations of spin qubit operation, this significant advancement is poised to finally catalyze the operation of large, previously unexplored quantum circuits.

Recent advances underscore the potential of qubits in semiconductors for universal quantum computation [4–6, 10–14]. These include the achievement of two-qubit gates showcasing fidelities that surpass thresholds essential for fault-tolerant computing [4, 15, 16], and hot qubits that address the bottleneck of millikelvin refrigeration [12, 17, 18]. Strides in wafer-scale manufacturing of these devices [7, 19] and their efficient testing at cryogenic temperatures [20, 21] further highlight the rapid progress in this field. Still, semiconductor quantum circuits are limited to at most six qubits [22] in one device. This stands in stark contrast to the potential afforded by modern semiconductor fabrication techniques, which could support the integration of millions of qubits.

One of the reasons for this contrast is that a long-standing challenge remains: the intricate tuning required to reach and maintain qubit operation. Previous works have introduced diverse approaches for automating single stages of this process, such as defining double quantum dot (DQD) confinement potentials [9, 23–25], navigating to specific charge transitions [26–33], fine-tuning of charge transport features [34] or the inter-dot tunnel couplings [35, 36], as well as device characterisation [37–39]. Offering glimpses of the potential of machine learning for full qubit tuning automation, these works left the challenge unaddressed.

Here, we present a fully autonomous tuning process,

able to encode a qubit without the need for human intervention. The process of going from a fully de-energized device to the observation of Rabi oscillations, a definitive indicator of qubit functionality, usually takes human experts weeks, or even months, to complete. Our algorithm, deployed on a DQD device, can complete the tuning process within three days.

Our success in moving away from the manual tuning of semiconductor qubits marks a paradigm shift in quantum device scalability. Key to this success is the algorithm's ability to navigate through various stages of the tuning process, efficiently handling challenges and making accurate decisions. Our findings, underpinned by deep learning, Bayesian optimisation, and computer vision techniques, would finally allow for the operation and characterisation of complex, large-scale semiconductor qubit circuits.

DEVICE ARCHITECTURE AND READOUT TECHNIQUE

We consider a common layout for a DQD device (inset of Fig. 1b). We use a Ge/Si core/shell nanowire device in which holes are confined in depletion mode [40]. The electrical potential is set by a number of gate electrodes. Two plunger gate electrodes predominantly shift the electrochemical potential in the left and right dots with voltages $V_{\rm LP}$ and $V_{\rm RP}$. The rest of the gate electrodes primarily control the barriers between the DQD and the leads as well as the inter-dot coupling. One of the plunger gates is

---

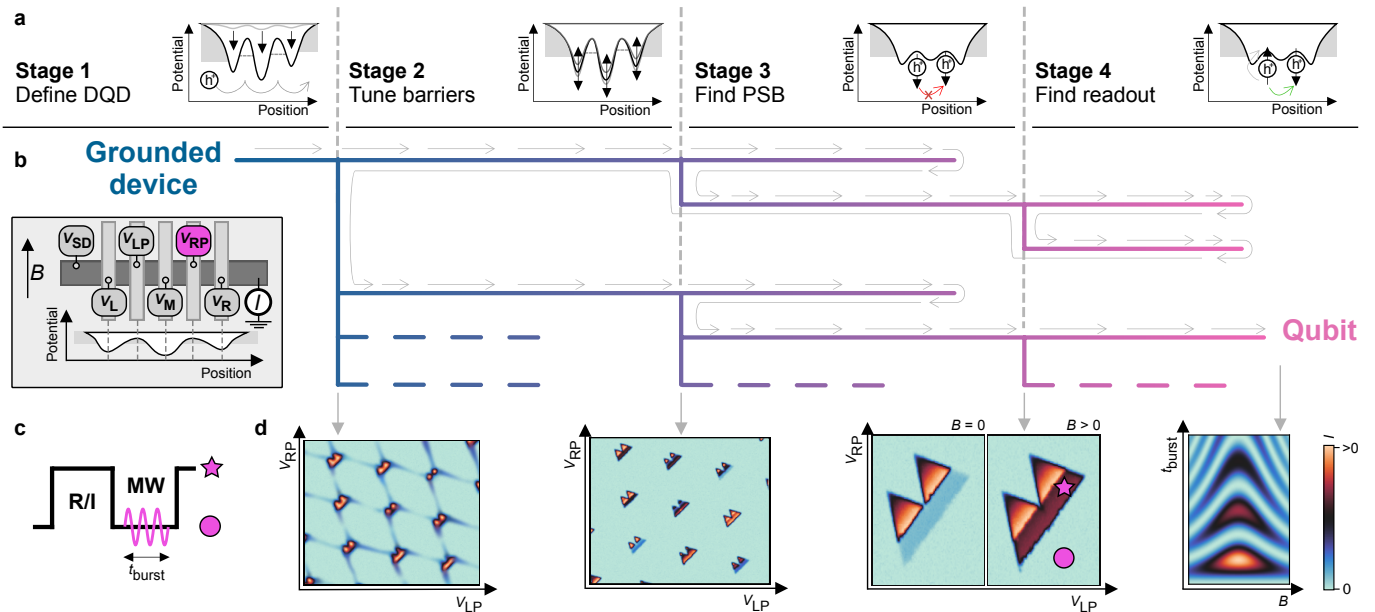[*] jonas.schuff@materials.ox.ac.uk
[**] natalia.ares@eng.ox.ac.uk

FIG. 1: **Algorithm overview. a,** There are four stages that the algorithm needs to successfully navigate to reach qubit operations. The goal of each stage is illustrated with a confinement potential diagram. **b,** Each stage can either be successful and produce candidates (leading to one or more branches), or unsuccessful (leading to a backtracking to the closest stage that still has candidates). The search is therefore conducted in a tree structure. Some branches of the tree might be left unexplored. This is indicated by dashed lines. Inset: Illustration of the device. Five different voltages V_i can be applied to a linear confinement. **c,** A pulsing scheme can be applied with a microwave burst of length t_{burst}. The fast line is connected to plunger gate RP. **d,** Simulated measurement illustrations that mark the successful transition between stages. Starting from a stability diagram with mere DQD features (far left), the algorithm tunes parameters until promising bias triangles (middle left), triangles exhibiting PSB (middle right) and finally Rabi oscillations (far right) are obtained. The extremal points of the pulse scheme are indicated as a star and circle in the middle right illustration.

connected to a high-frequency line via a bias-tee, allowing for the application of voltage pulses and microwave bursts.

The device can be probed by applying a bias voltage V_{SD} to the source lead and recording the current I_{SD} at the drain lead. The algorithm navigates to a DQD occupation that exhibits Pauli spin blockade (PSB) for spin-to-charge conversion. To achieve this, the DQD does not need to be depleted to the single hole regime. The charge occupation on each dot is estimated to be in the range of several dozens [41–44].

The algorithm uses a two-stage pulsing protocol [18, 45–47] which is parameterised by a microwave (MW) pulse frequency and a duration t_{burst} (Fig. 1c). This protocol allows for qubit manipulation if the spin resonance conditions are met. Details on the pulsing scheme for coherent spin control and the device are described in Methods and by Carballido *et al.* [40].

THE ALGORITHM

The autonomous tuning algorithm is structured into four main stages. Starting from a completely de-energized device, i.e. with all gate voltages set to 0, the first two stages define the DQD potential by tuning the inter-dot barrier and the reservoir coupling. The third stage narrows the search space by looking for distinct signatures

of PSB, an initialisation and readout requirement. The last stage fine tunes the plunger voltages and finds the frequency and duration of a microwave pulse needed to drive the qubit. The effect of each stage on the DQD confinement potential is illustrated in Fig. 1a. Measurement illustrations exemplifying those taken by the algorithm are shown in Fig. 1d.

As a result of the algorithm design a search tree emerges, as shown in Fig. 1b. Once a stage is successfully completed, a list of candidates is generated. A candidate consists of all information needed for the next stage to investigate it, usually containing locations or ranges of gate voltages, or information on the suspected g-factor and Rabi frequency f_{Rabi}. The candidates are ordered by a dedicated score in each stage and a single candidate is passed on to the next stage. If a stage is unsuccessful, the algorithm backtracks to the previous stage and investigates the next candidate in that stage's list of candidates. This process dynamically creates a search tree. If a different branch has proven to lead to a qubit, some branches of the tree may be left unexplored. These are indicated by dashed lines in the tree in Fig. 1.

We describe each stage in this section. Details on the stage structures, substages, and composition of candidates for each stage can be found in Methods and the Supplementary Material.
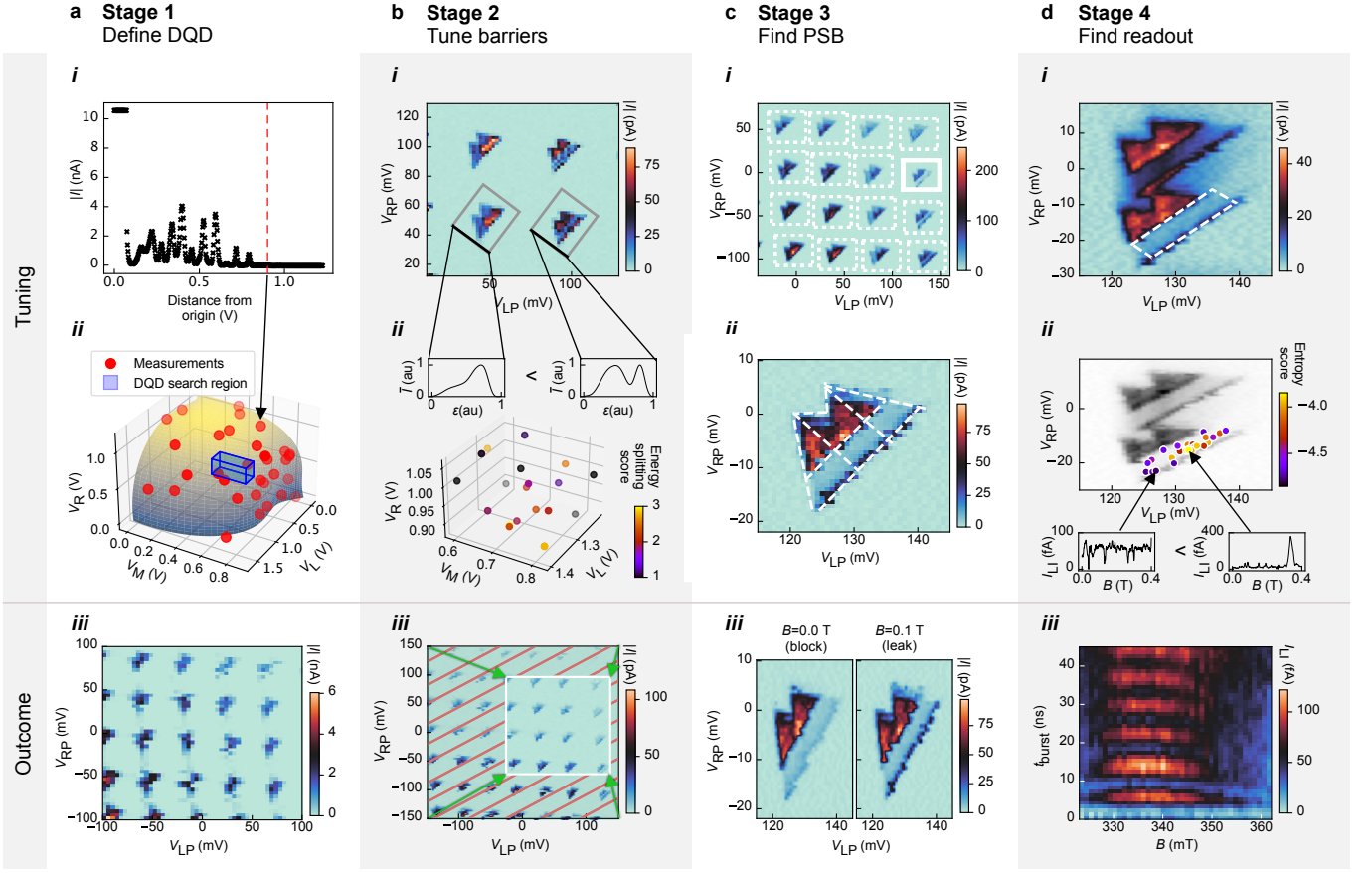
FIG. 2: Summary of each stage. a, Stage 1: Definition of a DQD potential. **a-*i*** Current measurements along random directions in gate voltage space to determine the points at which conducting and non-conducting regions meet, so-called pinch-off points. **a-*ii*** Gaussian processes model the hypersurface after collecting sufficient pinch-off points. **a-*iii*** A neural network confirms the presence of a DQD analysing the aquired stability diagrams. **b,** Stage 2: Optimisation of bias triangle features. **b-*i*** Analysis of stability diagrams, segregating individual transitions and averaging them along segments orthogonal to the detuning axes. **b-*ii*** Distribution of optimisation scores and averages of bias triangles along segments orthogonal to their detuning axes. We aim to increase the singlet-triplet energy splitting via a proxy score that measures the dip of current between the baselines and the rest of the triangles. **b-*iii*** Identification of plunger voltage windows unaffected by charge switches aided by neural networks. **c,** Stage 3: Finding PSB **c-*i*** Initial low-resolution, wide-range detection of PSB using neural networks. **c-*ii*** Detailed high-resolution scans employing segmentation algorithms. **c-*iii*** Bias triangles that show signatures of PSB and can be used to optimise readout. **d,** Stage 4: Readout spot identification within a promising transition **d-*i*** Acquisition of stability diagrams pulsing gate RP to locate the readout region (indicated by a white dashed box). **d-*ii*** Entropy-based scoring of magnetic field traces within the readout region, optimized through Bayesian methods. **d-*iii*** Rabi oscillations for different magnetic fields around the resonance condition to confirm qubit operation. The measurements marked with I_{LI} were amplified with a lock-in amplifier, see Methods for details.

Stage 1: Define DQD

The first stage identifies the gate voltage settings that define the DQD confinement potential. It determines a lower and upper limit for each barrier gate voltage, which is used in subsequent stages.

Building upon the methodologies of Moon *et al.* [9] and Severin *et al.* [24], a hypersurface model is created to distinguish between conducting and non-conducting regions within the three-dimensional barrier gate voltage space. The algorithm takes current measurements along random directions within this space (Fig. 2 a-*i*), and models the hypersurface with a Gaussian process, as depicted in Fig. 2 a-*ii*. We expect a DQD potential

forming near a corner of the hypersurface in the first octant. To pinpoint this corner, three specific current measurements are conducted using only one of the gate electrodes at a time. The resulting coordinates are then projected onto the model of the hypersurface, setting the lower bounds of the region where DQDs are likely to be found. The upper bounds of the region are given by the coordinates of the single gate pinch-off voltages, i.e. when the current drops to a value that is indistinguishable from the noise floor. The resulting box is labelled as 'DQD search region' in Fig. 2 a-*ii*.

A methodical search within this box is conducted by the algorithm. The algorithm samples locations in the DQD search region and investigates them, starting from the

point nearest to the projected corner and progressing to higher gate voltages. At each location a one-dimensional trace of the plunger gate voltages is taken and checked for Coulomb peaks, a signature for quantised charge transport and a first requirement for a DQD. If Coulomb peaks are found, a measurement varying both plunger gate voltages, a so-called stability diagram, is acquired and analyzed via a neural network for DQD characteristics; see Fig. 2 a-*iii* for a stability diagram that shows the desired features. Successful identification of DQD features at a location in gate voltage space establishes it as a lower limit for the subsequent search.

Stage 2: Tune barriers

The goal in the second stage is to adjust the tunnel barriers to enhance the singlet-triplet energy splitting, an important requirement for qubit operation. Additionally, this stage needs to avoid regions of gate voltage space with following characteristics: regions with high currents above a generous threshold; regions that are susceptible to charge switch noise; regions that show co-tunneling lines.

Within the gate voltage bounds established in the first stage, Bayesian optimisation is employed to search gate voltage combinations. The figure of merit for this optimisation is based on the degree to which it reduces current between the triangle baseline and excited states, compared to the current throughout the rest of the bias triangles. This acts as an easy-to-compute proxy score of the singlet-triplet energy splitting. After Bayesian optimisation suggests a voltage setting, a stability diagram is measured (Fig. 2 b-*i*). We first use a segmentation algorithm [48] to find the outlines of individual pairs of bias triangles. The score is then computed using averages of the stability diagram along the common baseline, as illustrated in Fig. 2 b-*ii*. A neural network identifies bias triangles that are impacted by charge switches. Charge switches distort the stability diagrams and make the area unsuitable for qubit operation; see the hatched area in Fig. 2b-*iii* or Extended Data Fig. 6 for examples. These bias triangles are excluded from the optimisation. The gate voltage regions explored by the optimisation are shown in Fig. 2 b-*ii*.

A significant challenge of this stage is the need for numerous stability diagrams, which are time-intensive to measure. To address this, we implement an adaptive, efficient measurement algorithm designed to specifically focus on gate voltage regions where bias triangles are present, see Supplementary Material for details. Employing this method cuts down the measurement time by approximately two-thirds. The optimisation is performed using these efficient measurements.

As a final step, stability diagrams without the efficient measurement algorithm are taken in the most promising regions. This is done to ensure there are no charge switches, because the previous step ranked each location by the highest score of a bias triangle at that location. Therefore, some stability diagrams may have regions affected by charge switches. Bounding boxes are then created in the plunger voltage space, encompassing primarily stable bias triangles with current below the previously mentioned threshold (Fig. 2 b-*iii*). These triangles are each evaluated and scored, and the bounding boxes are ranked based on the highest score they contain.

The algorithm has up to this point only used a positive bias voltage. This stage proposes both positive and negative bias voltages for each candidate it creates. The gate voltage coordinates including the bias voltage are passed on to the subsequent stage.

Stage 3: Find PSB

In this stage, our algorithm searches for charge transitions exhibiting PSB, a necessary condition for qubit initialisation and readout in this setup. A candidate has to pass three different classifiers to be judged as exhibiting PSB. This is necessary to avoid false positives entering the time-intensive last stage.

We begin with low-resolution stability diagrams of bias triangle candidates, both with $B = 0$ T and $B = 0.1$ T. In these devices, PSB is expressesd as a suppressed baseline of the bias triangles in low magnetic field compared to high magnetic fields (Fig. 2 c-*iii*).

We use a routine based on the auto-correlation of the stability diagram to pinpoint bias triangle locations, see Methods for details. The identified triangles are then analyzed using a neural network [39](Fig. 2 c-*i*). Subsequently, the algorithm measures a stability diagram to precisely delineate the bias triangle, followed by high-resolution stability diagrams with $B = 0$ T and $B = 0.1$ T.

The algorithm invokes a routine to segment the bias triangles [48], illustrated in Fig. 2 c-*ii*. A further PSB classification based on the segmentation is performed. The routine from [48] defines the direction in plunger voltage space that controls the detuning ε of the dot energies, known as the detuning axis of the bias triangles. Scanning along this line with varying magnetic fields, we expect to observe a current drop at the baseline at zero magnetic field, which another classifier detects. Upon meeting all criteria, the gate voltage coordinates are forwarded to the next stage.

Stage 4: Find readout

The final stage of the process is dedicated to finding an operating point for qubit readout and manipulation. This stage not only identifies a suitable location in plunger voltage space but also determines the optimal driving frequency and duration of the pulse.

Based on the segmentation from the previous stage, a predicted readout gate voltage region within the bias triangle is defined (Fig. 2 d-*i*). The algorithm optimizes
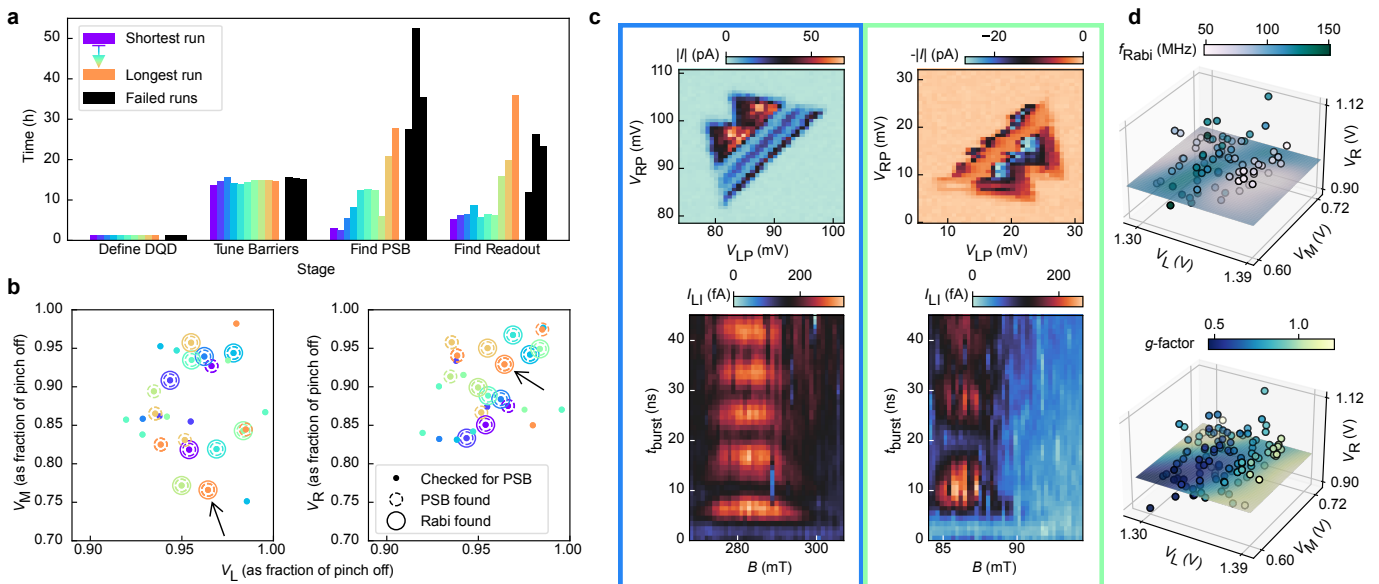
FIG. 3: Benchmarking. a, The duration of each stage for ten successful runs is shown, sorted by the total time taken to tune a qubit in operation. In black: the three runs that did not lead to a qubit. **b,** Locations in the barrier voltage space (as a fraction of the individual pinch-off voltage) is shown. In some locations, no PSB was identified. The qubit locations are a subset of locations that exhibit PSB. **c,** Examples of transitions from two different runs, showing the associated bias triangles (upper row) and Rabi chevrons (lower row). **d,** Characterisation. Varying the barrier gate voltages, we can build a map of f_{Rabi} (top row) and the g-factor (bottom row) of one of the found qubits (marked with arrows in b) using automated measurements. For illustrative purposes we show slices of a Gaussian Process model that was fitted to the data.

across the four-dimensional space of the two plunger gate voltages, the driving frequencies and pulse durations. It samples a point within this space and then measures the current as a function of the magnetic field (Fig. 2 d-*ii*). The optimisation's goal is to identify a current peak in these scans, indicative of the qubit's resonance condition (Fig. 2 d-*ii*, lower right plot). This is achieved by evaluating the entropy of current traces; traces exhibiting a peak correspond to lower entropy values, see Methods for details.

Bayesian optimisation then proposes potential candidates for further analysis. These candidates are filtered based on the presence of one or two peaks (corresponding to the number of qubits addressed), as determined by a simple peak-finding algorithm [49]. Noisy measurements might also show peaks. Therefore, a follow-up step involves retaking the measurement to confirm the presence of a peak.

Once a candidate is verified, several measurements are taken by the algorithm to establish the qubit's operational functionality. These include a spectroscopy measurement, a Rabi chevron experiment (Fig. 2 d-*iii*), and repeated high resolution Rabi oscillations at the Larmor frequency.

Hyperparameters

Each stage requires a set of hyperparameters. They control various aspects of the measurements such as: resolution and safe gate voltage ranges for stability diagrams;

aspects of the signal processing algorithms such as the required prominence of peaks; and steering parameters such as the number of candidates that a stage can suggest. The measurement aspects can be derived from some weak prior knowledge about device such as the magnitude of lever arms which informs the resolution of stability diagrams. We also assume a g-factor larger than 0.5 (limiting B and f_{MW}) and a f_{Rabi} between 30 MHz and 250 MHz (limiting t_{burst}). The requirements on the prior knowledge can be easily softened by widening the search space.

The set of hyperparameters influences the length of the runs and the way the algorithm manages trade-offs between exploration and exploitation. Regardless of the hyperparameters, the algorithm will always terminate once all candidates of each stage have been exhausted.

The choice of hyperparameter was made during development and not optimised for total run time or efficiency. We provide a full list of all hyperparameters in the Supplementary Material.

Characterisation

After a qubit has been found, we may choose to study the qubit further. Routines from Stages 3 and 4 enable the tracking of a known readout spot and the resonance condition in gate voltage space. This allows for the recording the dependency of qubit metrics as we change the confinement potential. We provide details of this characterisation algorithm in Methods.

RESULTS

We fixed the hyperparameters and gathered 13 runs. Rabi oscillations were found in ten of those runs. In successful runs, the time spent in each stage varied (Fig. 3a). total time required ranged from 22 to 80 hours, with a mean of 38 hours (median 34 hours) (Table I). Each stage relies on the exploration and accuracy of the previous stage. The variation in the time required in each stage gets progressively larger. Almost all time is spent on measuring the device, not on the decision algorithms. This is due to the measurement of current through the DQD which requires long integration times. A setup that allows for fast measurements via, e.g. radiofrequency reflectometry could be tuned orders of magnitudes faster.

Run #	1	2	3	4	5	6	7	8	9	10
Time (h)	22.8	24.5	28.5	32.0	33.3	34.4	34.5	37.8	56.5	79.2
f_{Rabi} (MHz)	31 ±1	97 ±2	109 ±1	47 ±1	51 ±1	90 ±1	56 ±1	115 ±3	49 ±1	63 ±1
g	1.52 ±0.07	0.72 ±0.05	0.70 ±0.05	2.20 ±0.12	2.74 ±0.22	0.73 ±0.04	2.31 ±0.14	0.67 ±0.08	2.12 ±0.14	0.74 ±0.03
V_{L} (V)	1.34	1.34	1.34	1.37	1.36	1.35	1.35	1.32	1.35	1.38
V_{M} (V)	0.69	0.81	0.79	0.80	0.69	0.78	0.79	0.77	0.65	0.71
V_{R} (V)	0.91	1.02	0.95	1.01	1.04	0.94	0.95	0.89	1.00	1.02

TABLE I: **Metrics of successful runs.** For each qubit found, we show the total time it took, the Rabi frequency f_{Rabi}, the g-factor, and the settings for the barrier gates V_{L}, V_{M} and V_{R}. The errors of f_{Rabi} are estimated from the fit uncertainty and the errors of g are calculated from the width of the resonance peak.

The three failed runs terminated after their exploration on a similar timescale to the longest successful runs. They took between 56.0 h and 94.9 h to complete and are shown in black in Fig. 3a. The runs found transitions that showed Pauli spin blockade but the algorithm was not able to find Rabi oscillations, possibly due to unfortunate settings of the tunnel barrier strengths in Stage 2.

The qubit locations in gate voltage space in terms of the three barrier gate voltages are depicted in Fig. 3b. For comparability, we normalise each voltage by the voltage at which each barrier gate electrode pinches off the current individually. At each point, Stage 2 (Tune barriers) passed a candidate for further analysis (solid dots). In some cases, PSB was detected (dashed circles), passing Stage 3, and a subset of these also yielded a qubit (solid circles), successfully completing Stage 4.

Fig. 3c presents examples from two runs, showing the transitions found and Rabi chevron measurements. The discovery of qubits in both bias directions evidences the algorithm's adaptability and its non-specificity to certain transitions. Both Rabi chevrons were obtained using the same given driving power and driving frequency, but varied in magnetic fields and Rabi frequencies, highlighting the algorithm's generalization capability. All depicted measurements were autonomously executed by the algorithm, accounting for the non-centered chevron measurements.

Analyzing the locations in gate voltage space where qubits were found provides insight into the device physics (Table I). By fitting a convex hull around the qubit locations in the barrier gate voltage space, we can estimate the volume of the region where qubits can be found. For this device, the volume of the convex hull is approximately $3.5 \times 10^{-4} \text{V}^3$, translating to a fraction of the safe ranges $[(2\text{V})^3]$ of about 4×10^{-5}. The space is further restricted by the plunger voltage location, which is a box of roughly $(10\text{mV})^2$. Given a search space of $(300\text{mV})^2$, this brings down the size of the volume to around 2×10^{-7} as a fraction of the 5-dimensional gate voltage space. That is roughly equivalent to a needle in a $(2\text{m})^3$ haystack.

Once a qubit has been found, the algorithm allows for extensive characterisation. We can study f_{Rabi} and the g-factor as a function of the barrier gate voltages (Fig. 3d). The resulting maps gives insights into qubit properties and can be extended to measure, e.g. the Hahn-echo coherence time.

CONCLUSIONS

We have demonstrated fully automated tuning of spin qubits, progressing from a de-energised device to qubit control. Our algorithm autonomously achieved Rabi oscillations in 10 out of 13 trials. Most tuning processes concluded within three days, with the primary speed constraint being the integration times required to perform DC transport measurements, which could be replaced by fast readout alternatives. Maps of the g-factor and f_{Rabi} serve as evidence for the potential of this approach for high-throughput qubit characterisation.

The methodology is versatile and can be adapted for use with similar quantum devices, such as silicon FinFETs. The modular design of the algorithm allows for rapid adaptation to other architectures. For example, devices that use charge sensors and reflectometry for measurements would only require different signal-processing algorithms.

We anticipate that the mass tuning and characterization of qubits, facilitated by our fully autonomous tuning algorithm, will establish a productive feedback loop between measurement and fabrication processes. Wafer-scale, high-throughput characterisation of quantum devices, already feasible in early tuning stages, can mitigate device variability. This, in turn, improves the tuning process, bolstered by expanding datasets.

The first successful experiments on quantum computing with semiconductors were conducted nearly twenty years ago. We have now confirmed the feasibility of fully automatic tuning of a semiconductor qubit. This breakthrough would allow us to move forward by leveraging the ability to mass characterise qubits to advance quantum circuit scalability in semiconductors.

[1] D. Loss and D. P. DiVincenzo, "Quantum computation with quantum dots," *Phys. Rev. A*, vol. 57, no. 1, p. 120, 1998.

[2] J. R. Petta, A. C. Johnson, J. M. Taylor, E. A. Laird, A. Yacoby, M. D. Lukin, C. M. Marcus, M. P. Hanson, and A. C. Gossard, "Coherent Manipulation of Coupled Electron Spins in Semiconductor Quantum Dots," *Science*, vol. 309, no. 5744, pp. 2180–2184, 2005.

[3] F. H. Koppens, J. A. Folk, J. M. Elzerman, R. Hanson, L. W. Van Beveren, I. T. Vink, H.-P. Tranitz, W. Wegscheider, L. P. Kouwenhoven, and L. M. Vandersypen, "Control and Detection of Singlet-Triplet Mixing in a Random Nuclear Field," *Science*, vol. 309, no. 5739, pp. 1346–1350, 2005.

[4] X. Xue, M. Russ, N. Samkharadze, B. Undseth, A. Sammak, G. Scappucci, and L. M. Vandersypen, "Quantum logic with spin qubits crossing the surface code threshold," *Nature*, vol. 601, no. 7893, pp. 343–347, 2022.

[5] A. Noiri, K. Takeda, T. Nakajima, T. Kobayashi, A. Sammak, G. Scappucci, and S. Tarucha, "Fast universal quantum gate above the fault-tolerance threshold in silicon," *Nature*, vol. 601, no. 7893, pp. 338–342, 2022.

[6] M. T. Mądzik, S. Asaad, A. Youssry, B. Joecker, K. M. Rudinger, E. Nielsen, K. C. Young, T. J. Proctor, A. D. Baczewski, A. Laucht, *et al.*, "Precision tomography of a three-qubit donor quantum processor in silicon," *Nature*, vol. 601, no. 7893, pp. 348–353, 2022.

[7] S. Neyens, O. Zietz, T. Watson, F. Luthi, A. Nethwewala, H. George, E. Henry, A. Wagner, M. Islam, R. Pillarisetty, *et al.*, "Probing single electrons across 300 mm spin qubit wafers," *arXiv preprint arXiv:2307.04812*, 2023.

[8] J. P. Zwolak and J. M. Taylor, "Colloquium: Advances in automation of quantum dot devices control," *Rev. Mod. Phys.*, vol. 95, no. 1, p. 011006, 2023.

[9] H. Moon, D. Lennon, J. Kirkpatrick, N. van Esbroeck, L. Camenzind, L. Yu, F. Vigneau, D. Zumbühl, G. A. D. Briggs, M. Osborne, *et al.*, "Machine learning enables completely automatic tuning of a quantum device faster than human experts," *Nat. Commun.*, vol. 11, no. 1, pp. 1–10, 2020.

[10] M. Veldhorst, C. Yang, J. Hwang, W. Huang, J. Dehollain, J. Muhonen, S. Simmons, A. Laucht, F. Hudson, K. M. Itoh, *et al.*, "A two-qubit logic gate in silicon," *Nature*, vol. 526, no. 7573, pp. 410–414, 2015.

[11] G. Burkard, T. D. Ladd, A. Pan, J. M. Nichol, and J. R. Petta, "Semiconductor spin qubits," *Rev. Mod. Phys.*, vol. 95, no. 2, p. 025003, 2023.

[12] L. Vandersypen, H. Bluhm, J. Clarke, A. Dzurak, R. Ishihara, A. Morello, D. Reilly, L. Schreiber, and M. Veldhorst, "Interfacing spin qubits in quantum dots and donors - hot, dense, and coherent," *npj Quantum Inf.*, vol. 3, no. 1, pp. 1–10, 2017.

[13] T. Hensgens, T. Fujita, L. Janssen, X. Li, C. Van Diepen, C. Reichl, W. Wegscheider, S. D. Sarma, and L. M. Vandersypen, "Quantum simulation of a Fermi–Hubbard model using a semiconductor quantum dot array," *Nature*, vol. 548, no. 7665, pp. 70–73, 2017.

[14] X. Zhang, E. Morozova, M. Rimbach-Russ, D. Jirovec, T.-K. Hsiao, P. C. Fariña, C.-A. Wang, S. D. Oosterhout, A. Sammak, G. Scappucci, *et al.*, "Universal control of four singlet-triplet qubits," *arXiv preprint arXiv:2312.16101*, 2023.

[15] A. Noiri, K. Takeda, T. Nakajima, T. Kobayashi, A. Sammak, G. Scappucci, and S. Tarucha, "A shuttling-based two-qubit logic gate for linking distant silicon quantum processors," *Nat. Commun.*, vol. 13, no. 1, p. 5740, 2022.

[16] A. R. Mills, C. R. Guinn, M. J. Gullans, A. J. Sigillito, M. M. Feldman, E. Nielsen, and J. R. Petta, "Two-qubit silicon quantum processor with operation fidelity exceeding 99%," *Science Advances*, vol. 8, no. 14, p. eabn5130, 2022.

[17] L. Petit, H. Eenink, M. Russ, W. Lawrie, N. Hendrickx, S. Philips, J. Clarke, L. Vandersypen, and M. Veldhorst, "Universal quantum logic in hot silicon qubits," *Nature*, vol. 580, no. 7803, pp. 355–359, 2020.

[18] L. C. Camenzind, S. Geyer, A. Fuhrer, R. J. Warburton, D. M. Zumbühl, and A. V. Kuhlmann, "A hole spin qubit in a fin field-effect transistor above 4 kelvin," *Nature Electronics*, vol. 5, no. 3, pp. 178–183, 2022.

[19] A. Zwerver, T. Krähenmann, T. Watson, L. Lampert, H. C. George, R. Pillarisetty, S. Bojarski, P. Amin, S. Amitonov, J. Boter, *et al.*, "Qubits made by advanced semiconductor manufacturing," *Nat. Electron.*, vol. 5, no. 3, pp. 184–190, 2022.

[20] M. de Kruijf, S. Geyer, T. Berger, M. Mergenthaler, F. Braakman, R. J. Warburton, and A. V. Kuhlmann, "A compact and versatile cryogenic probe station for quantum device testing," *Rev. Sci. Instrum.*, vol. 94, no. 5, 2023.

[21] E. J. Thomas, V. N. Ciriano-Tejel, D. F. Wise, D. Prete, M. de Kruijf, D. J. Ibberson, G. M. Noah, A. Gomez-Saiz, M. F. Gonzalez-Zalba, M. A. Johnson, *et al.*, "Rapid cryogenic characterisation of 1024 integrated silicon quantum dots," *arXiv preprint arXiv:2310.20434*, 2024.

[22] S. G. Philips, M. T. Mądzik, S. V. Amitonov, S. L. de Snoo, M. Russ, N. Kalhor, C. Volk, W. I. Lawrie, D. Brousse, L. Tryputen, *et al.*, "Universal control of a six-qubit quantum processor in silicon," *Nature*, vol. 609, no. 7929, pp. 919–924, 2022.

[23] J. Darulová, S. Pauka, N. Wiebe, K. Chan, G. Gardener, M. Manfra, M. Cassidy, and M. Troyer, "Autonomous Tuning and Charge-State Detection of Gate-Defined Quantum Dots," *Phys. Rev. Appl.*, vol. 13, no. 5, p. 054005, 2020.

[24] B. Severin, D. T. Lennon, L. C. Camenzind, F. Vigneau, F. Fedele, D. Jirovec, A. Ballabio, D. Chrastina, G. Isella, M. de Kruijf, *et al.*, "Cross-architecture Tuning of Silicon and SiGe-based Quantum Devices Using Machine Learning," *arXiv preprint arXiv:2107.12975*, 2021.

[25] B. van Straaten, F. Fedele, F. Vigneau, J. Hickie, D. Jirovec, A. Ballabio, D. Chrastina, G. Isella, G. Katsaros, and N. Ares, "All rf-based tuning algorithm for quantum devices using machine learning," *arXiv preprint arXiv:2211.04504*, 2022.

[26] S. S. Kalantre, J. P. Zwolak, S. Ragole, X. Wu, N. M. Zimmerman, M. D. Stewart, and J. M. Taylor, "Machine learning techniques for state recognition and auto-tuning in quantum dots," *npj Quantum Inf.*, vol. 5, no. 1, pp. 1–10, 2019.

[27] J. P. Zwolak, T. McJunkin, S. S. Kalantre, J. Dodson, E. MacQuarrie, D. Savage, M. Lagally, S. Coppersmith, M. A. Eriksson, and J. M. Taylor, "Autotuning of Double-Dot Devices in Situ with Machine Learning," *Phys. Rev.*

Appl., vol. 13, no. 3, p. 034075, 2020.

[28] J. P. Zwolak, T. McJunkin, S. S. Kalantre, S. F. Neyens, E. MacQuarrie, M. A. Eriksson, and J. M. Taylor, "Ray-Based Framework for State Identification in Quantum Dot Devices," *PRX Quantum*, vol. 2, no. 2, p. 020335, 2021.

[29] V. Nguyen, S. Orbell, D. T. Lennon, H. Moon, F. Vigneau, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, *et al.*, "Deep reinforcement learning for efficient measurement of quantum devices," *npj Quantum Inf.*, vol. 7, no. 1, pp. 1–9, 2021.

[30] T. A. Baart, P. T. Eendebak, C. Reichl, W. Wegscheider, and L. M. Vandersypen, "Computer-automated tuning of semiconductor double quantum dots into the single-electron regime," *Appl. Phys. Lett.*, vol. 108, no. 21, p. 213104, 2016.

[31] S. Czischek, V. Yon, M.-A. Genest, M.-A. Roux, S. Rochette, J. C. Lemyre, M. Moras, M. Pioro-Ladrière, D. Drouin, Y. Beilliard, *et al.*, "Miniaturizing neural networks for charge state autotuning in quantum dots," *Machine Learning: Science and Technology*, vol. 3, no. 1, p. 015001, 2021.

[32] R. Durrer, B. Kratochwil, J. V. Koski, A. J. Landig, C. Reichl, W. Wegscheider, T. Ihn, and E. Greplova, "Automated Tuning of Double Quantum Dots into Specific Charge States Using Neural Networks," *Phys. Rev. Appl.*, vol. 13, no. 5, p. 054019, 2020.

[33] M. Lapointe-Major, O. Germain, J. C. Lemyre, D. Lachance-Quirion, S. Rochette, F. C. Lemyre, and M. Pioro-Ladrière, "Algorithm for automated tuning of a quantum dot into the single-electron regime," *Phys. Rev. B*, vol. 102, no. 8, p. 085301, 2020.

[34] N. van Esbroeck, D. Lennon, H. Moon, V. Nguyen, F. Vigneau, L. Camenzind, L. Yu, D. Zumbühl, G. Briggs, D. Sejdinovic, *et al.*, "Quantum device fine-tuning using unsupervised embedding learning," *New Journal of Physics*, vol. 22, no. 9, p. 095003, 2020.

[35] J. D. Teske, S. S. Humpohl, R. Otten, P. Bethke, P. Cerfontaine, J. Dedden, A. Ludwig, A. D. Wieck, and H. Bluhm, "A machine learning approach for automated fine-tuning of semiconductor spin qubits," *Appl. Phys. Lett.*, vol. 114, no. 13, p. 133102, 2019.

[36] C. Van Diepen, P. T. Eendebak, B. T. Buijtendorp, U. Mukhopadhyay, T. Fujita, C. Reichl, W. Wegscheider, and L. M. Vandersypen, "Automated tuning of inter-dot tunnel coupling in double quantum dots," *Appl. Phys. Lett.*, vol. 113, no. 3, p. 033101, 2018.

[37] A. Chatterjee, F. Ansaloni, T. Rasmussen, B. Brovang, F. Fedele, H. Bohuslavskyi, O. Krause, and F. Kuemmeth, "Autonomous Estimation of High-Dimensional Coulomb Diamonds from Sparse Measurements," *Phys. Rev. Appl.*, vol. 18, no. 6, p. 064040, 2022.

[38] D. L. Craig, H. Moon, F. Fedele, D. T. Lennon, B. van Straaten, F. Vigneau, L. C. Camenzind, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, D. Sejdinovic, and N. Ares, "Bridging the Reality Gap in Quantum Devices with Physics-Aware Machine Learning," *Phys. Rev. X*, vol. 14, p. 011001, Jan 2024.

[39] J. Schuff, D. T. Lennon, S. Geyer, D. L. Craig, F. Fedele, F. Vigneau, L. C. Camenzind, A. V. Kuhlmann, G. A. D. Briggs, D. M. Zumbühl, *et al.*, "Identifying Pauli spin blockade using deep learning," *Quantum*, vol. 7, p. 1077, 2023.

[40] M. J. Carballido, S. Svab, R. S. Eggli, T. Patlatiuk, P. Chevalier Kwon, J. Schuff, R. M. Kaiser, L. C. Camenzind, A. Li, N. Ares, E. P. A. M. Bakkers, S. Bosco, J. C. Egues, D. Loss, and D. M. Zumbühl, "Compromise-free scaling of qubit speed and coherence," *In preparation*, 2023.

[41] F. N. M. Froning, M. J. Rančić, B. Hetényi, S. Bosco, M. K. Rehmann, A. Li, E. P. A. M. Bakkers, F. A. Zwanenburg, D. Loss, D. M. Zumbühl, and F. R. Braakman, "Strong spin-orbit interaction and g-factor renormalization of hole spins in ge/si nanowire quantum dots," *Phys. Rev. Res.*, vol. 3, p. 013081, Jan 2021.

[42] F. N. M. Froning, M. K. Rehmann, J. Ridderbos, M. Brauns, F. A. Zwanenburg, A. Li, E. P. A. M. Bakkers, D. M. Zumbühl, and F. R. Braakman, "Single, double, and triple quantum dots in Ge/Si nanowires," *Appl. Phys. Lett.*, vol. 113, p. 073102, 2018.

[43] J. Ungerer, P. Chevalier Kwon, T. Patlatiuk, J. Ridderbos, A. Kononov, D. Sarmah, E. Bakkers, D. Zumbühl, and C. Schönenberger, "Charge-sensing of a Ge/Si core/shell nanowire double quantum dot using a high-impedance superconducting resonator," *Materials for Quantum Technology*, vol. 3, no. 3, p. 031001, 2023.

[44] R. S. Eggli, S. Svab, T. Patlatiuk, D. A. Trüssel, M. J. Carballido, P. C. Kwon, S. Geyer, A. Li, E. P. Bakkers, A. V. Kuhlmann, *et al.*, "Cryogenic hyperabrupt strontium titanate varactors for sensitive reflectometry of quantum dots," *Phys. Rev. Appl.*, vol. 20, no. 5, p. 054056, 2023.

[45] R. Maurand, X. Jehl, D. Kotekar-Patil, A. Corna, H. Bohuslavskyi, R. Laviéville, L. Hutin, S. Barraud, M. Vinet, M. Sanquer, *et al.*, "A CMOS silicon spin qubit," *Nature communications*, vol. 7, no. 1, p. 13575, 2016.

[46] N. Hendrickx, W. Lawrie, L. Petit, A. Sammak, G. Scappucci, and M. Veldhorst, "A single-hole spin qubit," *Nature communications*, vol. 11, no. 1, p. 3478, 2020.

[47] F. N. Froning, L. C. Camenzind, O. A. van der Molen, A. Li, E. P. Bakkers, D. M. Zumbühl, and F. R. Braakman, "Ultrafast hole spin qubit with gate-tunable spin–orbit switch functionality," *Nat. Nanotechnol.*, vol. 16, no. 3, pp. 308–312, 2021.

[48] M. Kotzagiannidis, J. Schuff, and N. Korda, "The Automated Bias Triangle Feature Extraction Framework," *arXiv preprint arXiv:2312.03110*, 2023.

[49] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nat. Methods*, vol. 17, pp. 261–272, 2020.

[50] C. E. Rasmussen and C. K. I. Williams, "Gaussian Processes for Machine Learning," in *Adaptive computation and machine learning*, 2005.

[51] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[52] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams, and N. de Freitas, "Taking the Human Out of the Loop: A Review of Bayesian Optimization," *Proceedings of the IEEE*, vol. 104, pp. 148–175, 2016.

[53] J. Danon and Y. V. Nazarov, "Pauli spin blockade in the presence of strong spin-orbit coupling," *Phys. Rev. B*, vol. 80, no. 4, p. 041301, 2009.

METHODS

Here we provide details of the experimental set up, and algorithmic choices made in this study. We begin by describing the device used for our experiments. We then follow with a discussion of the intelligent computational approaches used in the algorithm. Finally, details of each stage are discussed. An overview of all the stages of the algorithm is given in Table II.

Hyperparameters that are not explicitly given here can be found in the Supplementary Material.

I. THE DEVICE, MEASUREMENT APPARATUS AND PULSE SEQUENCE

The device consists of a Ge/Si core/shell nanowire lying on top of nine bottom gates measured in a variable temperature insert (VTI) in a liquid helium bath with the sample mounted below the 1-K pot (base temperature 1.5 K). By applying positive voltages to the first five bottom gates from the left, an intrinsic hole gas inside the nanowire is depleted to form a hole DQD. An SEM image of a device similar to the one used here is shown in Fig.4. The device, measurement apparatus and pulse sequence are the same as used by Carballido *et al.* [40] and are described in detail in their work.

To amplify the measurements that rely on a microwave pulse, we applied a pulse-modulation by a lock-in amplifier at 87.777 Hz. The measurements therefore have an in-phase and out-of-phase component. We apply principal component analysis (PCA) to these measurements and project each measurement onto the principal axis. We further offset the measurements such that they are strictly positive. Measurements that are obtained this way are marked with I_{LI}, as opposed to currents that were measured conventionally which are marked with $|I|$.

II. TECHNIQUES USED IN THE ALGORITHM

Useful automation of tuning from a de-energised device to identification of Rabi oscillations requires an algorithm that can adapt to different data capture regimes, and be transferred to other, similar devices. To achieve this, we have made extensive use of intelligent, adaptive and data-driven subroutines. Nowadays, there is a plethora of such techniques to choose from, and in order to choose the right technique to apply to each stage of the algorithm, we considered:

- the need for expert labelled training data, which was not always possible or realistic to source;

- the need of being efficient both in the total number of measurements taken during a stage, and in the resources needed for computing decisions about which measurements to take;

- the minimal accuracy needed in each stage for the whole algorithm to be able to achieve its overall goal of identifying operating parameters for a qubit.

To address the above considerations across the many stages of the algorithm, a non-exhaustive list of techniques we have found useful to employ includes Gaussian process (GP) inference, convolutional neural networks (CNNs), unsupervised computer vision (CV) and computational geometry techniques, and Bayesian optimisation (BO). We now briefly introduce these, highlighting their strengths and weaknesses.

GPs are a popular form of non-parametric Bayesian inference [50]. They can be thought of as a method for doing principled Bayesian inference over a space of functions. GPs can be tailored to any specific domain or problem by making a choice of the so-called kernel (or covariance) function, a part of this model which describes prior knowledge about the possible space of functions in which inference is to occur. This choice can allow practitioners to encode important domain knowledge before capturing any data, such as specifying knowledge of periodicities, symmetries or the expected degree of smoothness of the underlying process that is being observed. This constitutes the main strength of GP modelling, often enabling highly data-efficient inference. However, both model fitting and model prediction can be computationally intensive, typically growing cubically [50] in the number of observed data points.

Over the past two decades, CNN architectures have proved to be go-to models for solving computer vision tasks. Their strengths lie in their adaptability across different computer vision tasks, their robustness in the face of unknown noise, and their computational efficiency at training time. Their weakness lies in always requiring substantial amounts of training data. In this work, we use some standard architectures, such as ResNet [51] as tools for extracting properties from or making assessments of stability diagrams. Where we have applied them, training data has been either generated by a sufficiently good simulator, or gathered from this or similar devices and then labelled by an expert.

Often, the use of CNNs is neither required nor appropriate for the particular computer vision task at hand. In particular, it has been of crucial importance to a number of stages through the algorithm to be able to automatically locate and segment bias triangles within stability diagrams using a coordinate-wise approach. To achieve this we have employed a number of unsupervised computer vision techniques that can mitigate noise in, and localize features of the geometric figures present in stability diagrams. Whereas CNNs would require large amounts of labelled training data to achieve this result, the requirements can be met by computer vision techniques that need no training data, and require only a few hyperparameter choices to be made. All together we have called this a bias triangle segmentation framework, and it is described in [48], where the specific application to PSB detection is also detailed.

Bayesian optimisation [52] is a general, iterative approach to black-box function optimisation. At each iteration, it constructs a surrogate model of the function being approximated using the data already gathered, and uses this surrogate model to efficiently compute the next most informative location from which to sample the unknown function. In order to apply this technique one must specify a score function to optimise and a parameter space over which the search for an optimum is conducted. The choice of the surrogate model is also influential in the accuracy, efficiency and reliability achieved using this method. In this work we have made consistent use of GP surrogate models using a Matern 5/2 kernel.

III. STAGES OF THE ALGORITHM

1. Define DQD

a. Hypersurface building

As a first step, the algorithm determines a current that it considers to be pinched off by ramping to the high end of the safe ranges. We take repeated current measurements there to characterise the noise floor. From the noise floor we compute a pinch off current.

Next, we sample several points within the safe ranges using a Sobol sequence for quasi-random locations. The points are used to define rays from the origin that are then investigated for pinch off. To avoid overloading the current amplifier, we search for pinch off from the origin towards the upper end of the safety ranges with a low bias voltage. Once pinch off is found, we retrace with a higher bias voltage to confirm the exact pinch off location.

Finally, we use this data to construct the hypersurface model as outlined in the main text.

b. Double dot detection

The previous stage defines a region within which we can look for a DQD potential. We sample quasi-random points via a Sobol sequence for investigation. For each point, we vary both plunger voltages simultaneously and measure the current, following the method described by Moon *et al.* [9]. We use the random forest classifier developed by Severin *et al.* [24] to check for the presence of Coulomb peaks. If Coulomb peaks are found, we then measure a stability diagram. This diagram is analyzed with a neural network to detect features of the DQD. The neural network was trained on data from a variety of devices, mainly from the data gathered by Moon *et al.* [9] and Severin *et al.* [24], and additional data from a nanowire device different from the one used in this work. In total, there were 4,611 stability diagrams of which 726 showed double dot features.

2. Tune barriers

a. Barrier optimisation

Upon formation of separated pairs of bias triangles, we perform coordinate-wise segmentation and polygon fitting (see Sect. III 3 b for more details, [48]) in order to facilitate their tracking and feature extraction throughout the remainder of the tuning pipeline.

The segmentation and shape extraction enable the assessment of the current intensity difference between the base line and gap formed between the base and first excited state line. By quantifying this intensity difference in a base separation score, we can employ a Bayesian optimisation framework which seeks to maximize the intensity difference, as a promising indicator for detecting viable candidates for PSB.

Each pair of bias triangles should present a base well separated from the main body. The wider the separation, the better the protection of the ground state and the more likely it is that we are in the presence of a qubit. The separation score is computed by averaging the current along the detuning axis, see the two one-dimensional traces in Fig. 2 b-ii, and computing the ratio of the intensity between the peaks and the lowest point in the valley between them. In the case of multiple triangles, the highest separation is used as score.

Certain potential landscapes can lead to situations in which charge configurations are affected by charge switch noise. Since these potentials are unlikely to be used as a qubit and the resulting bias triangles can skew our base separation scoring, we excluded them by leveraging a neural network classifier. This classifier was trained to distinguish between normal bias triangles and ones that are affected by charge switch noise, as illustrated in Fig. 6. The training dataset for this classifier was obtained as follows: initially, potential bias triangles were identified using our segmentation routine. Subsequently, we hand-labeled 2,302 of these (1,539 samples showed no switch noise) to create a robust training set. The classifier itself was then obtained by fine-tuning a ResNet-based architecture with this dataset.

Once the voltage space has been explored through BO we have a clear understanding of the landscape, as in Fig. 2 b-ii. The measurements in this optimisation were obtained using an efficient measurement algorithm, see Supplementary Materials for details. We sample the most promising regions again without the efficient measurement algorithm and analyse them as described in the next section.

b. Plunger window selection

Given a sampled stability diagram containing bias triangles, the aim is to select the region that contains as many bias triangles with scores as high as possible, no areas with current that is too high, and as few switches

as possible as depicted in Fig. 2 b-iii. High current bias triangles are unlikely to be able to be used for qubit operations. This is a heuristic and we set a conservative threshold of 200 pA. To ease the downstream steps, the region should be a rectangular window. Through an iterative approach, starting from the smallest bounding boxes containing each single pair of bias triangles, larger windows are constructed by merging the existing ones in case they satisfy the conditions about switch absence and low currents. For the switches absence we used a soft constraint, allowing for bias triangles affected by switch noise in case their total area was less than 25% of the area covered by all triangles in the window. The algorithm complexity scales exponentially with respect to the number of triangles and some heuristics have been leveraged to reduce substantially complexity and therefore execution time. In particular, at each iteration, only the top 100 bounding boxes by the number of contained triangles without switches were kept, to ensure a manageable upper limit on the number of possible merges. The routine halts when no further merges are possible. Once the plunger windows have been selected they are ranked by highest separation score.

3. Find PSB

a. Wide shot PSB detection

To identify each bias triangle's location, we first leverage the fact that they sit on a honeycomb or skewed rectangle pattern. We use autocorrelation on the stability diagram to identify this pattern. The largest two peaks in the autocorrelation help us establish a vector that spans this pattern of skewed rectangles. To fix the pattern in place, we employ a blob detection algorithm, using the first blob it identifies. This helps us accurately overlay the skewed rectangle pattern and estimate the locations of the bias triangles (Fig. 5).

Next, we extract these bias triangles using the identified locations, with side lengths informed by the pattern dimensions. These extracted bias triangles are then input into a neural network for further analysis. We used autonomously gathered data that was taken during the initial development phase. In total, we used 626 pairs of bias triangles taken from 70 stability diagrams. 55 of the pairs of bias triangles showed PSB. For more detailed information on this procedure, see Schuff *et al.* [39].

b. Re-centering and high resolution PSB detection

In an effort to filter the previously detected candidates for PSB and eliminate false positives, a second set of higher resolution measurements is performed. For that purpose, a dedicated low resolution stability diagram of the candidate bias triangles is taken and used to update the plunger voltage extent based on the detected

contours, effectively performing a re-centering. With the updated voltage extent, high resolution stability diagrams with $B = 0$ T and with $B = 0.1$ T are taken.

A second sub-stage of PSB-classification is applied through a segmentation-based detection and feature extraction framework, which facilitates the coordinate-wise quantification of geometric and physical properties of bias triangles [48]. In particular, given the high-resolution stability diagram with $B = 0.1$ T, this framework fits minimum-edge polygons to the detected contours of bias triangle pairs by utilizing a relaxed extension of the Ramer-Douglas-Peucker algorithm [48]. Once the segmented shape mask is identified, further geometric properties such as the base and excited state lines can be automatically extracted solely based on the prior knowledge of the bias voltage sign, which predicts the direction in which bias triangles point.

For the identification of PSB, an analytical classifier based on the above framework was devised [48]. PSB expresses itself as a suppressed base line of the bias triangles at $B = 0$ T. At $B > 0$ T, there is a leakage current at the base line. The routine extracts the segment enclosing the base and a prominent excited state line on the stability diagram with leakage current ($B = 0.1$ T). Subsequently, the average pixel intensity of the segment normalised by the intensity of the entire pair of bias triangles is computed. By superimposing the detected segment on the scan with blocked current ($B = 0$ T), normalised intensity values are compared and, if their difference exceeds a specified threshold, the charge transition is identified as positive for PSB.

Based on the segmented bias triangle shape mask, further geometrical properties can be automatically extracted, which enables the tuning of bias triangle features in stage 2.
The detuning axis, utilized for the Danon gap measurement, is automatically extracted by identifying the bias triangles base midpoints and tips and computing the lines between them.

c.   Danon gap check

As a further filter for possible candidates, we check to magnetic field dependence of the leakage current at the base of the bias triangles in a different way. As a function of the applied magnetic field B, we expect the leakage current to be minimal at $B = 0$ T and higher away from this point. We call this the Danon gap [53]. A current measurement of the detuning line while varying the magnetic field gives us a two-dimensional input (Fig. 7), which we can analyse as follows: Ignoring the noise signal, the current is roughly constant along the magnetic field axis, whereas the detuning line axis is information-rich: away from the Danon gap there are two local extrema

(to one side the noise floor outside the bias triangles, to the other side the gap due to singlet-triplet energy splitting) whilst the Danon gap region is characterised by a monotonic behaviour, with roughly a constant value.
To detect the presence of the Danon gap, the current I is firstly processed with a Gaussian filter, to smooth out the noise, and then the absolute slopes along the detuning line axis are integrated $g(B) := \sum_\varepsilon \left| \frac{\partial \tilde{I}(\varepsilon, B)}{\partial \varepsilon} \right|$, where the derivative is the discrete derivative along the detuning line axis. The function g is minimised in the areas where the smoothed signal $\tilde{I}$ shows a constant value. We show the normalised function $\bar{g} = \frac{g}{|\varepsilon|}$ in Fig. 7. To detect the presence of the Danon gap from g two tunable hyperparameters are used, validating the depth and width of the basin of the global minimum of g: in case the basin is not prominent enough there is no Danon gap. As the last check, the location of the minima has to be in proximity of zero magnetic field.

4.   Find readout

a.   Tracking & entropy optimisation

In subsequent steps, we apply a pulse sequence to the right plunger electrode. As it is a two-stage pulse, the bias triangles will have a 'shadow' in the stability diagram. We need to identify the original bias triangles and find a suitable region where we can expect to find qubit readout. In light of resulting shape distortions and further degrading effects to the measurement quality, we opt for template matching as opposed to performing re-segmentation for bias triangle tracking in order to ensure robustness.

The relative direction in which the shadow bias triangles appear with respect to the original one is known in advance due to the applied pulse shape. This is incorporated into the shape matching approach as the cardinal direction to uniquely identify and track the triangles.
We perform shape matching by comparing the edge map of a stability diagram prior to pulsing, functioning as the template, to the edge map of a subsequent stability diagram with pulsing, functioning as the source for current information. Further, we extract the segmented shape mask from the template. The method slides the template over the source edge map, thereby comparing the template with individual patches of the stability diagram with pulsing, and returns a result matrix (of the same size as the source) whose individual entries quantify the similarity with the template patch. The employed similarity metric is the normalized correlation coefficient and the patch with maximum correlation is selected. Once the appropriate patch has been identified, the initial segmentation mask of the stability diagram

without pulsing is mapped to the stability diagram with pulsing and used for subsequent processing.

In order to identify the optimal readout spot, we extract the segment enclosing the base and prominent excited state lines on the obtained segmented mask of bias triangles with pulsing, and perform Bayesian optimisation of a readout quality score over four parameters: the constrained two-dimensional plunger gate voltage space, frequency of the driving pulse f_{MW} and burst time t_{burst}. Optimal readout candidates are those which meet the resonance conditions of the qubit. If they are met, there is a leakage current that we record using the lock-in amplifier. For a given burst time t_{burst} (relating to the Rabi frequency f_{Rabi}) and a given frequency of the driving pulse f_{MW} (relating to the g-factor), the leakage is characterized as a peak in leakage current for a certain magnetic field B. Thus, for the readout optimisation, we use measure the current with varying magnetic fields. Instead of applying PCA, as explained above, we use the L2 norm of the in-phase and out-of-phase components to retrieve a one-dimensional trace $l(B)$. This guarantees peaks to be higher than the background, as opposed to processing with PCA, which can lead to dips rather than peaks.

To quantify the sharpness of these peaks, we developed a score based on the Shannon entropy $H = -\sum_B [l(B)\log(l(B)]$ of the trace. For the calculation of the entropy of the score, we first subtract the median and then clip values at zero. This particular pre-processing turns the trace into something more akin to a distribution and enhances the robustness of our score, making it less susceptible to potential noise disturbances in the trace. This method results in a smooth score landscape suitable for Bayesian optimisation.

b. Resonance confirmation

This verification step acts as a final filter and the last stages are all executed once a candidate passes this filter. The previous stage sends a candidate with a suspected resonance condition. The stage re-measures the leakage current as a function of the magnetic field. If the resonance condition was truly found, a peak should appear at the same magnetic field again. Should a peak with a specified prominence at this magnetic field within a set margin of error be detected, the resonance condition is considered confirmed and all downstream measurements are executed. We note that a noisy candidate might pass this stage. Repeating the verification step can reduce such occurrences.

c. Qubit measurements

Once a resonance condition is found, we vary the magnetic field and the burst duration. The characteristic Rabi chevron can be analysed by considering the frequency spectrum for each magnetic field. The frequency should have a minimum at the magnetic field that meets the resonance condition of the qubit. The amplitude should also be maximal there due to decoherence for off-resonant driving. We can therefore simply look for the maximum amplitude in the Fourier transformed Rabi chevron (Fig. 8). This information will give us the precise resonance conditions for the last step, repeated measurements of Rabi oscillations on resonance.

IV. CHARACTERISATION

The maps in Fig. 3 were generated using automated measurements. Initially, upon identifying a qubit, we record its readout spot, g-factor, and f_{Rabi}. We then alter the confinement potential by slightly adjusting the barrier gate voltages. This adjustment may shift the bias triangles, consequently moving the readout spot, g-factor, and f_{Rabi}. Our method involves tracing these transitions to locate the readout spot in its new position. At this new location, we conduct an EDSR check scan. Any changes in the peak's location inform us about variations in the g-factor. Furthermore, measuring Rabi oscillations at this point helps update our understanding of f_{Rabi}.

As we progressively deviate from the initial measurement point, we utilize our closest prior qubit data to infer the properties at the new location. This step is crucial as a shift in the g-factor necessitates modifying the magnetic field range, while a change in f_{Rabi} requires adjusting the t_{burst} duration for the EDSR check to accurately detect resonances.

AUTHOR CONTRIBUTIONS

J.S. developed the algorithm, in particular the modular framework and the overall algorithm flow. J.S. performed the experiments and analysed the results with M.J.C.'s guidance and help from T.P. and R.S.E. in D.M.Z.'s lab. M.J.C., S.S. and P.C.K. fabricated the device. J.S., M.J.C., J.C.C., M.K., M.C., J.R., N.K., D.C., B.v.S. and

B.S. developed the routines for the various stages of the algorithm, including analysing and interpreting data, making algorithm design choices, and writing the software used to run and record the experiments. J.C.C., M.K., M.C., J.R., and N.K. made substantial independent contributions to the Methods and Supplementary Materials sections. J.S. wrote the manuscript with inputs from all authors. N.A. conceived of the project.

DATA AND CODE AVAILABILITY

The data and code will be made available upon final publication.

EXTENDED DATA
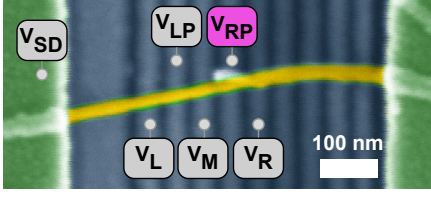
FIG. 4: **Device.** False color micrograph of a nanowire device similar to the one used in this work. The Ge/Si nanowire is colored in yellow.
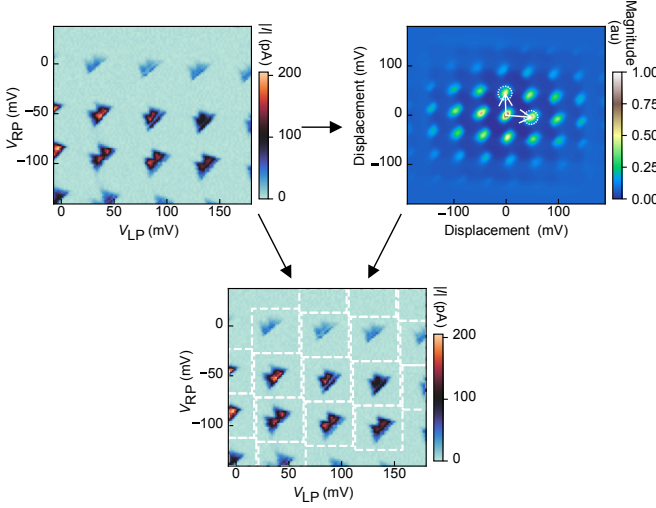

FIG. 5: **Bias triangle locations via auto-correlation.** A stability diagram (top left) is processed using its auto-correlation (top right). Within the auto-correlation, we can find the highest values to span a grid of skewed rectangles. The extracted information is used to locate bias triangles in the original stability diagram (bottom).
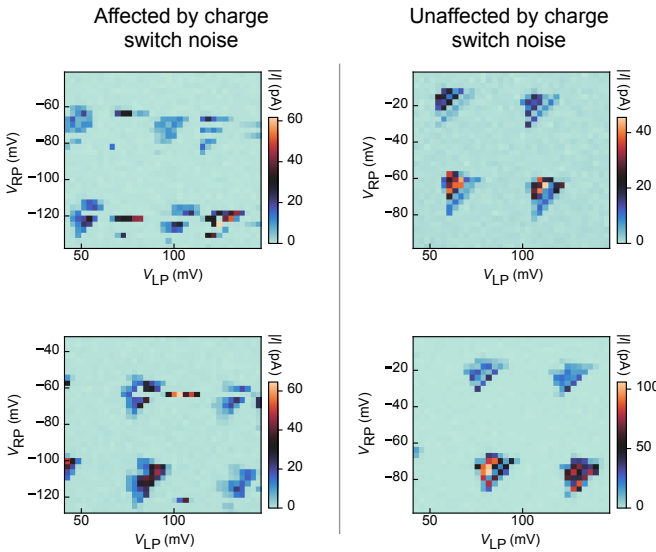

FIG. 6: **Effect of charge switch noise.** Stability diagrams can be affected by charge switches. Their effect can be seen in the left column. For comparison, the right column is unaffected by these charge switches.
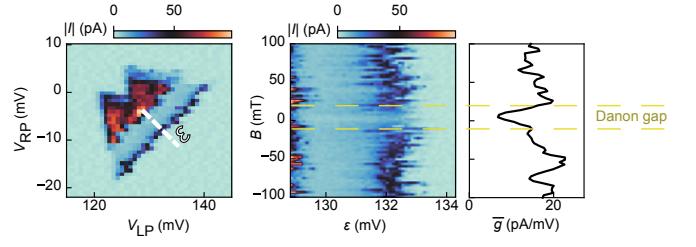

FIG. 7: **Danon gap check.** As final check to confirm the presence of PSB, the algorithm takes measurements of the detuning line ε (marked in the left panel) as a function of the magnetic field B (middle). The measurement is analysed, as described in III 3 c, by first computing the averaged sum of absolute derivatives $\bar{g}$ (right). If a dip is present in $\bar{g}$ close to $B = 0$ T, PSB is confirmed.
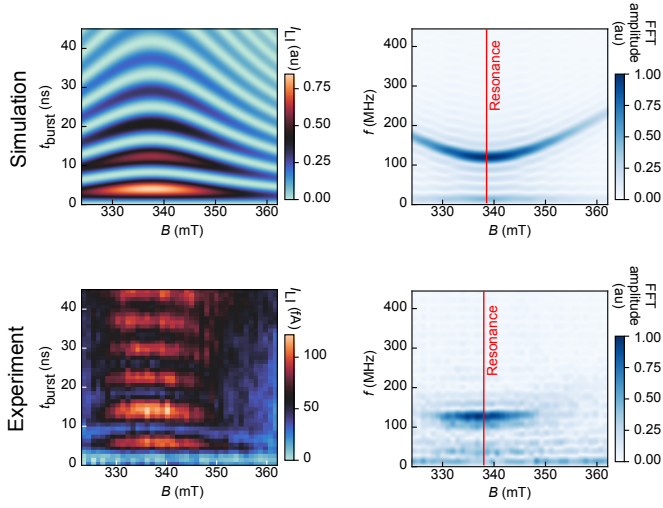
FIG. 8: Rabi chevron analysis. A qubit that is driven close to the resonance condition exhibits so-called Rabi chevrons, see left column for examples (top: simulation, bottom: experimental data). To find the resonance frequency, we analyse the frequency spectrum for each magnetic field (which controls the detuning from resonance). The frequencies have a maximum amplitude and a minimal point at the resonance condition.

Stage	Description	Techniques used	Candidate information
Stage 1: Define DQD			
a, Hypersurface building	Building a model of the surface that separates pinch-off from conducting	Gaussian processes	Upper and lower bounds of barrier voltages
b, Double dot detection	Identifying Coulomb peaks and double dot signatures	Random forests and neural networks	Corrected lower bound of barrier voltages
Stage 2: Tune Barriers			
a, Barrier optimisation	Search over the barrier voltage space for ideal settings	Bayesian optimisation, computer vision, neural networks	Promising barrier voltage location
b, Plunger window selection	Determine a window for plunger voltages	Computer vision, neural networks	Barrier voltage location and wide plunger voltage range
Stage 3: Find PSB			
a, Wide shot PSB detection	Identify locations of transitions with PSB	Neural networks, computer vision	Narrow plunger voltage range
b, Re-centering	Get precise plunger voltage window	Computer vision	Corrected narrow plunger voltage range
c, High res. PSB detection	Confirm PSB with high resolution measurement	Computer vision	Narrow plunger voltage range, link to high res. measurement, detuning line definition
d, Danon gap check	Confirm PSB by measuring detuning line as a function of magnetic field	Computer vision	Narrow plunger voltage range, link to high res. measurement
Stage 4: Find Readout			
a, Entropy optimisation	Find readout spot, g-factor and Rabi frequency by optimisation of an entropy score	Bayesian optimisation, computer vision	Precise plunger voltage locations, magnetic field, drive frequency and burst time
b, Resonance confirmation	Confirm resonance from previous stage/filter out noise	Peak finding	Passed on from previous
c, Spectroscopy	Measurement of current while varying t_{burst} and f_{MW} for documentation	-	Passed on from previous
d, Rabi chevron	Measuring Rabi oscillations close to the resonance condition	Frequency analysis	Corrected magnetic field for resonance condition
e, Rabi oscillations	Take repeated Rabi oscillations on resonance	-	-

TABLE II: List of all stages. We list all stages and sub-stages with a short description, a rough overview of what techniques were used and what the information is passed downstream for a candidate from each stage. Besides information on the parameters that are directly needed to operate a qubit, the stages also pass down meta-information that other stages might need to use, e.g. the high resolution stability diagram of Stage 3c is needed in Stage 4a for template matching.

Supplementary Information: Fully autonomous tuning of a spin qubit

Jonas Schuff,[1] Miguel J. Carballido,[2] Madeleine Kotzagiannidis,[3] Juan Carlos Calvo,[3] Marco Caselli,[3] Jacob Rawling,[3] David L. Craig,[1] Barnaby van Straaten,[1] Brandon Severin,[1] Federico Fedele,[4] Simon Svab,[2] Pierre Chevalier Kwon,[2] Rafael S. Eggli,[2] Taras Patlatiuk,[2] Nathan Korda,[3] Dominik Zumbühl,[2] and Natalia Ares[4]

[1]*Department of Materials, University of Oxford, Oxford OX1 3PH, United Kingdom*
[2]*Department of Physics, University of Basel, 4056 Basel, Switzerland*
[3]*Mind Foundry Ltd, Summertown, Oxford OX2 7DD, United Kingdom*
[4]*Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, United Kingdom*
(Dated: February 7, 2024)

CONTENTS

S1.   QUBIT MEASUREMENTS OF ALL SUCCESSFUL RUNS

We show measurements that confirm that we found a qubit in the ten successful runs, ordered by total run time as in the main text, see Fig. S1. The measurements were all taken autonomously. We show the associated pair of bias triangles (upper left in each panel), a measurement varying the magnetic field and the driving frequency (upper right), a Rabi chevron measurement where we vary the magnetic field and the burst duration (lower left), and an averaged measurement of Rabi oscillations (lower right) at the magnetic field indicated with dashed lines in the Rabi chevron measurement.

The diversity in the plunger voltage settings, the magnetic field settings, and the Rabi frequencies showcase the versatility of our algorithm.
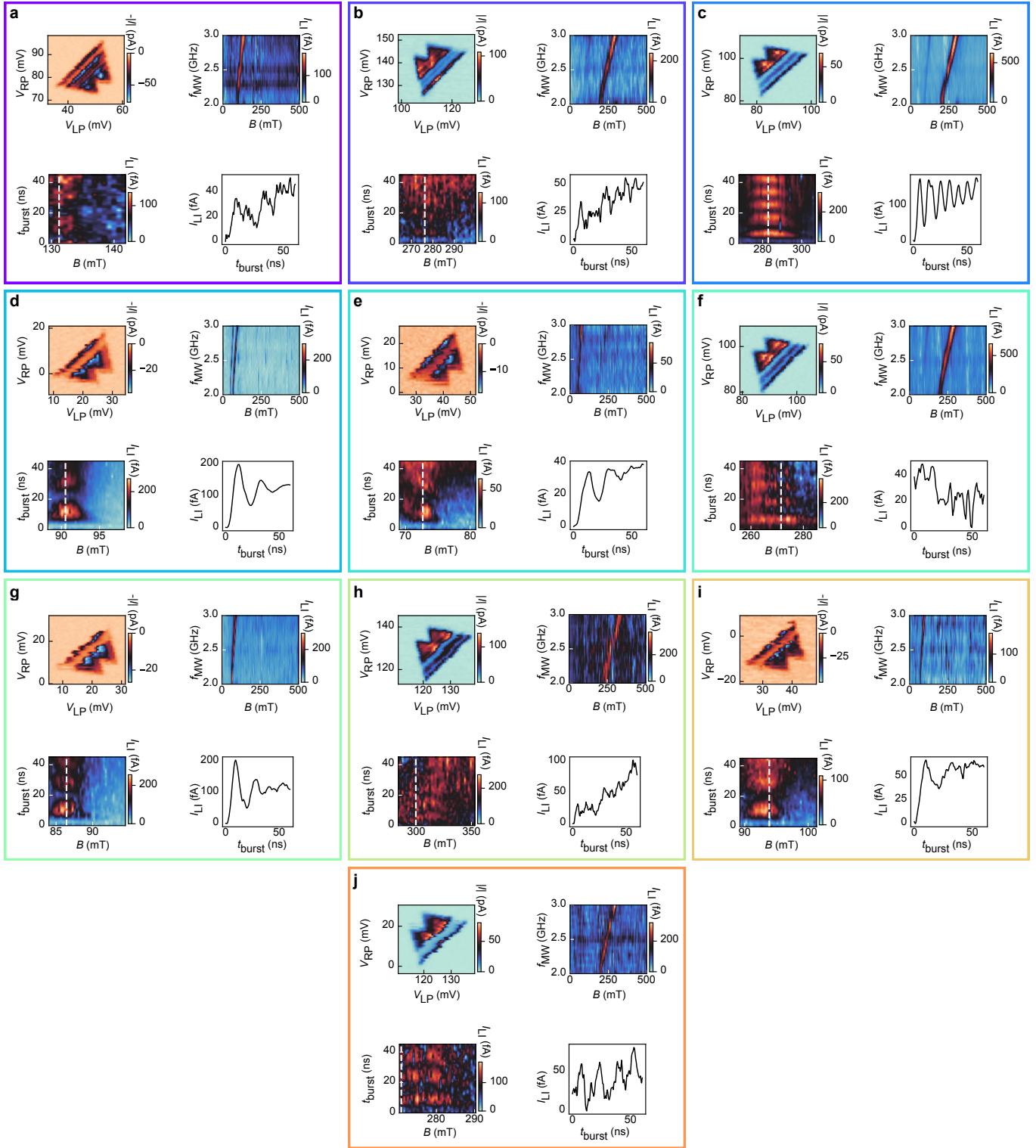
FIG. S1. **Qubit measurements of all successful runs.** The panels **a,-j,** are ordered by the total run time of the algorithm for each qubit respectively. Each panel includes four current measurements: the pair of bias triangles (upper left), spectroscopy measurement, varying magnetic field and driving frequency (upper right), Rabi chevron pattern, varying magnetic field and burst duration (lower left), and averaged Rabi oscillations (lower right) taken at the dashed lines in the Rabi chevron measurement. All measurements were performed autonomously. The Rabi chevron measurement does not have a dedicated re-centering stage, accounting for the off-centered measurements. The spectroscopy measurements were purely taken for documentation and always with the same ranges; these measurements did not inform any other part of the algorithm. Some measurements for panels **d**, **e**, and **f** were taken again using automated measurements after the initial runs finished because a setting of the lock-in amplifier led to slight measurement artifacts.

S2. HYPERPARAMETERS

List of all hyperparameters and comment on what they do. We do not report on hyperparameters that are irrelevant to the functionality of the algorithm, such as how often measurements are plotted to a documentation file.

1. Define DQD

a. Hypersurface building

- Steering parameters:
  - `number_of_rays` = 32 // Number of rays used to build hypersurface.
  - `n_noise_floor` = 100 // Number of measurements of the noise floor.

- Measurement parameters:
  - `lower_bounds` = [0, 0, 0] // Defines lower bound of area in which the hypersurface model is built, in V.
  - `upper_bounds` = [1.8, 1.8, 1.8] // Defines upper bound of area in which the hypersurface model is built, in V.
  - `bias_low` = 0.0007 // Low bias voltage used to find pinch off starting from the conducting region, in V.
  - `bias_high` = 0.005 // Bias voltage used to in subsequent stages. Used to confirm pinch off going from non-conducting to conducting region, in V.
  - `d_r` = 0.003 // Step length of pinch off search, in V.
  - `len_after_pinchoff` = 0.250 // Length after last point above threshold before pinch off is considered true (past last Coulomb peaks), in V.
  - `max_dist` = 2 // Additional safe range how far the ray is maximally ramped, in V.

- Analysis parameters:
  - `threshold_as_multiple_of_noise_high` = 100 // Used to define the noise threshold based on the noise floor measurement

b. Double dot detection

- Measurement parameters:
  - `magnetic_field` = 0.1 // At which magnetic field the measurements are taken, in T.
  - `plunger_location` = [0, 0] // Center of the plunger gate voltages, in V.
  - One-dimensional scan (to detect Coulomb peaks):
    * `window_right_plunger` = 0.1 // Defines side length of window in which measurements us taken, in V.
    * `window_left_plunger` = 0.1 // Defines side length of window in which measurements us taken, in V.
    * `n_px` = 128 // Number of points to be taken.
    * `wait_time` = 0.051 // Delay after setting parameter before measurement is performed, in s.
  - Two-dimensional scan (to detect DQD features):
    * `window_right_plunger` = 0.2 // As above.
    * `window_left_plunger` = 0.2 // As above.
    * `n_px_rp` = 48 // Number of points to be taken for right plunger axis.
    * `n_px_lp` = 48 // Number of points to be taken for left plunger axis.
    * `wait_time_slow_axis` = 0.5 // Delay after setting parameter before measurement is performed, in s.
    * `wait_time_fast_axis` = 0.051 // Delay after setting parameter before measurement is performed, in s.

- Analysis parameters:
  - `max_distance_between_locations` = 0.1 // Used to determine number of points sampled within DQD search region. Sets maximal distance between each sampled point, in V.
  - `path_to_nn` = `local_path` // Path to weights of neural network used for DQD feature detection.

2. Tune barriers

a. Entropy optimisation

- Steering parameters:

  - `seeding` = 15 // Parameter for Bayesian optimisation that informs exploration period.
  - `n_required_results` = 30 // Number of stability diagrams to be taken.

- Measurement parameters:

  - `rp_start` = −0.15 // Starting point of measurement for right plunger, in V.
  - `rp_end` = 0.15 // End point of measurement for right plunger, in V.
  - `lp_start` = −0.15 // Starting point of measurement for left plunger, in V.
  - `lp_end` = 0.15 // End point of measurement for left plunger, in V.
  - `n_points_plungers` = 100 // Number of points in each dimension.

b. Plunger window detection

- Steering parameters:

  - `number_of_full_scans_threshold` = 10 // Maximum number of full scans (i.e., without the efficient measurement algorithm) to be taken.
  - `number_of_candidates` = 10 // Maximum number of candidates Stage 2 can suggest in each bias direction, i.e., 10 can lead to up 20 candidates.
  - `bias_directions` = [`positive_bias`, `negative_bias`] // Candidates are built in those bias directions.

3. Find PSB

a. Wide shot PSB detection

- Steering parameters:

  - `max_number_candidates` = 5 // Maximum number of candidates this sub-stage can create.

- Measurement parameters:

  - `low_magnetic_field`= 0.0 // Magnetic field at which the stability diagram with blocked current shall be taken, in T.
  - `high_magnetic_field`= 0.1 // Magnetic field at which the stability diagram with leakage current shall be taken, in T.
  - `resolution` = 0.002 // Resolution of stability diagram in each axis, in V.
  - `padding` = 0.03 // Padding added to the plunger window suggestion from previous stage. Needed to have a slight margin around bias triangles, in V.

- Analysis parameters:

  - `psb_threshold` = 0.5 // Threshold for PSB detection. Neural network returns a value between 0 and 1 for each pair of bias triangles.
  - `folder_path_to_nn` = `local_path` // Path to neural network model that predicts signatures of PSB from low resolution measurements.
  - `offset_px` = 10 // Parameter used in the location detection via auto-correlation. The highest peak will always be in the center, so peaks within a certain distance (given in pixel here) from the center are disregarded.

<div align="center">*b. Re-centering*</div>

- Measurement parameters:

  - `magnetic_field`= 0.0 // Magnetic field at which the stability diagram shall be taken, in T.
  - `resolution` = 0.002 // Resolution of stability diagram in each axis, in V.
  - `wait_time_slow_axis` = 0.5 // As above.
  - `wait_time_fast_axis` = 0.051 // As above.

- Analysis parameters (all related to routine from Kotzagiannidis *et al.* [1]):

  - `segmentation_upscaling_res` = 2 // Image is upscaled by this factor to improve segmentation.
  - `relative_min_area` = 0.01 // Computes the `min_area` as a fraction of the total area. `min_area` sets a threshold for the minimum area of contour to be detected to avoid outliers.
  - `denoising` = true // Apply Gaussian smoothing.
  - `allow_MET` = false // Determines whether the 'Minimal enclosing triangle' technique is used or not. If true facilitates enclosing triangle shape approximation for disconnected contours.
  - `thr_method` = 'triangle' // Thresholding method for contour detection.

<div align="center">*c. High resolution PSB detection*</div>

- Measurement parameters:

  - `low_magnetic_field`= 0.0 // As above.
  - `high_magnetic_field`= 0.1 // As above.
  - `resolution` = 0.00075 // As above.
  - `wait_time_slow_axis` = 0.5 // As above.
  - `wait_time_fast_axis` = 0.051 // As above.
  - `padding` = 0.005 // As above.

- Analysis parameters:

  - `slope_tol` = 0.4 // Tolerance for deviation in absolute value between slopes of detected lines.
  - `int_tol` = 0.05 // Tolerance for PSB metric (absolute value difference between normalized segment intensities).
  - `seg_tol` = 0.05 // Gives percentage of image length as threshold for segments that are too small.
  - `median` = false // If true, selects the median of detected lines (ordered by y-intercept); false by default, so that the line with largest y-intercept (outmost) is selected.
  - `segmentation_upscaling_res` = 2 // As above.
  - `relative_min_area` = 0.01 // As above.
  - `denoising` = true // As above.
  - `allow_MET` = false // As above.
  - `thr_method` = 'triangle' // As above.

<div align="center">*d. Danon gap check*</div>

- Measurement parameters:

  - `magnetic_field_min` = - 0.1 // Start of magnetic field, in T.
  - `magnetic_field_max` = 0.1 // End of magnetic field, in T.
  - `resolution_magnet` = 0.003 // Resolution of magnetic field, in T.

- – `resolution_detuning` = 0.0001 // Resolution of detuning line axis, in V.

- – `detuning_base_offset` = 0.002 // We add this to the detuning line measurement to include the full base as the segmentation algorithm can lead to detuning line definitions that end on the base line, therefore missing valuable information.

- – `extra_wait_time_slow_axis` = 0.5 // The slow axis (magnetic field) is delayed by the time needed for the magnet to ramp one position, plus this given time.

- – `wait_time_fast_axis` = 0.077 // As above.

- Analysis parameters:

  - – `segmentation_upscaling_res` =2 // As above.

  - – `min_area` = 3 // As above.

  - – `thr_method` = 'triangle' // As above.

  - – `allow_MET` = false // As above.

  - – `padding_factor` = 1 // As above.

  - – `minimum_det_line_length_ratio` = 0.33 // The detuning line is determined via the segmentation algorithm. It also determines a cutoff within the triangles so that the algorithm only takes measurements at the base line of the triangle. If the detuning line that is determined is less than `minimum_det_line_length_ratio` of the full detuning line (from base line to the tip of the triangles), we extend the detuning line definition to avoid detuning lines definitions that are too short.

  - – `peak_offset_tolerance` = 0.025 // The gap can be at most offset from 0T by this much and still be accepted as a true gap, in T.

  - – `sigma` = 1 // For the gap detection, Gaussian smoothing factor.

  - – `field_gap_size` = 0.002 // For the gap detection, parameter that controls maximal gap width.

  - – `relative_depth` = 1.0 // For the gap detection, parameter that controls maximal gap depth.

4. Find readout

a. Entropy optimisation

- Steering parameters:

  - – `number_of_candidates` = 3 // Maximum number of candidates this sub-stage can create.

  - – `seeding` = 15 // As above.

  - – `iterations` = 30 // Number of total measurements taken by the Bayesian optimisation.

  - – `freq_vs.minimum` = 2.6e9 // Minimum driving frequency f_{MW} used in Bayesian optimisation, in Hz.

  - – `freq_vs.maximum` = 2.9e9 // Maximum driving frequency f_{MW} used in Bayesian optimisation, in Hz.

  - – `burst_time_ns.minimum` = 2 // Minimum burst time t_{burst} used in Bayesian optimisation, in ns.

  - – `burst_time_ns.maximum` = 16 // Minimum burst time t_{burst} used in Bayesian optimisation, in ns.

- Measurement parameters:

  - – `magnetic_field` = 0.1 // As above.

  - – `resolution` = 0.00075 // As above.

  - – `padding` = 0.005 // As above.

  - – `wait_time_slow_axis` = 0.5 // As above.

  - – `wait_time_fast_axis` = 0.051 // As above.

  - – `lockin_tc` = 1 // Time constant of lock-in amplifier, in s.

  - – `field_setpoint.start` = 0.0 // Minimum magnetic field, in T.

  - – `field_setpoint.stop` = 0.4 // Maximum magnetic field, in T.

- `field_setpoint.num_points` = 300 // Number of points in magnetic field axis.
  - `field_setpoint.delay` = 1 // Delay parameter, as above.

- Analysis parameters:

  - `segmentation_upscaling_res` = 2 // As above.
  - `relative_min_area` = 0.001 // As above.
  - `thr_method` = 'triangle' // As above.

b. Resonance confirmation

- Measurement parameters:

  - `magnetic_field_window` = 0.1 // Window symmetric around the assumed peak, in T.
  - `resolution_magnet` = 0.001 // Resolution of scan, in T.
  - `wait_time` = 2.5 // Delay for measurement, needs to be longer than the lock-in time constant.
  - `lockin_tc` = 2 // As above.

- Analysis parameters:

  - `prominence` = 0.9 // Minimum prominence of peaks.
  - `sigma` = 1 // Gaussian smoothing factor.
  - `peak_offset_tolerance` = 0.025 // Peaks with a maximum offset of this parameter from the assumed position are accepted.

c. Spectroscopy

- Measurement parameters:

  - `min_magnetic_field` = 0 // Start of magnetic field, in T.
  - `max_magnetic_field` = 0.5 // End of magnetic field, in T.
  - `resolution_magnet` = 0.005 // Resolution of magnetic field, in T.
  - `min_freq_vs` = 2e9 // Start of driving frequency f_{MW}, in Hz.
  - `max_freq_vs` = 3e9 // End of driving frequency f_{MW}, in Hz.
  - `resolution_freq` = 0.5e8 // Resolution of driving frequency f_{MW}, in Hz.
  - `extra_wait_time_slow_axis` = 1 // As above.
  - `wait_time_fast_axis` = 1.1 // As above.
  - `lockin_tc` = 1 // As above.

d. Rabi chevron

- Measurement parameters:

  - `magnetic_field_window_multiplier` = 2 // The window is determined by the width of the peak that resonance confirmation (Stage 4b) takes and multiplied with this factor.
  - `n_px_magnet` = 40 // Number of points in the magnetic field axis.
  - `resolution_burst_time` = 1e-9 // Resolution of the burst time t_{burst} axis, in s.
  - `min_burst_time` = 0 // Minimum of the burst time t_{burst}, in s.
  - `max_burst_time` = 45e-9 // Maximum of the burst time t_{burst}, in s.
  - `dead_burst_time` = 10e-9 // The total length of the pulse is twice the maximum t_{burst}, plus this factor, in s.
  - `extra_wait_time_slow_axis` = 6 // As above. Needs to be significantly larger than the lock-in time constant to avoid spill-over effects.
  - `wait_time_fast_axis` = 2.5 // As above.
  - `lockin_tc` = 2 // As above.

e.   Rabi oscillations

- Steering parameters:

  - `n_repetitions` = 5 // Number of repetitions of the same Rabi oscillation measurement.

- Measurement parameters:

  - `resolution_burst_time` = 0.5e-9 // As above.
  - `min_burst_time` = 0 // As above.
  - `max_burst_time` = 60e-9 // As above.
  - `dead_burst_time` = 10e-9 // As above.
  - `lockin_tc` = 2 // As above.

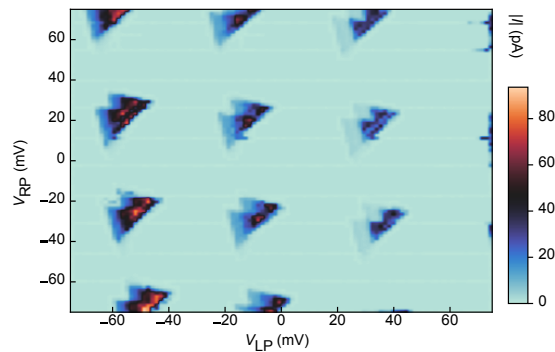## S3. EFFICIENT MEASUREMENT ALGORITHM

FIG. S2. **Examples of measurements taken with the efficient measurement algorithm.** Areas where no measurement have been taken have been filled with the threshold value.

Taking multiple charge transition stability diagram measurements is notably time-consuming, largely due to the predominance of featureless areas, as bias triangles are typically embedded within a skewed rectangular pattern. To address this, we have developed an efficient measurement algorithm. By rephrasing the measurement of bias triangles as an image processing task, whereby the goal is to determine a contour that traces the outline of a bias triangle, we were able to reduce the measurement time to 33% of a brute force scan.

For a binary image represented by a matrix composed entirely of zeros and ones, the perimeter of any grouping of non-zero elements that form a contiguous region is known as a *contour*. The Moore-Neighbour contour tracing algorithm provides a method of evaluating a complete contour given a starting point within the contour [2]. The Moore-Neighbour contour achieves this by only ever examining pixels adjacent to a previously examined pixel. As the location of pixels corresponds to plunger gate voltages, measurements of well separated pixels are both costly in time and present a risk of introducing noise such as switches. Once the edge of a bias triangle has been identified, its contour can therefore be quickly measured with minimal overhead from the device. After the contour has been evaluated, each pixel inside the contour can be measured sequentially to complete the bias triangle.

To construct a binary image from a series of measured current values at differing gate voltages, a threshold must be determined. Current values above and below this threshold are considered to be ones and zeros in the binary image, respectively. The threshold can either be set manually, using prior knowledge of the system, or determined on-the-fly. To provide a fully automated system we took a calibration scan using a sparse sampling and evaluated the median absolute deviation threshold from these measured points. The sampling routine was a so-called *snake scan*, where measurements are performed horizontally left-to-right until a boundary of the measurement region is reached, then proceed vertically for a fixed length and continue horizontally in the opposite direction until the entire image has been covered.

With a threshold determined, a second sparse sampling across the measurement region is performed. The Moore-Neighbour contour tracing routine is triggered on any measurements that exceed the threshold, followed by a routine to measure the inside of the contour. This process systematically captures a complete pair of bias triangles. Post completion, the scan resumes until the next cluster necessitates flood filling. In order to minimize wasted measurements, current values were cached. The second sampling of the routine also used a snake scan to explore the measurement region, however it was offset vertically compared to the original to maximise the chance of encountering a bias triangle.

Bias triangles are organised on a skewed grid pattern. The bias triangles evaluated in the second sparse sampling stage can be used to fit a skewed rectangular grid and infer the location of any missing bias triangles. Triangles can be missed by the initial sparse sampling if they reside between the horizontal lines of the two snake scans. A skewed grid can be represented by two vectors that describe the spatial separation between points on the grid, and the location of one grid point. These parameters were determined by minimising the total distance between the barycenter of all contours evaluated in the second sampling stage and points on the fitted grid. After fitting, each point on the grid within the measurement region that did not have a bias triangle was measured sequentially.

Employing this method has proven to significantly streamline the process, cutting down the measurement time by approximately two-thirds. For the hyperparameters as reported above, the measurement of a 100 by 100 point stability diagram takes 14.5 min ± 3.0 min with this efficient measurement algorithm, and 43.7 min ± 0.1 min with a conventional grid scan.

S4. SEARCH TREE EXAMPLES

In Fig. 1b of the main text, we show an illustrative example of a search tree. Here, we show the search trees that were actually constructed for the longest and shortest runs in our experiments.
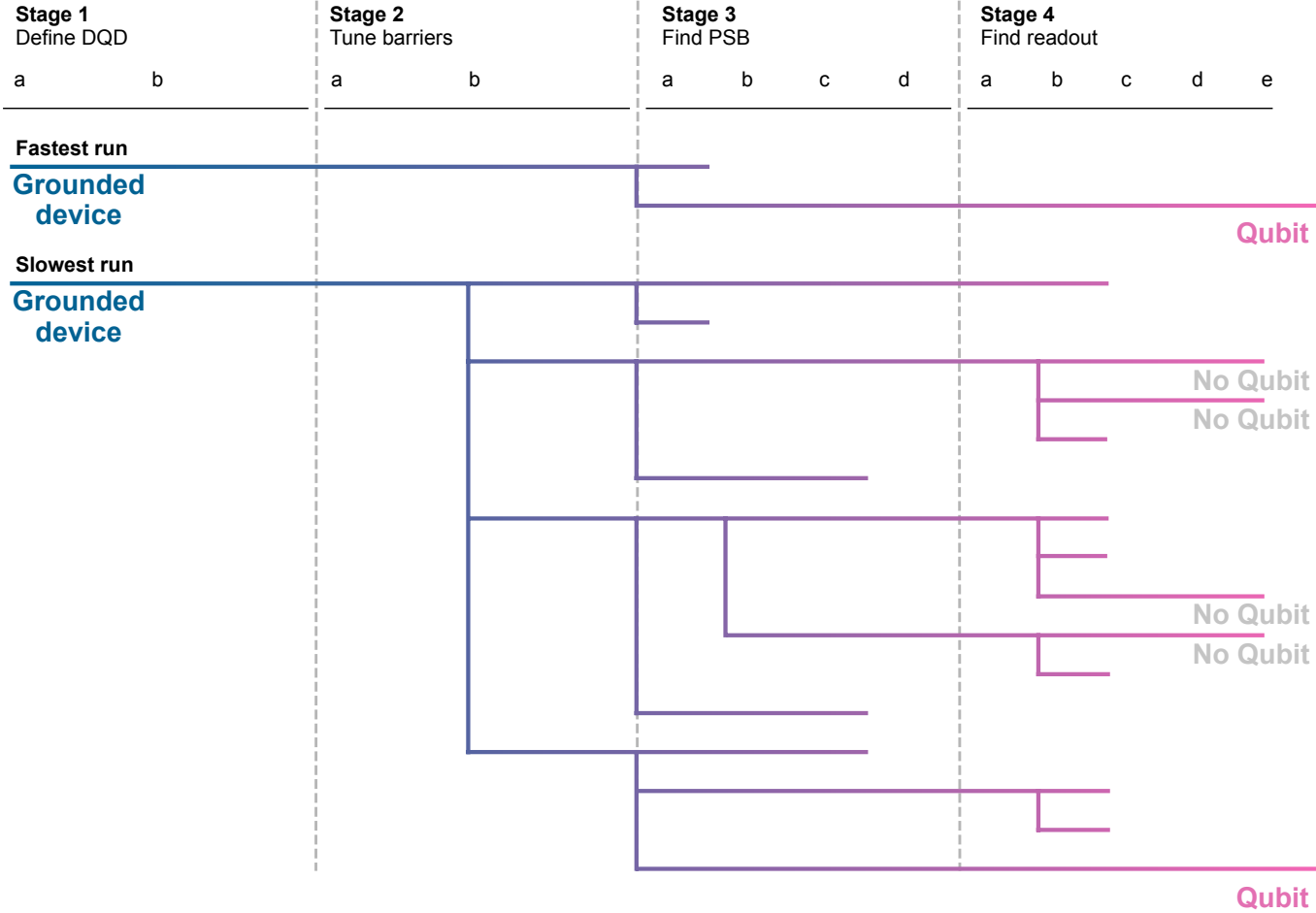

FIG. S3. **Examples of search trees from full runs.** The fastest run only has two branches and then successfully found a qubit. The slowest run explored much more, with several branches reaching all the way to qubit measurements. However, only the last branch shows conclusive qubit signatures. We rejected the first tries as noise.

S5. MODULAR FRAMEWORK

We implemented several design choices to standardize the framework across all stages, achieving a cohesive and modular architecture. Each stage exhibits these common characteristics:

1. Stage structure

1. Integration with QCoDeS [3]: All stages have access to the station object of QCoDeS, allowing each stage to take measurements and change experimental parameters.

2. Data management: A data access object manages (in addition to the QCoDeS database) costum data saving, such as the structure of the tree that was created so far, and automated documentation of the run.

3. Hyperparameter configuration: Each stage possesses specifically tailored hyper-parameters to fulfill its requirements, as detailed in the Section S2.

4. Candidate management: A list of candidates that were passed to a stage and that are sent off to another stage is kept.

a. Functions

1. Investigation function: Stages are primarily invoked through an `investigate` function, managing candidate lists, orchestrating measurements and data analysis, and forwarding candidates to the subsequent stage.

2. Experimental setup: A `prepare_experiment` function sets up the experimental parameters as needed, for example, setting certain voltages, ramping the magnet to a starting position, or stopping the AWG from outputting a pulse sequence.

3. Experiment execution: The function `perform_experiment` unction conducts the stage-specific measurements.

4. Data analysis: `determine_candidate` function analyses the acquired data to assemble a viable set of candidates for further exploration.

2. Candidates

1. Data association: Once a stage has taken data relating to a specific candidate, it will keep a note of the global unique identifier (GUID) that is recorded in the QCoDeS database.

2. Parameter storage: Critical parameter information is stored flexibly in a dictionary format to adapt to various experimental scenarios.

3. Metadata storage: Candidates carry metadata, such as their position within the search tree.

4. Stage timing: The duration required for each stage's process is recorded.

5. Resulting candidates list: A distinct list is maintained for candidates resulting from the stage's analysis.

---

[1] M. Kotzagiannidis, J. Schuff, and N. Korda, arXiv preprint arXiv:2312.03110 (2023).
[2] T. Pavlidis, Algorithms for graphics and image processing (Springer Science & Business Media, 2012).
[3] Copenhagen / Delft / Sydney / Microsoft quantum computing consortium, "QCoDeS," .