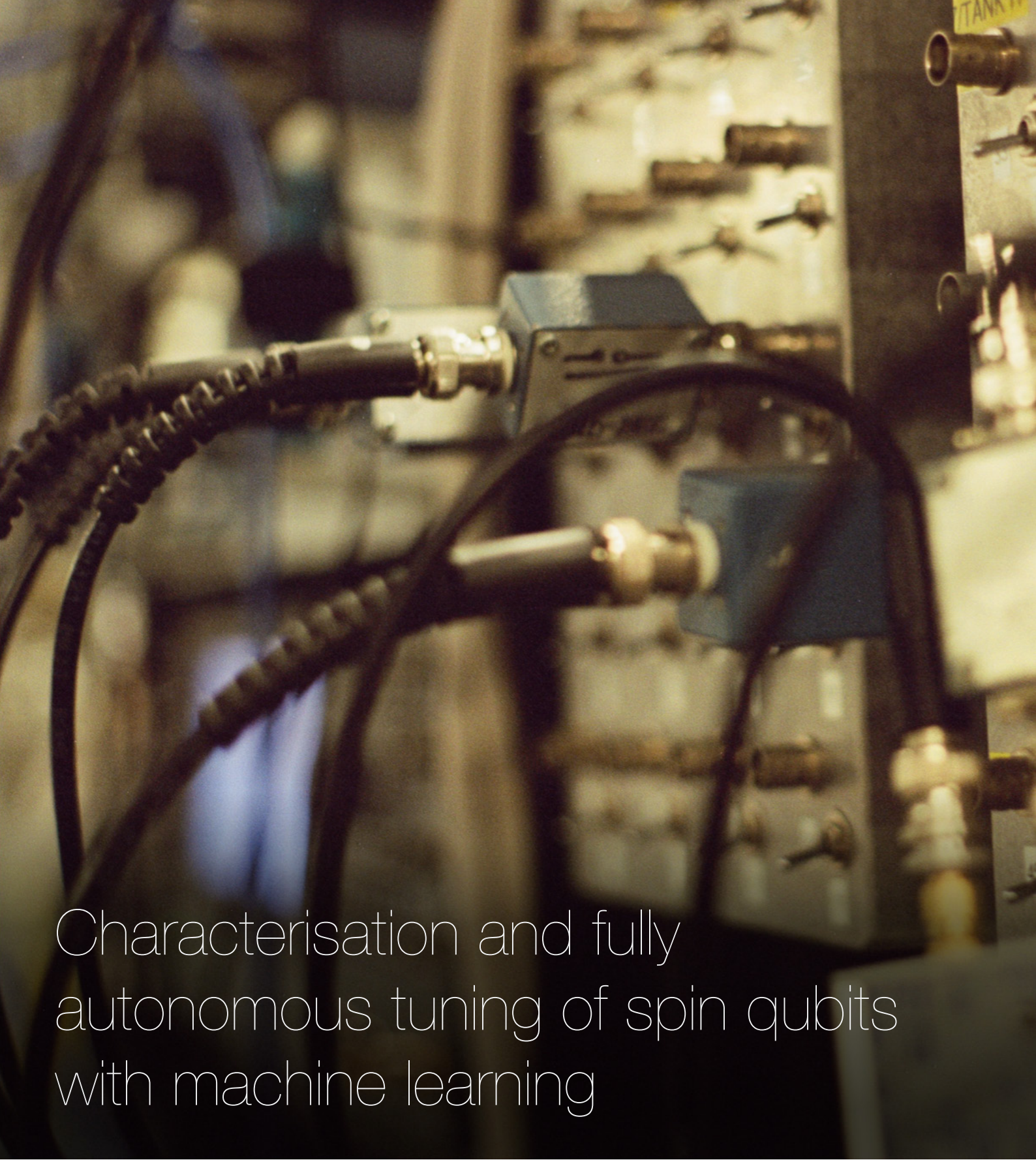


Characterisation and fully autonomous tuning of spin qubits with machine learning

Jonas Schuff
Harris Manchester College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy
Trinity Term 2024

UNIVERSITY OF OXFORD

Acknowledgements

There are countless individuals who have supported me throughout this journey. While I cannot thank everyone in detail here, I offer my sincere gratitude to all who have been part of this path.

Reflecting on my journey, I often find it challenging to comprehend the twists and turns it has taken. As Peter Medawar aptly noted, "What actually turns one's thoughts in one direction rather than another is to quite a large extent luck and opportunity, and the influence of the people you happen to be working with" [1]. These acknowledgements are dedicated to those influential people. Thank you all.

First and foremost, I would like to thank Natalia Ares, who made so much possible for me: from enabling me to come to Oxford as a visiting student to supporting my DPhil projects. Her guidance, encouragement, and trust have been invaluable.

I am grateful to those who offered advice and guidance along the way: Andrew Briggs, Florian Vigneau, Hyungil Moon, Yikai Yang, and Federico Fedele. Your insights and support have been essential.

To those who walked the path alongside me: Jonathan Dexter, Josh King, Sebastian Orbell, Anna Pearson, Joel Pendleton, Marek Rychetský, Lucas Schorling, Brandon Severin, Pranav Vaidhyanathan, Vivek Wadhia. Your companionship enriched these years immensely.

Cornelius Carlsson, thank you for your company during the late-night shifts. Kushagra Aggarwal, your presence helped me navigate both the office and our time in India. David Craig, thank you for your wide-ranging wisdom and for introducing me to the charm of a Northern Irish accent. Joseph Hickie, you always manage to lift my spirits; I wish you the very best in your future endeavors. Dominic Lennon, our discussions on tuning problems have been a continuous source of inspiration. Nina van Esbroeck, thank you for welcoming me to the group as a fellow visiting student. Barnaby van Straaten, you were always ready to tackle any problem or challenge I brought to you. I extend my heartfelt thanks to all of you.

I am thankful for my academic collaborators at IST Austria, Georgios Katsaros and Jaime Saez Mollejo, for the insightful discussions and guidance. At the University of Basel, my sincere gratitude goes to Dominik Zumbühl, Taras Patlatiuk, Simon Geyer, and Rafael Eggli for their invaluable contributions to the experiments and enriching discussions. Special thanks to Miguel Carballido for generously offering

his cherished device, which has turned into an absolute hero device—well done! The jokes and giggles we shared made the journey all the more enjoyable. This dissertation would not have been possible without your support and camaraderie.

To the Mind Foundry team, your expertise and collaboration were instrumental in developing the tuning pipeline. Thank you to Juan Carlos Calvo, Marco Caselli, Nathan Korda, Madeleine Kotzagiannidis, and Jacob Rawling.

Werner Vogelsang, you were the first to support my academic career, and your teaching of the foundations of quantum mechanics has left an indelible mark on me. Daniel Braun, thank you for entrusting me with my first project at the intersection of quantum and machine learning. Lukas Fiderer, your collaboration and perseverance in tackling that project were irreplaceable. Thank you all for helping me at the beginning of my academic path.

I am grateful to Riverlane for their financial support and the stimulating internship experience. Special thanks to Anton Buyskikh, Nick Chen, Gabriel Gallardo, Samin Ishtiaq, and Eve Townsend.

I would like to thank the EPSRC for funding my DPhil studies. I also gratefully acknowledge the Otto A. Wipprecht Stiftung and the E. W. Kuhlmann Stiftung for their generous financial assistance, which enabled me to come to Oxford as a visiting student. Your support was instrumental in making this path possible.

One of the best parts of studying in Oxford is undoubtedly the people you meet. To the friends I made at Harris Manchester College, I cannot put into words how much I cherish the memories we created together. Heartfelt thanks to Bianca, Femke, Jan, Jan, Josef, Katie, Lea, Leoni, Leonie, Madeline, Marie, Maxime, Melody, Monty, and Penny. Your friendship and shared experiences made these years unforgettable.

To the wonderful friends I met playing ultimate frisbee: Kay, who is one of the best friends one could ever hope for; Mateo, who is there for me when it matters most; and Andrew, whose inspiring attitude and community-building talent continually motivate me.

I formed and deepened friendships while hiking the West Highland Way with Arlette, Hameem, and Merlin. Our shared challenges strengthened our bonds. Thank you for the conversations, both deep and light-hearted.

To the exceptional housemates I lived with in Jericho—Ania, Carlotta, Delia, Haruna, Laura, Nico, and Sophie: You provided me with a safe space to share my thoughts and a sanctuary during the challenges of the pandemic. You transformed our house into a home.

To my friends from my undergraduate years, Pano and Santiago: I can always count on you, and your unwavering support means the world to me.

A special thanks to Andrew, Cornelius, Joseph, Kay, Lucas, and Natalia for providing feedback on this dissertation.

Abschließend gilt mein tiefster Dank meinen Eltern, die mich von Anfang bis Ende bedingungslos ermutigt haben. Eure Unterstützung ist mein Fundament.

Danke.

Abstract

Spanning over two decades, the study of qubits in semiconductors for quantum computing has yielded significant breakthroughs. 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. This presents a real 'needle in the haystack' problem, which, until now, has resisted complete automation due to device variability, fabrication imperfections, and limited data availability.

In this thesis, I first present a machine learning algorithm capable of automatically identifying Pauli spin blockade (PSB) using charge transport measurements. PSB can be employed as a resource for spin qubit initialisation and readout but it can be difficult to identify. The scarcity of PSB data is circumvented by training the algorithm with simulated data and by using cross-device validation. I demonstrate the approach on a silicon field-effect transistor device and report a high accuracy on test devices, giving evidence that the approach is robust to device variability.

Then, I present the first fully autonomous tuning of a semiconductor qubit, from a grounded device to Rabi oscillations, a clear indication of successful qubit operation. This automation, achieved without human intervention, is demonstrated in a Ge/Si core/shell nanowire device and integrates deep learning, Bayesian optimisation, and computer vision techniques.

Further, I demonstrate the potential of full automation by characterising how the Rabi frequency and g-factor depend on barrier gate voltages for qubits in four different charge transitions. The data reveals that certain features in charge transport measurements can enhance automated tuning processes by recognising the conditions for successful readout.

Finally, I present the autonomous optimisation of two qubits for maximal Rabi frequency. This optimisation extends the tuning algorithm, enabling it to enhance the performance of a qubit.

I hope the mass tuning and characterisation of qubits, enabled by the methods in this thesis, will create a productive feedback loop between measurement and fabrication processes. Wafer-scale, high-throughput characterisation of quantum devices can mitigate device variability and contribute to their scalability. The continued evolution of these methods will support the growing demands of the field, ultimately – and hopefully – leading to practical and powerful quantum computers.

Contents

Appendices

List of Abbreviations

1D, 2D, 3D . . One-, two-, or three-dimensional

AC . . . . . Alternating current

AWG . . . . . Arbitrary waveform generator

BO . . . . . Bayesian optimisation

CI . . . . . Constant interaction (model)

CMOS . . . . . Complementary metal-oxide semiconductor

CNN . . . . . Convolutional neural network

DAC . . . . . Digital-to-analog converter

DC . . . . . Direct current

DQD . . . . . Double quantum dot

EDSR . . . . . Electric dipole spin resonance

EI . . . . . Expected improvement

ESR . . . . . Electron spin resonance

FET . . . . . Field-effect electron transistor

GP . . . . . Gaussian process

GUID . . . . . Global unique identifier

LI . . . . . Lock-in (amplification)

ML . . . . . Machine learning

PI . . . . . Probability of improvement

PSB . . . . . Pauli spin blockade

SEM . . . . . Scanning electron microscope

UCB . . . . . Upper confidence bound

The only way of discovering the limits of the possible is to venture a little way past them into the impossible.

— Arthur C. Clarke [2]

1 Introduction

Richard Feynman once pondered the challenge of simulating quantum mechanics with computers. He concluded that to truly mimic nature—especially the quantum realm—we need a quantum-based simulation [3]. In other words, instead of relying on classical computer bits that are either 0 or 1, we need quantum bits, or qubits, which can exist as a superposition of both 0 and 1 simultaneously.

One of the most exciting prospects for quantum computers lies in their ability to simulate quantum systems. This has potential applications in fields like chemistry and materials science, where quantum computers can be used to calculate the ground state energies of molecules [4, 5]. However, the real game-changer came from Peter Shor, who demonstrated that a large-scale quantum computer could factorise large products of primes [6]. This breakthrough implied that quantum computers could potentially break current encryption methods, catching the attention of scientists and intelligence agencies alike. Beyond cryptography, other promising applications have been proposed, including optimisation [7] and machine learning [8, 9].

Building a quantum computer is no small feat, and various approaches have been explored. Regardless of the method, all implementations must meet the stringent criteria outlined by DiVincenzo [10]. These include scalability, reliable qubit initialisation, long coherence times, a universal set of quantum gates, and qubit-specific measurement.

Recent advances underscore the potential of qubits in semiconductors [11] for universal quantum computation [12–19]. These include the achievement of two-qubit gates showcasing fidelities that surpass thresholds essential for fault-tolerant computing [16, 20, 21], and hot qubits that address the bottleneck of millikelvin refrigeration [14, 22, 23]. Strides in wafer-scale manufacturing of these devices [24, 25] and their efficient testing at cryogenic temperatures [26, 27] further highlight the rapid progress in this field. Still, semiconductor quantum circuits are limited to at most six simultaneous qubits [28] 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 [29–32], navigating to specific charge transitions [33–40], fine-tuning of charge transport features [41] or the inter-dot tunnel couplings [42, 43], as well as device characterisation [44–46]. Offering glimpses of the potential of machine learning for full qubit tuning automation, however, the key challenge remains unaddressed.

I want to change that with this dissertation.

Before we embark on this adventure, let me introduce a metaphor to illuminate the concepts behind semiconductor device tuning. Imagine the tuning problem as trying to find a landmark, such as the Radcliffe Camera in Oxford. To achieve this, you need two key elements:

1. A method to determine your current location. In this metaphor, this means being able to read street signs or ask a passerby for directions.

2. A strategy for deciding your next move. With a map of Oxford and knowledge of your current location and destination, you can determine the best route to take.

Similarly, a successful tuning algorithm consists of classifiers to ascertain the current state of a device and a strategy to decide the next steps based on the classifiers' outcomes.

I will first cover the fundamentals of spin qubits in quantum dots, the machine learning techniques employed in this dissertation, and review the existing tuning literature in Chapter 2.5.8.

Addressing the first key element of automated tuning (determining your location), we will encounter the first obstacle that hinders the way forward on our journey. This challenge is the classification of Pauli spin blockade, crucial for the initialisation and readout of spin qubits. Accurate recognition is vital for automated tuning. In Chapter 3.4, I delve into this problem, highlighting the pitfalls and demonstrating how to navigate around them.

Armed with this knowledge, I will guide you through the journey from a grounded device to observing Rabi oscillations, all without human intervention. This feat requires a fusion of Bayesian optimisation, deep learning, computer vision techniques, and a novel framework that integrates these elements. This comprehensive process is detailed in Chapter 4.4.

What lies beyond these triumphs? As with any scientific endeavor, we often end with more questions than we began with, albeit different and more refined ones. One application of the tuning algorithm, beyond bringing up new devices, is to extensively characterise qubit devices. This exploration is presented in Chapter 5.2.

The final leg of our journey is about optimising the performance of a qubit. I show how to autonomously maximise the Rabi frequency of a spin qubit, detailed in Chapter 6.4.

The methods and findings in this dissertation lay the foundation for more efficient and scalable quantum computing, addressing long-standing challenges in qubit tuning and device characterisation, and bringing us closer to realising practical quantum computers.

And I knew exactly what to do. But in a much more real sense, I had no idea what to do.

— Michael Scott

2 Background

Contents

2.1 Quantum dots

This thesis deals with spin qubits which are encoded in the spin of charges that are confined in semiconductor quantum dots (QDs) [11]. A QD is a potential well that confines charges in three dimensions, leading to discrete energy levels. Two such successive levels are spaced by the addition energy

$$E_{\text{add}} = E_C + \Delta E \tag{2.1}$$

with the purely electrostatic charging energy E_C and the energy spacing between two quantum levels ΔE [47]. The charging energy is defined by the electron charge e and the total capacitance C of the QD [48]:

$$E_C = \frac{e^2}{C}. \tag{2.2}$$

There are various ways how such a confinement can be achieved. Commonly, a combination of electrostatic gating and band structure engineering is used. For this dissertation, QDs defined with metallic gate electrodes are the most relevant. The QD can be located within a sandwich of semiconductor materials (such as a 2D heterostructure or a 1D nanowire), or it can be trapped at the interface between a semiconductor and an insulating layer (e.g., for silicon FinFETs) [13, 49–51].

Once a QD is formed, it can be probed using tunnel-coupled source and drain reservoirs. For sufficiently low temperatures $T \ll \frac{E_\text{C}}{k_\text{B}}$ and small bias voltages $V_\text{bias} < \frac{E_\text{C}}{e}$, where k_B is the Boltzmann constant, only single charge transport is possible. If there are no energy levels energetically available, the charge transport is blocked, the so called Coulomb blockade. Using a plunger gate, the potential of the QD can be adjusted until an energy level becomes available and Coulomb blockade is lifted, allowing for sequential tunneling of single charges.

In double quantum dots (DQD), two QDs are tunnel-coupled. Electron transport then depends on the alignment of energy levels in both dots and the strength of the coupling between them (see section 2.1.2). When plotting the current as a function

of the plunger gate voltages controlling each dot, regions of conductance emerge with triangular shape, so called bias triangles. They can be analysed to determine gate lever arms, i.e., how a change in the voltage applied to a gate electrode shifts the energy levels of a quantum dot, as well as charging and orbital energies [50, 52].

I will describe the features seen in transport measurements in more detail, as their interpretation is central to this dissertation. Thinking back on the landmark metaphor: We want to understand their signatures to orient ourselves.

2.1.1 Single quantum dots

A single quantum dot (SQD) is an isolated region of electrons or holes connected to source and drain reservoirs. The constant interaction (CI) model provides a simplified description of this system (see Figure 2.1) [47, 53]. Figure 2.1**a** illustrates a SQD in the CI model. This model assumes that Coulomb interactions within the dot and between the dot and its environment can be described by a single capacitance $C = C_s + C_d + C_g$, where C_s, C_d, and C_g are the capacitances between the dot and the source, drain, and gate, respectively. In this model, the capacitance is independent of the number of charges already on the dot.

The electrochemical potential $\mu(N)$, which is the energy difference between a dot with N charges and a dot with $N-1$ charges, depends linearly on the gate voltage while the energy has a quadratic dependence. The dependence is the same for all N, so we can move the whole "ladder" of electrochemical potentials up and down while the difference between levels remain constant.

Charges tunnelling through the dot crucially depend on the electrochemical potential levels of the source μ_s and drain μ_d to align appropriately with the electrochemical potential $\mu(N)$ of the dot. When applying a bias voltage $V_{\text{bias}} = V_s - V_d$, an energy window, also called bias window, $\mu_s - \mu_d = -|e|V_{\text{bias}}$ between source and drain opens up. If the electrochemical potential μ of a transition between two states lies in this window, a charge can tunnel into the dot from the source and then off again into the drain reservoir.

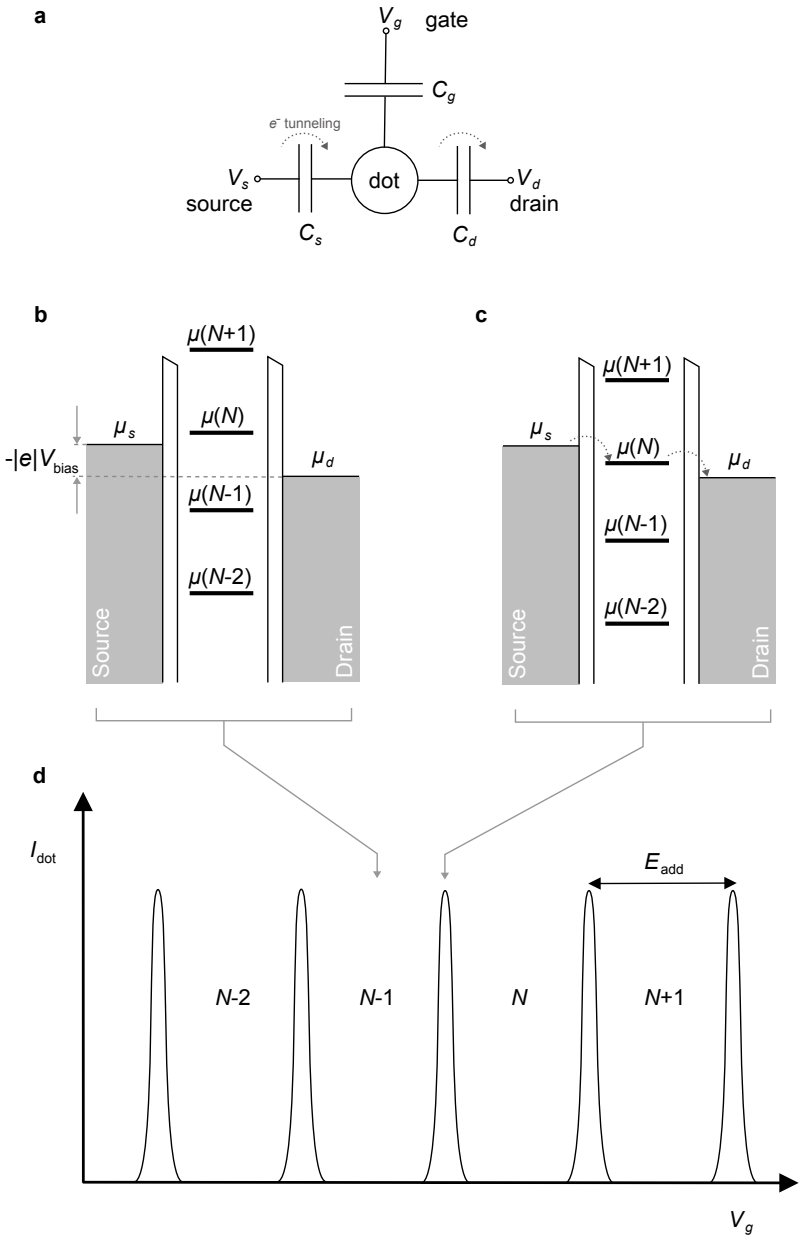


Figure 2.1: Single quantum dot. a Constant interaction model of a single quantum dot. The dot is tunnel-coupled to a source and a drain, modelled with capacitances C_s and C_d. Additionally, the quantum dot is coupled to a gate through the capacitance C_g. The gate controls the electrochemical potential of the dot. Charges can tunnel between the dot and source and drain. **b** Single quantum dot in Coloumb blockade. No electrochemical potential $\mu(N)$ lies within the bias window. **c** Lift of Coloumb blockade. Current can flow because the levels align and single charge transport is activated. **d** Current through the dot I_{dot} when varying the gate voltage V_g, which changes the electrochemical potential of the dot. Thus, the current flow and the number of charges on the dot can be controlled through the gate voltage. Adapted from Ref. [47].

Transport through the dot is possible if the electrochemical potential of a transition between successive ground states lies within the bias window: $\mu_s \geq \mu(N) \geq \mu_d$. If this condition is not satisfied, charges cannot tunnel on or off the dot. Therefore the number of charges on the dot remains fixed. This is called Coloumb blockade. Figure 2.1**b** provides an example of this.

Coloumb blockade can be lifted by changing the gate voltage V_g. Once a level lies within the bias window, a charge can tunnel from the source to the dot, changing the number of charges from $N-1$ to N. After that, a charge can tunnel to the drain, and the cycle starts again. This is called single-electron or single-hole transport, as illustrated in Figure 2.1**c**.

By sweeping the gate voltage, we can obtain the current trace shown in Figure 2.1**d**. Transport occurs whenever a level aligns appropriately, resulting in a peak. Between the peaks, the dot is in Coulomb blockade. This mechanism allows precise control over the number of charges on the quantum dot. The peaks in this type of measurement are referred to as Coulomb peaks.

2.1.2 Double quantum dots

We consider a double quantum dot (DQD) as two SQDs that are connected in series (see Fig. 2.2). A circuit in the CI model is shown in Fig. 2.2**a**. Dot 1 is tunnel-coupled with the source to capacitance C_s. It is also tunnel-coupled to dot 2 with capacitance C_m. Dot 2 is tunnel-coupled to the drain with capacitance C_d. Each dot has its designated gate, where a voltage $V_{g,1}$ (gate 1) or $V_{g,2}$ (gate 2) can be applied. These gates are coupled to their designated dots but can also exhibit cross-coupling, leading to cross-capacitances labeled C_{ij} where i indicates the gate and j indicates the dot.

We can map out the number of charges on both dots (N_1, N_2) depending on the gate voltages $V_{g,1}$, $V_{g,2}$. This mapping results in a charge stability diagram, shown in Fig. 2.2**b, c**. Lines indicate voltages where the number of charges on one dot changes. In the case of uncoupled dots, i.e. $C_m = 0$, and no cross-capacitances, i.e. $C_{12} = C_{21} = 0$, the perpendicular lines indicate that changing one gate voltage

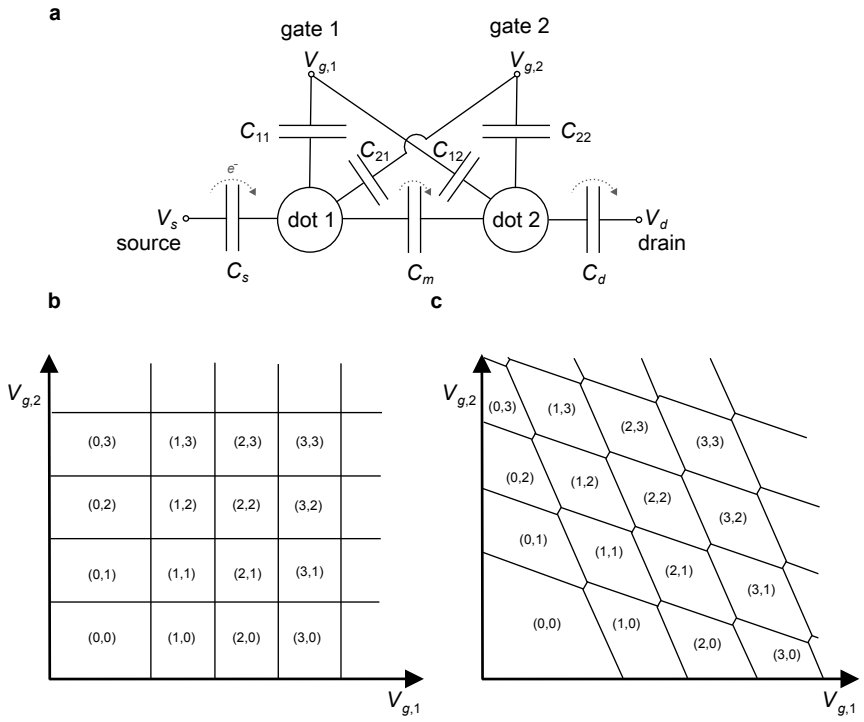


Figure 2.2: Double quantum dot a Constant interaction model of a double quantum dot. The double dot is tunnel-coupled to a source and a drain, modelled with capacitances C_s and C_d. Each dot is coupled to a gate to which gate voltages $V_{g,1}$ and $V_{g,2}$ can be applied. Those gates couple capacitively with their designated dot with C_{11} and C_{22}, and can also couple with the other dot, giving cross-capacitances C_{12} and C_{21}. The gates control the electrochemical potential of each dot. The dots are also tunnel-coupled with C_m. Charges may tunnel between source and dot 1, drain and dot 2, or between the 2 dots. **b**, **c** Charge stability diagrams for a double quantum dot. The number of charges in the double dot (N_1, N_2) for $N_1(N_2)$ charges in the first (second) dot are shown. At the lines, the number of charges in at least one dot changes. **b** Uncoupled dots with no cross-capacitances, $C_m = C_{12} = C_{21} = 0$. **c** Coupling of dots ($C_m > 0$) leads to splitting of cross-points and cross-capacitances ($C_{12}, C_{21} > 0$) lead to a honeycomb pattern. Adapted from Ref. [47].

does not affect the electrochemical potential of the other dot, and adding a charge to one dot does not influence the other (Fig. 2.2**b**).

However, double dots can be coupled, i.e. $C_m > 0$, meaning that adding a charge to one dot shifts the electrochemical potential of the other. This coupling leads to the splitting of cross points into two separate points known as triple points, where three different charge configurations become degenerate. If the cross-capacitances C_{12} and C_{21} are non-zero, the lines skew, resulting in a hexagonal pattern called a honeycomb lattice (Fig. 2.2**c**).

Low bias regime A detailed view of the charge stability diagram for one pair of triple points is shown in Fig. 2.3**a**. The separation of the triple points is due to the electrostatic coupling energy E_{C_m}. This coupling energy describes the change in energy of one dot when a charge is added to the other dot. The solid line connecting both triple points corresponds to the electrochemical potential of each dot being shifted simultaneously, meaning that in the example in the figure the electrochemical potentials states (0,1) and (1,0) align. The remaining lines indicate where the potential of one dot aligns with either source or drain. At the triple points, a charge can tunnel through both dots from source to drain, allowing current flow. At all other points, the double dot is in Coloumb blockade.

In the low-bias regime, only aligned electrochemical potential levels are involved in transport processes. This is called elastic tunnelling. Misaligned levels can still allow tunneling, known as inelastic tunneling [47]. Inelastic tunneling requires energy exchange with the environment to conserve energy. One example is photon-assisted tunneling, where photons from microwave or radio-frequency radiation are absorbed [54]. Energy emission typically occurs through phonons in the surrounding lattice [47]. Inelastic tunneling is a second-order process and contributes to the current, dominating charge transport when aligned levels do not permit elastic tunneling.

High bias regime Opening the bias window expands the triple points into bias triangles, as illustrated in Fig. 2.3**b**. Within the triangle, the electrochemical potential levels align such that at least one path, i.e. a succession of charge configurations, is allowed through the dot causing charge transport from source to drain. Assuming the drain is grounded, at the lower left tip of triangles the electrochemical potentials $\mu_1(1,0)$ and $\mu_2(0,1)$ align. Moving up the left side of the triangle, the electrochemical potential of dot 1 remains aligned with the source, $\mu_1(1,0) = -|e|V_{\text{bias}}$. As the electrochemical potential ladder of dot 2 lowers, an excited state may enter the bias window, leading to increased current, indicated by darker regions within the triangle. From the tip towards the base along the line where $\mu_2(0,1) = 0$, the electrochemical potential ladder of dot 1 lowers until the excited state of dot 2 is

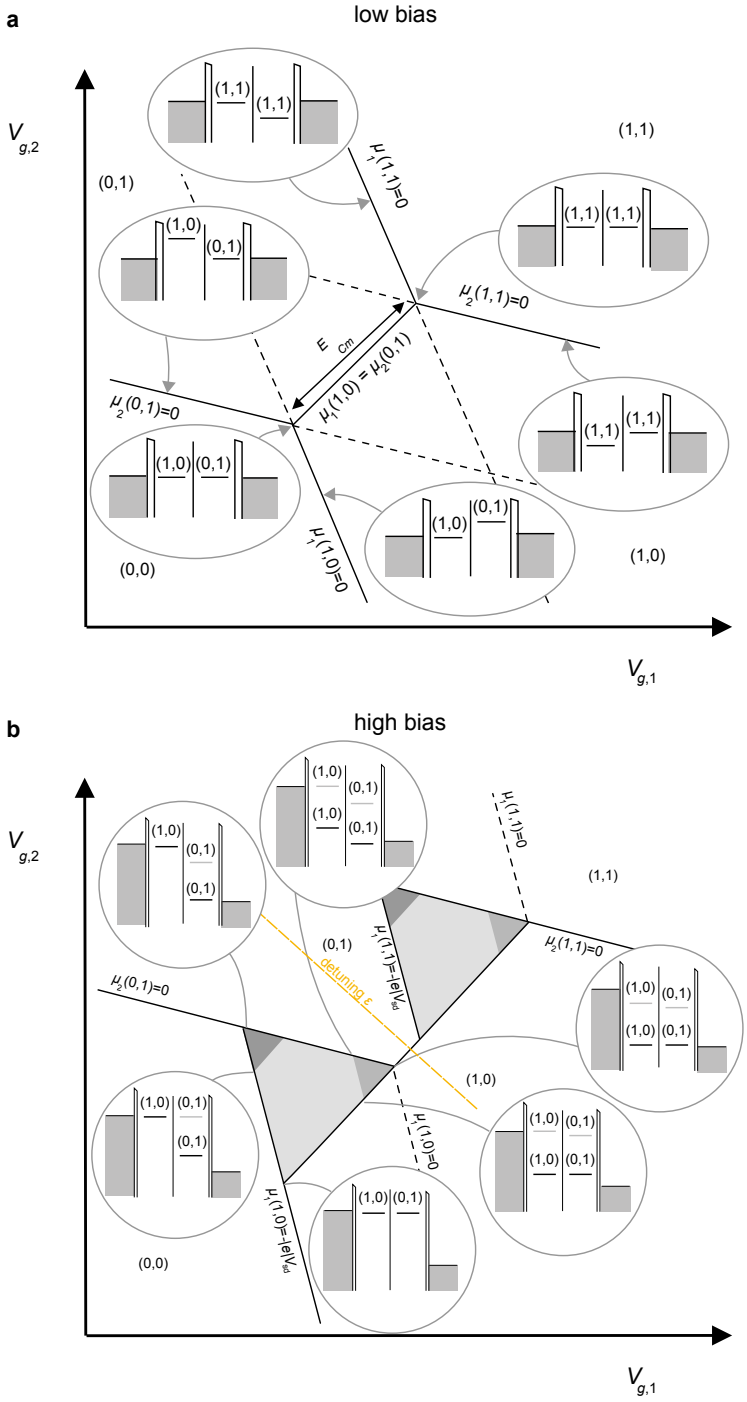


Figure 2.3: Charge stability diagrams. a Detailed view of the charge stability diagram around one pair of triple points. Lines indicate where the electrochemical potential of one dot aligns with the source or drain, or with each other in the case of the line connecting the triple points. Schematic representations of the electrochemical potential alignment is shown for several configuations of $V_{g,1}$ and $V_{g,2}$. At the triple points the electrochemical potentials align such that current can flow. **b** Bias triangles. For a non-zero bias window $-|e|V_{\text{bias}}$ triple points expand into bias triangles. Within those triangles, charge transport is possible and current can flow. If excited states align within the bias window, charge transport through those states becomes accessible, leading to a higher current indicated by darker regions in the triangle. Between the triangles, the detuning axis is plotted as a dashed orange line. Along this line, the levels of the dots only shift with respect to each other. Adapted from Ref. [47].

no longer accessible, causing a current drop. Further along this line, the excited state of dot 1 may enter the bias window, raising the current again.

The visibility of different regions within the bias triangles highly depends on factors such as the widths and height of the three tunnel barriers between the dots and the source and drain, relaxation within the dots, and the efficiency of inelastic tunnelling processes [47].

Another essential concept is the detuning axis, plotted in Fig. 2.3**b** on the right triangle as a dashed orange line perpendicular to the shared basis of both triangles. Along this axis, the total electrochemical potential of both dots does not change, but their relative alignment shifts. The magnitude of detuning is denoted by ε.

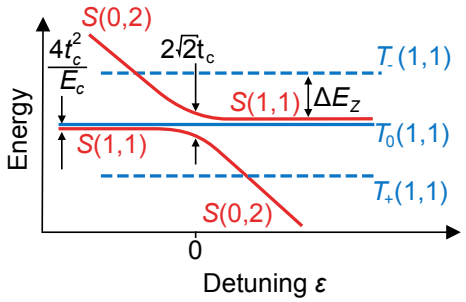


Figure 2.4: Energy spectrum of singlets and triplets in a DQD as a function of the detuning ε. The degeneracy between the $S(1,1)$ and $T(1,1)$ is lifted at finite tunnel coupling $t_c > 0$ and there is an avoided crossing due to the hybridisation of the S(1,1) and S(0,2) states. The triplet states split with a magnetic field $B > 0$ according to the Zeeman splitting ΔE_Z. Adapted from Ref. [47].

A DQD with two charges is of particular interest due to the spin selection rules that govern the quantum dot states and transitions. When two electrons occupy the DQD, their spins can form singlet or triplet states, with distinct energy configurations. These singlet-triplet states can be used to encode a qubit and for qubit readout via Pauli spin blockade, see section 2.1.3.

Two charges can either reside in the same QD, denoted as (0,2), or split between the two QDs, denoted as (1,1). They form singlet and triplet states. The energy spectrum of these singlets and triplets is shown for small detunings in Figure 2.4. The two QDs are considered independent if the interdot tunnel coupling t_c is negligible. For charges in a spin triplet configuration, they can only occupy the same QD if

one charge is in a higher orbital state, resulting in the triplet $T(0,2)$ states being separated by the single QD orbital level splitting. If the charges are distributed between the two dots and within the detuning range shown in Figure 2.4, they must be in one of the three triplet states: T_+, T_-, or T_0. These states are energetically degenerate at zero magnetic field but split in a finite magnetic field due to the Zeeman effect, resulting in an energy difference of $\Delta E_Z = g\mu_B B$.

A finite, spin-conserving tunnel coupling t_c between the two QDs, leads to an avoided crossing at $\varepsilon = 0$ of the $S(1,1)$ and $S(0,2)$ states which is proportional to t_c. In the absence of spin-mixing mechanisms the states $T(1,1)$ and $S(0,2)$ do not couple. For $\varepsilon < 0$, the T(1,1) state is energetically separated from the S(1,1) state by the singlet-triplet splitting $J \propto \frac{t_c^2}{E_C}$ [55].

2.1.3 Pauli spin blockade

The spin states of the charges in a DQD give rise to an effect called Pauli spin blockade (PSB). First experimentally observed in 2002 [56], PSB has since been used to explore fundamental physics, including spin T_1 relaxation times [57, 58], the coupling of electron spins to lattice nuclear spins [59], and spin-orbit effects [60]. In the realm of quantum information, PSB is often used for the read-out and initialisation of spin states [61–63].

PSB is a current rectification mechanism that occurs due to spin selection rules. Depending on the charge states involved in transport, the current through a DQD can flow only for either a positive or negative bias voltage and is blocked for the opposite bias voltage.

This effect is best understood by considering the triple point at the degeneracy of the (0,1), (1,1) and (0,2) charge configurations. Figure 2.5**a** shows the bias voltage setting in which the current can flow. Charges move through the dot in the cycle $(0,1) \rightarrow (0,2) \rightarrow (1,1) \rightarrow (0,1)$. There is always at least one charge in the right dot with an additional charge tunneling onto the dot from the drain reservoir. Depending on the available energy levels, a singlet or triplet is formed. Since the (1,1) transitions in the left dot are energetically accessible, the charge can tunnel to the left dot

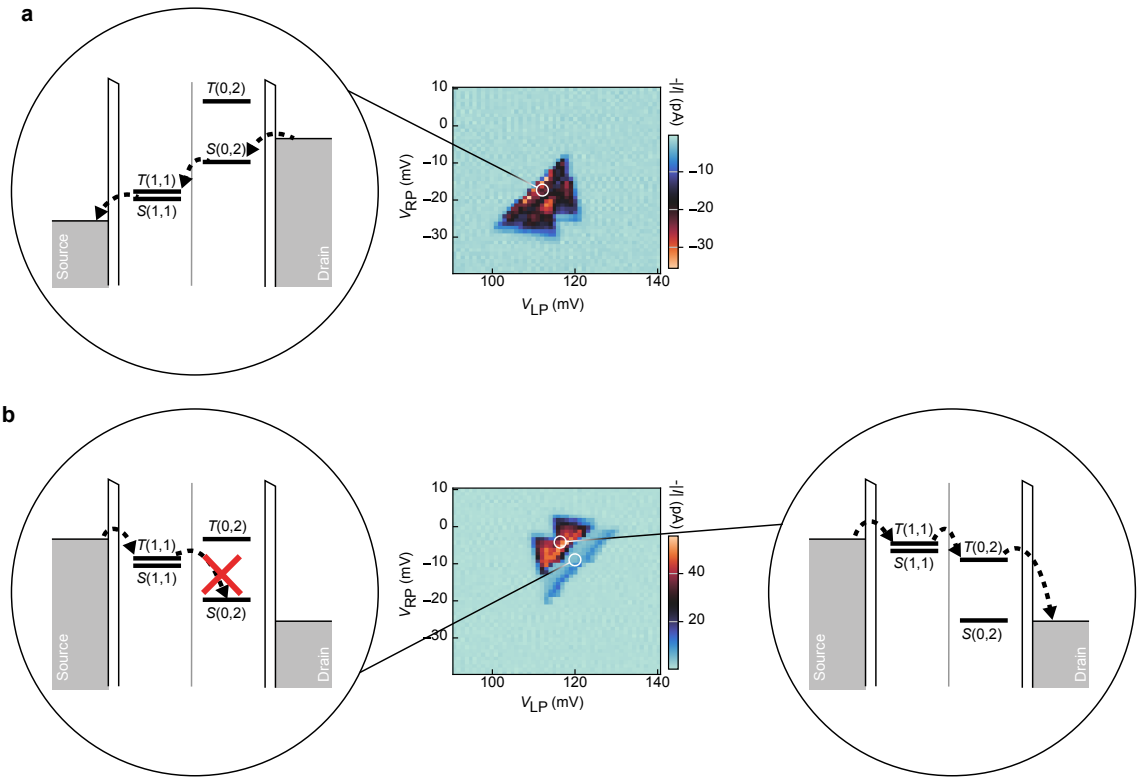


Figure 2.5: Demonstration of Pauli spin blockade. **a** Unblocked configuration. Charges can move through the DQD and a bias triangle can be observed. **b** Blocked configuration. When a charge enters from the left and forms a triplet with the electron in the right dot, it cannot tunnel to the right dot because of spin conservation. The electrochemical potentials corresponding to this case are shown on the left. At the tips of the triangles, the triplet state $T(0,2)$ becomes accessible and the blockade is lifted. This is shown on the right. The data is taken from the device in Chapters 4.4, 5.2 and 6.4.

and exit to the source reservoir. Here, the terms source and drain are used for consistency but do not carry their literal meanings.

When the bias voltage is reversed, charges follow the cycle $(0,1) \rightarrow (1,1) \rightarrow (0,2) \rightarrow (0,1)$. This is shown in Figure 2.5**b**. Again, at least one charge always stays in the right dot. When a charge enters from the drain, it forms either a singlet or a triplet with the charge in the right dot. If they form a singlet, i.e. $S(1,1)$, the configuration can access $S(0,2)$, allowing transport. However, if they form a triplet $T(1,1)$, the charge in the left dot cannot access the right dot. The $T(0,2)$ state is inaccessible due to its higher energy, and $S(0,2)$ is forbidden by the Pauli exclusion principle.

Since the relaxation time T_1 for a relaxation from $T(1,1)$ to $S(1,1)$ can reach milliseconds [47], the current is effectively blocked on experimental timescales. An

electron tunneling every 1 ms results in a current of approximately 160 fA, which is too low to be measured with conventional equipment.

If the bias window is large enough, the $T(0,2)$ state can become accessible at sufficiently large detunings. This is shown in the right diagram of Fig. 2.5(b).

The effect relies on the asymmetry of the energy levels, which can also cause PSB in the opposite direction. At the $(1,1)-(2,0)$ transition, the sign of the bias voltage where the current is blocked and allowed to flow is reversed compared to the $(1,1)-(0,2)$ transition.

Lift of PSB with a magnetic field

Any mechanism that induces a spin flip in one of the charges can remove PSB. Such a flip changes the (1,1) state from a triplet to a singlet configuration, which is not blocked. This can occur through deliberate spin rotations, which enable coherent spin manipulation essential for qubit functionality. PSB can also be lifted by hyperfine interactions, spin-flip cotunneling, differences in the g-factor between the two quantum dots, or spin-orbit interactions [64]. I will discuss the lift via spin-orbit interaction in more detail, as it predominantly affects the transport measurements in the devices used [65–68]. Further details on spin-orbit interaction can be found in Section 2.1.5.

To understand this intuitively, consider that without an external magnetic field, the spins align with the internal effective spin-orbit field $\mathbf{B}_{\text{eff}}$. When an external magnetic field is applied, the spins start to precess around it, enabling spin-flip tunneling.

For a more technical explanation, consider the energy diagram of the (1,1) and (0,2) states shown in Figure 2.6. When a magnetic field B is applied, the (1,1) triplet states split into three separate states: T_-, T_0, and T_+, due to the Zeeman effect. The $T_0(1,1)$ state mixes with $S(1,1)$, creating two hybrid states: a $T_0(1,1)$-like state $|M_\alpha\rangle$ and a $S(1,1)$-like state $|M_\beta\rangle$ [69, 70]. Transport is permitted through the singlet-like state $|M_\beta\rangle$ but is blocked through the triplet-like state $|M_\alpha\rangle$.

The spin aligns with the spin-orbit field, and when an external magnetic field is applied, it begins to precess. This precession enables transitions from $T_\pm(1,1)$ to

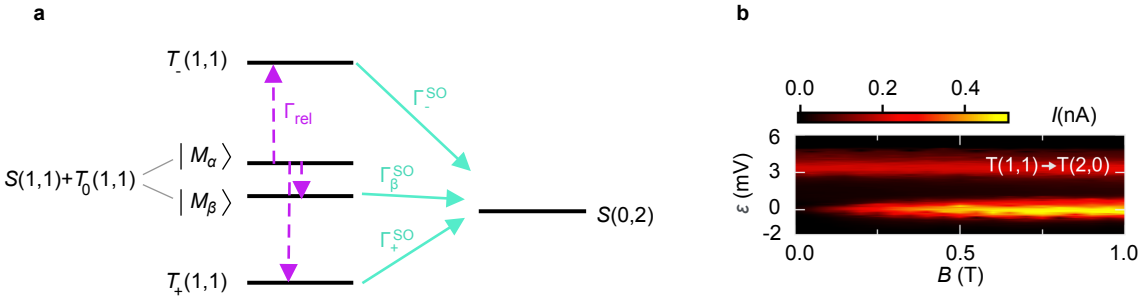


Figure 2.6: Lift of Pauli spin blockade with magnetic field. **a** Energy levels and transitions of the (1,1) states into the $S(0,2)$ state. $S(1,1)$ and $T_0(1,1)$ are coupled in presence of strong spin-orbit interaction and the mixed states $|M_\alpha\rangle$ and $|M_\beta\rangle$ become new basis states. The state $|M_\alpha\rangle$ can relax into another state with (1,1) charge configuration with a transition rate Γ_{rel}. Those states, i.e. $|M_\beta\rangle$ and $T_\pm(1,1)$, can relax to $S(0,2)$ with transition rates $\Gamma_\beta^{\mathrm{SO}}$ and $\Gamma_\pm^{\mathrm{SO}}$. **b** Current through a double quantum dot in a GeSi nanowire as a function of the detuning ε and an external magnetic field B with $V_{\mathrm{bias}} = -2.5$ mV. The spin blockade at $\varepsilon = 0$ is lifted with increasing magnetic field. The triplet transitions $T(1,1)$ to $T(0,2)$ are unaffected by the magnetic field. **a** is adapted and **b** reprinted from [71].

$S(0,2)$ and from $|M_\beta\rangle$, allowing for non-spin-conserving tunneling due to spin flips, as depicted in Figure 2.6**a**. The coupling rate $\Gamma_\beta^{\mathrm{SO}}$ between $|M_\beta\rangle$ and $S(0,2)$ is unaffected by the magnetic field strength, as both states have a spin component of zero. On the other hand, the transition rates $\Gamma_\pm^{\mathrm{SO}}$ from $T_\pm(1,1)$ to $S(0,2)$ increase with the magnetic field strength, since a stronger field leads to more rapid spin precession.

A transition involving $|M_\alpha\rangle$ is always blocked. Thus, at high fields transport is limited by the time the system spends in $|M_\alpha\rangle$. It will become unblocked once it relaxes into an unblocked state with a transition rate Γ_{rel}. The maximum leakage current I_{max} is therefore given by

$$I_{\mathrm{max}} = 4e\Gamma_{\mathrm{rel}} \tag{2.3}$$

in the high field limit where $\mu_B B \gg \hbar\sqrt{\Gamma_\pm^{\mathrm{SO}}\Gamma_{\mathrm{rel}}}$ with the Bohr magneton μ_B and the reduced Planck constant $\hbar$ [70].

At zero detuning, i.e. at the common base of the pair of bias triangles, the leakage current exhibits a Lorentzian shape:

$$I = I_{\mathrm{max}}\left(1 - \frac{8}{9}\frac{B_c^2}{B^2 + B_c^2}\right) \tag{2.4}$$

where B_c is related to the spin-orbit field [70].

Figure 2.6**b** shows transport measurements through a DQD as a function of detuning ε and the external magnetic field. As predicted by equation 2.4, at $\varepsilon = 0$ there is a transition from low current to a saturated current with increasing magnetic field. The inter dot triplet transitions $T(1,1)$ to $T(0,2)$ are unaffected by the magnetic field. Typical values of B_c range between $\sim$100 mT and $\sim$1 T [65].

2.1.4 Hole spins

The devices studied in this dissertation utilise electron holes instead of electrons, which is the absence of an electron in the valence band. When an electron is removed, the remaining state behaves as a positively charged particle. Electrons in the conductions band have s-like atomic wave functions and therefore a spin quantum number $s = \frac{1}{2}$. In contrast, holes are described by a p-type wave function. This results in reduced hyperfine interaction because of the reduced overlap of the wave function with the nuclear sites of their host. Additionally, they experience a strong spin orbit interaction which can be exploited to allow for all-electrical control of qubits [72–74].

When describing holes, we must account for their orbital angular momentum $l = 1$ in addition to their electron spin $s = \frac{1}{2}$. This means the system is characterised by the total angular momentum operator $J = L + S$, where L represents the orbital momentum operator and S represents the spin operator. However, when considering strain and confinement effects, the description becomes simpler, allowing hole quantum dots to be treated using the same formalism as electron quantum dots [75, 76]. An external magnetic field causes Zeeman splitting, with the energy expressed as $E_Z = g^* \mu_B B$, where g^* is the effective g-factor for the hole [55].

The all-electrical driving of hole spin qubits via strong spin orbit interaction reduces the fabrication complexity of devices by eliminating the need for additional components like microwave strip lines or magnetic field gradients required for electron spin qubits. Hole spin qubits have demonstrated ultra-fast Rabi frequencies (f_{Rabi}) in the gigahertz range [77], significantly surpassing the megahertz-range Rabi frequencies typically observed in electron spin qubits [78]. Despite these high operation speeds, hole spins may maintain coherence times (T_2) in the tens of microseconds [79],

comparable to those of electron spins, though there is often a trade-off between coherence time and driving speed [80]. Additionally, the effective g-factor g^* for holes is highly tunable through electric fields, providing greater control over qubit properties and enabling more flexible qubit manipulation [81]. These advantages make hole spins a promising platform for scalable and high-performance quantum computing.

2.1.5 Spin-orbit interaction

The term spin-orbit interaction (SOI) originates from the concept of an electron moving in the central potential of an atomic core. As the electron "orbits" the nucleus, it experiences an effective magnetic field due to its motion, leading to the coupling of its spin and orbital degrees of freedom. This effect is not limited to isolated atoms but is also observed in bulk solid-state materials, where electrons or electron holes experience an electric field generated by the charged atoms in the lattice.

In solid-state physics, there are two primary types of SOI arising from different kinds of inversion asymmetries. The first type is Dresselhaus SOI (DSOI), which results from bulk inversion asymmetry inherent in the crystal structure. The second type is Rashba SOI (RSOI), which is due to structural inversion asymmetry, often introduced by external factors such as interfaces or electric fields in heterostructures [47, 82, 83].

For the materials used in this dissertation, DSOI is absent due to the inversion-symmetric lattices. The dominating effect is of RSOI type. It stems from the strong confinement of the holes in at least once spatial dimension [49, 64].

2.2 Spin qubits

Spin qubit operation involves the precise manipulation of electron or hole spins within quantum dots, which are essential units in quantum information processing. There are various methods to achieve this manipulation. One might initially consider directly addressing the spin state using magnetic resonance techniques—applying oscillating magnetic fields to induce spin transitions between spin-up and spin-down states. However, this approach is challenging due to the small magnetic moment of

electron spins, which requires strong magnetic fields that are difficult to generate and localise at the nanoscale.

Alternative techniques offer improvements [47, 84, 85]. Of particular importance for this thesis is electric dipole spin resonance (EDSR) [64, 81]. EDSR leverages the coupling between the spin and an external electric field via spin-orbit interaction, allowing for spin manipulation using oscillating electric fields instead of magnetic ones. This method provides more efficient and localised control over qubit states because electric fields can be more easily confined and controlled at the quantum dot scale. See Section 2.2.2 for more details.

2.2.1 Single spin control

The essence of a qubit lies in its ability to encode information within a two-level system, which can be realised through different physical mechanisms, such as the charge states [86] or the spin states of electrons [61] in quantum dots. A spin-1/2 electron inherently offers two distinct spin states, creating a natural two-level system. Furthermore, other configurations, like the spin singlet and triplet states, also serve as effective two-level systems for qubit representation.

The Bloch sphere is a powerful geometric representation of a qubit, providing an intuitive visualisation of its quantum state (Figure 2.7**a**). In this model, the state of a qubit is depicted as a point on the surface of a unit sphere. The north and south poles of the sphere represent the classical binary states $|0\rangle$ and $|1\rangle$, respectively. Any point on the sphere's surface corresponds to a superposition of these basis states $|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, where α and β are complex numbers satisfying the normalisation condition $|\alpha|^2+|\beta|^2 = 1$. The angles θ and φ, known as the polar and azimuthal angles, define the position of the qubit state on the sphere. This representation is particularly useful for visualising qubit operations, such as rotations and state transitions.

Larmor precession and Rabi oscillations are fundamental phenomena in the study of qubits and their dynamics. A static magnetic field defines the quantisation axis of the qubit (represented by the blue dashed vertical line in Figure 2.7**a**). Larmor precession describes the evolution of a qubit's spin state under the influence of such

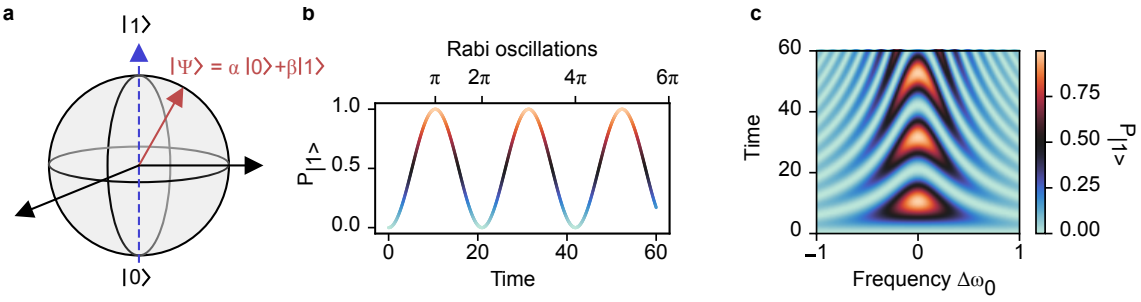


Figure 2.7: Rabi Oscillations in a two-level system a The Bloch sphere represents the qubit state $|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$, a superposition of the basis states $|0\rangle$ and $|1\rangle$ with complex coefficients α and β that satisfy the normalisation condition $|\alpha|^2 + |\beta|^2 = 1$. **b** The time evolution of the probability of finding the qubit in state $|1\rangle$ (refer to equation 2.5 with $\Delta\omega_0 = 0$). This probability oscillates between 0 and 1 with the Rabi frequency ω_R. **c** Here, the probability of the qubit being in state $|1\rangle$ is shown as a function of time and the detuning of the drive frequency. On resonance, the Rabi oscillations resemble those in **b**. When the drive frequency is off-resonance, the amplitude of the oscillations decreases, while the frequency increases. These dynamics are explained by equation 2.5. Adapted from Ref. [64].

a constant external magnetic field. In the Bloch sphere, this can be visualised as a precession around the north-south axis. The precession frequency is determined by the energy difference between the $|0\rangle$ and $|1\rangle$ states, which, for a spin-1/2 particle, corresponds to the Zeeman splitting.

Rabi oscillations occur when a qubit is exposed to an oscillating magnetic or electric field, typically in resonance with the qubit's energy splitting. When the qubit is driven on resonance with the Larmor frequency, this results in periodic transitions between the qubit's energy states, as seen in Figure 2.7**b**, allowing for control over the qubit's state through the duration and amplitude of the applied field. The frequency of these oscillations are known as the Rabi frequency ω_R. The Rabi formula captures these dynamics [87]:

$$P_{|1\rangle}(t) = \frac{\omega_\mathrm{R}^2}{\omega_\mathrm{R}^2 + (\Delta\omega_0)^2} \sin^2\left(\sqrt{\omega_\mathrm{R}^2 + (\Delta\omega_0)^2}\,\frac{t}{2}\right) \tag{2.5}$$

This formula describes the probability $P_{|1\rangle}(t)$ of a qubit being in the excited state $|1\rangle$ at time t during a Rabi oscillation. When detuned by $\Delta\omega_0$ the amplitude and frequency of this oscillation changes, with a maximum possible oscillation amplitude and slowest oscillations on resonance ($\Delta\omega_0 = 0$). The resulting pattern is shown in Figure 2.7**c** and is called "Rabi chevron".

2.2.2 Electric dipole spin resonance

Electron spin resonance (ESR) is a technique for controlling spin qubits in quantum computing. ESR involves the application of an alternating current (AC) magnetic field at a frequency matching the energy difference between the spin states of an electron. The AC magnetic field can be generated by driving a current through a coplanar stripline integrated into the device, near the location of the qubit [88]. Key challenges in ESR include maintaining generating magnetic fields with sufficiently high frequency to meet the resonance condition, preventing photon-assisted tunneling due to accompanying electric fields, and applying the method while mitigating heating effects at low temperatures [55].

Given the difficulties associated with ESR, alternative strategies focus on eliminating the reliance on high-frequency magnetic fields by using electric fields to control the spin. High-frequency electric fields are simpler to generate, apply, and confine at the nanoscale than magnetic fields. However, since electric fields do not directly interact with the spin, a mediating mechanism is needed to couple the spin to the electric dipole of the AC electric field.

The spin-orbit interaction experienced by holes enables the coupling of electric field oscillations to the spin of the hole qubit. This process, called electric dipole spin resonance (EDSR), allows for significantly faster manipulation of spin qubits compared to traditional ESR. EDSR is typically implemented by varying the electric potential of nearby metallic gates in a semiconductor device, causing the qubit to oscillate within its confinement potential. Due to the spin-orbit interaction, the hole experiences an effective AC magnetic field, leading to spin rotations [89].

2.3 Machine learning

Artificial intelligence as a research discipline started with a 2-month workshop at Dartmouth College, where a group of mathematicians and scientists invited fellow researchers to work on making machines “solve kinds of problems now reserved

for humans, and improve themselves"[1][90]. They thought they could significantly advance the field in "a summer".

The problems they were working on are still being researched today. This field has evolved significantly, driven by advances in computational power, the availability of vast datasets, and the development of sophisticated algorithms. Among these algorithms, neural networks and Bayesian optimisation stand out as particularly influential and widely applied methods.

Neural networks, inspired by the structure and function of the human brain, consist of layers of interconnected nodes or "neurons." These networks are capable of learning complex patterns and representations from data, making them suitable for a wide range of tasks, from image and speech recognition to natural language processing. The flexibility and power of neural networks stem from their ability to approximate virtually any function, given sufficient data and computational resources. I use neural networks to classify measurements throughout this dissertation.

Bayesian optimisation (BO), on the other hand, is a strategy for optimising complex functions that are expensive to evaluate. It employs principles from Bayesian statistics to build a probabilistic model of a function of interest and uses this model to make informed decisions about where to evaluate the function next. I use BO in Chapter 4.4 to guide a tuning algorithm and in Chapter 6.4 to optimise qubits.

2.3.1 Neural networks

Artificial neural networks (ANNs) are functions that map a feature space F to a label space L:

$$f : F \to L \tag{2.6}$$

Typically, $F \subset \mathbb{R}^n$ and $L \subset \mathbb{R}^k$, where $n, k \in \mathbb{N}$. Neural networks can approximate any function [91]. Features can be vectors or tensors, particularly when dealing with images, where the feature space can be $\mathbb{R}^{h_1} \times \mathbb{R}^{h_2} \times \mathbb{R}^c$ (height, width, and channels).

[1]The full proposal can be viewed at www-formal.stanford.edu/jmc/history/dartmouth.html.

Layer structure

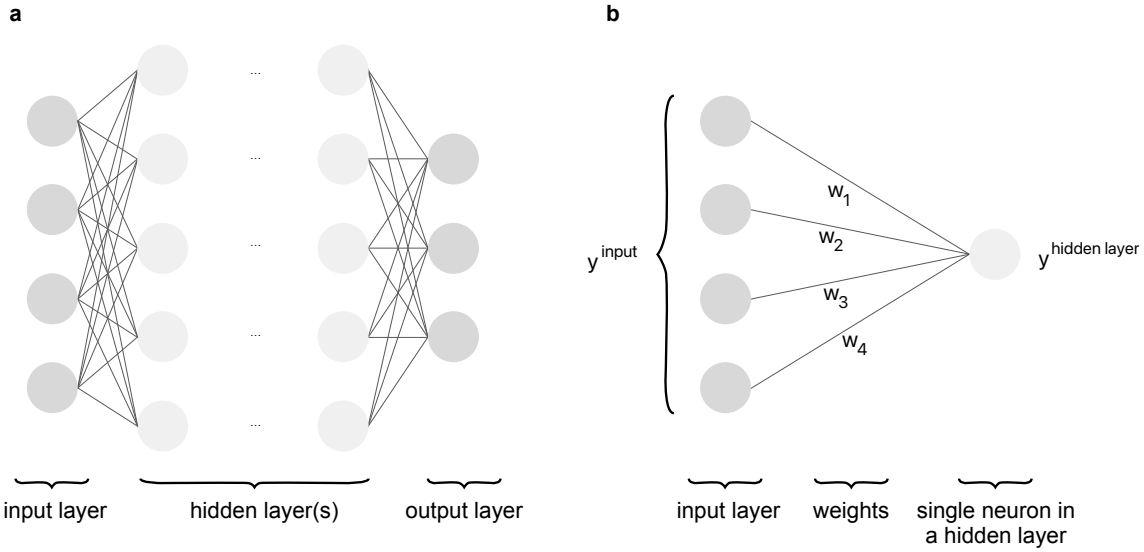


Figure 2.8: Illustration of neural networks. a A typical representation of a neural network with 4 input neurons, 5 neurons in the first and last hidden layer and 3 output neurons. **b** A part of the full neural network.

Figure 2.8 depicts typical neural network structures. The input vector $\boldsymbol{y}^{\text{input}}$ represents activations of the input layer. Each neuron in the hidden layer computes its activation $y^{\text{hidden layer}}$ by:

$$y^{\text{hidden layer}} = \sigma\left(\sum_i (y_i^{\text{input}} w_i + b_i)\right) \equiv \sigma(z), \tag{2.7}$$

with the activation function σ, the activation of the i-th neuron y_i, and the associated weight w_i and bias b_i.

This process is repeated across all neurons and layers until reaching the output layer.

Activation function

The activation function introduces non-linearity into the model, which is crucial for learning complex patterns. Without non-linearity, the neural network would merely perform linear mappings, limiting its ability to model non-linear data relationships. Different layers can use different activation functions or none at all.

A commonly used activation function is the sigmoid function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad \text{(sigmoid)} \tag{2.8}$$

However, the sigmoid function's gradient diminishes for large inputs, leading to slow and inefficient learning. The rectified linear unit (ReLU) function overcomes this:

$$\sigma(z) = \max(0, z) \quad \text{(ReLU)} \tag{2.9}$$

Cost function

To train a neural network, it is essential to define a cost function that measures the discrepancy of the network output and the ground truth. One of the most common cost functions is the mean squared error (MSE):

$$C(\boldsymbol{w}, \boldsymbol{b}) = \frac{1}{N} \sum_{i=0}^{N} |\boldsymbol{y}_i^{\text{label}} - \boldsymbol{y}_i^{\text{output}}|^2 \tag{2.10}$$

Here, N is the number of training samples, $\boldsymbol{y}^{\text{label}}$ represents the true labels, and $\boldsymbol{y}^{\text{output}}$ represents the neural network's outputs. The cost function depends on the network's weights and biases.

To optimise the weights and biases, we compute the gradient of the cost function, ∇C, using backpropagation, which applies the chain rule to calculate these gradients efficiently.

The weights and biases are updated using the gradient descent rule:

$$\boldsymbol{w}_{\text{new}} = \boldsymbol{w}_{\text{old}} - \eta \nabla C(\boldsymbol{w}_{\text{old}}) \tag{2.11}$$

where η is the learning rate. This process is repeated iteratively to minimise the cost function. A common variant, stochastic gradient descent (SGD), improves computational efficiency by evaluating the cost function on mini-batches, subsets of the training set, which vary with each iteration.

Advanced optimisation methods like Adam [92] further enhance learning by adjusting the learning rate for each parameter based on estimates of the first and second moments of the gradients.

2.3.2 Gaussian Processes

Gaussian processes (GPs) provide a flexible, non-parametric framework for making predictions about uncertain functions. At their core, GPs define a distribution over functions, allowing us to model complex relationships without assuming a specific functional form. This is achieved by considering every point in the input space to be associated with a normally distributed random variable, where the covariance between any two points captures our belief about how similar the function values should be. By leveraging this probabilistic approach, GPs can incorporate prior knowledge and handle noisy observations effectively. Their strength lies in the ability to provide not only mean predictions but also uncertainty estimates, making them particularly valuable in fields where understanding the confidence in predictions is crucial, such as in Bayesian optimisation. Through the use of kernel functions, GPs can model a wide variety of patterns and structures in data, ensuring flexibility and adaptability to various tasks.

In formal terms, a GP defines a collection of random variables, where any finite subset has a joint Gaussian distribution [93]. For regression tasks using GPs, the distribution is specified by a mean function $\mu(x)$ and a covariance function (also known as the kernel) $k(x, x')$. Thus, it is defined as:

$$f(\mathbf{x}) \sim \mathcal{GP}(\mu(\mathbf{x}), k(\mathbf{x}, \mathbf{x}'))$$

Here, f denotes the function we aim to model, with $f(\mathbf{x}) = [f(x_1), f(x_2), \ldots, f(x_n)]$ representing the function values at points $\mathbf{x} = [x_1, x_2, \ldots, x_n]$. The mean and covariance functions are evaluated for each element, hence

$$\mu(\mathbf{x}) = [\mu(x_1), \mu(x_2), \ldots, \mu(x_n)]$$

and

$$k(\mathbf{x}, \mathbf{x}') = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \cdots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \cdots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \cdots & k(x_n, x_n) \end{bmatrix}$$

Here, $\mu(\mathbf{x})$ is the mean vector and $k(\mathbf{x}, \mathbf{x}')$ is the covariance matrix of the GP.

The mean function $\mu(x)$ and covariance function $k(x, x')$ are central to the definition of a GP. The mean function, $\mu(x)$, represents the expected value of the function at any given point x, providing a baseline prediction. The covariance function, or kernel, $k(x, x')$, defines the structure of the relationship between pairs of points x and x', encapsulating our assumptions about the function's smoothness and variability. A commonly used kernel is the radial basis function (RBF) kernel, also known as the Gaussian kernel [94], defined as

$$k(x, x') = \exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right).$$

The RBF kernel assumes that points closer together are more strongly correlated, leading to smooth and continuous function estimates.

In GP regression, making predictions for new points given some observed data involves updating our prior beliefs with the information from the observed data. Given a set of observations $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ represents the input points and $\mathbf{y}$ represents the corresponding outputs, we want to predict the output y_* for a new input x_*. The prediction is achieved by conditioning the joint Gaussian distribution of the observed and new points on the observed data. This results in a predictive distribution for y_*, which is also Gaussian, with a mean and variance that can be computed using the observed data and the kernel function. The mean provides the best estimate for y_*, while the variance gives a measure of uncertainty about this prediction, reflecting how the new input relates to the observed data.

2.3.3 Bayesian optimisation

Bayesian optimisation (BO) is a probabilistic framework for optimising expensive and noisy black-box functions, particularly when evaluations are costly and limited. BO leverages GPs as surrogate models to approximate the unknown objective function $f(\mathbf{x})$. The GP provides a posterior predictive distribution over $f(\mathbf{x})$, which is iteratively updated using Bayes' theorem as new data is observed.

At each iteration n, given observations $\mathcal{D}_n = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $y_i = f(\mathbf{x}_i) + \epsilon_i$ and ϵ_i represents noise, the posterior distribution $p(f|\mathcal{D}_n)$ is updated via Bayes' rule:

$$p(f|\mathcal{D}_n) = \frac{p(\mathcal{D}_n|f)\, p(f)}{p(\mathcal{D}_n)},$$

where $p(f)$ is the prior over functions defined by the GP, $p(\mathcal{D}_n|f)$ is the likelihood of the observed data given f, and $p(\mathcal{D}_n)$ is the marginal likelihood (also known as the evidence), ensuring the posterior distribution is properly normalised.

BO efficiently explores the input space by balancing exploration and exploitation through an acquisition function $\alpha(\mathbf{x}; \mathcal{D}_n)$, which quantifies the utility of evaluating f at point $\mathbf{x}$. The next evaluation point is chosen by maximising the acquisition function:

$$\mathbf{x}_{n+1} = \arg\max_{\mathbf{x}} \alpha(\mathbf{x}; \mathcal{D}_n).$$

Commonly used acquisition functions include the expected improvement (EI), probability of improvement (PI), and upper confidence bound (UCB). These functions are designed to guide the search towards regions with high potential for improvement or high uncertainty.

The EI acquisition function focuses on areas where the potential for improvement over the current best observation is high. It is defined as:

$$\alpha_{\mathrm{EI}}(\mathbf{x}; \mathcal{D}_n) = \mathbb{E}_{f(\mathbf{x}) \sim p(f(\mathbf{x})|\mathcal{D}_n)}[\max(0, f(\mathbf{x}) - f_{\mathrm{best}} - \xi)],$$

where f_{best} is the best observed value so far, and $\xi \geq 0$ is a trade-off parameter encouraging exploration.

PI targets regions with a high probability of surpassing the best known value. The PI is given by:

$$\alpha_{\mathrm{PI}}(\mathbf{x}; \mathcal{D}_n) = \mathbb{P}(f(\mathbf{x}) \geq f_{\mathrm{best}} + \xi) = \Phi\left(\frac{\mu(\mathbf{x}) - f_{\mathrm{best}} - \xi}{\sigma(\mathbf{x})}\right),$$

where $\mu(\mathbf{x})$ and $\sigma(\mathbf{x})$ are the predictive mean and standard deviation from the GP, respectively, and Φ is the cumulative distribution function of the standard normal distribution.

UCB combines the mean prediction and uncertainty, encouraging exploration of regions with high uncertainty and potential high rewards. It is defined as:

$$\alpha_{\text{UCB}}(\mathbf{x}; \mathcal{D}_n) = \mu(\mathbf{x}) + \kappa\sigma(\mathbf{x}),$$

where $\kappa > 0$ controls the degree of exploration.

An illustrative example of these acquisition functions is presented in Figure 2.9. The true function $f(x)$, the GP mean prediction $\mu(x)$, and the 95% confidence interval given by $\mu(x) \pm 1.96\sigma(x)$ are shown in Figure 2.9**a**. The initial sample points are indicated by red dots. The EI, PI, and UCB acquisition functions are shown in Figure 2.9**b**, each suggesting a different next sampling point based on their respective criteria.

By iteratively selecting evaluation points that maximise the acquisition function and updating the surrogate model with new observations, BO can efficiently identify the global maximum of complex functions with relatively few evaluations. This process is particularly useful in fields like hyperparameter tuning and experimental design [95, 96].

2.4 Measurement techniques

In this thesis I will mostly use simple direct current (DC) measurements to inform my tuning algorithms. They involve applying a bias voltage and measuring the resulting current through the DQD, allowing direct observation of conductance changes. This method is straightforward but can be limited by low current levels. In contrast, charge sensing measurements use a nearby quantum point contact or single-electron transistor to detect changes in the charge state of the QD without requiring current flow through the DQD itself [47]. This approach is highly sensitive, enabling the detection of single-electron tunneling events and providing detailed

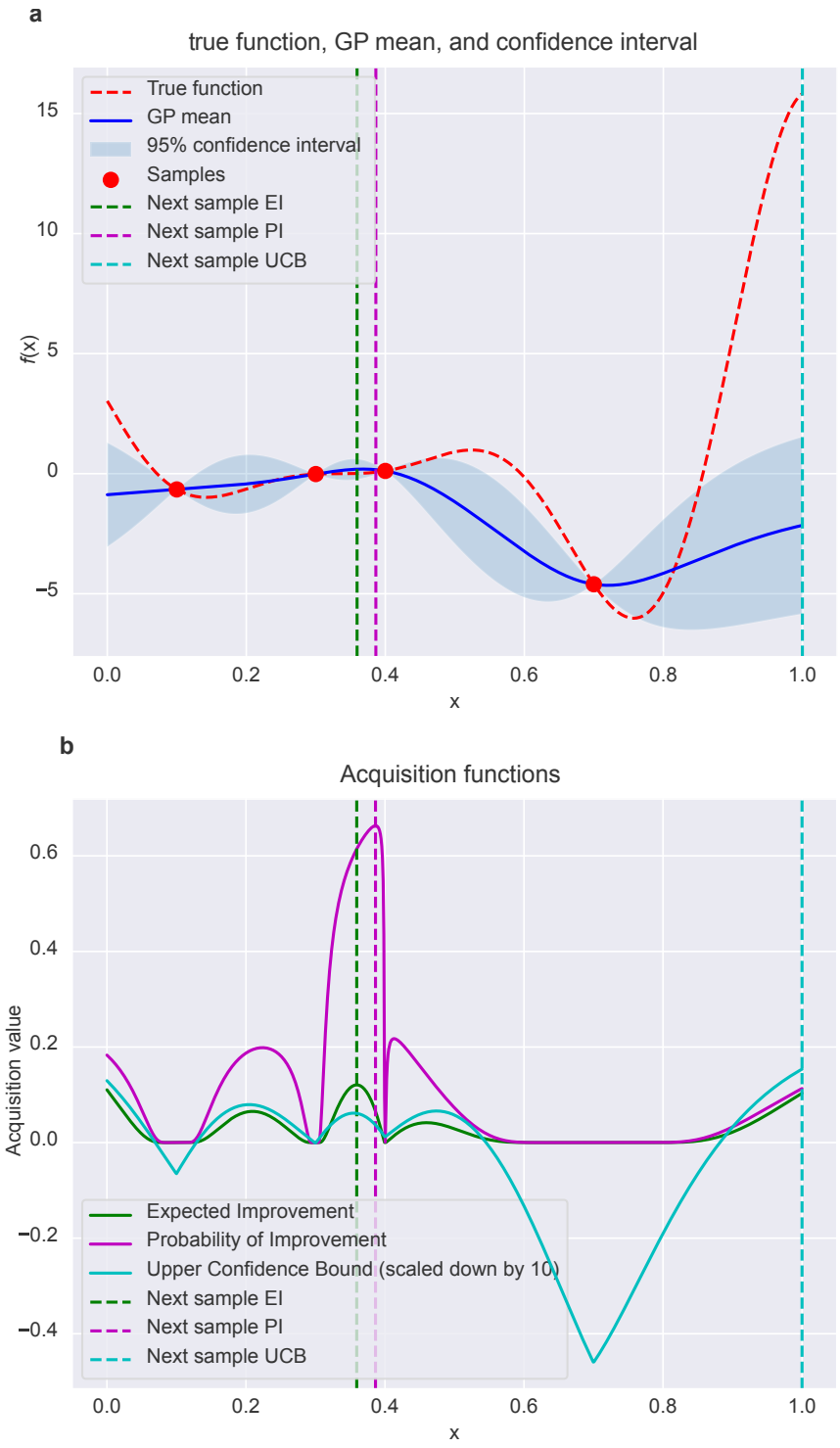


Figure 2.9: Illustration of Bayesian Optimisation using Gaussian Processes. **a** The true function $f(x)$ (red dashed line), the GP mean prediction $\mu(x)$ (blue line), and the 95% confidence interval (shaded area). The initial sample points are shown as red dots. The vertical dashed lines indicate the next suggested sampling points by each acquisition function: EI (green), PI (magenta), and UCB (cyan). **b** The acquisition functions EI (green), PI (magenta), and UCB (cyan), with their respective maxima marked by vertical dashed lines, suggesting the next points to sample.

information about charge configurations and Coulomb blockade phenomena but requires more complex devices or measurement setups.

When reading out a qubit, I want to measure the transport of a single charge that is either blocked by PSB, or the block has been lifted by the manipulation pulse. Since we are measuring charge transport, we cycle this process to measure a current. If the cycle length is 100 ns for the whole time it takes to initialise, manipulate and read out, this would give a current of about 1 pA given the elementary charge of about 1.6×10^{-19} C. This assumes a perfect conversion rate. At the same time, unwanted leakage currents exists in the readout point. This current is on the order of pA to tens of pA in the experiments I will be presenting. This means that the current contribution by the qubit readout will almost certainly be blocked out by noise.

To improve the signal-to-noise ratio and effectively detect the qubit readout signal despite the presence of noise and leakage currents, I employ a lock-in amplifier (Figure 2.10). The lock-in amplifier modulates the driving pulse at a specific frequency (the lock-in frequency) and demodulates the measurement signal from the sample. This technique allows me to measure the difference in current through the DQD when the driving pulse is applied versus when it is blanked, enhancing the detection of the qubit signal.

In my measurement setup, the lock-in amplifier provides a trigger signal to the arbitrary waveform generator (AWG) at a frequency on the order of Hertz. The rising edge of this trigger activates channel A of the AWG, while the falling edge activates channel B. Depending on which channel is active, the AWG controls the microwave (MW) source via IQ modulation, turning it on or off accordingly. The AWG also outputs a Coulomb blockade pulse in the MHz range. Both the Coulomb blockade pulse and the MW pulse are combined and sent to one of the plunger gates. At 1.5 K, this combined signal is merged with the DC voltage from the digital-to-analog converter (DAC) via a bias-tee. The current through the nanowire is then amplified by a current-to-voltage (I/V) converter and measured by an acquisition card.

The signal coming from the device is split and separately multiplied with the reference signal from the lock-in amplifier and a 90° phase-shifted copy of it. The

outputs of these mixers pass through configurable low-pass filters, resulting in two outputs, X and Y, termed the in-phase and quadrature components. Due to factors such as the length of the cables, the signal might appear in either X, Y, or a combination of both. In principle, the phase of the lock-in reference signal could be adjusted to compensate for this. However, I avoid this calibration step by applying principal component analysis (PCA) to the lock-in signal, a method previously used for the same purpose [32, 97].

2.4.1 PCA of lock-in measurements

Principal components analysis (PCA) is a statistical technique used to reduce the dimensionality of a dataset while retaining most of its variability. Intuitively, PCA works by identifying the directions (principal components) along which the data varies the most, enabling us to project the data onto a lower-dimensional space without significant loss of information.

Let $\mathbf{X}$ be a dataset with n observations and p variables, represented as an $n \times p$ matrix. The first step in PCA is to standardise the data if the variables are on different scales. This is typically done by subtracting the mean of each variable and dividing by the standard deviation, resulting in a zero-mean and unit-variance dataset.

Next, we compute the covariance matrix $\mathbf{S}$:

$$\mathbf{S} = \frac{1}{n-1} \sum_{i=1}^{n} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

with $\mathbf{x}$ the individual elements of $\mathbf{X}$ and $\bar{\mathbf{x}}$ the mean value of $\mathbf{x}$. The covariance matrix $\mathbf{S}$ is a $p \times p$ symmetric matrix that contains the covariances between each pair of variables.

The core of PCA involves finding the eigenvalues and eigenvectors of the covariance matrix $\mathbf{S}$. The eigenvalues λ_i indicate the amount of variance captured by each principal component, and the eigenvectors $\mathbf{v}_i$ provide the directions of these components. We solve the following equation to obtain the eigenvalues and eigenvectors:

$$\mathbf{S}\mathbf{v}_i = \lambda_i \mathbf{v}_i$$

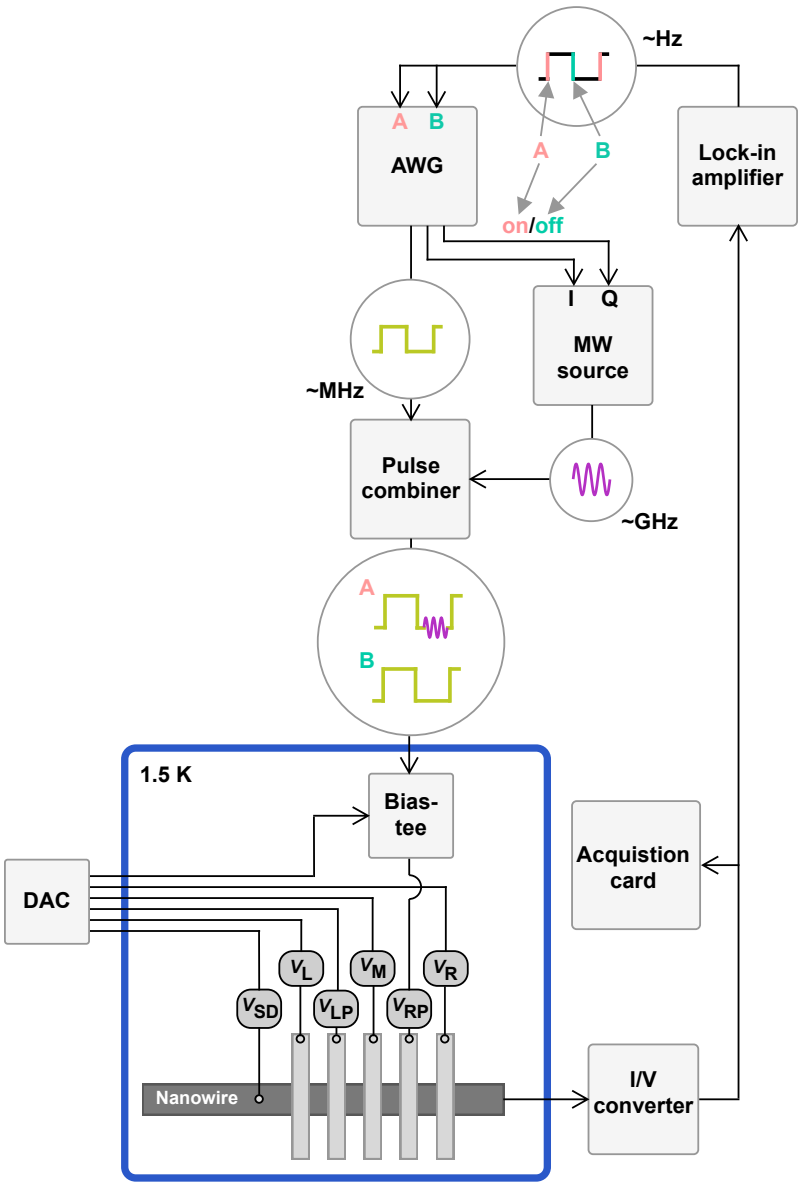


Figure 2.10: Measurement setup with lock-in amplified signal. Diagram of the experimental setup used for measurements in Chapters 4.4, 5.2, and 6.4, highlighting the lock-in amplification of the qubit signal. All DC voltages are set by a digital-to-analog converter (DAC). The current through a nanowire is amplified by a current-to-voltage (I/V) converter and measured by an acquisition card. The lock-in amplifier provides a trigger signal to the arbitrary waveform generator (AWG) at a lock-in frequency on the order of Hertz. The rising edge triggers channel A of the AWG, while the falling edge triggers channel B. The AWG controls the microwave (MW) source via IQ modulation, turning it on or off depending on whether channel A or B is activated. The AWG always outputs a Coulomb blockade pulse in the MHz range. Both the Coulomb blockade pulse and the MW pulse are combined and sent to one of the plunger gates. At 1.5 K, the signal is combined with the DC voltage from the DAC via a bias-tee. The current through the nanowire can then be demodulated by the lock-in amplifier.

The eigenvalues are then sorted in descending order, and the corresponding eigenvectors form the principal components. The first principal component is the eigenvector associated with the largest eigenvalue, capturing the most variance in the data. Subsequent principal components capture the remaining variance under the constraint that they are orthogonal to the previous components.

To transform the original dataset $\mathbf{X}$ into the principal component space, we project the data onto the eigenvectors. Let $\mathbf{V}$ be the matrix of eigenvectors, then the transformed data $\mathbf{Y}$ is given by:

$$\mathbf{Y} = \mathbf{XV}$$

Here, $\mathbf{Y}$ is the dataset represented in the new coordinate system defined by the principal components. The dimensionality of the data can be reduced by selecting a subset of the top k eigenvalues and their corresponding eigenvectors, thus forming a lower-dimensional representation while retaining most of the original variance.

In the case of the measurements from the lock-in amplifier, we simply want to reduce a set of 2D data points to 1D data points. Fig. 2.11**a** shows an example measurement. Lock-in amplified measurements are taken for varying magnetic fields B. These types of measurements will be analysed in Chapters 4.4, 5.2 and 6.4.

Fig. 2.11**b** shows the distribution of the same data in a plane with the magnetic field as the colour coding. Further signal processing requires us to, e.g., find peaks in these measurements so it is preferrable to reduce their dimensionality. For this, we can use PCA as a linear projection describes the data well.

The first principal axis is shown in Fig. 2.11**b** as a green arrow. The projection on this axis is shown in Fig. 2.11 **c** in blue. For comparison, I show another simpler way to reduce the dimensionality, which is simply the Euclidean distance from the origin (orange curve in Fig. 2.11**c**). That metric in general retains less of the variability of the data.

One problem with PCA in combination with this measurement technique is that it will not tell by itself whether deviations will present itself as positive or as negative

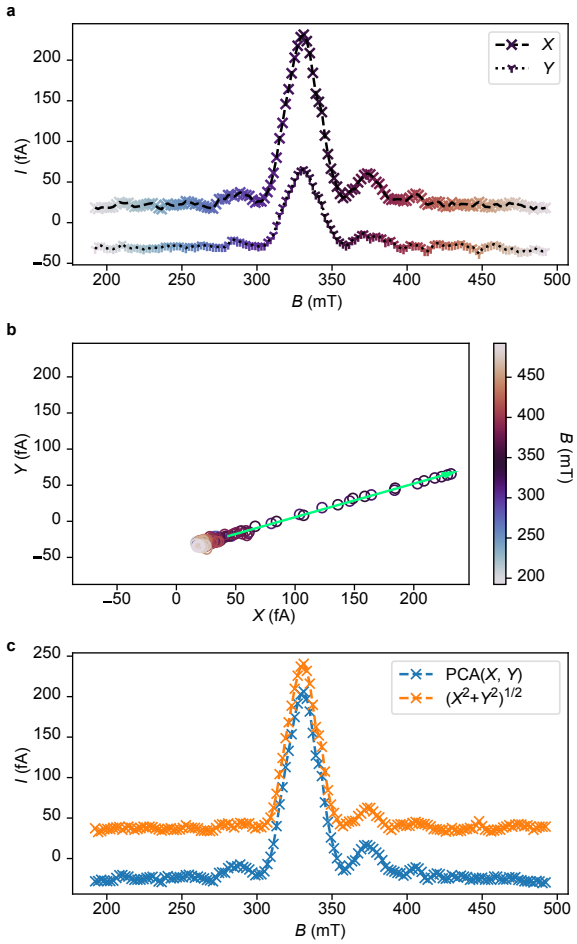


Figure 2.11: Signal processing of lockin amplified measurements. **a** A typical measurement of the current that is enhanced with a lock-in amplifier while varying magnetic field B. **b** The in-phase component X and quadrature component Y is shown with B as the colour corresponding to **a**. The first principal component is plotted as a green arrow over the data. That arrow describes the direction of variability in data well, thus PCA is a well suited dimensionality reduction tool here. **c** Data processed with the Euclidean distance from the origin for X and Y in orange and with PCA in blue.

currents. This can lead to problems if we, e.g., apply a peak finding algorithm downstream. This case is shown in Fig. 2.12.

The Euclidean distance sidesteps this problem but loses some of the variability, see Fig. 2.12**c**. However, we can use a simple rule to correct the sign of the signal by considering the mean value of the signal and comparing it to the mean of the range. If the mean of the range is higher than the mean of the signal, we do nothing, otherwise we negate the signal, leading to the "corrected PCA" in Fig. 2.12**c**.

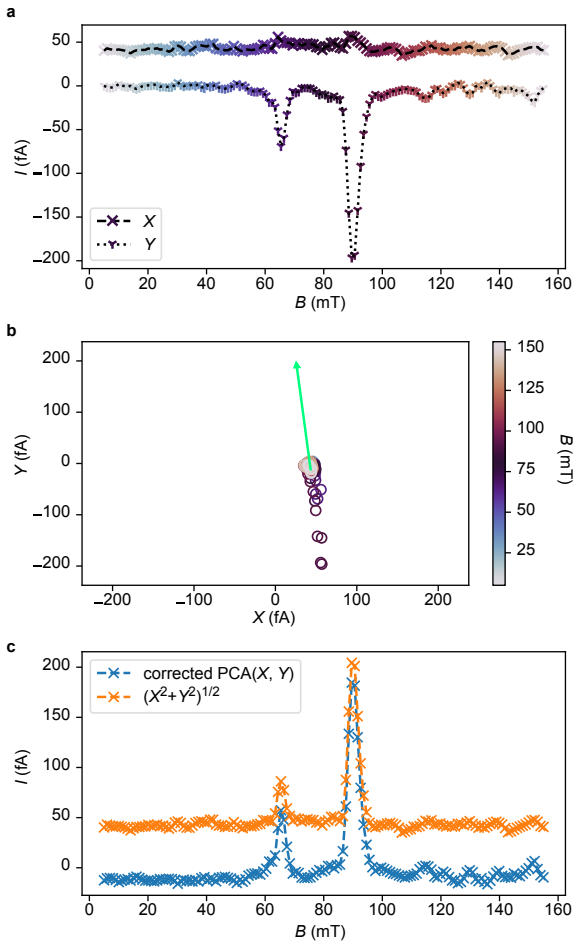


Figure 2.12: Signal processing for measurements with negative peak currents. **a, b** as in Fig. 2.11. **c** Instead of simply showing the projection on the first principal component, I show it with the sign corrected.

2.5 Automated tuning – the playing field

The process of bringing a new semiconductor quantum dot device into a state in which it can be used as a quantum bit is notoriously difficult because of fluctuations in the disorder potential, fabrication variances, and material defects.

To solve the problem of automating this, there are numerous techniques available, and selecting the appropriate one for a given task requires careful consideration of several factors:

- The necessity for expert-labeled training data, which may not always be feasible or practical to obtain.
- The need for efficiency, both in the total number of measurements taken for a given task and in the computational resources required to decide which measurements to take.
- The minimum accuracy required for the broader algorithm to achieve its overall objective.

To address these considerations at various stages of the algorithm, a range of commonly used techniques includes GP inference (see Section 2.3.2), CNNs (see Section 2.3.1), unsupervised computer vision (CV) and computational geometry techniques, and BO (see Section 2.3.3).

There have been suggestions on how to structure this tuning process by Ares (2021) [98] and Zwolak and Taylor (2023) [99]. These approaches are illustrated in Fig. 2.13. The terminology differs slightly between the two and I will elaborate on these differences below. It is important to note that both of those frameworks did not come with a demonstration of the full tuning functionality. The jury is therefore still out on how to best think about the tuning process. I will structure the literature overview according to these two suggestions as they provide a good foundation.

We begin with a grounded device where the electrostatic potential landscape is mostly flat. Zwolak and Taylor (2023) [99] introduce a preliminary "bootstrapping" phase, which includes tasks such as checking the functionality of gates, testing for

leakage currents that indicate a fabrication problem, and establishing safe operating voltage ranges that will not damage the device. To my knowledge, these specific tasks have not been the focus of any publication, as these steps are generally considered trivial and not warranting detailed investigation.

Ares (2021) [98] proposed the first stage as "super-coarse tuning" which overlaps with the "bootstrapping" stage of Zwolak and Taylor (2023) [99] and their subsequent "coarse tuning" stage. The overarching theme of this stage is to set the rough potential of the device, i.e., form a DQD. Following this, it may be beneficial for downstream tasks to "establish controllability" [99], by, e.g., determining cross-talk between gate electrodes.

Subsequently, we move to the "coarse tuning and characterisation" stage as described by Ares (2021) [98], or "charge state tuning" according to Zwolak and Taylor (2023) [99]. At this stage, the goal is to deplete the quantum dots, allowing operation in the few-electron regime.

Both frameworks agree on a stage termed "fine tuning" immediately prior to encoding a qubit. This involves further manipulation of the potential landscape to adjust specific physical parameters, such as the inter-dot coupling. Ares (2021) [98] extends the process by suggesting an additional stage focused on optimising a qubit once it has been established.

The linear representation of the process is a significant simplification. Each publication addresses various challenges, sometimes fitting well within this linear framework and at other times only touching on specific aspects. The publications use different device architectures, material platforms, and measurement techniques. Therefore, while one might borrow ideas from specific publications to unify this pipeline, each step ultimately requires tailoring to the specific context of the device in question.

I will now highlight the progress in the literature for the stages of the tuning process. I have placed most of them in Fig. 2.13 but as discussed above, the publications do not always fit neatly into the proposed frameworks, so please enjoy this figure with a pinch of salt. I added more depth to the pipeline by showing which

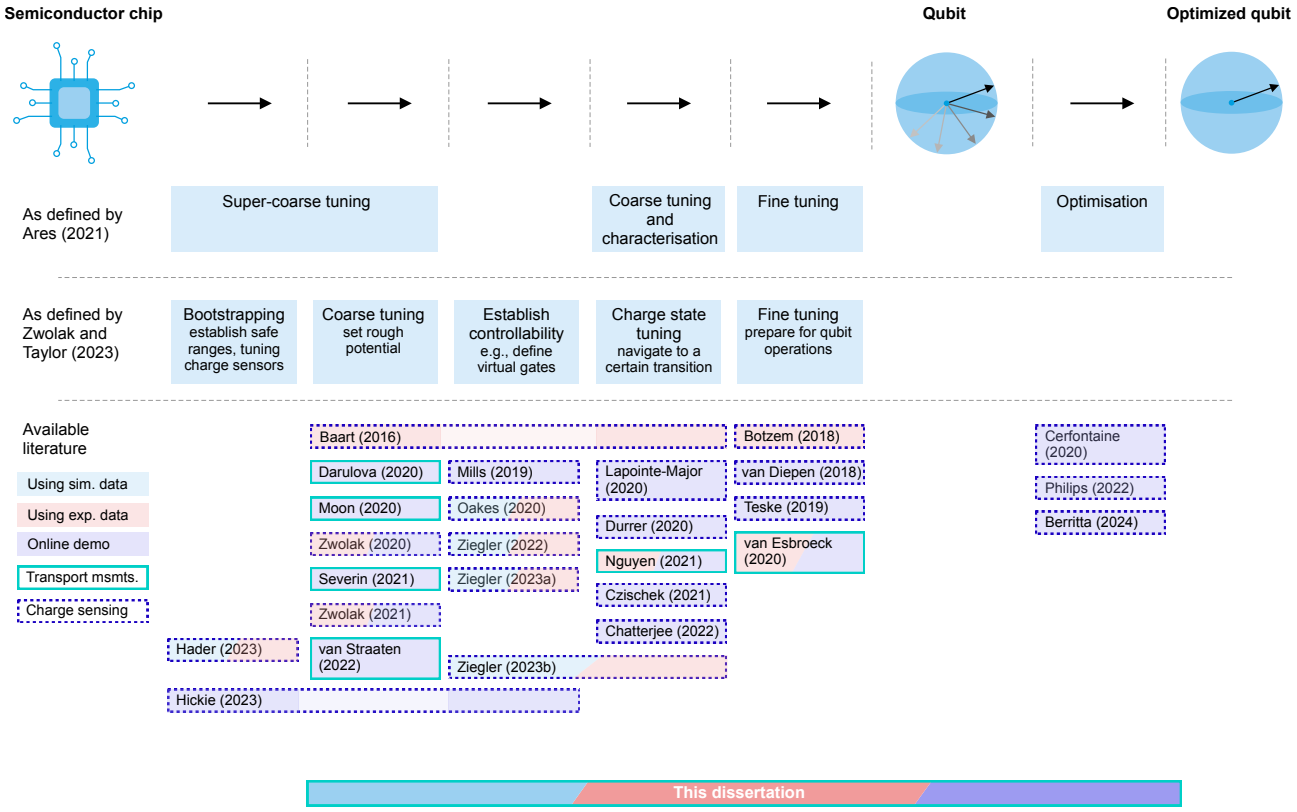


Figure 2.13: Autonomous tuning literature. There are two suggestions in the literature on how to structure the tuning process. I show both frameworks and how they align with each other. Publications automating particular steps of the tuning pipeline are listed below the two proposed frameworks. I colour-code each publication by whether the authors used simulated data (blue), experimental data (red), or demonstrated their approach live with a real device (purple). Further the border of each publication indicates whether they employ transport measurements (green solid line) or a charge sensing scheme (dashed purple line). Graphics at the top are adapted from Ares (2021) [98].

of the publications were done using transport measurements and which used charge sensing. This is indicated by the colour of the edge of the corresponding boxes. On top of that I use the background colour of the boxes to hint at whether the work was done using simulated data, experimental data (but offline), or online with a "live" demo.

2.5.1 Setting the rough potential

The first algorithm to address autonomous tuning was introduced by Baart *et al.* (2016) [37]. This algorithm is highly device-specific and may not generalise to other devices. In terms of the landmark metaphor this means: they already have a very detailed map, and the street signs are always clearly visible, which is not always

the case in practice. The authors demonstrated the ability to tune a device from scratch to a double dot in the single-electron regime through a three-step process. First, they find the pinch-off of each gate, i.e., the voltage which divides the voltage space into a region of low or no current and a region of high current. Secondly, they take 2D scans of the current as a function of the barrier gates, fit a tetragon to the region of large current and use this to take a higher resolution scan where they expect a single dot to form. A point in gate voltage space is selected based on post-processed charge stability diagrams, and the process is repeated to form a double dot. As their last step they template match a cross pattern in a charge sensing measurement to fix the number of electrons in the double quantum dot (see Section 2.5.4 for more details).

They demonstrate the overall strategy on how to find a double dot automatically, i.e., finding pinch-off, looking for Coulomb peaks which are characteristic QD features, and mapping out the region of high current in voltage space. This is an important contribution as later literature builds upon those findings.

Darulova *et al.* (2020) [29] picked up the ideas developed by Baart *et al.* (2016) [37] and combined it with various classifiers that guide the tuning, making it applicable to a larger range of devices. Through a series of binary decisions that are guided by heuristics and by the results of classifiers that were trained on real data, they arrive in a single or double dot. Of the five devices that pass the initial quality tests, three were successfully tuned in two cool-downs and two were successfully tuned in one of the two cooldowns.

Darulova *et al.* (2020) [29] rely on a lot of device-specific knowledge, e.g., the heuristics in the tuning process and the way the tuning algorithm steps through voltage space. This hampers the generality of their approach. However, it may lead to very fast tuning times which they unfortunately do not report.

Device-specific algorithms are not necessarily bad in the context of tuning up a large-scale quantum computer because we can assume that all devices in such a quantum computer are very similar. Speaking in terms of the metaphor, if we want to find the Radcliffe Camera in Oxford over and over again, we will have a very good map and strategy to find it at some point. A highly specialised approach

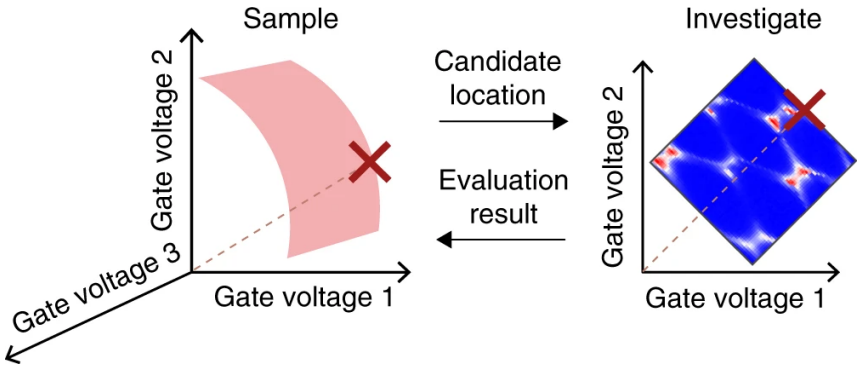


Figure 2.14: Visualisation of hypersurface construction. Many algorithms rely on mapping the area between conducting and non-conducting regions. In certain parts, we expect to find DQDs, which exhibit recognisable features in the stability diagram (see measurements on the right). This figure is reprinted from Ref. [30].

might therefore be favourable, even though a more general approach is more robust to variations in the task and to noise.

Moon *et al.* (2020) [30] show what a more general approach can look like. Their algorithm revolves around the idea of mapping out the surface between high and low current areas, similar to Baart *et al.* (2016) (see Fig. 2.14). However, they assume little prior knowledge about the device. In terms of the landmark metaphor, this means that they first build a map instead of assuming one from previous knowledge.

They claim that the time needed to find a double quantum dot is only 70 minutes, and therefore less than the time needed by a human expert (3 hours). Another advantage of this method is that it is robust against failing gates. In their study, one gate showed leakage current and was therefore excluded from the tuning. Other approaches [29] would have sorted out this device as untuneable but Moon *et al.* (2020) were still able to tune that device into a double quantum dot.

Severin *et al.* (2021) [31] take what is essentially the same algorithm and show that it can tune three different silicon-based device architectures into a the double quantum dot regime, further showing the generality of the approach.

While very general, there are some issues with the algorithm of Moon *et al.* (2020). It does not stop on its own but only when it is timed out. The recognition of double quantum dot is done *a posteriori* by human experts. This could be fixed by combining the algorithm with the classifiers developed by Darulova *et al.* (2020) [29].

Taking the idea of mapping out the gate voltage space to the extreme is the work of Euler *et al.* (2020) [100]. They use a semiconductor device with a circular arrangement of gates and create a large data set of current measurements while controlling input gate voltages. They use that data set to train a neural network that can predict the current given a set of gate voltages. The neural network is essentially a virtual representation of the device. They can use it to determine a set of gate voltages that will lead to a specified behaviour, e.g., for the device to act as a logical gate or as a classifier.

A different angle was taken by van Straaten *et al.* (2021) [32]. In terms of the metaphor, they managed to find the Radcliffe Camera by moving extremely fast but in a random walk. They rely on fast measurements based on radio-frequency reflectometry and can take a device from being grounded to showing DQD features in about 15 minutes. This measurement technique allows them to speed up the process to a point where they don't need much guidance anymore, they can simply search through the parameter space with a random walk and some guidance from a GP. They use a score function based on the Fourier transform of the measurement signal to inform their algorithm.

2.5.2 From a single to a double quantum dot

Several publications have suggested a way to automatically tune a device from a single dot to a double dot. The algorithms here start with a single dot or close to the single dot regime. In a charge stability diagram, i.e., a 2D scan of the current as a function of the gates that control the electrochemical potential of each dot (so-called plungers), this expresses itself as diagonal lines. Double dots show a honeycomb pattern and eventually bias triangles instead. Kalantre *et al.* (2019) [33] showed on mostly simulated data that it is possible to train classifiers to distinguish the two features. With the trained neural network able to tell in which gate voltage region a device is, they use a simple numerical optimiser to find the desired configuration. They demonstrate this algorithm on simulated data and on a real

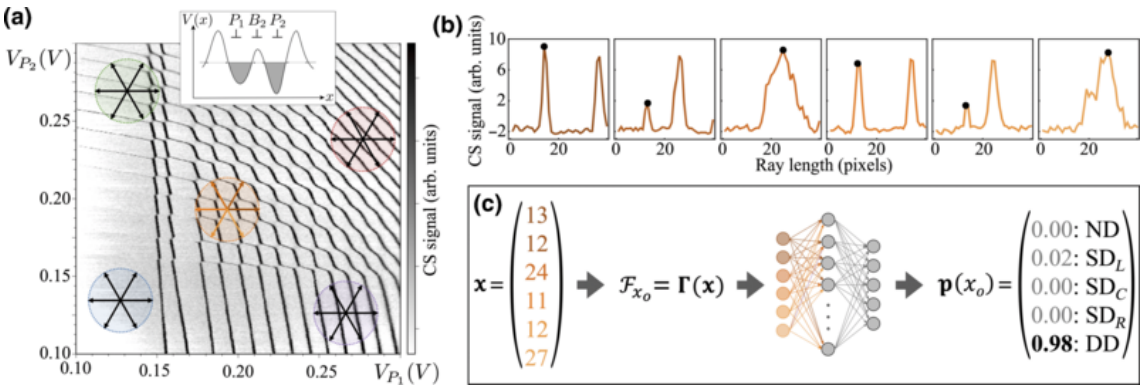


Figure 2.15: Visualisation of ray-based fingerprinting. **a** A stability diagram measured via a charge sensor is displayed. The stability diagram includes regions of DQDs, SQDs, and areas with no dots. The figure illustrates how the rays effectively sub-sample a 2D scan. **b** Measurements from the rays closest to the center in **a** are shown. These measurements are post-processed to identify the transition line nearest to the measurement center, which is marked with a black dot. **c** The post-processed data from **b** is input into a neural network, which predicts whether there is a DQD, no dot, or a SQD under the left plunger gate, right plunger gate, or centrally between them. Reprinted from Ref. [35].

device to a limited extent. Zwolak *et al.* (2020) [34] systematically demonstrate that algorithm online, i.e., in real time.

The same authors have also proposed an improved version of their algorithm, in Zwolak *et al.* (2021) [35]. Instead of measuring full two-dimensional scans and feeding them into neural networks, they introduce "ray-based fingerprints" that reduce the amount of measurements needed by about half (Fig. 2.15). The scheme leverages one-dimensional projections of the device response in the multidimensional parameter space. They report improved accuracy and a reduced complexity in the classifier structure, i.e., a smaller neural network. They argue that these smaller neural networks are needed for on-chip tuning in the future.

Nguyen *et al.* (2021) [36] show that it is possible to solve the problem of tuning into the double dot regime using reinforcement learning (RL). They train a so-called RL agent, in this case a neural network, to move in the plunger voltage space to find bias triangles.

2.5.3 Establish controllability

There is considerable cross talk between the gate electrodes because of their proximity. In my analogy, this means that we need to adjust the way we walk so that if you think you are moving North, you are actually moving North and not also slightly West. In the context of quantum dot tuning, this means determining the virtual gates such that each gate only changes what it was intended to change.

Some work has been done to introduce methods for the extraction of virtual gates on the device. Mills *et al.* (2019) [101] demonstrate the extraction of virtual gates with simple methods such as Hough transforms and template matching in a triple quantum dot. Using virtual gates, they can tune the inter-dot tunnel couplings of both pairs of DQD configurations.

Ziegler *et al.* (2023a) [102] show that neural networks can be used to classify stability diagrams on a pixel level which in turn enable the automatic extraction of virtual gates.

Oakes *et al.* (2020) [103] demonstrate this characterisation task for more quantum dots. They extract virtual gates of 2x2 dot array using stability diagrams that are processed with a Hough transform and then fed into a neural network. Similarly, Chatterjee *et al.* (2022) [44] develop an active learning scheme that characterises a 2x2 array by estimating the polytope of Coulomb blockade boundaries. From these, virtual gates could be extracted.

When using sensor dots to measure the device quantum dots, one also needs to consider the cross talk of those sensors with the device quantum dots. Hickie *et al.* (2023) [97] show an automated method to do this. They also show the autonomous tuning of the sensor dot for optimal sensibility. Similarly, Hader *et al.* (2023) [104] study the optimal operating regimes of sensor dots by modeling the noise of measurements of sensor dots.

2.5.4 Finding the correct charge configuration

To encode a spin qubit, we typically want to reduce the number of electrons or holes in the QDs to a few or single electron/hole regime. Finding a suitable charge transition

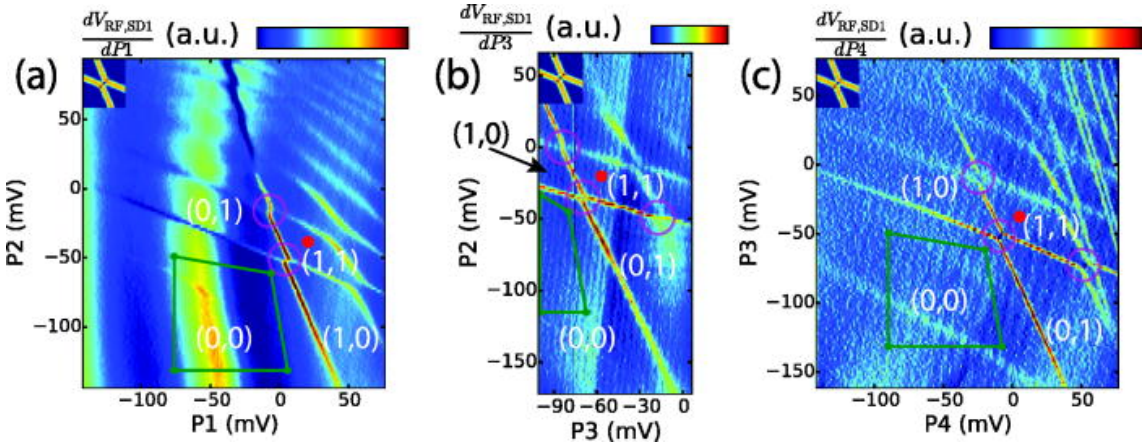


Figure 2.16: Demonstration of template matching. Three stability diagrams of three different DQDs measured with charge sensing are shown. The insets show the template that is used to find the inter-dot transitions. The ones found by the algorithm are encircled. The single-electron regime is found by checking the region to the lower left of the most bottom-left detected cross. If no more crosses are found, it is assumed that the DQDs are depleted in that area, allowing for the assignment of the number of charges. Reprinted from Ref. [37].

is usually done by first completely depleting the quantum dots, i.e., there are no more electrons or electron holes in the dots, and then counting charge-transitions until the number of charges is right in each dot. Charge-transitions can be measured using a nearby sensor dot, or a quantum point contact. In the landmark metaphor this means that we try to find the Radcliffe Camera by finding a different landmark we can recognise directly, say, Magdalen Bridge, and then counting intersecting roads on the way towards the Radcliffe Camera.

As previously mentioned, Baart *et al.* (2019) [37] identify the crossing of two charge-transitions by matching the relevant features in the stability diagrams to a template. Assuming we also know when the dots are both empty, those crossings can be used to determine and set the number of electrons in each dot (Fig. 2.16).

Durrer *et al.* (2020) [39] train neural networks to recognise transitions in charge stability diagrams and use this to tune into a desired charge state. They first deplete the dots until they are empty and then fill them to the desired number of electrons by counting the relevant features. They report a success rate of only 57% and attribute it to a low signal-to-noise ratio.

Lapointe-Major *et al.* (2020) [40] report that it is possible to tune into the single electron regime using simpler techniques. They remove background noise and then use the Hough transform to detect transitions. The authors only demonstrate their method on a single stability diagram and they don't systematically benchmark the algorithm, e.g., for random starting positions. Thus, it is not clear how well the algorithm works.

Czischek *et al.* (2021) [38] show a similar approach to Lapointe-Major *et al.* (2020) but use very small neural networks to detect transitions. In contrast to Lapointe-Major *et al.* (2020), they study the success rate of their algorithm which finds the single-electron regime in about 75% of the times.

Ziegler *et al.* (2023b) [105] present an approach that integrates multiple tuning stages. They begin with a coarse-tuning module that assumes pre-set barrier voltages and can transition from a single to a double quantum dot using previously developed algorithms [33–35]. Next, they establish controllability as described by Ziegler *et al.* (2023a) [102] and navigate to a specified charge state following ideas from Zwolak *et al.* (2021) [35]. Their methods are primarily tested on simulators and only validated offline with experimental data. The real-world performance of this approach remains to be demonstrated.

2.5.5 Fine tuning

Van Diepen *et al.* (2018) [43] show the automatic tuning of the inter-dot tunnel coupling using fitted measurement data and a simple feedback loop. Botzem *et al.* (2018) [106] show different techniques that can be used to characterise the device. Additionally to the inter-dot tunnel coupling t, they describe techniques to extract the tunnel coupling to the leads, and how to locate certain operating points. They claim that their measurements can be done with almost no human interference.

Teske *et al.* (2019) [42] use the techniques from Botzem *et al.* (2018) [106] to automatically fine tune parameters. They estimate the gradient of the parameters which are being tuned (the width of the inter-dot transition in gate voltage space, w, and the time required to reload a singlet t_{sr}) with respect to the gate voltages. Based

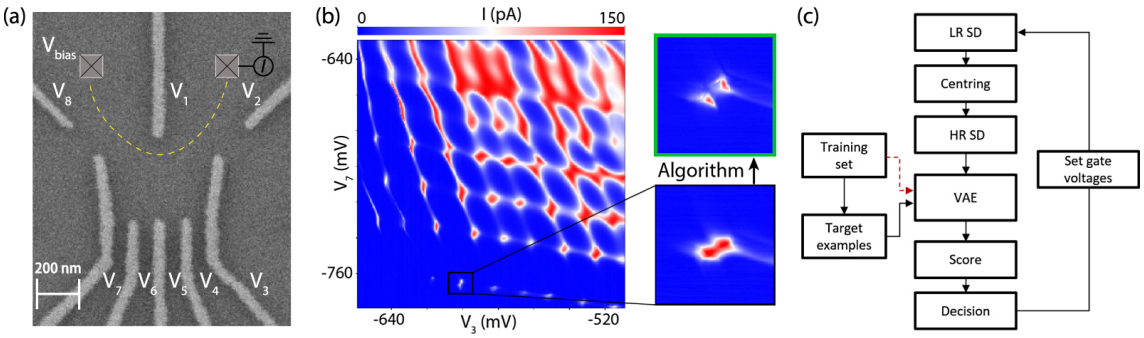


Figure 2.17: Fine tuning using a variational autoencoder. a The device used in this study by van Esbroeck *et al.* (2020), probed using transport measurements. **b** A large stability diagram serving as the starting point. The pairs of triple points lack the typical triangular features. The algorithm adjusts the barrier voltages to transform a pair of triple points into bias triangles. **c** The algorithm flow. The variational autoencoder (VAE) is used as a way to score the measurements and inform a decision making algorithm. Reprinted from Ref. [41].

on that gradient, they calculate a new set of voltages and iterate. The advantage of this method is that the estimation of the gradient of the QD's characteristics as a function of gate voltages provides insight into the tunability of a quantum dot.

Van Esbroeck *et al.* (2020) [41] have taken a deep learning approach for fine tuning. They train a deep generative model, a variational autoencoder, on bias triangle measurements. The model provides a latent representation of those measurements. By introducing targets labelled by experts in this latent space, they can reduce the distance between an untuned pair of bias triangles and the targets in latent space (Fig. 2.17).

2.5.6 Classification approaches

Some publications are only concerned with the aspect of classifying measurements. This emphasises the importance of the aspect. Think back on the Radcliffe Camera metaphor: If we can't see where we are, then we don't know where to go.

Schug *et al.* (2024) [107] propose a method for the explainable classification of a measurement type called "triangle plot" (as dubbed by the authors) which can be used in the early stages of tuning when setting the rough potential. Using so-called explainable boosting machines, they devise a method that does not need a neural

network to function. Since it is an explainable technique, they argue that this may work better in a hybrid tuning model where humans intervene from time to time.

Weber and Zwolak (2023) [108] propose a way to annotate charge stability diagrams with the label of single, or double quantum dots, or no dots. They base their scheme on the "fingerprinting" method of Zwolak *et al.* (2021) [35].

I previously introduced the work of Ziegler *et al.* (2023) [102] when talking about establishing controllability. Their work also falls in this category of publications focused on classifications as their neural network does a pixel-wise classification of charge stability diagrams.

Muto *et al.* (2024) [109] show a neural network approach to recognise charge states, i.e., single, or double quantum dots, or no dots. The largest contribution they make is that they gain insights into the inner workings of the neural network and show that the neural network indeed looks for the transitions lines when making a decision, much in the same way as humans.

Most coarse tuning methods rely on training a classifier, which usually requires a large amount of data. Since real experimental data can be scarce, Darulova *et al.* (2021) [110] investigate the performance of such classifications depending on whether real or simulated data is used. They specifically look at the classification between single and double dot features. They find that training with real data, or with real and simulated data mixed, gives the best accuracy. They conclude that real data is more important and that realistic noise models need to be included in the simulated data for them to prove helpful. On a similar note, Ziegler *et al.* (2022b) [111] emphasise the importance of noise models and show how appropriate noise models can improve the quality of classifiers trained on simulated data.

2.5.7 Beyond the initial tune-up

If the initial tuning is successful, the story does not end here. For one, having successfully tuned a qubit does not say anything about its quality. We therefore may wish to optimise things like the fidelity of one- or two-qubit gates. Multiple qubits also require multiple (double) quantum dots to be tuned in vicinity of each

other, i.e., the tuning of arrays would need to be addressed. This is complex due to the interaction between the gates and complex features that occur when measuring multiple quantum dots at once.

The optimisation of single-qubit gates has been demonstrated by Cerfontaine *et al.* (2020) [112]. They measure eight error syndromes in a singlet-triplet qubit defined in GaAs and use those to optimise pulse sequences using a simple numerical optimiser. Using this technique, they achieve a gate fidelity of 99.5%. They do not optimise this fidelity directly but optimise via error syndromes. It might therefore be possible to push the fidelity even higher by directly optimising it.

Berritta *et al.* (2024) [113] stabilise a singlet-triplet qubit and optimise the coherence time by leveraging fast computations that estimate local fluctuations that affect the qubit performance.

Although many algorithms for various of the steps have been proposed, it remains unclear how to combine them. Looking at the calibration of superconducting quantum computers can offer some insights into this. Kelly *et al.* (2018) [114] developed a framework in which the calibration process is represented as a directed acyclic graph that corresponds to dependencies in the calibration routine. By traversing this graph, a system can be calibrated and maintained, effectively representing and utilising human knowledge about the dependencies of different steps. However, translating this algorithm to the tuning of semiconductor quantum devices is not straightforward.

The largest spin qubit device to date was presented by Philips *et al.* (2022)[28]. They operate a six qubit processor. To do this they need extensive calibration and they use a framework similar to the one introduced by Kelly *et al.* (2018) [114].

Tuning of arrays has been demonstrated and first steps towards automation have been taken. Volk *et al.* (2019) [115] show how an array of eight dots can be systematically tuned by sequentially tuning single dots and using virtual gates, i.e., combinations of the real gates that aim to counteract cross-talk effects. This approach, however, was not automated. Liu *et al.* (2022) [116] use a very similar approach to show that they can tune up a linear array of four quantum dots. They

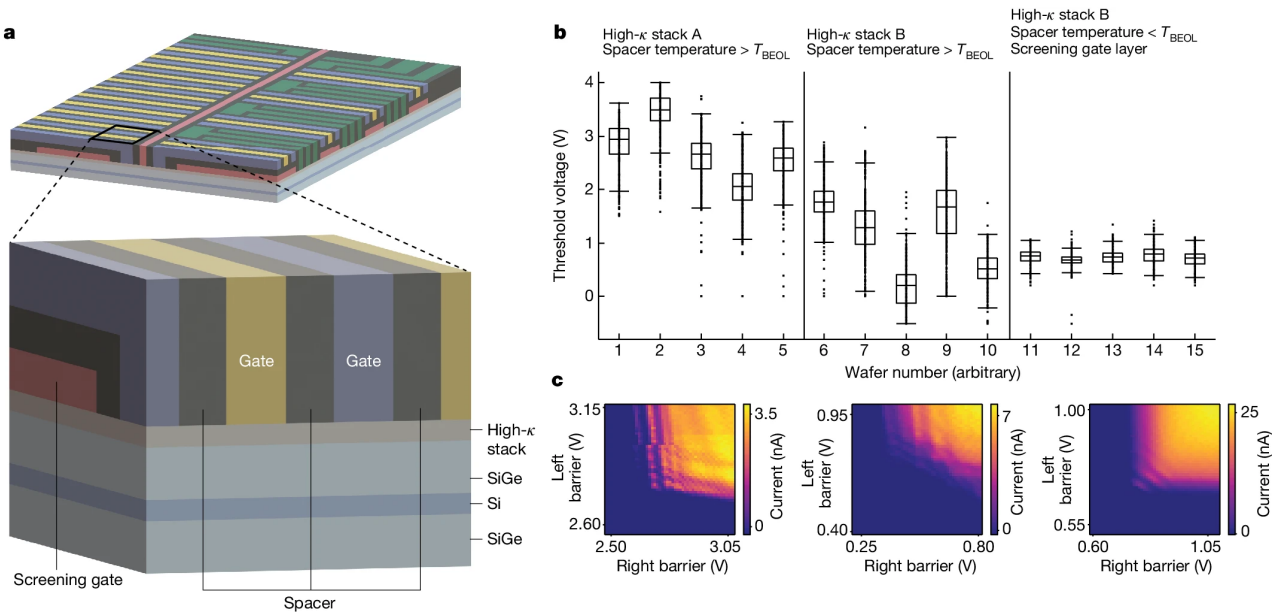


Figure 2.18: Improvements of devices using high-throughput characterisation. **a** A model of the devices developed by Neyens *et al.* (2024). **b** Over multiple generations, device fabrication improved. The voltage required to turn current on and off with a particular gate, the threshold voltage, is shown. The variance in threshold voltage is significantly reduced in later generations. **c** Stability diagrams of SQDs representative of each generation are displayed. The uniformity of these stability diagrams improves due to the feedback loop. Reprinted from Ref. [24].

combine the ideas of Volk *et al.* (2019) [115] with neural networks that estimate the charge state and the tunnel coupling between dots.

2.5.8 Alternative approaches

Looking on from that, the need for tuning can be reduced by other means. Neyens *et al.* (2024) [24] show that improving the fabrication can make devices more predictable. They show that a median of 63% of the quantum dots per device could be set to the single electron regime with a single common plunger gate voltage after improving the fabrication of devices with rapid and large-scale characterisation. Fig. 2.18 illustrates the improvement of the devices using a feedback loop between measurements and fabrication.

A different approach is taken by Meyer *et al.* (2023) [117]. Using hysteretic effects, they show that the voltages to have a single electron on a quantum dot are actually tunable and can be tuned to a certain value. If properly exploited,

this could reshape tuning pipelines. However, it remains to be shown that this effect can actually improve tuning.

Finally, it is worth mentioning the work done by Klimov *et al.* (2020, 2024) [118, 119] on superconducting qubits. They show that on larger quantum processors with tens of qubits, optimisation can be done to enhance the performance on a systems level rather than for single qubits. Similar work will have to be done for spin qubits once they reach this size.

The automation of the tuning process is an active field of research and many groups have made significant contributions. The whole pipeline, however, remains to be automated from end to end. There is, therefore, an opportunity to address those shortcomings. A framework needs to be established that connects the different parts of the tuning process and possible gaps in the tuning pipeline need to be filled. This is the focus of this dissertation.

3

Automated identification of Pauli spin blockade

Contents

The challenges in constructing automated tuning algorithms are plentiful. One of the main challenges is to make decisions based on noisy data with complex structure. Identifying Pauli spin blockade (PSB) in charge transport measurement is an example. Its signatures can be subtle and there is not a lot of data available to train a decision algorithm.

In terms of the landmark metaphor introduced in Chapter 1, this is trying to answer the first question: How can we interpret what we see? I explain how a machine learning approach is able to solve this. I evaluate the performance of

two different neural network architectures trained on simulated and experimental data, examine the impact of the number of training samples, and show how this algorithm can be integrated into a tuning flow.

This chapter is adapted from my publication

> **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, D. Sejdinovic, and N. Ares, "Identifying Pauli spin blockade using deep learning"

published in the journal *Quantum* in 2023 [46]. Parts are taken verbatim from that publication.

This work was done in collaboration with the University of Basel. To honour the collaborative nature, I use the first person plural form in this chapter. My contribution was to perform the experiments, write the simulator, perform the training and evaluation of the neural networks, and write the manuscript, see also the author contributions statement of Ref. [46].

3.1 Context of Pauli spin blockade in spin qubits

PSB is often a crucial requirement for spin qubit initialisation and readout. It allows for spin-to-charge conversion, as spin-conserved tunneling leads to current rectification [47]. We can rely on PSB even at elevated temperatures (>1K) that do not require a dilution refrigerator and ease the requirements on the experimental setup [22, 23, 120, 121]. It is thus essential to reliably and efficiently detect PSB.

However, detecting PSB is challenging even for experienced experimenters because evidence for PSB is often subtle. In the few-charge regime, PSB can occur at unexpected gate voltages or might be entirely absent due to variations in device fabrication and material defects. In the multi-charge regime, identifying PSB is like finding a needle in a haystack; the regions in the stability diagram where PSB manifests can be extremely small and easily overlooked. Moreover, PSB can be partially lifted, e.g., by spin-orbit coupling, which can allow some current to flow even when PSB is expected. These partial lifts make it harder to distinguish PSB

from other transport phenomena. The appearance of PSB relies on several factors, including the precise details of the confinement potential, which are affected by fluctuations in the disorder potential due to fabrication variances and defects within the material. All these complexities contribute to the elusiveness of PSB.

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

We demonstrate how to detect PSB using deep neural networks. Deep neural networks were used in the context of charge state identification, coarse and fine tuning and readout [29–31, 33–43, 45, 106, 122], with some approaches using simulated data to train their algorithms [33, 34, 110]. Unlike these cases, automatic PSB detection required us to make use of extremely scarce and unbalanced quantum device data. We developed a physics-inspired simulator and introduced cross-device validation to address this challenge.

We demonstrate our algorithm in a silicon fin field-effect transistor (FinFET) confining holes [23]. We show that we can achieve an accuracy of over 96% on identifying signs of PSB on unseen devices. The data stems from four silicon FinFET devices with different gate dimensions. We designate training devices, from which we extract training data, and testing devices, from which we extract data to test our algorithm. We discuss the performance of the algorithm for different types of training data, using simulated training data, measured training data, and a combination of both.

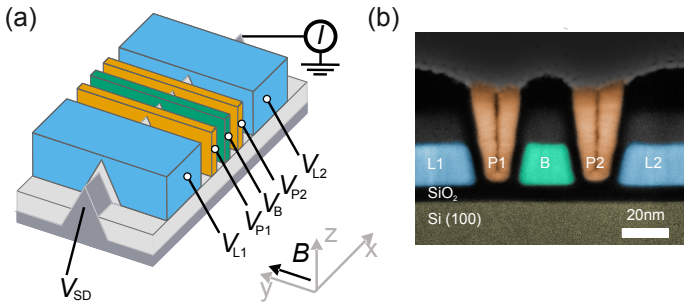


Figure 3.1: Silicon FinFET device. a Schematic and **b** cross-sectional TEM image of a silicon FinFET device. The plunger gates (P1 and P2) accumulate holes in a DQD, the inter-dot barrier is controlled by B and source and drain reservoirs are accumulated using lead gates (L1 and L2).

3.2 Experiment

A schematic representation of a silicon FinFET device similar to the ones used in this work and a cross-sectional transmission electron microscope (TEM) image are shown in Figs. 3.1. The devices are fabricated using a CMOS-compatible fabrication process, where a self-alignment technique allows for ultra-small gate length and intrinsically perfect layer-to-layer alignment [123]. The fin provides a quasi 1D confinement for holes and a double quantum dot (DQD) can be defined using gate electrodes. Source and drain reservoirs are formed by lead gates L1 and L2, which accumulate p-type carriers. The plunger gates P1 and P2 allow for control of the hole occupancy. The inter-dot coupling is controlled by gate B. We perform transport measurements by applying a bias voltage V_{SD} between source and drain drawing a current I through the device. Measurements of current as a function of the plunger gate voltages V_{P1} and V_{P2} are called stability diagrams. Energetically allowed charge transitions appear as two bias triangles in stability diagrams, see Fig. 3.2c. Bias triangles indicate that the device is tuned into the DQD regime.

Four devices with different dimensions (for details see Appendix A.1) were measured at different bias voltages and at temperatures ranging from 20 mK to 1.5 K. In all measurements the magnetic field was applied in-plane and perpendicular to the fin as indicated in Fig. 3.1a.

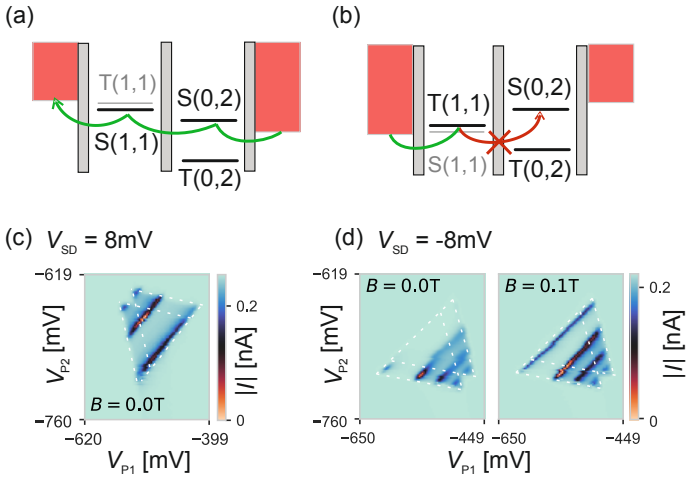


Figure 3.2: Pauli spin blockade mechanism and measurements. a, b Schematic of the transport cycle in the unblocked (**a**) and blocked (**b**) configurations, respectively. In the unblocked configuration (**a**), holes can easily tunnel through the device. In the blocked configuration (**b**), spin conservation blocks the transport through the ground state transition when inverting V_{SD} due to the forbidden $\mathrm{T}(1,1)-\mathrm{S}(0,2)$ transition. **c** Bias triangles with positive V_{SD}. **d** Bias triangles with negative V_{SD}. The current at the base line of the triangles, i.e. current due to the $\mathrm{T}(1,1)-\mathrm{S}(0,2)$ transition, is blocked (left). A finite magnetic field $B = 0.1\,\mathrm{T}$ lifts the blockade (right). We show the absolute value of current for ease of comparison between figures. We outline the bias triangles with white dashed lines to guide the eye.

3.2.1 Pauli spin blockade

PSB has been introduced in Chapter 2.1.3. The description there applies to both holes and electrons. We reiterate the main points here and give more context for PSB in the devices used in this chapter. For positive V_{SD}, we expect transport cycles to be possible (see Fig. 3.2a). When applying a negative V_{SD}, we expect transport through the DQD is blocked since the $\mathrm{T}(1,1)-\mathrm{S}(0,2)$ transition is forbidden by spin conservation (see Fig. 3.2b). The effect is also possible in the opposite bias direction, i.e. a blockade can occur with positive bias voltage for a $(1,1)-(2,0)$ charge transition.

The blockade can be lifted by processes that allow transitions out of the T(1,1) state. The mechanism for holes and strong spin orbit coupling in a magnetic field is described in detail in Ref. [124] and Chapter 2.1.3.

Signatures of PSB can be observed in Figs. 3.2c, d. Two bias triangles are visible for $V_{\mathrm{SD}} = 8\,\mathrm{mV}$. For opposite polarity $V_{\mathrm{SD}} = -8\,\mathrm{mV}$ the current at the common base line of the triangles, i.e. current due to ground state transitions, is strongly

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

3.2.2 Simulator

A simulator allows us to generate large and diverse data sets needed to train the deep learning algorithm. Our goal is to simulate pairs of stability diagrams, as introduced in Section 3.2.1. To achieve this, we first calculate the steady state current [125] with one energy level in each quantum dot. Next, we consider multiple energy levels in each dot and sum the contribution of every possible combination of energy levels to determine the total current.

Let us consider the steady state current through a double quantum dot coupled to fermionic reservoirs. For an energy level E_A in the left dot and an energy level E_B in the right dot we define the detuning as $\epsilon = E_A - E_B$. These energy levels are seen as the combined charge and spin states of an excess electron. We describe the simulator in terms of electrons but it holds true for holes as in the silicon FinFET used in the experiment. We consider an electron to already occupy the right dot. The tunneling rates from the left reservoir (the source) and to the right reservoir (the drain) are Γ_L and Γ_R, and the tunneling rate between the dots is Γ_T. This situation is shown in Fig. 3.3a. The stationary solution from Ref. [125] is given by

$$I_{\text{partial}} = \frac{\Gamma_T^2 \Gamma_R}{\Gamma_T^2(2 + \Gamma_R/\Gamma_L) + \Gamma_R^2/4 + \epsilon^2}. \tag{3.1}$$

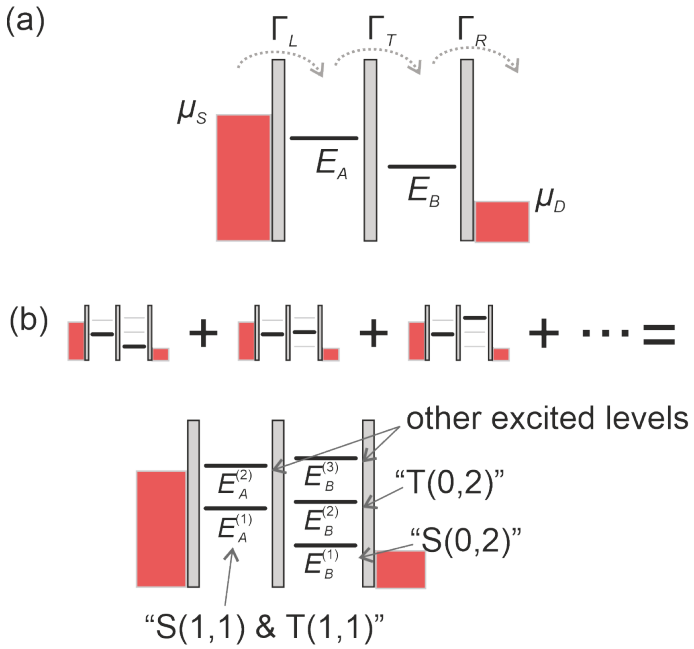


Figure 3.3: Explanation of the simulator. a Fundamental building block. If only one energy level is available in each dot, we can compute the current through a double quantum dot. **b** Simulation with several levels in each dot. We simply add the contribution of each pair according to 3.3a. If the DQD is in Pauli spin blockade, lower levels can be identified with singlet and triplet levels, higher levels are excited levels. The excited states are associated with the same number of charges in the DQD as the lowest states, but with additional energy, e.g., due to one charge occupying a higher orbital (see Fig. 2.3). To simulate the effect of Pauli spin blockade, transitions from a singlet to a triplet state are suppressed.

The total current consists of contributions due to different energy levels in the left and right dot, as illustrated in Fig. 3.3b. The excited states are associated with the same number of charges in the DQD as the lowest states, but with additional energy. This can be due to, e.g., one charge occupying a higher orbit (see Fig. 2.3). The current is given by

$$I = \sum_{E_A} \sum_{E_B} I_{\text{partial}} = \sum_i \sum_k \frac{\left(\Gamma_T^{(i,k)}\right)^2 \Gamma_R^{(k)}}{\left(\Gamma_T^{(i,k)}\right)^2 (2 + \Gamma_R^{(k)}/\Gamma_L^{(i)}) + \left(\Gamma_R^{(k)}\right)^2/4 + \epsilon_{(i,k)}^2}. \tag{3.2}$$

Each level in the left dot $E_A^{(i)}$ is associated with a source tunneling rate $\Gamma_L^{(i)}$. Accordingly, each level in the right dot $E_B^{(k)}$ is associated with a drain tunneling rate $\Gamma_R^{(k)}$ and each pair of energy levels $(E_A^{(i)}, E_B^{(k)})$ is associated with an inter-dot tunneling rate $\Gamma_T^{(i,k)}$. This equation is only used if the lowest energy levels in each dot are in the bias window. Otherwise the DQD is in Coulomb blockade and current will be suppressed by setting I to zero.

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

$$E_A^{(0)} = L_A \cdot V_A + C_B \cdot V_B, \tag{3.3}$$

$$E_B^{(0)} = C_A \cdot V_A + L_B \cdot V_B. \tag{3.4}$$

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

$$E_A^{(i)} = E_A^{(i-1)} + E_{\text{split}}^{(i)}, \tag{3.5}$$

$$E_B^{(k)} = E_B^{(k-1)} + E_{\text{split}}^{(k)}. \tag{3.6}$$

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

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

$$\Gamma_L = \tilde{\Gamma}_L f(E_A, \mu_S), \tag{3.7}$$

$$\Gamma_R = \tilde{\Gamma}_R[1 - f(E_B, \mu_D)], \tag{3.8}$$

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

$$\Gamma_T = \begin{cases} \tilde{\Gamma}_T, & \text{if } E_A - E_B \geq 0 \\ 0, & \text{else.} \end{cases} \tag{3.9}$$

Equations 3.1 to 3.9 produce one triangle of the pair we need. We can think of this as the cycle $(0,1) \rightarrow (1,1) \rightarrow (0,2) \rightarrow (0,1)$ with (m,n) indicating m electrons (holes) in the left dot and n electrons (holes) in the right dot. There is another cycle possible, namely $(1,2) \rightarrow (1,1) \rightarrow (0,2) \rightarrow (1,2)$. The bias triangle associated with this cycle is shifted due to the electrostatic coupling energy between the dots E_{C_m}. This second triangle is simulated by shifting all energy levels by E_{C_m} and repeating the current simulation discussed above. If the bias triangles overlap, only the maximum current is used.

To mimic experimental observations and to create a diverse data set, we get a set of two bias triangles by randomly sampling the scalar parameters

$$\{L_A, L_B, C_A, C_B, \sigma, \tilde{\mu}_S, \tilde{\mu}_D, \mathcal{T}, E_{C_m}\},$$

the vector parameters

$$\{\tilde{\mathbf{\Gamma}}_L, \tilde{\mathbf{\Gamma}}_R, \mathbf{E}_{\text{split}}\},$$

and the elements of the matrix $\tilde{\mathbf{\Gamma}}_T$. The dimensions of the vectors and the matrix depend on the number of energy levels we consider in each dot, which is also sampled randomly. This means that the total number of parameters varies between 24 (when the minimal number of energy levels is used) and 48 (when the maximal number of energy levels is used). All these parameters are significant as they define the physical properties of the simulated quantum dot system. The simulator then generates a two dimensional charge stability diagram. A list of all sampling ranges can be found in Appendix A.2.

Finally, we add different types of noise to the simulations further to the noise already described. Gaussian blurring is used to smooth the edges of the triangles. The triangles can also move in voltage space due to charge switches or other drift effects. Charge switch noise at random points simulates both effects, see Figure 3.4b bottom left for an example. We also add white noise to the final current values at each simulated point.

To simulate the effect of PSB we add simple rules about where and how much current is allowed. The right lowest level represents the S(0,2) level and we neglect the small splitting of the left lowest energy level into S(1,1) and T(1,1), see Figure 3.3b, making it only one level. With PSB, tunneling between these two states is prohibited because an electron will eventually occupy the T(1,1) state and will not be able to tunnel to S(0,2). This leads to the rule:

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

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

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

While this can lead to simulated measurements that may be unphysical, it enables efficient training of the algorithm. We sample each parameter of the simulator, e.g. lever arms, from a given range of possible values. We did not perform an optimisation over those sampling ranges to avoid introducing a greater bias to the output of the simulator.

Fig. 3.4 shows examples of simulated bias triangles. PSB is introduced for $B_{\text{sim}} = 0$ while it is not considered for $B_{\text{sim}} \neq 0$ (Fig. 3.4a). Each pair of simulated stability diagrams is marked by a grey box in Fig. 3.4. Examples of bias triangles where PSB does not occur are shown in Fig. 3.4b. In this case the difference between measurements in a pair is due to the added noise.

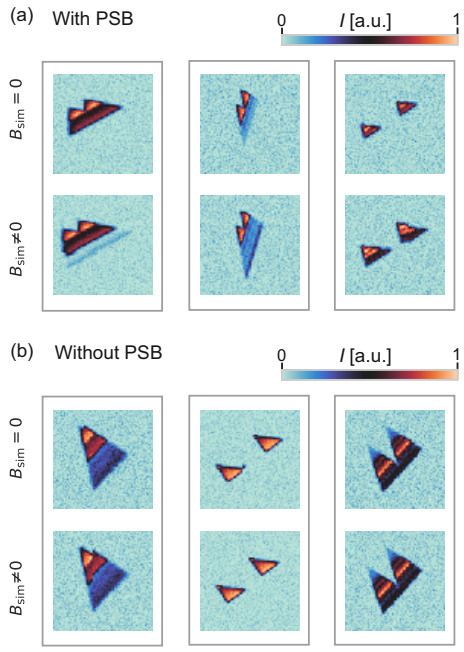


Figure 3.4: Examples of simulated training data. Each pair in a and b (grey boxes) represents a different set of parameters. The simulated current values are normalised for each of these pairs. **a** Examples displaying PSB. **b** Examples in which PSB is not introduced in the simulation. Running the simulator twice with the same parameters results in two similar measurements that only differ due to the noise added to the simulator.

3.2.3 Deep learning

The neural network used is a deep residual network (ResNet) [126] with 18 layers. A ResNet is a standard and well-established convolutional neural network architecture that benchmarks very well on image recognition and classification tasks [127, 128]. It introduces shortcut connections to mitigate the vanishing gradient problem, allowing for the effective training of deeper networks. This architecture was chosen because it enables the model to learn more features from the stability diagrams without the degradation problems that typically occur in very deep networks, and because it is a proven, reliable choice in the field.

The input to the neural network is a pair consisting of two stability diagrams, as defined in Section 3.2.1. The current input values are jointly normalised between 0 and 1 for each pair. The neural network outputs a score between 0 and 1, where a score of 1 corresponds to maximum confidence in the occurrence of PSB, and a score of 0 corresponds to minimum confidence. The threshold for classification is set at 0.5.

Due to the random sampling of parameters in the simulations and the randomness in the data augmentation process, we expect a high variance in the results of the classification if we train the neural network more than once. We thus train the neural network ten times to obtain ten individual classifiers that we combine into an ensemble of classifiers. By ensemble classifier we mean a classifier that uses the average score of all individual classifiers as the score of the ensemble. This approach is expected to produce more robust results than individual classifiers. We will later compare the performance of the neural network based on the ResNet architecture with the performance of a smaller network based on the LeNet architecture. Details on the architectures and the training of the neural networks can be found in Appendices A.3 and A.4.

3.3 Results

To test our algorithm, we consider a device that is tuned to a DQD regime. Often, the next step is to identify bias triangles which might exhibit PSB. We select gate voltage windows enclosing bias triangles as indicated by rectangles labelled A-C in Fig. 3.5a. All other bias triangles that could be observed in this device are displayed in Figure 3.6. To create the pairs introduced in Section 3.2.1 we combine the bias triangles delimited by the chosen rectangles with an equivalent version of these measurements at low magnetic field.

Magnetic field hysteresis can shift the PSB signature away from $B = 0$. This can sometimes occur when using superconducting coils. We thus do not set $B = 0$ for these measurements but we choose 9 different equidistant magnetic fields between $B = -0.08\,\mathrm{T}$ and $B = 0.08\,\mathrm{T}$. In this way, we create nine pairs of stability diagrams for each gate voltage window A to C. Example pairs are displayed in Fig. 3.5b.

The predictions obtained by the ensemble of classifiers trained on simulated data as described in Section 3.2.3 can be seen in Fig. 3.5c. All predictions for charge transitions A and B are negative, i.e. no signs of PSB are detected. For charge transitions in C PSB is detected for pairs with low magnetic field values $B = -0.06, -0.04, -0.02\,\mathrm{T}$ even though the base line is very faint when PSB is

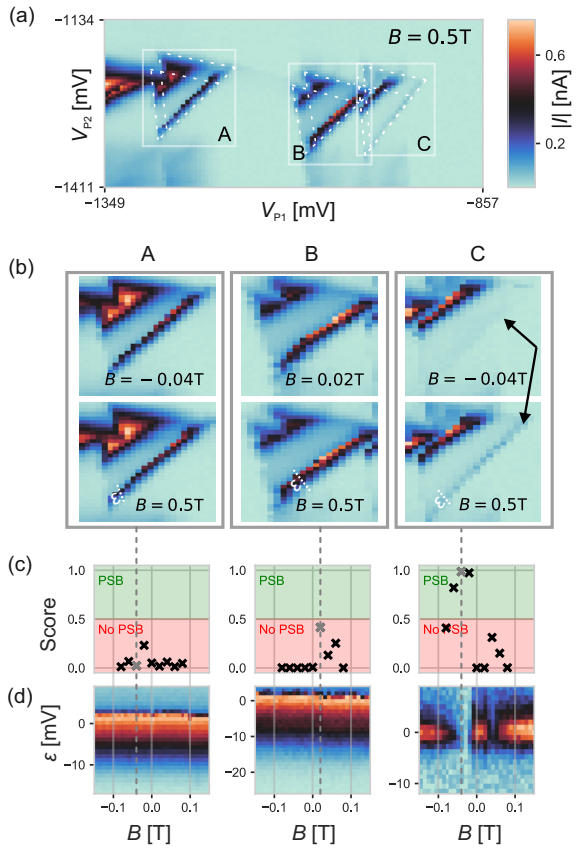


Figure 3.5: Using the classifier to find bias triangles exhibiting PSB. a Stability diagram taken at $B = 0.5\,\mathrm{T}$. White rectangles are drawn by a human wishing to classify bias triangles A-C. As in Fig. 3.2, we display the absolute value of current and outline the bias triangles with dashed white lines. **b** Examples of pairs that are inputs to the classifier. The bottom row shows the bias triangles from Fig. 3.5a, which serve as a reference, since we expect PSB to be lifted at this magnetic field value. The same bias triangles are shown in the top row at low magnetic field. Black arrows point at a vanishing common base line indicating the presence of PSB. **c** Classification of bias triangles A-C. The ensemble of classifiers produces a score for each of the nine pairs corresponding to charge transitions A-C, which are composed by paired measurements at $B = 0.5$ T and at values of B close to zero ($B = -0.08$ to 0.08 T). A score of over 0.5 predicts the occurrence of PSB. **d** Magnetic field dependence of current measurements along the detuning axis, indicated by white dotted lines in Fig. 3.5b. A reduction in current confirms the predicted PSB for three pairs of C. We draw a dashed line from the pairs in Fig. 3.5b through Fig. 3.5c and Fig. 3.5d to indicate the magnetic field values corresponding to those pairs. More results can be found in Figure 3.6.

lifted. Our algorithm classifies pairs, not transitions. For charge transition C, some pairs are classified as not exhibiting PSB for those magnetic fields for which the signature of PSB is not apparent. Also, for this transition, the blocked current occurs at $B \neq 0$. We assume that the effective magnetic field is 0 for the blocked

case and the offset is due to hysteresis.

To confirm the predictions, we measure the current at the base of the bias triangles as a function of the magnetic field and detuning, see Fig. 3.5d. The detuning axes, i.e. the sweep direction of gate voltages V_{P1} and V_{P2}, are indicated as white dotted lines in Fig. 3.5b. For charge transitions in C, the current suppression is evident at low magnetic field values. This verifies that the corresponding pairs are correctly identified as displaying PSB. Conversely, no current reduction is observed for bias triangles A and B, confirming the absence of PSB. An automated tuning pipeline could use this information to perform further experiments, such as measurements of Rabi oscillations.

More classification results

Fig. 3.6 shows more classification results. Fig. 3.6a shows the stability diagrams with the identified bias triangles as white boxes and Fig. 3.6b shows the predictions of the classifier, which predicted no PSB for charge transitions D-F.

We repeat the same experiment for reversed bias. Fig. 3.6c shows the stability diagram with identified bias triangles. We call them G-L to distinguish them from the measurements in the main text even though they correspond to charge transitions A-F.

Here, all bias triangles were classified as not having PSB, shown in Fig. 3.6d.

3.3.1 Benchmarking

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

We built a data set consisting of 53 pairs, as defined in Section 3.2.1, originating from measurements of four different devices. In this data set, we only included examples that exhibit well shaped bias triangles and measurements that show either clear signatures of PSB or no signatures of PSB at all, so that human experts can verify the correctness of the label. Table 3.1 shows how many pairs are associated with each device. All pairs used are shown in Fig. 3.8.

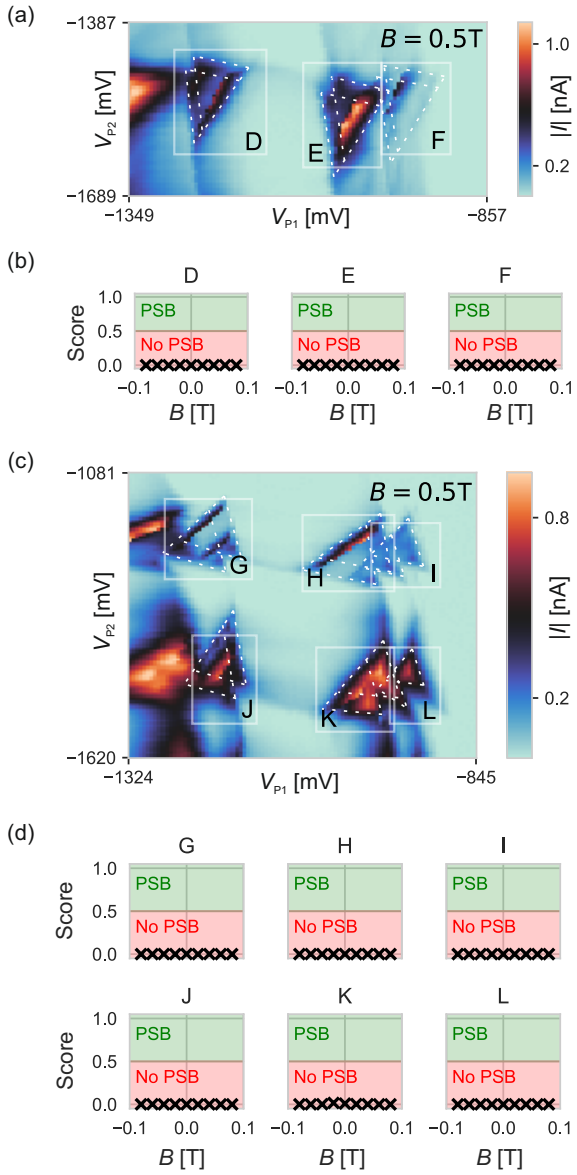


Figure 3.6: More classification results. **a** Stability diagram for more bias triangles. **b** Classification results corresponding to charge transitions in **a**. **c** Stability diagram for reversed bias. **d** Classification results in the reversed bias case. As in Figs. 3.2 and 3.5, the absolute value of current is shown and dashed white lines outline the bias triangles in **a** and **c**.

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

Table 3.1: Structure of experimental data used to benchmark the algorithm. For each of the devices considered, number of pairs displaying PSB (positive) and not displaying PSB (negative) as assessed by a human judge. Data from device ii was collected over multiple cool-downs. A few individual pairs might show the same charge transitions in different locations in gate voltage space, e.g. for different tunnel coupling strengths. The data set includes measurements corresponding to $B = -0.04\,\mathrm{T}$ from Fig. 3.5 and Fig. 3.6.

To study the effect of different training data on the performance of the algorithm, we investigate three cases, which we refer to as Simulated data (**Sim**), Experimental data (**Exp**) and Mixed data (**Mix**). **Sim** corresponds to the case of training the classifiers with only simulated data. These classifiers are the same as those used for the predictions in Fig. 3.5. For **Exp**, training is performed only with experimental data from the devices listed in Table 3.1. **Mix** is a mix of training with experimental and simulated data; half of the training data is experimental and the other half is simulated.

We augment the training data by random shearing, stretches, crops, contrast and brightness such that there are 50,000 pairs. Details of training and augmentation can be found in Appendix A.4.

Since our data set of measurements is small we employ a form of cross-validation in the cases **Exp** and **Mix** which we call cross-device validation. This means that each classifier is tested on data from a device that is different from the devices it was trained on. Each of these groups of training and testing data forms a *fold*. The process is repeated until all devices have served as a testing device once. Because we don't have both positive and negative pairs from device i we don't use that device as a testing device as computing some specific performance metrics is not possible. In computation of the cross-validation performance metrics, we weight each fold according to the number of pairs it holds. This can be seen as a form of inverse-variance weighting. In the case **Sim** we do not need cross-validation since we use only experimental data for testing.

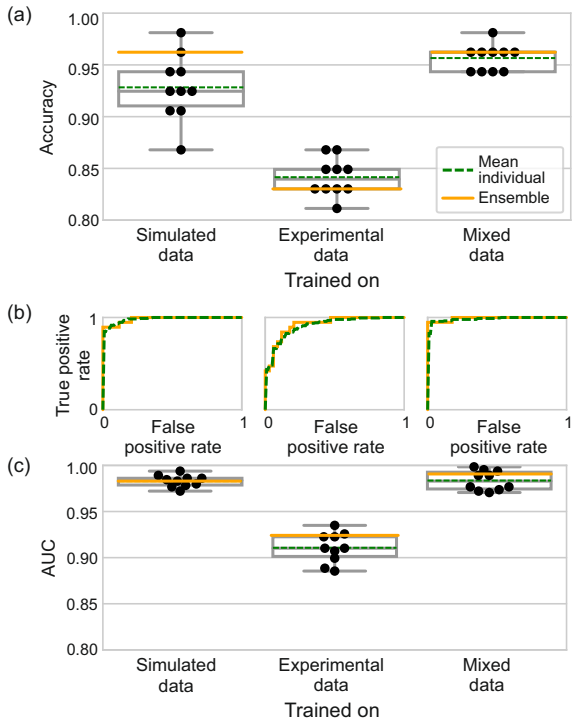


Figure 3.7: Benchmarking results. a Accuracy. Box plots with results from individual classifiers plotted as dots, the mean of those is plotted as a dashed green line, and the performance of the ensemble of classifiers as a solid orange line. **b** Receiver operating characteristic (ROC). For each case, the ensemble ROC is plotted as a solid orange line and the mean individual ROC as a dashed green line. **c** Area under the curve (AUC). Legend as in Fig. 3.7a.

We define accuracy as the proportion of correctly classified data in the complete data set. The accuracies of single and ensemble classifiers are shown in Fig. 3.7a. Training the classifier with only simulated data (case **Sim**) leads to an ensemble accuracy of 96.2%. This high accuracy indicates that our simulator produces data that are similar to experimental measurements. The ensemble classifier outperforms the mean of individual classifiers (92.8%), justifying the use of ensemble classifiers. Previous classifiers that classify data from quantum devices show accuracies below 90% [110]. The mean accuracy of individual classifiers decreases to 84.2% for classifiers trained purely on experimental data (case **Exp**) and the ensemble of classifers achieves an accuracy of only 83.0%, showing the advantage of a simulator. These lower accuracies indicate over-fitting as a result of small training data sets. The simulated data provides a more representative and diverse data set that prevents this problem. Mixing the two data types (case **Mix**) leads to the same ensemble classifier

accuracy as using only simulated data (96.2%). The mean of the individual classifiers improves to 95.7% and a lower variance of the individual classifiers is observed.

We find that neural networks trained only on experimental data strongly underperform those that were trained with only simulated data or a mix of both types of data. Thus, the simulated data seems to be the main driver of performance in contrast to the findings in [110]. The superiority of classifiers trained on simulated data may be due to the scarcity of experimental data and specific to the problem of detecting PSB. We train with between 31 and 48 experimental measurements before augmentation in the cases **Exp** and **Mix** depending on the fold. In the case **Sim** (**Exp**) we use 25,000 (12,500) simulated pairs before augmentation. The influence of the number of pairs used in training is discussed later in Section 3.3.4.

The accuracy can be affected by the choice of the score threshold so we use other metrics to further analyse our results. Choosing a score threshold means navigating a trade-off between true positive rate (TPR) and false positive rate (FPR). The receiver operating characteristic (ROC) curve, a plot of TPR against FPR, illustrates this trade-off, see Fig. 3.7b. The area under the ROC curve (AUC) is independent of the score threshold and is 1 for a perfect classifier. An arbitrary classifier would produce a ROC that is a diagonal with an AUC of 0.5. In the case **Sim** we obtain an AUC of 0.983 for the ensemble classifier, see Fig. 3.7c. This can be slightly improved by mixing in experimental data (case **Mix**), leading to an AUC of 0.991. In comparison, only using experimental data (case **Exp**) gives an ensemble AUC of 0.924. The mean individual classifiers achieve an AUC of 0.983, 0.910, and 0.984 in cases **Sim**, **Exp** and **Mix**, respectively. The results obtained by estimating the AUC are similar to those obtained by calculating the accuracy of the classifiers; training only on experimental data results in under-performing classifiers.

We show all experimental data used for the training and testing of the classifier in Fig. 3.8. We split them according to device source (row) and whether they show signs of PSB or not (column). Data from device iv corresponds to charge transitions A, B, C, D, and H with a low magnetic field of $B = -0.04\,\mathrm{T}$ (see Fig. 3.5 and Fig. 3.6). We show three ensemble classifier scores (one for each type of training

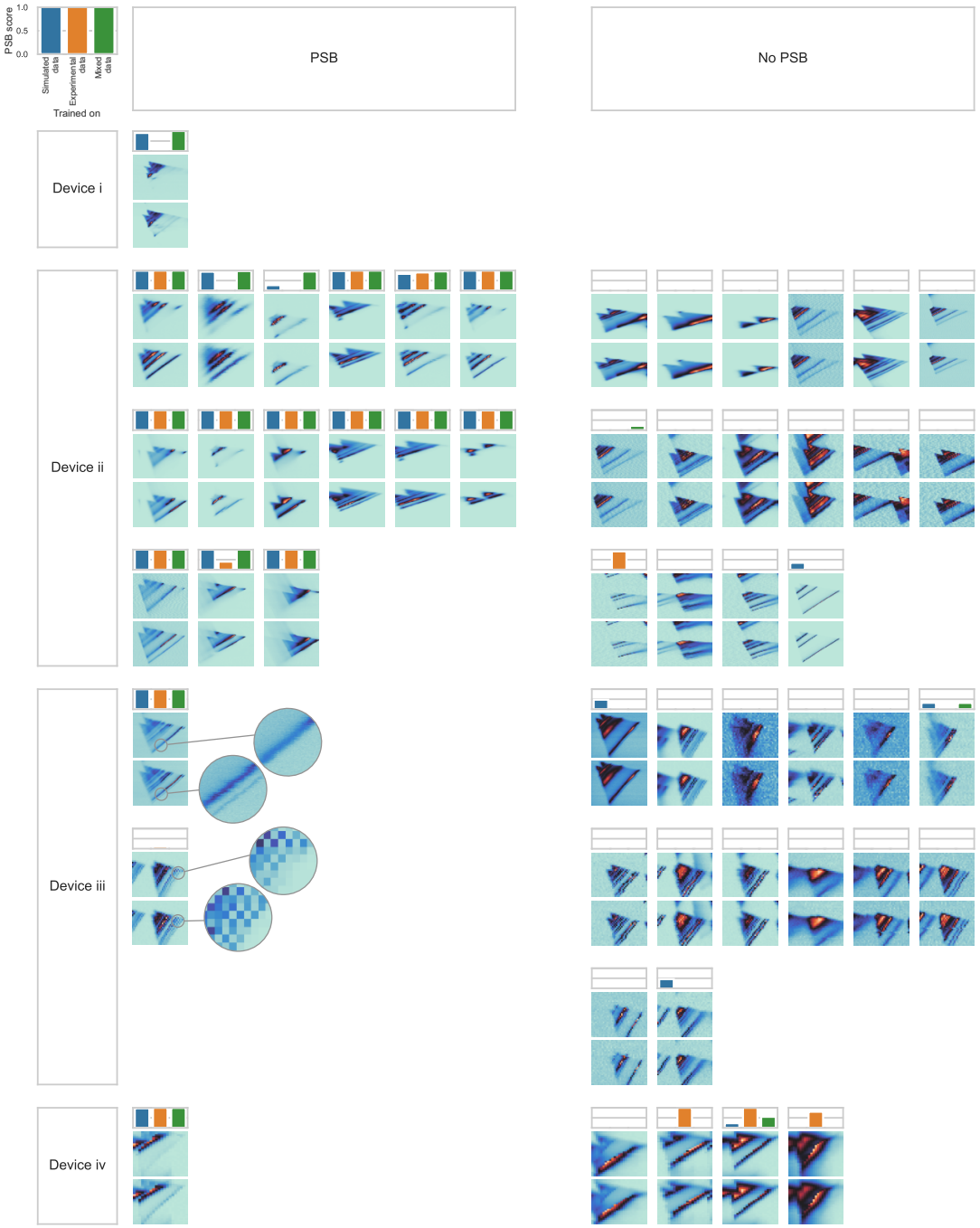


Figure 3.8: All experimental data used for benchmarking and the prediction of the ensembles of classifiers. There are two main columns (sets with and sets without PSB) and four main rows (devices i to iv). The measurements shown in the left column (marked "PSB") are not definitive proof that PSB is present but show the signs we expect and want to detect with a classifier. In the top left corner there is an example prediction plot with all labels: We show ensemble scores for the case of training on simulated data (blue), experimental data (orange), or a mix of both (green). The prediction threshold is plotted in the background as a horizontal line at a score of 0.5. Each set consists of a current measurement with low or zero magnetic field (top) and with a large magnetic field (bottom) and is jointly normalised between 0 and 1. Above each set the score plot is shown for that set without labels. The sets with PSB of device iii additionally show a magnification of a part of the base of the triangles due to visibility.

data) for each example. In the left column, we expect a perfect classifier to predict PSB (which would mean a full bar), in the right column it would predict no PSB (which would mean a missing bar).

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

Some bias triangles show only weak signs of PSB, such as the ones from device iii. Even though we can not be sure from those measurements that this is indeed due to PSB and not due to other effects, e.g. orbital effects, we still label those bias triangles as showing PSB. The point is that the signature could be due to PSB and should therefore be caught by a classifier. Confirmation measurements such as the ones done in Fig. 3.5d need to be performed after that to verify the prediction.

3.3.2 Benchmarking with smaller neural network

We repeat the benchmarking as in Section 3.3.1 with the smaller LeNet architecture described in Appendix A.3.2. The results are shown in Fig. 3.9. The performance in the case **Exp** is improved significantly, reaching an ensemble accuracy of 90.6%. This indicates that the size of the ResNet18 architecture leads to over-fitting. However, given the size of our dataset, we are limited in the creation of a holdout dataset to report generalisation capability and to tune the hyperparameters of the training without risking overfitting (see Appendix 3.3.3). Still, the ensemble trained in the **Mix** setting is outperforming the **Sim** and **Exp** case, showcasing the usefulness of the simulator. The ensemble classifier of **Mix** reaches an accuracy of 98.1% and an AUC of 1.0.

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

3.3.3 Testing on simulated data

To further judge the generalisation performance of the models, we test them on new simulated data. We simulate 1000 pairs of stability diagrams. For **Exp**

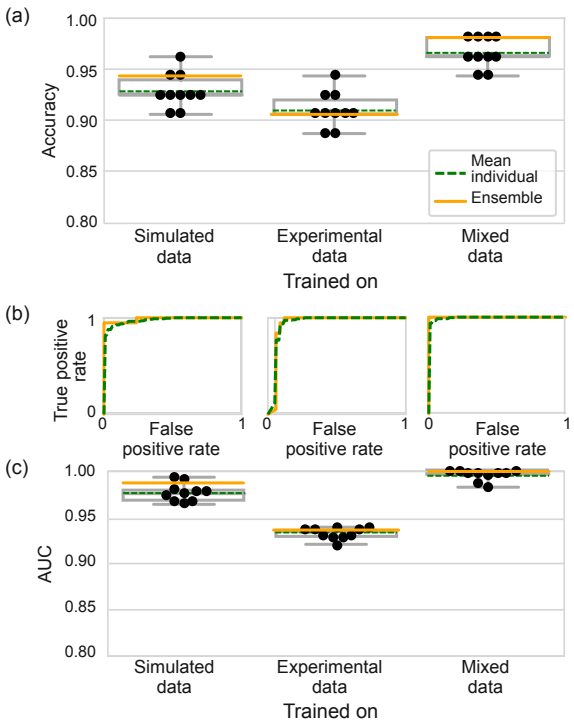


Figure 3.9: Benchmarking results with LeNet architecture. This is the same analysis as in Fig. 3.7 but with a different neural network architecture. **a** Accuracy. Box plots with results from individual classifiers plotted as dots, the mean of those is plotted as a dashed green line, and the performance of the ensemble of classifiers as a solid orange line. **b** Receiver operating characteristic (ROC). For each case, the ensemble ROC is plotted as a solid orange line and the mean individual ROC as a dashed green line. **c** Area under the curve (AUC). Legend as in Fig. 3.9a.

and **Mix** we use models that were trained on all available data, i.e. we do not separate data into folds.

The results are similar to those found testing on experimental data but with notable changes when comparing the models with different sizes. Fig. 3.10 shows the performance of the smaller LeNet models and Fig. 3.11 of the larger ResNet models. With an ensemble accuracy of 86.5% and an AUC of 0.956, the smaller LeNet model is performing very similarly to the larger ResNet model in the **Exp** case with an ensemble accuracy of 87.5% and an AUC of 0.943. For the other cases, the larger ResNet (**Sim**: ensemble accuracy 98.4% and AUC 1.0; **Mix**: ensemble accuracy 98.5% and AUC 1.0) performs better than the LeNet (**Sim**: ensemble accuracy 96.5% and AUC 0.999; **Mix**: ensemble accuracy 97.3% and AUC 0.997) which hints at a greater generalisation capability of the larger network when trained

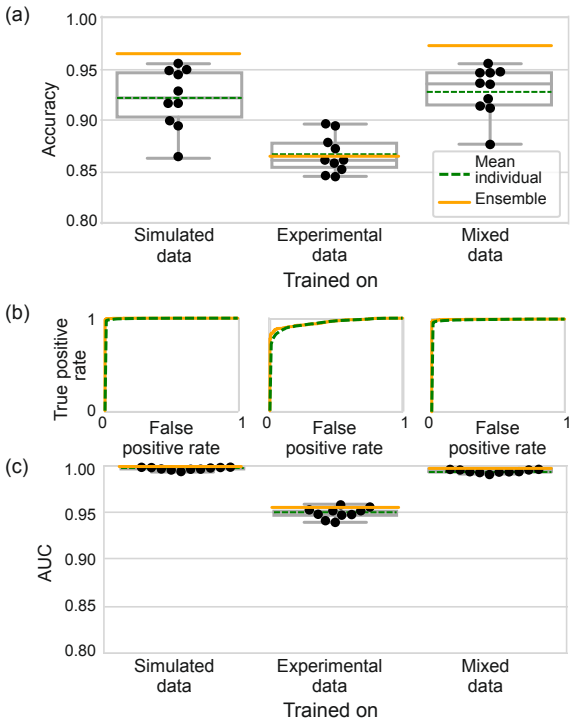


Figure 3.10: Benchmarking results on simulated data with LeNet architecture. Description as in Fig. 3.9 but for testing on simulated data.

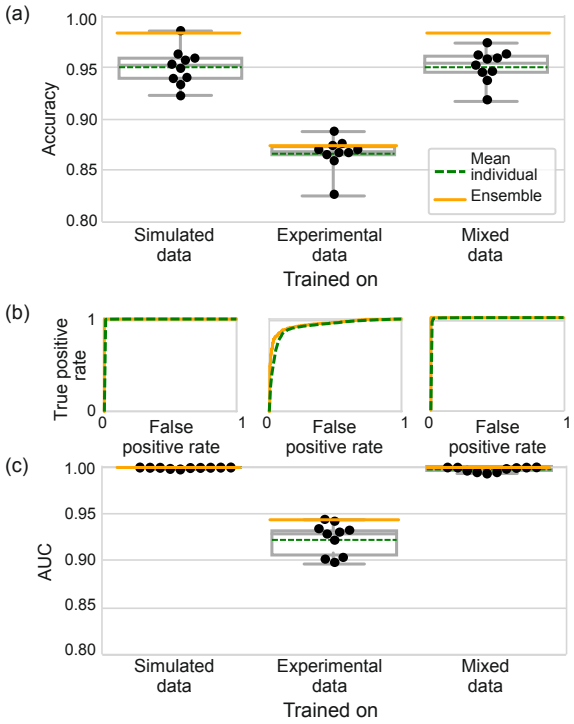


Figure 3.11: Benchmarking results on simulated data with ResNet architecture. Description as in Fig. 3.9 but for testing on simulated data.

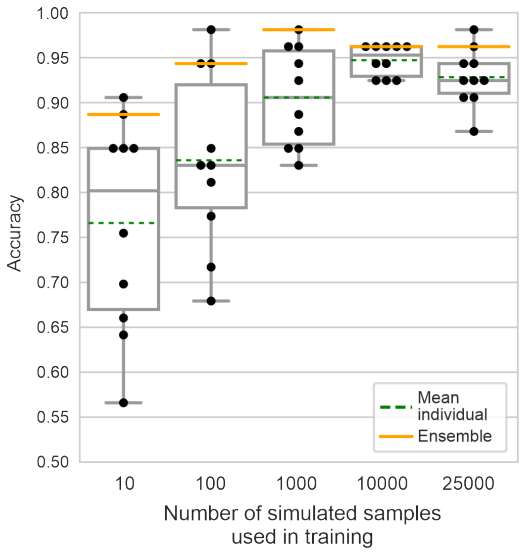


Figure 3.12: Varying simulated sample size. We show the accuracy for varying sizes of the simulated data set. In all cases the simulated pairs were augmented until 50,000 total pairs are created. 25,000 pairs in the training set corresponds to the analysis shown in the main text in Fig 3.7.

with sufficiently diverse data.

3.3.4 Influence of the number of simulated pairs

We investigate the role of the number of simulated pairs when training with only simulated data. We simulate 10, 100, 1,000, or 10,000 pairs and augment them until we have 50,000 pairs. The accuracy for classifiers trained on those data sets in comparison with the sample size used in the main text (25,000 sampled pairs) is shown in Fig. 3.12.

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

3.4 Discussion

We train deep neural networks with simulated and experimental data to detect bias triangles that show signs of PSB. We demonstrate that even in the case of extremely limited data, a neural network can be successfully trained to solve this

intricate task. Cross-device validation allows us to show that the method performs well on unseen devices.

We find a higher variance of accuracy of individual classifiers when trained on simulated data compared to classifiers trained on real data. This might be due to the limitations of the simulator and could be mitigated by increasing the number of simulations used in training at a larger computational cost. Forming an ensemble prediction leads to a high accuracy. In contrast to previous work [110], simulated data seems to be more important for training than experimental data. This might be due to the scarcity of experimental data available for the classification of PSB. The small experimental data set potentially lacks comprehensive information, which could hinder the neural network's ability to learn.

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

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

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

The simulator developed in this study holds the potential for various applications, including the development of an algorithm capable of recognising energy splittings or defining an energy detuning axis. The integration of simple simulators to train machine learning models can be applied to tackle a wide range of quantum

device challenges, including feature extraction and efficient measurement of stability diagrams, Rabi chevrons, and EDSR spectroscopy.

In light of the subtlety of the problem of identifying PSB this method has promising applications in the automation of tuning procedures for spin qubit devices. In the next chapter we embed the classifier in a larger tuning algorithm to determine whether a charge transition is promising for PSB.

There is a finite number of ways you can screw it up,
so you will eventually finish.

— Shane Wighton

4
Fully autonomous tuning

Contents

In the previous chapter I explained how to teach a computer to make a difficult decision by itself, in the landmark metaphor this means it can now "see the Radcliffe Camera". An algorithm that can tune a device from "from scratch" needs to make many of those decisions. Additionally, it requires routines that navigate through the parameter space and a framework that connects everything.

The algorithm in this chapter achieves all of those things. It is adapted from my publication

> **J. Schuff**, M. J. Carballido, M. Kotzagiannidis, J. C. Calvo, M. Caselli, J. Rawling, D. L. Craig, B. van Straaten, B. Severin, F. Fedele, S. Svab, P. Chevalier Kwon, R. S. Eggli, T. Patlatiuk, N. Korda, D. Zumbühl, N. Ares, "Fully autonomous tuning of a spin qubit"

which has been published in *Nature Electronics* [129]. Parts are taken verbatim from that publication. This work was done in collaboration with the University of Basel and Mind Foundry. I will present the work in the "we" form to reflect this. I developed the modular framework and the overall algorithm flow. I performed the experiments and analysed the results. I developed some of the routines for the various stages of the algorithm completely myself, and some of them in collaboration with Mind Foundry, and I wrote the manuscript. I refer the reader to the author contributions statement of Ref. [129] for a detailed list.

The process of going from a fully de-energised 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 without the need for human intervention.

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.

4.1 Device architecture and readout technique

We consider a common layout for a DQD device (inset of Fig. 4.1**b**). 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. 4.2.

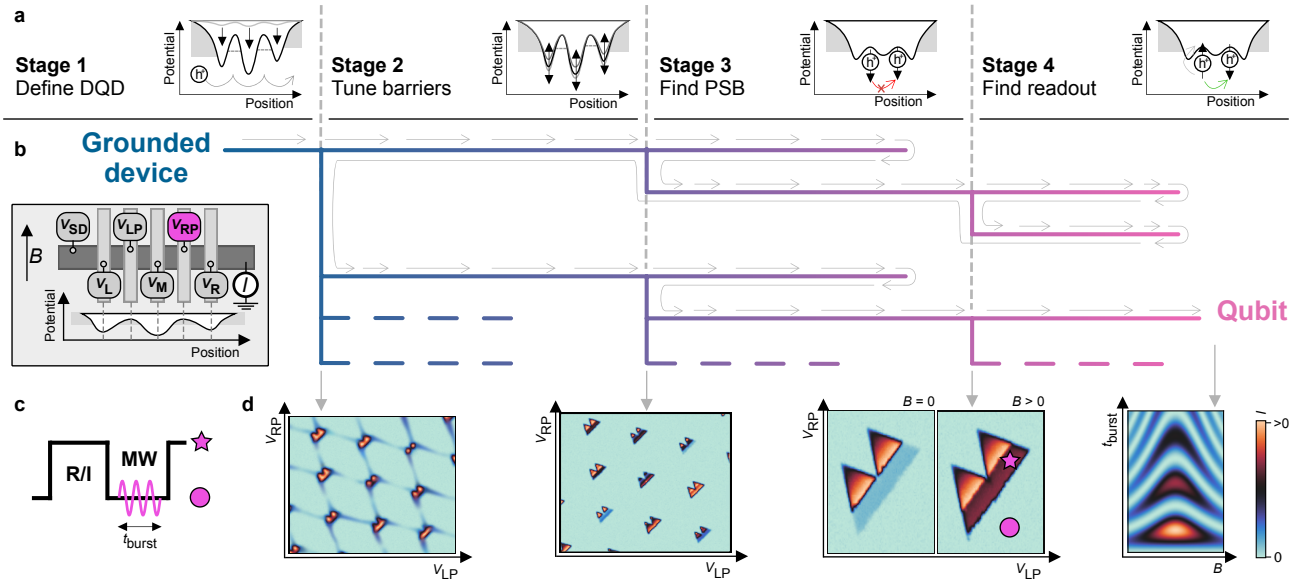


Figure 4.1: Algorithm overview. a, There are four stages that the algorithm needs to successfully navigate to reach qubit operations. The goal of each stage is illustrated with a confinement potential diagram. **b,** Each stage can either be successful and produce candidates (leading to one or more branches), 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.

Two plunger gate electrodes predominantly shift the electrochemical potential in the left and right dots with voltages V_{LP} and V_{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 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 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 [64, 124, 130, 131].

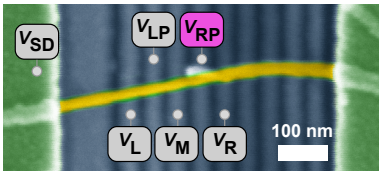


Figure 4.2: Device. False color micrograph of a nanowire device similar to the one used in this work. The Ge/Si nanowire is colored in yellow.

The algorithm uses a two-stage pulsing protocol [23, 132–134] which is parameterised by a microwave (MW) pulse frequency and a duration t_{burst} (Fig. 4.1c). This protocol allows for qubit manipulation if the spin resonance conditions are met.

To amplify the measurements that rely on a microwave pulse, we applied a pulse-modulation by a lock-in amplifier at 87.777 Hz. This specific frequency was chosen to minimise interference from ambient electrical noise, such as power line frequencies (typically 50 Hz) and their harmonics. By selecting a frequency that is not an integer multiple of these common noise sources, we reduce the likelihood of noise contaminating our measurements. The measurements have an in-phase and a quadrature component. We apply PCA to these measurements and project each measurement onto the principal axis, see Section 2.4.1. 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 $|I|$. The device, measurement apparatus and pulse sequence are the same as used by Carballido *et al.* [81] and are described in detail in their work.

4.2 The algorithm

The autonomous tuning algorithm is structured into four main stages. Starting from a completely de-energised 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

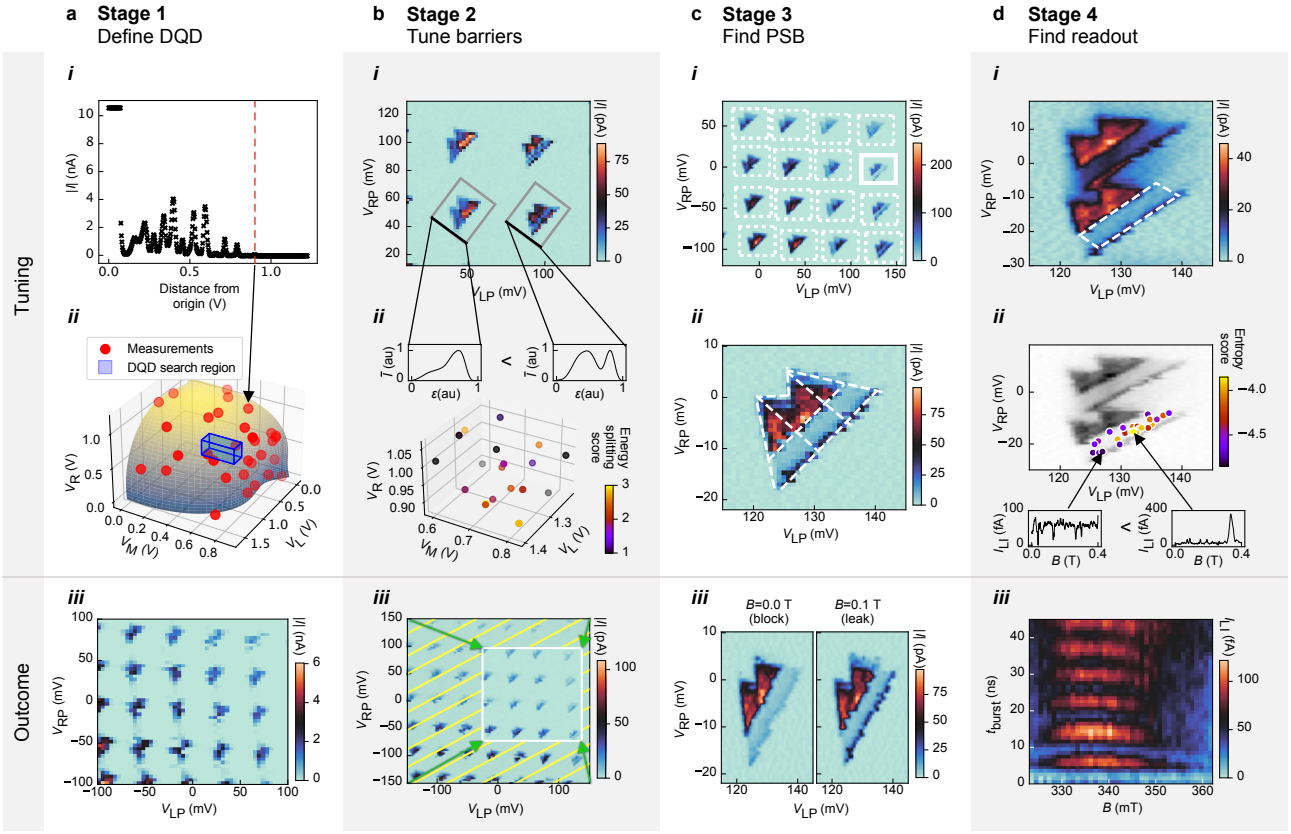


Figure 4.3: Summary of each stage. **a,** Stage 1: Definition of a DQD potential. **a-*i*** Current measurements along random directions in gate voltage space to determine the points at which conducting and non-conducting regions meet, so-called pinch-off points. **a-*ii*** Gaussian processes model the 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, optimised 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 Section 2.4.1 for details.

to drive the qubit. The effect of each stage on the DQD confinement potential is illustrated in Fig. 4.1**a**. Measurement illustrations exemplifying those taken by the algorithm are shown in Fig. 4.1**d**.

As a result of the algorithm design a search tree emerges, as shown in Fig. 4.1**b**. 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. 4.1. The underlying modular framework is described in Appendix B.4.

4.2.1 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.

Hypersurface building

Building upon the methodologies of Moon *et al.* [30] and Severin *et al.* [31], a hypersurface model is created to distinguish between conducting and non-conducting regions within the three-dimensional barrier gate voltage space. 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. The algorithm takes current measurements along random directions within this space (Fig. 4.3 a-i). 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. Then, the algorithm models the hypersurface with a Gaussian process, as depicted in Fig. 4.3 a-*ii*.

Double dot detection

We expect a DQD potential forming near a corner of the hypersurface in the first octant. To pinpoint this corner, three pinch off measurements as described above are conducted but using only one of the barrier gate electrodes at a time instead of a combination of them. 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. 4.3 a-*ii*.

A methodical search within this box is conducted by the algorithm. The algorithm samples quasi-random locations via a Sobol sequence in the DQD search region and investigates them, starting from the point nearest to the projected corner and progressing to higher gate voltages. Following the method described by Moon *et al.* [30], 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. We use the random forest classifier developed by Severin *et al.* [31] to check for the presence of Coulomb peaks.

If Coulomb peaks are found, a measurement varying both plunger gate voltages, a so-called stability diagram, is acquired and analysed via a neural network for DQD characteristics; see Fig. 4.3 a-*iii* for a stability diagram that shows the desired features. The neural network was trained on data from a variety of devices, mainly from the data gathered by Moon *et al.* [30] and Severin *et al.* [31], 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. Successful

identification of DQD features at a location in gate voltage space establishes it as a lower limit for the subsequent search.

4.2.2 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.

Barrier optimisation

Within the gate voltage bounds established in the first stage, BO is employed to search gate voltage combinations. After BO suggests a voltage setting, a stability diagram is measured (Fig. 4.3 b-*i*). We then use a segmentation algorithm [135] to find the outlines of individual pairs of bias triangles.

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.

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. 4.3 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.

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. 4.3b-*iii* or Fig. 4.4 for examples. These

bias triangles are excluded from the optimisation. The gate voltage regions explored by the optimisation are shown in Fig. 4.3 b-*ii*.

This classifier for switch detection was trained to distinguish between normal bias triangles and ones that are affected by charge switch noise, as illustrated in Fig. 4.4. 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.

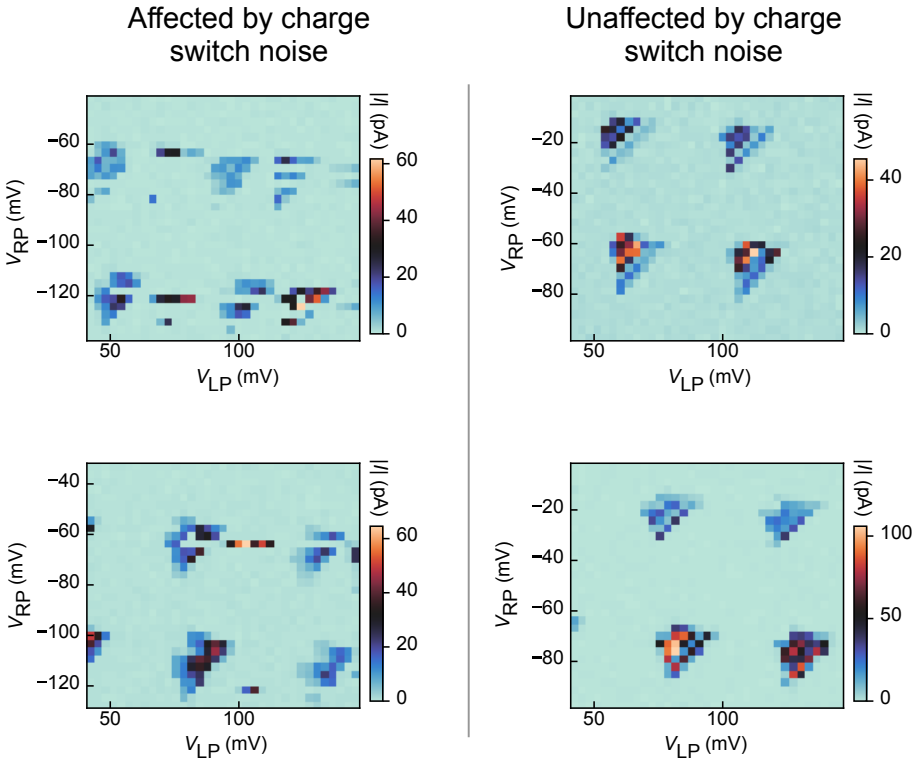


Figure 4.4: Effect of charge switch noise. Stability diagrams can be affected by charge switches. Their effect can be seen in the left column. For comparison, the right column is unaffected by these charge switches.

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 Appendix B.2 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.

Plunger window selection

Given a sampled stability diagram containing bias triangles, the aim is to select the region that maximises the number of high-scoring bias triangles, avoids areas with excessively high current, and minimises the number of switches, as illustrated Fig. 4.3 b-*iii*. High current bias triangles are typically unsuitable for qubit operations; therefore, a conservative current threshold of 200 pA is applied. To simplify downstream processing, the selected region should be a rectangular window.

The selection process follows an iterative approach, beginning with the smallest bounding boxes containing each individual pair of bias triangles. Larger windows are constructed by merging these initial boxes, provided they meet the conditions of switch absence and low current. Switch absence is treated as a soft constraint, allowing for some bias triangles affected by switch noise, as long as the total area impacted by switches is less than 25% of the area covered by all triangles in the window.

The algorithm's complexity scales exponentially with the number of triangles, so heuristics are employed to reduce complexity and execution time. Specifically, at each iteration, only the top 100 bounding boxes, ranked by the number of contained triangles without switches, are retained. This limits the number of potential merges to a manageable size. The process halts when no further merges are possible.

Once the plunger windows are selected, they are ranked by their highest separation score. This way, bounding boxes are created in the plunger voltage space, encompassing primarily stable bias triangles with current below the previously mentioned threshold (Fig. 4.3 b-*iii*).

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.

4.2.3 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.

Low-resolution wide shot classification

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. 4.3 c-*iii*).

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. 4.5).

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.* [46].

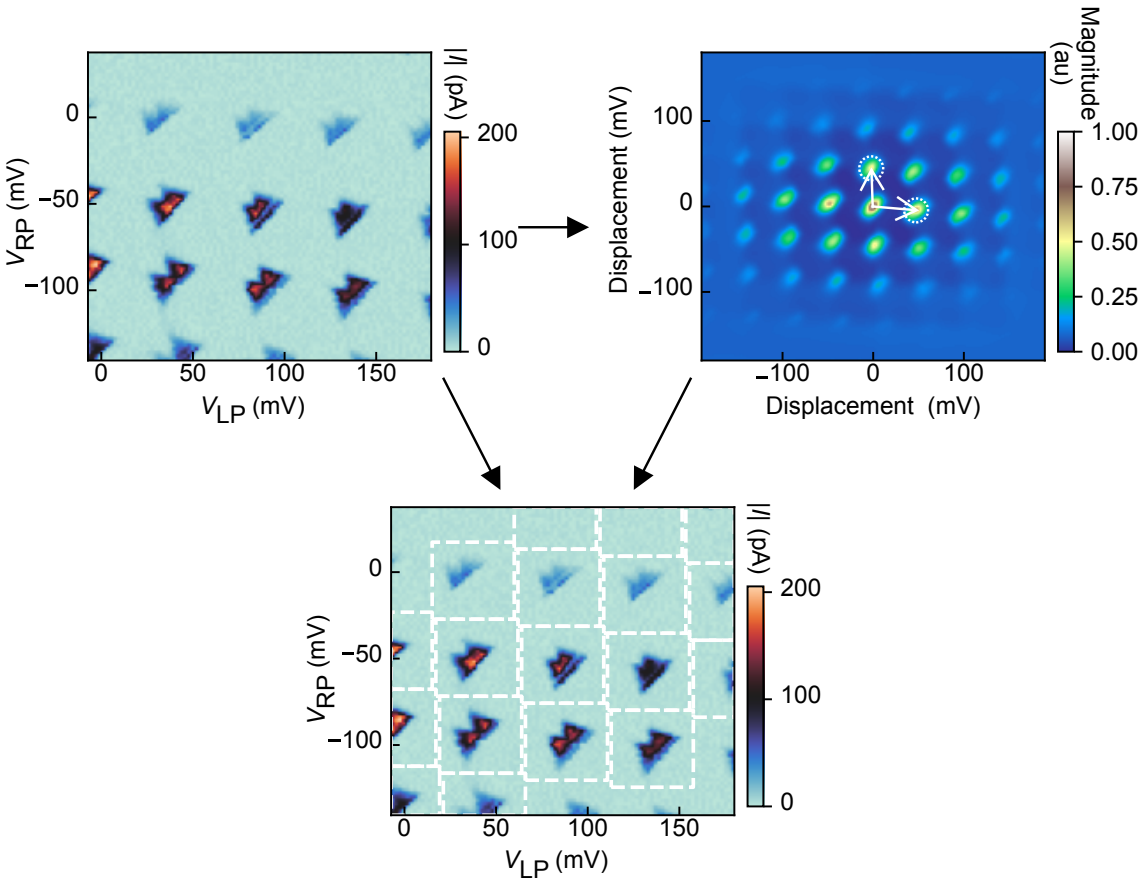


Figure 4.5: Bias triangle locations via auto-correlation. A stability diagram (top left) is processed using its auto-correlation (top right). Within the auto-correlation, we can find the highest values to span a grid of skewed rectangles. The extracted information is used to locate bias triangles in the original stability diagram (bottom).

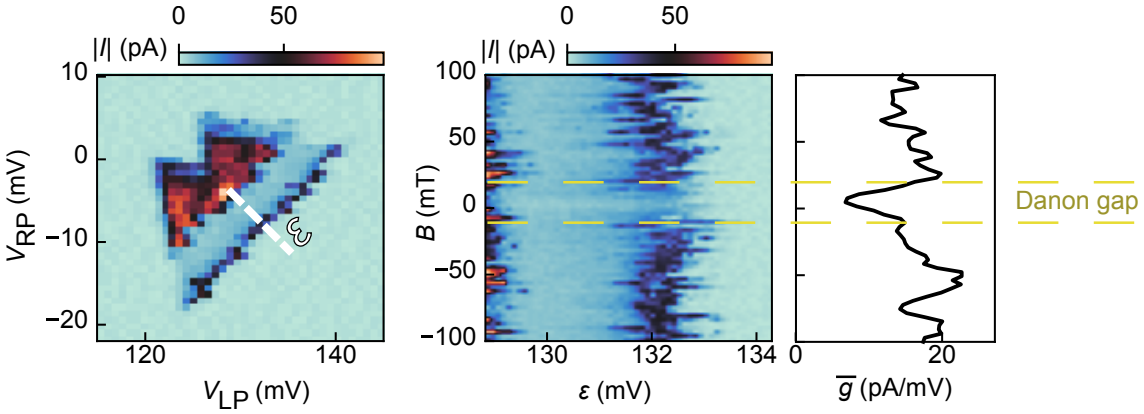


Figure 4.6: Danon gap check. As final check to confirm the presence of PSB, the algorithm takes measurements of the detuning line ε (marked in the left panel) as a function of the magnetic field B (middle). The measurement is analysed, as described in 4.2.3, by first computing the averaged sum of absolute derivatives $\bar{g}$ (right). If a dip is present in $\bar{g}$ close to $B = 0$ T, PSB is confirmed.

High-resolution classification

To filter previously detected candidates further and eliminate false positives, a second set of higher resolution measurements is performed. Initially, a dedicated low-resolution stability diagram of the candidate bias triangles is captured to update the plunger voltage extent based on detected contours, effectively re-centering the measurements. With this updated voltage extent, high-resolution stability diagrams are taken at $B = 0$ T and with $B = 0.1$.

In the subsequent sub-stage of PSB classification, a segmentation-based detection and feature extraction framework is applied to quantify geometric and physical properties of the bias triangles in a coordinate-wise manner [135]. Specifically, using the high-resolution stability diagram at $B = 0.1$ T, this framework fits minimum-edge polygons to the detected contours of bias triangle pairs by employing a relaxed extension of the Ramer-Douglas-Peucker (RDP) algorithm [135].

The RDP algorithm is a discrete curve approximation method that simplifies a curve by reducing the number of points based on a specified tolerance, effectively approximating the original shape with fewer line segments. In our context, we use a relaxed RDP (rRDP) algorithm to approximate the detected shapes of bias triangles with polygons that have the minimal necessary number of edges. This relaxation involves setting an initial approximation tolerance ε, defined as a fraction of the total perimeter of the detected shape, which is gradually increased until one of two stopping criteria is met: either the Intersection over Union (IoU) score between the original detection and the approximation falls below a certain threshold (e.g., 85%), or the number of edges falls below a predefined minimum (e.g., six edges).

We employ this relaxed version of the RDP algorithm to handle imperfections in the detected shapes, such as noise or disconnected components, which are common in experimental data. By fitting minimum-edge polygons to these shapes, we obtain a robust shape identifier that not only simplifies the contour but also preserves the essential geometric features of the bias triangles. This approach allows for the automatic extraction of further geometric properties, such as the base and excited

state lines, based on the known bias voltage sign, which predicts the direction in which the bias triangles point.

To identify PSB, an analytical classifier based on the aforementioned framework was devised [135]. This method involves extracting the segment that encloses the base and a prominent excited state line on the stability diagram with leakage current ($B = 0.1$ T). The average pixel intensity of this segment is then normalised by the intensity of the entire pair of bias triangles. This detected segment is subsequently superimposed on the scan with blocked current ($B = 0$ T), and the normalised intensity values are compared. If the difference between these values exceeds a specified threshold, the charge transition is classified as positive for PSB.

Danon gap check

As a further filter for possible candidates, we check the 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 [70]. By measuring the current along the detuning line while varying the magnetic field, we obtain a two-dimensional input (Fig. 4.6). The detuning lines are defined by identifying the bias triangles base midpoints and tips and computing the lines between them using the routine from Ref. [135]. We analyse the 2D plot as follows: Ignoring noise, the current remains roughly constant along the magnetic field axis. However, the detuning line axis has more features: away from the Danon gap, there are two local extrema—one side corresponding to the noise floor outside the bias triangles, and the other to the gap due to singlet-triplet energy splitting. In contrast, the Danon gap region is characterized by a monotonic behavior with a roughly constant value.

To detect the presence of the Danon gap, the current I is first processed with a Gaussian filter to smooth out the noise. Then, the absolute slopes along the detuning line axis are integrated as $g(B) := \sum_\varepsilon \left|\frac{\partial \tilde{I}(\varepsilon,B)}{\partial \varepsilon}\right|$, 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. 4.6.

To confirm the presence of the Danon gap from $\bar{g}$, two tunable hyperparameters are used to validate the depth and width of the basin of the global minimum of g. If the basin is not prominent enough, there is no Danon gap. The hyperparameters used for this are listed in Appendix B.4.2. Finally, the location of the minimum must be in proximity to zero magnetic field.

4.2.4 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.

Entropy optimisation

We now apply a pulse sequence to the right plunger electrode. As this 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 locate a suitable region for qubit readout. Considering the resulting shape distortions and further degrading effects on measurement quality, we opt for template matching instead of re-segmentation to ensure robust tracking of the bias triangles.

The relative direction in which the shadow bias triangles appear, with respect to the original ones, is predetermined by the applied pulse shape. This known cardinal direction is incorporated into the shape-matching approach to uniquely identify and track the triangles.

We perform shape matching by comparing the edge map of a stability diagram prior to pulsing, serving as the template, with the edge map of a subsequent stability diagram with pulsing, which provides current information. Additionally, we extract the segmented shape mask from the template. The method slides the template over the source edge map, comparing the template with individual patches of the stability

diagram with pulsing. This process generates a result matrix (of the same size as the source) whose individual entries quantify the similarity to the template patch. The employed similarity metric is the normalised correlation coefficient, and the patch with the maximum correlation is selected. Once the appropriate patch is identified, the initial segmentation mask from the stability diagram without pulsing is mapped onto the stability diagram with pulsing for subsequent processing.

To identify the optimal readout spot, we extract the segment enclosing the base and prominent excited state lines from the segmented mask of the bias triangles with pulsing. We then perform BO over four parameters: the constrained two-dimensional plunger gate voltage space, frequency of the driving pulse f_{MW} and burst time t_{burst}.

We sample a point within this four-dimensional space and then measure the current as a function of the magnetic field (Fig. 4.3 d-*ii*). The goal of the optimisation is to identify a current peak in these scans, indicative of the qubit's resonance condition (Fig. 4.3 d-*ii*, lower right plot). This is achieved by evaluating the entropy of current traces; traces exhibiting a peak correspond to lower entropy values.

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. Here, $l(B)$ is a one-dimensional trace obtained by taking the Euclidean norm of the in-phase and quadrature components of the lock-in amplified measurements. For the calculation of the entropy score, we first subtract the median and then clip values at zero. This preprocessing step transforms the trace into a distribution-like form and enhances the robustness of our score, making it less susceptible to noise disturbances. This method results in a smooth score landscape suitable for BO.

Based on all points investigated and ranked by their entropy score, we propose 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 [136]. Noisy measurements might also show peaks. Therefore, a follow-up step is to retake the measurement to confirm the presence of a peak.

Resonance confirmation

This verification step acts as a final filter and the last sub-stages are all executed once a candidate passes this filter. The previous stage sends a candidate with a suspected resonance condition. The verification stage re-measures the leakage current as a function of the magnetic field. If the resonance condition is indeed met, a peak should reappear at the same magnetic field. If a peak with a specified prominence is detected within a set margin of error, the resonance condition is 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.

Various qubit measurements

Once a candidate is verified, several measurements are taken by the algorithm to establish the qubit's operational functionality. These include a spectroscopy measurement, and a Rabi chevron experiment (Fig. 4.3 d-*iii*). We vary the magnetic field and the burst duration. The characteristic Rabi chevron is analysed by considering the frequency spectrum for each magnetic field. The frequency should reach a minimum at the magnetic field that meets the qubit's resonance condition, and the amplitude should be maximal at this point due to reduced decoherence for off-resonant driving. We can therefore simply look for the maximum amplitude in the Fourier transformed Rabi chevron (Fig. 4.7). This information allows us to perform the final step: repeated measurements of Rabi oscillations at the resonance condition.

An overview over all stages is shown in Table 4.1.

4.2.5 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

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 4.1: 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.

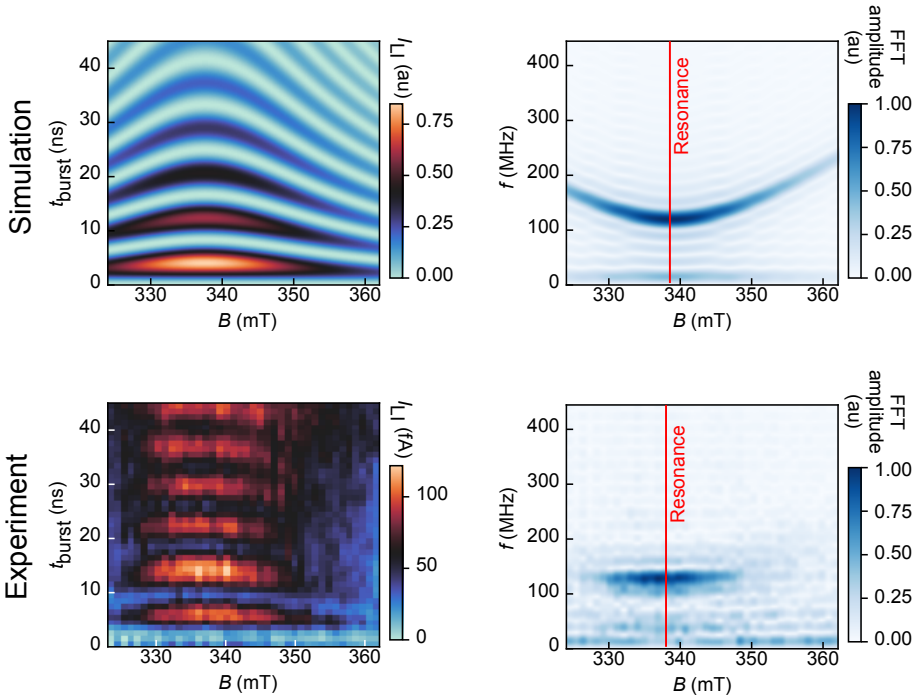


Figure 4.7: Rabi chevron analysis. A qubit that is driven close to the resonance condition exhibits so-called Rabi chevrons, see left column for examples (top: simulation, bottom: experimental data). To find the resonance frequency, we analyse the frequency spectrum for each magnetic field (which controls the detuning from resonance). The frequencies have a maximum amplitude and a minimal point at the resonance condition.

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 Appendix B.1.

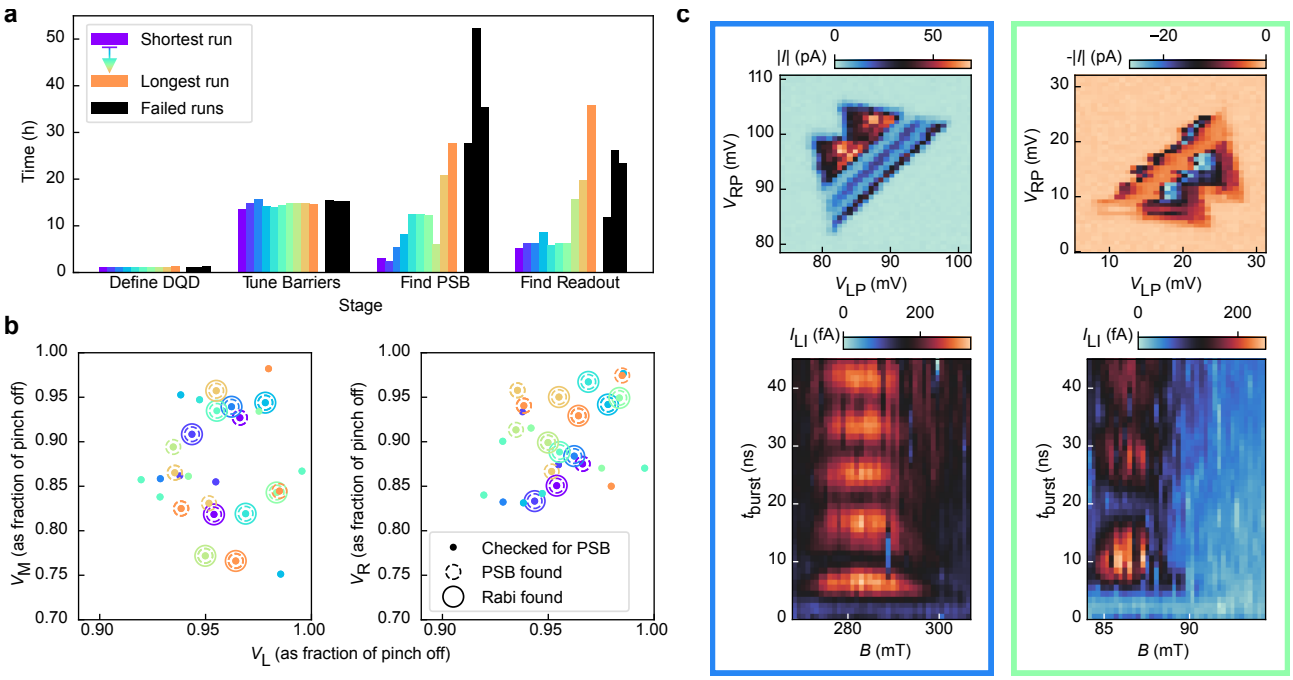


Figure 4.8: Benchmarking. a, The duration of each stage for ten successful runs is shown, sorted by the total time taken to tune a qubit in operation. In black: the three runs that did not lead to a qubit. **b,** Locations in the barrier voltage space (as a fraction of the individual pinch-off voltage) 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).

4.3 Results

We fixed the hyperparameters and gathered 13 runs. Rabi oscillations were found in ten of those runs. Success was manually judged by the existence of qubit signatures in all of the measurements in Stages 4 c, d, and e, i.e., Rabi oscillations, Rabi chevrons, and EDSR spectroscopy measurements. In successful runs, the time spent in each stage varied (Fig. 4.8a). total time required ranged from 22 to 80 hours, with a mean of 38 hours (median 34 hours) (Table 4.2). 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 4.2: 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. 4.8a. 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. 4.8b. 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. 4.8**c** 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 generalisation capability. All depicted measurements were autonomously executed by the algorithm, accounting for the non-centered chevron measurements.

We show measurements that confirm that we found a qubit in the ten successful runs, ordered by total run time, see Fig. 4.9. 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.

Occasionally, the algorithm attempted to take qubit measurements but was unsuccessful. This happened a few times during our experiments. For comparison to the successful qubit measurements in Fig. 4.9, we show these unsuccessful attempts in Fig. 4.10. Except for the bias triangles (upper left in each panel), the measurements display only noise.

Analysing the locations in gate voltage space where qubits were found provides insight into the device physics (Table 4.2). 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. Detailed calculations are in Appendix B.3.

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. The resulting maps gives insights into qubit properties and can be extended to measure, e.g. the Hahn-echo coherence time. These measurements will explored in Chapter 5.2.

Figure 4.9: 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.

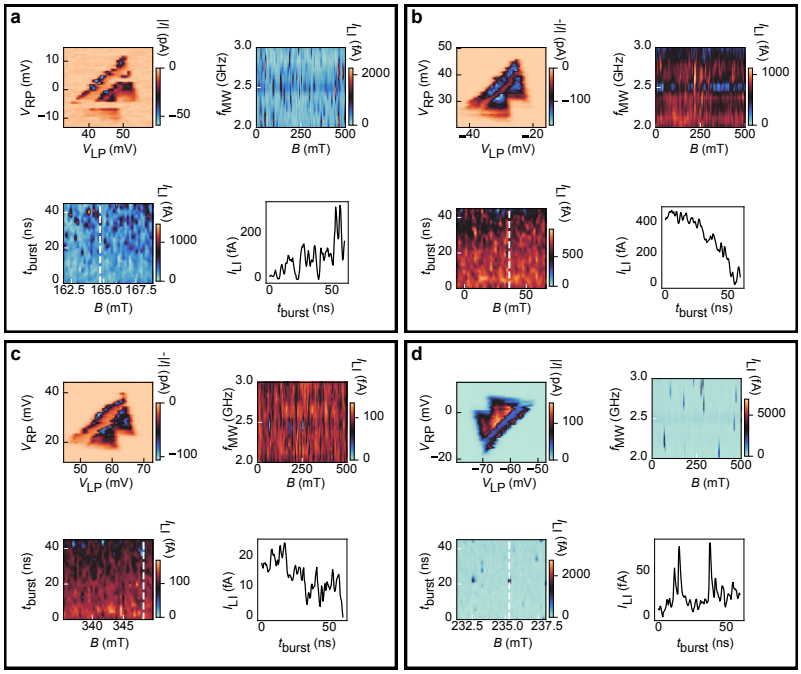


Figure 4.10: Attempted qubit measurements. For comparison with successful runs, these measurements are from runs before a qubit was actually found but that were ultimately successful. Each panel is organised the same way as in Fig. 4.9. The measurements show noise, except for the bias triangles (upper left in each panel).

In Fig. 4.1**b**, we show an illustrative example of a search tree. In Fig. 4.11 we show the search trees that were actually constructed for the longest and shortest runs in our experiments.

4.4 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.

The tuned qubits showed exceptional temporal stability. Once the settings to operate them were determined, they remained stable over extended periods, minimising the need for frequent recalibration. This stability simplifies the maintenance of qubit operations over time. In cases where drift might occur, the modular design

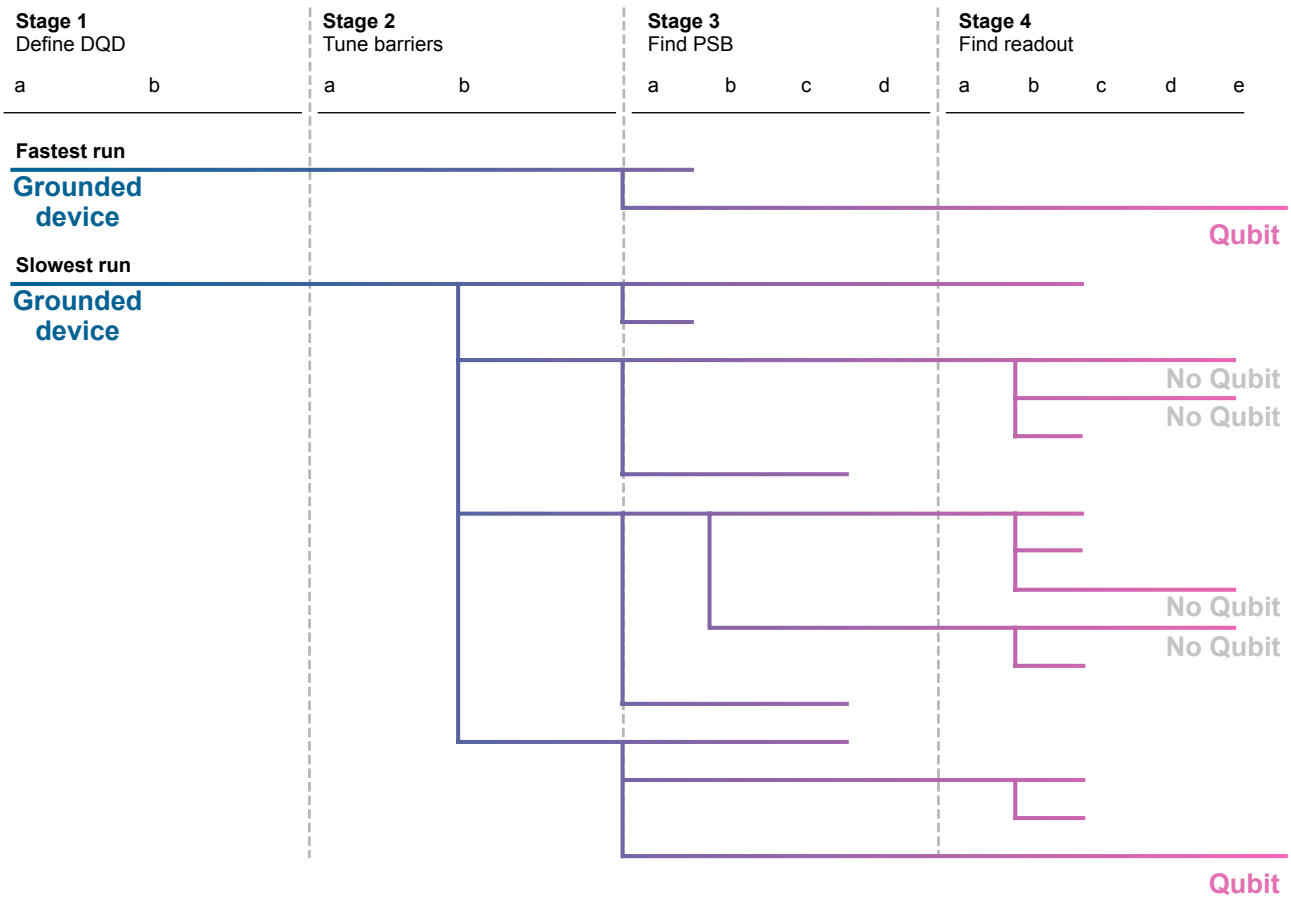


Figure 4.11: Examples of search trees from full runs. The fastest run only has two branches and then successfully found a qubit. The slowest run explored much more, with several branches reaching all the way to qubit measurements. However, only the last branch shows conclusive qubit signatures. We rejected the first tries as noise.

of our algorithm allows for the incorporation of additional modules to compensate for such changes, although this was not necessary in our experiments.

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 characterisation 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.

5
Qubit characterisation

Contents

In the previous chapter I demonstrate how to autonomously tune a spin qubit device from a grounded state to measuring Rabi oscillations. This chapter is about what to do next. I use the same device and measurement setup as in Chapter 4.4. All software development (apart from some parts borrowed from earlier chapters) and experiment execution in this chapter was done by myself.

The work in this chapter has been partially published along with the tuning algorithm from Chapter 4.4 to demonstrate the power of automation. Significant portions remain unpublished, so the narrative is not as refined. Especially the second part of this chapter is somewhat of a side quest. The central message is that automation allows for data collection and statistical analysis that, while theoretically possible to perform manually (just like the tuning in Chapter 4.4), are typically too tedious and time-consuming to undertake without automation.

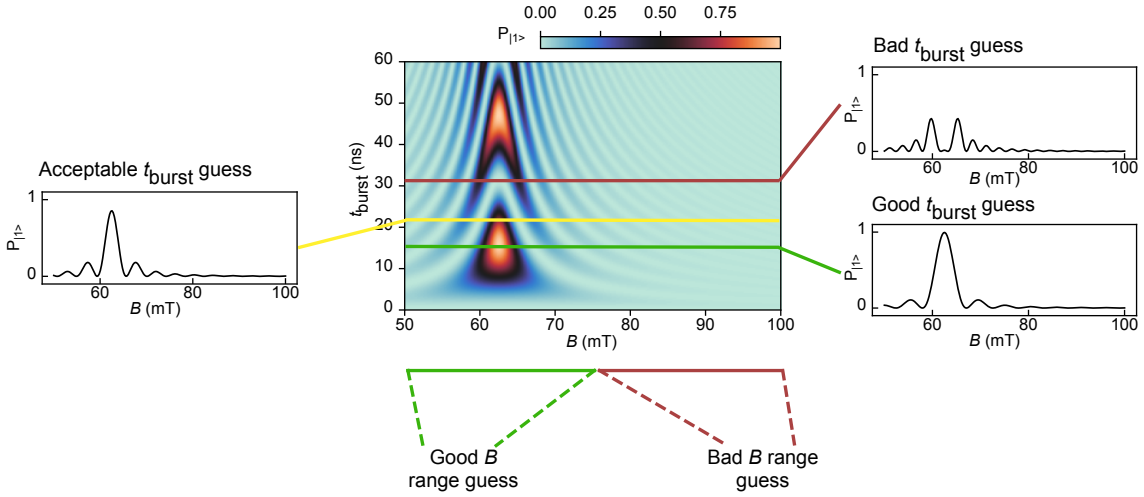


Figure 5.1: Informative line cuts of a Rabi chevron measurement. A good B range is one that allows the resonance to be found, which means it needs to start at a lower magnetic field and end at a higher magnetic field compared to where the resonance is. A good guess for the t_{burst} time is one that is close to a π pulse. However, the guess for t_{burst} can be slightly off and still be acceptable.

Initially, the work in this chapter aimed to inform the development of the tuning pipeline to determine the extent in gate voltage space within which qubit readout is possible. This question ultimately found an answer with the data from Chapter 4.4. During these investigations, it became evident that the scope of the study could be expanded to include two additional aspects.

First, I ventured into the construction of qubit maps—diagrams representing how qubit metrics, e.g., the Rabi frequency, change with variations in the three barrier voltages (to the source, drain, and between the dots). This is discussed in Section 5.1.

Secondly, by gathering sufficient data points from regions exhibiting both successful and unsuccessful readout, I explored a hypothesis suggested by our collaborators in Basel. This hypothesis proposes that by analysing and adjusting the currents within the bias triangle, one can find appropriate barrier voltages that allow qubit readout. This could be useful as it can inform tuning algorithms to find qubit readout faster. I present that aspect of the work in Section 5.2.

I will now discuss what needs to be done to find qubit readout for a given set of barrier voltages. I need to find four quantities:

- The two voltages of the plunger gates which sets the initialisation/read-out point.
- The g-factor.
- The Rabi frequency f_{Rabi}.

The initialisation/readout point can be found based on the computer vision techniques introduced in Chapter 4.4. As always, the devil is in the details, though, so I explain the exact mechanism I choose below.

To find f_{Rabi} and g, I designed a dedicated stage in Chapter 4.4 (Stage 4). However, that stage is very costly in time. We can exploit prior information of f_{Rabi} and g by taking a Rabi chevron scan (Figure 5.1) where its range is informed by guesses of f_{Rabi} and g. This is still quite costly. Faster yet is to instead take a guess of both f_{Rabi} and scan over a range of B that is informed by a guess of g via $f_{\text{Larmor}} = g\mu_{\text{B}}B/h$. I only need a decent guess of f_{Rabi} to find the resonance, as illustrated in Figure 5.1. Similarly, the range of B only needs to be a decent guess. This reduces the measurement overhead drastically. The measurement scheme that I describe next is centered around this idea.

5.1 Mapping qubit characteristics

Mapping the characteristics of qubits involves two primary steps: Initially, we must select the barrier voltage settings to investigate. Subsequently, we must measure g and f_{Rabi} via an appropriate set of measurements and analyses.

5.1.1 Sampling strategy

The selection of points within the barrier voltage space can be approached in several ways. Any method that samples the space is possible. I prefer to employ quasi-random sampling, such as using a Sobol sequence, because this ensures that the points are distributed approximately evenly across the sample space. A Sobol sequence is a type of low-discrepancy sequence used in quasi-Monte Carlo methods for numerical integration and sampling in high-dimensional spaces. It generates a

sequence of points that uniformly fills the multidimensional space, minimising gaps and clusters that are typical in purely random sampling. By using a Sobol sequence, we achieve a more uniform and deterministic coverage of the barrier voltage space, enhancing the efficiency of the sampling process and reducing the number of points needed to effectively explore the space. A sampling method gives a point cloud of locations in barrier voltage space that need to be investigated. Each point can carry information such as the location of the charge transition in plunger voltage space, the g-factor, f_{Rabi}, or whether readout was lost. I call this collection of data the "map knowledge". At the start, no point has this information except for a starting candidate. The starting candidate needs to have a known location in plunger voltage space, a known g-factor and f_{Rabi} and can either be manually supplied or by the tuning algorithm from Chapter 4.4.

Further, we order the investigation of a given point cloud by distance from a starting candidate (Fig. 5.2**a**). A barrier voltage point that is to be investigated will use information from the closest (in barrier gate voltage space) point to enable the estimation of the magnetic field (via the g-factor) and t_{burst} (via f_{Rabi}). A new result is added back to the map knowledge after investigation.

5.1.2 Iterative measurement pipeline

I designed a pipeline to acquire the qubit g-factor and Rabi frequency f_{Rabi} for a given barrier voltage setting, as illustrated in Fig. 5.2**b-h**.

Determining plunger gate voltages

First, the plunger voltage location of the readout and initialisation point needs to be determined. I obtain an estimate of the transition location in plunger voltage space via virtual gates (see Appendix C.1 for an explanation of virtual gates). I measure a coarse stability diagram to identify a bounding box around the bias triangles (Fig. 5.2**b**). Using these coordinates, a higher resolution stability diagram is measured (Fig. 5.2**c**). This detailed diagram helps define the detuning line of the bias triangles [135], which guides the subsequent measurement of the detuning

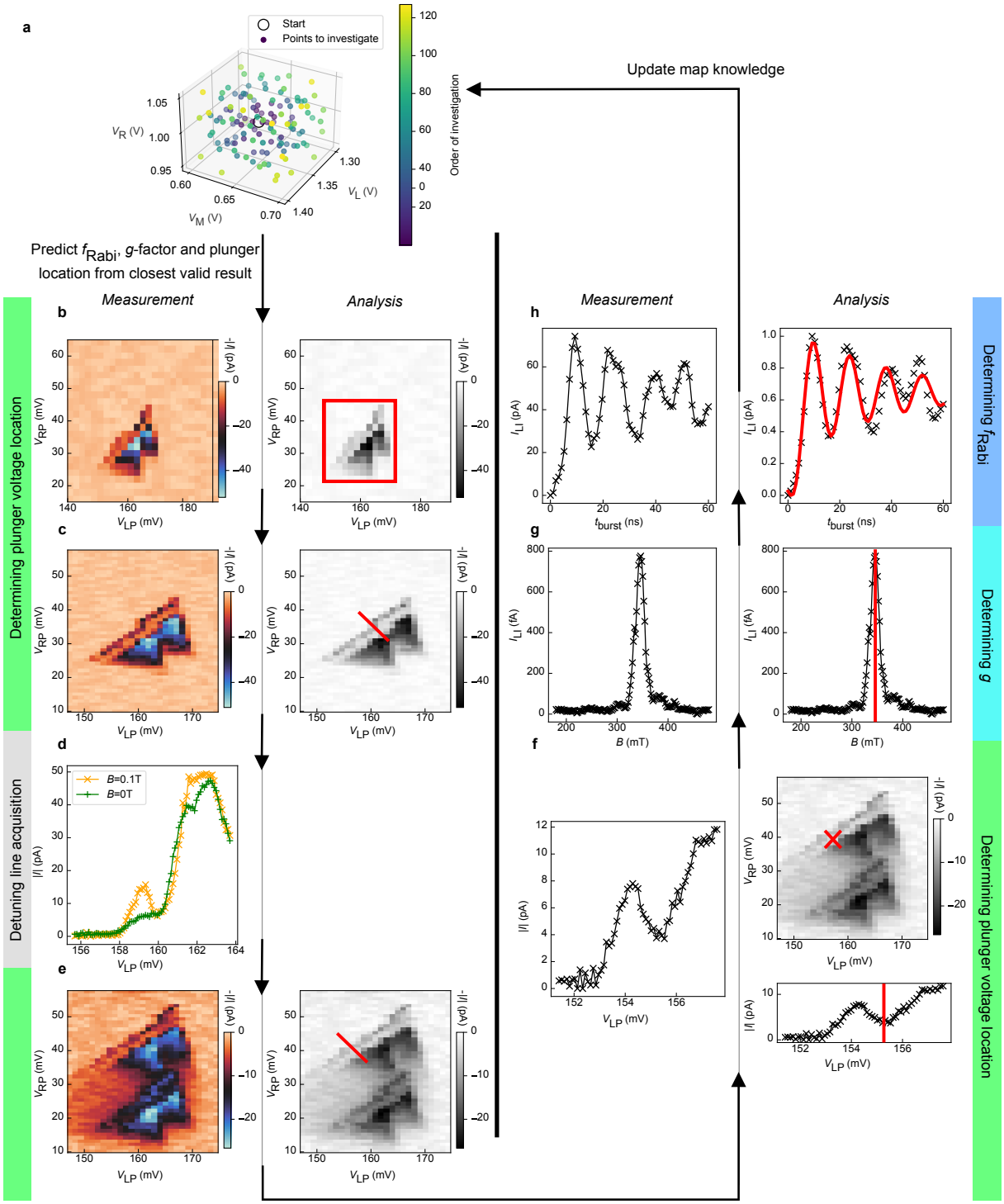


Figure 5.2: Pipeline to investigate the barrier voltage dependence of a qubit. **a** Barrier voltage locations and the sequence of investigation are chosen, e.g., by a Sobol sequence. **b** A coarse grained stability diagram is obtained, serving as a centering mechanism. **c** A high-resolution measurement defines the detuning line. **d** The detuning line is measured explicitly, both with and without an applied magnetic field. **e** A stability diagram taken with a pulse applied at the same time defines the detuning line under the influence of a pulse. **f** A measurement of the pulsed detuning line determines an optimal readout spot by identifying the valley between the baseline and the first excited state line. **g** The resonance condition in the magnetic field is identified. **h** Repeated Rabi oscillations are measured, fitted, and the results are fed back into the map knowledge to update estimates of g and f_{Rabi}.

lines with and without an applied magnetic field (Fig. 5.2**d**). This data is collected for later analysis, see Section 5.2.

Next, a stability diagram with an applied square pulse, the "pulsed stability diagram" from above, is obtained (Fig. 5.2**e**). Combining this with the detuning lines determined from Fig. 5.2**c**, the detuning line of the pulsed bias triangles is defined and measured (Fig. 5.2**f**). This measurement is analysed to find the point of lowest current between the baseline and the first excited state line. From experience with measuring these devices, this point in the current valley is expected to provide optimal readout conditions. This is because at this point the current due to unwanted leakage current is lowest and therefore the signal-to-noise ratio is best.

Determining g

Then, the g-factor needs to be determined. The suspected π-pulse burst duration $t_{\mathrm{burst},\pi}$, which should rotate the spin to lift PSB and lead to a current, is applied as described above. I sweep over a magnetic field range informed by the estimated g-factor (Fig. 5.2**g**). One could also scan over the driving frequency f_{MW} which can be done faster than changing the magnetic field, however the factor limiting speed is the averaging time we need to see a stable signal. Additionally, there can be resonances in the experimental setup which could be confused for qubit resonances. We avoid those by fixing f_{MW}. A peak finding algorithm [136] (see also Stage 4 of Chapter 4.4) identifies the peak and corresponding magnetic field.

Determining $f_{\mathbf{Rabi}}$

This allows for the final step, the determination of f_{Rabi}. At the magnetic field found in the previous step, repeated scans over the burst duration t_{burst} are conducted, and the current is recorded (Fig. 5.2**h**). The resulting measurement is fitted with a sinusoidal model to extract f_{Rabi}. The detailed model used for fitting the Rabi oscillations is provided in Appendix C.2.

This process repeats, iteratively updating the map knowledge with information from new points. A point is considered to have lost readout capability if either no EDSR peak is detectable during the magnetic field sweep or if the sinusoidal fit

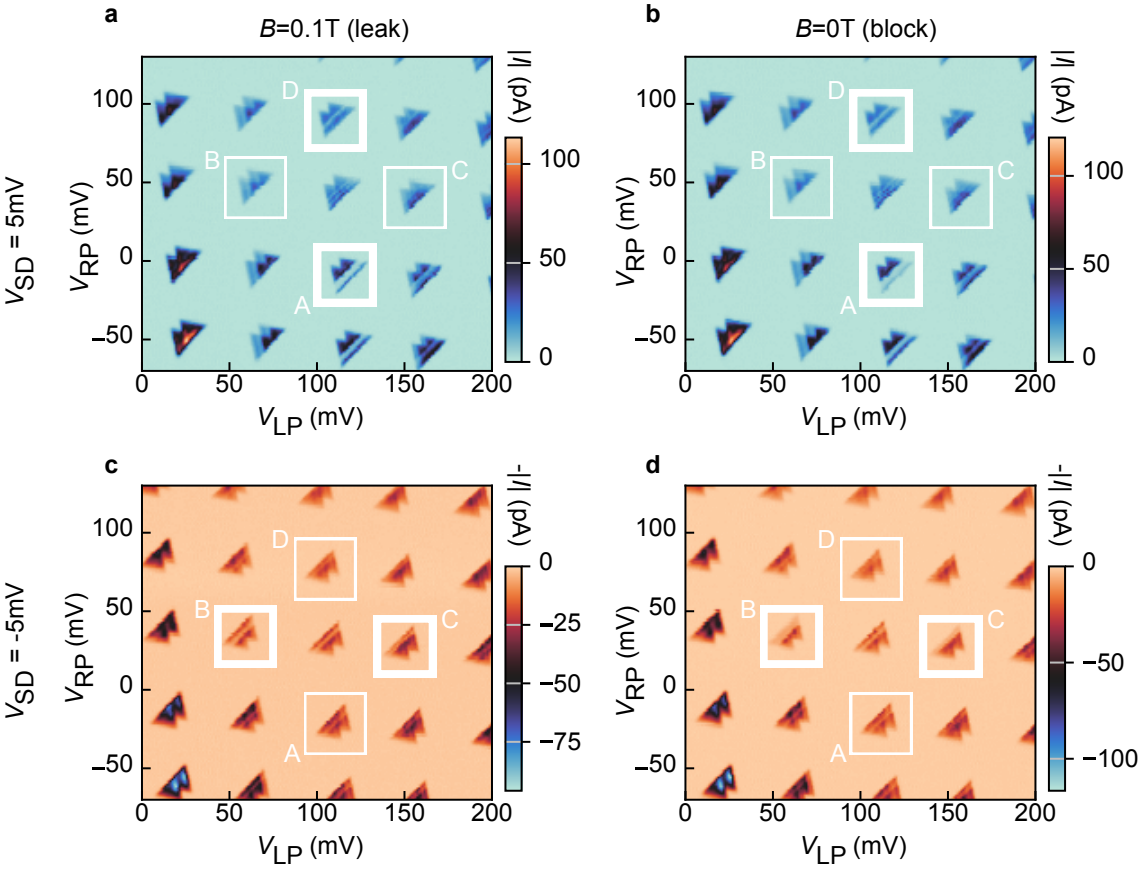


Figure 5.3: Charge transitions investigated. There are four charge transitions that we consider in the following, **A**-**D**. There are four measurements of each transition, with positive (**a, b**) and negative bias (**c, d**) voltage, and without a magnetic field (**b, d**) and with a magnetic field at $B = 0.1$T (**a, c**), which should lift PSB. Transitions **A** and **D** show PSB with a positive bias voltage. Transitions **B** and **C** show PSB with a negative bias voltage.

of the Rabi oscillations fails. This conservative criterion is chosen to avoid being misled by incorrect inferences that could arise from experimental noise or artifacts. By requiring both the detection of the EDSR peak and a successful fit of the Rabi oscillations, we minimise the risk of falsely identifying non-operational qubits as working ones. Chapter 6.4 describes a more refined filtering mechanism. For this work, I manually filter the results before analysing them further, as described below. The algorithm continues until it runs out of points to investigate.

5.1.3 Results

I study four different charge transitions, labeled **A** through **D**, as shown in Fig. 5.3. Transitions **A** and **D** show PSB in positive bias direction, while transitions **B** and **C** display it in the negative bias direction. This pattern in directionality of PSB is expected due to the required number of holes involved in the process, see Section 2.1.3.

The data collection spanned multiple acquisition runs, utilising a variety of suggestion mechanisms for the barrier voltages to optimise spatial sampling. The predominant method was Sobol sampling, which facilitated an even distribution of data points across the barrier voltage space. Another method was to intersperse measurements between known points where readout was lost and successfully detected. This aimed to find the extent of the barrier voltage space within which qubit operation was possible.

Post-experimental data processing involved manually labeling data to identify and correct for various anomalies such as charge switches, incorrect readout spots due to, e.g., failed segmentation, falsely identified peaks, or false fits. If an EDSR peak was present, the g-factor was determined. If, in addition, the fit of the Rabi oscillations was satisfactory, I accepted f_{Rabi}. The curated dataset, containing only points where the g-factor and/or f_{Rabi} could be conclusively determined, is visualised in Fig. 5.4. Charge transition **B** exhibited two distinct resonances, warranting two g-factor maps. The second qubit in this transition did not yield a measurable f_{Rabi} due to a bad signal-to-noise ratio, a limitation that could be overcome by integrating an additional calibration step into the measurement pipeline, as discussed in Chapter 6.4. For enhanced clarity and analysis, I fit each dataset with a Gaussian process, and the mean of each fit is depicted at a common voltage V_{R}=0.99V.

In the case of transition **D**, many points are aligned linearly as these measurements were initially intended to support hypotheses on the coherence and Rabi speed properties of these qubits, presented in Ref. [81]; however, this data was ultimately not included in that publication.

These maps give insights into the influence of the confinement potential on the g-factors and f_{Rabi}. Transitions **A**, **C**, and **D** share the characteristic that there

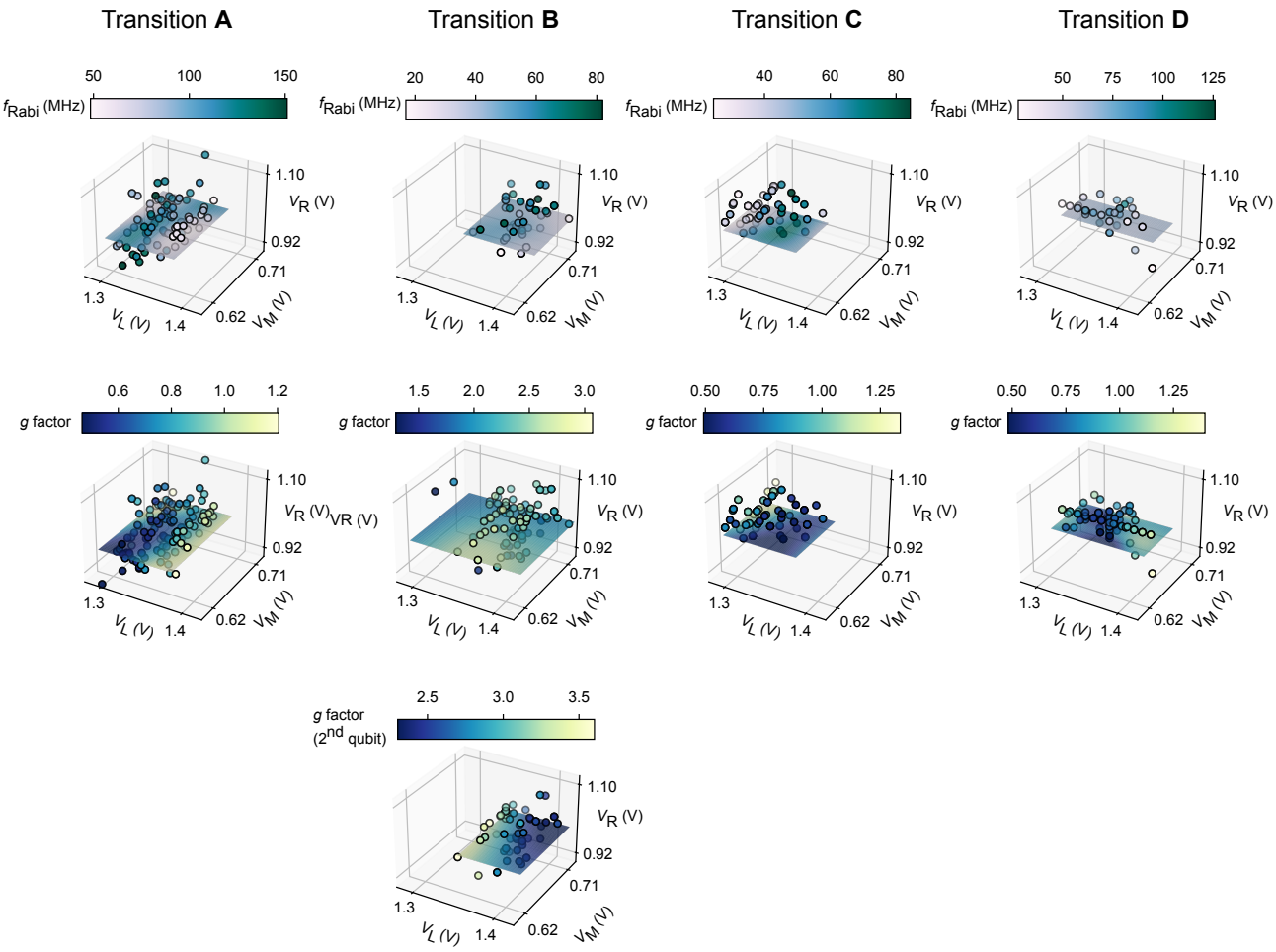


Figure 5.4: Maps of g-factor and f_{Rabi} for different charge transitions. The different transitions introduced in the beginning of the chapter are measured for their g-factor and f_{Rabi}. Transition **B** shows two qubits, explaining the second map of g-factors. We fit GPs to each map and show the GP mean for a slice at constant V_{R}=0.99V for the data range of each map.

seems to be an optimal operation point in terms of maximal f_{Rabi} which corresponds to a minimal g-factor [81]. Transition **B** shows a different dependence. For this charge transition, we see two resonances which can be attributed to two different qubits, one might be associated with a spin on the left dot, the other with a spin on the right dot [137]. The missing f_{Rabi} map of the the second qubit of that transition might show a similar behaviour as the qubits in the other transitions but this is subject to further studies. Further discussions of the mechanisms can be found in the publications by Carballido *et al.* [81, 137].

The data presented here builds the foundation to study spin qubits further. For

example, one could build theory models that connect the barrier voltages to qubit properties and test the predictions using this data. Another avenue for research could be to see if we could learn the dependence from one transition and map it to another transition. If this is possible, it could further improve the operation of these types of qubit by providing a shortcut to finding optimal operating points. The work can be employed to test the reproducibility and accelerate the development of spin qubits.

5.2 Study of charge transport signatures

Even if a given charge transition exhibits PSB, achieving a detectable qubit signal from it is not always guaranteed. There are several fundamental error sources to which this could be attributed, such as incorrect selection of the readout spot, inadequate scanning of the appropriate g-factor range, or microwave pulses with a miscalibrated t_{burst}. Other detrimental processes can affect the efficiency of the manipulation and readout process. The tunnel barriers to the source or drain might be too high, which can prevent timely charge loading or emission. Additionally, insufficient isolation from the non-blockaded triplet state (0, 2) can lead to unwanted interactions. Further, if the tunnel barriers to the source and drain are too low, spin-flip co-tunneling can reduce the efficiency of the coherent manipulation cycle, leading to the qubit signal being drowned in noise.

Figure 5.5 illustrates these concepts by presenting data from the same charge transition under three different barrier voltage settings. The stability diagrams, shown in panels **a**, **b**, and **c** include a white cross to mark the algorithmically chosen readout points. These diagrams show how different barrier settings influence the currents within the bias triangles. Panels **e**, **f**, and **g** display current measurements with varying magnetic fields, demonstrating that effective readout is achievable with the configurations of the first two columns (because one can determine a peak in current) but not with the third. Importantly, a robust signal-to-noise ratio is crucial for clearly observing Rabi oscillations. This necessity is evidenced in the last row of the figure, where only the Rabi frequency in panel **h** is determinable, whereas it is inconclusive for panels **i** and **j**. An explanation is that to determine

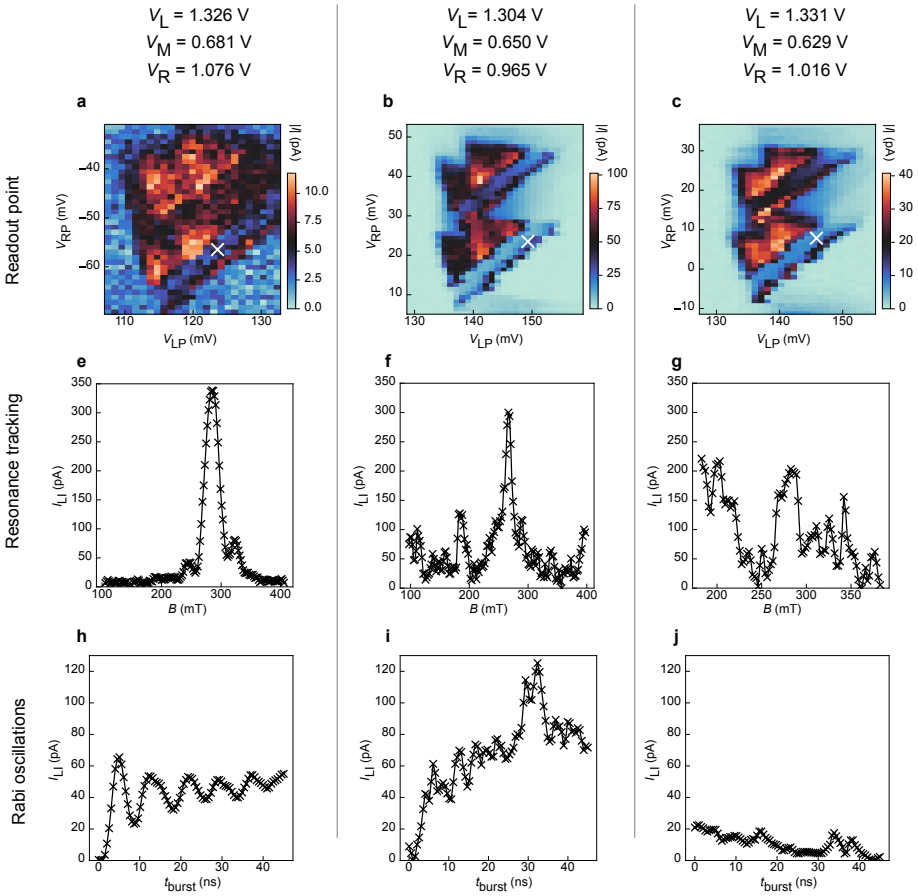


Figure 5.5: Demonstration of losing readout. These are measurements from the same charge transition at differing values for the barrier voltages. The first column shows the charge transition in a state where readout and coherent driving of a qubit is possible. The middle column shows a state in which the readout mechanism is almost lost and Rabi oscillations are no longer observed. The last column shows the complete loss of all readout. **a-c** show the bias triangles with an applied pulse and the readout spot marked with a white cross. **e-f** show the sweep over the magnetic field with a fixed driving frequency to identify the resonance condition and the g-factor. **h-j** show Rabi oscillation measurements, or the attempts thereof.

the g-factor we need only about half a coherent oscillation, whereas to determine the Rabi frequency we need at least a full coherent oscillation. In panel **i**, increasing the number of averages might reveal the Rabi oscillations. This is to show that we might be able to determine the g-factor but not f_{Rabi}.

When I started working on this experiment, I was told a hypothesis that proposes that the current at the readout point—i.e., within the region affected by PSB and away from the baseline—must be low, yet not so diminished as to fall below detection thresholds. This current does not necessarily vary linearly, or predictably, with

changes in the barrier voltages. This is due to, e.g., changing disorder potential (Appendix C.3). Gathering the current along the detuning lines (Fig. 5.2**d**) opens the possibility to check currents or current ratios in the bias triangles for a relationship with the ability to find readout.

My curiosity was piqued since this hypothesis was presented to me without any evidence. I realised that given the tools I developed, I could gather the data needed to see if the hypothesis is true.

The data presented in Fig. 5.4 exclusively represents instances where qubit readout was successful. Complementary data where readout failed is equally critical as it provides insight into the conditions under which qubit operations fail. Measurements of the detuning current traces, both with and without an applied magnetic field, enable comparison of currents with lost readout and found readout. For this analysis, I employed a lenient criterion to define successful readout: the presence of a discernible EDSR peak during the magnetic field sweep. Figure 5.6 displays these detuning current traces for transitions **A-C**, annotated with mean and median values.

Transition **D** is excluded from this analysis due to the lack of a clear separation of the base line from the first excited state in the experimental data, complicating any meaningful comparison of currents for this transition. This issue underscores the challenges and nuances of measuring and interpreting charge transport signatures.

The current within the valley of the detuning line current, marked by a green star, is particularly interesting. Comparing the current at this point for the cases of lost and found readout indicates that lower currents generally correlate with successful readout.

Currents in the valley for each condition are extracted and visualised in the first row of Fig. 5.7. To preserve the visual interpretability of these plots, I clip the current in valley plots at 50 pA as currents for transition **B** can far exceed this value for some cases. Notably, currents tend to be lower in the valley when readout is successfully achieved, although overlaps in the data distributions suggest that these measurements alone are not definitive indicators of readout success.

I wondered if other metrics might be more informative. Particularly in the context of tuning, just aiming for lower currents might be unwise because that would lead

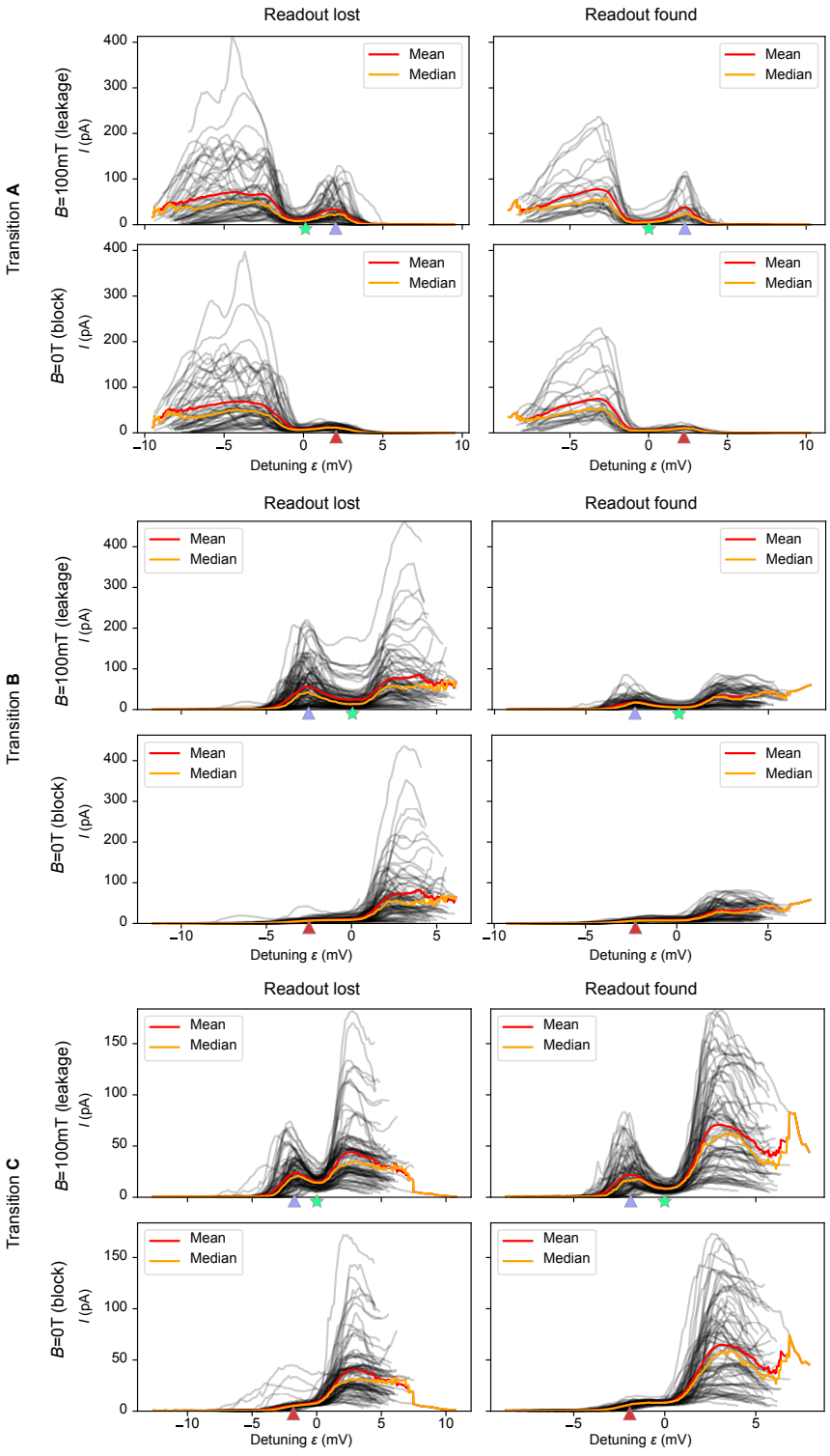


Figure 5.6: Detuning line measurements. For transitions **A-C**, I show measurements of current along the detuning line for varying barrier voltages for four different conditions: with and without a magnetic field, and with readout found and readout lost, as determined by manual labels. The traces are centered around the identified valley point, which is put to be at 0 detuning. This is unconventional because the energy levels align at the base line which usually would be identified with 0 detuning. This choice allows for improved visual clarity. The green star and purple and red triangles indicate the positions at which the current is taken for further analysis, see Figure 5.7.

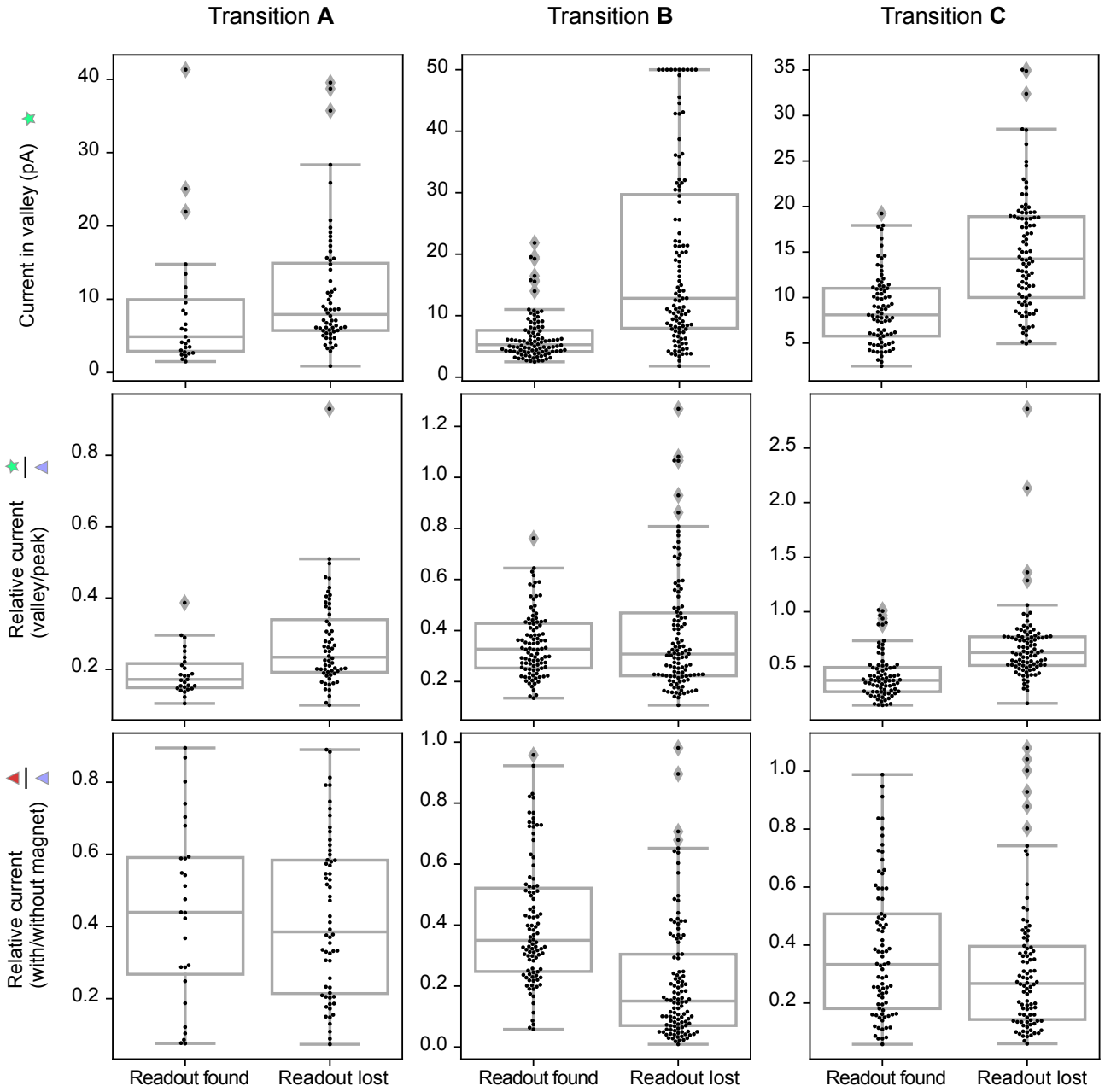


Figure 5.7: Comparisons of the currents on the detuning line. The underlying data from Figure 5.6 is analysed further by only considering certain currents or current ratios. We show the individual data as points and the statistics as underlying box plots. The three metrics under consideration are the current in the valley (top row), the ratio of currents in the valley, normalised by the current at the base line at high magnetic field (second row), and the ratio of currents of the base line with and without magnetic field (third row). The data for the valley current is clipped at 50 pA for visual clarity. For transition **A**, there are 60 data points with lost readout, 27 with found readout, for transition **B**, there are 94 with lost readout, 81 with found readout, for transition **C**, there are 112 with lost readout, 98 with found readout.

us into a fully pinched regime. A first idea would be to normalise the current in the valley by dividing it by the current at the baseline, marked by a purple triangle in Fig. 5.6. The resulting ratio is shown in the second row of Fig. 5.7. We see a similar effect, as when just considering the current in the valley, although especially for transition **B** the predictive insight seems weakened. Still, this result provides justification for the choice of the score function made in Stage 2 in Chapter 4.4.

Another metric under consideration is the ratio of baseline currents in the absence of a magnetic field, denoted by a red triangle, versus presence of a magnetic field. This comparison aims to quantify the magnetic field's effect on leakage currents, offering some proxy measure of the strength of PSB. However, the resulting current ratio, shown in the last row of Fig. 5.7, does not show a significant difference between the distributions. Reflecting on this, that might not be unexpected. This current ratio expresses how strongly the magnetic field lifts the leakage current. This property might therefore not affect the efficiency of the manipulation and readout cycle and, consequently, no clear relationship with the ability to coherently manipulate and readout a qubit is found.

Despite the insights gained, the signals derived from these measurements are not perfect predictors of readout success. Variabilities such as inaccurately chosen g-factors, suboptimal pulse durations, or misaligned readout points can significantly affect outcomes. Still, the idea behind these current signatures are built into Stage 2 in Chapter 4.4. To understand how much the analysis of these current signatures are helping in tuning, one needs to perform of an experiment that benchmarks them in a tuning context.

This chapter introduced a systematic approach to measure the g-factor and f_{Rabi} as a function of the three barrier gate voltages. I showed that this works qubits hosted in four different charge transitions. By analysing current signatures and the impacts of varying confinement potentials, we have gained insights into the factors influencing qubit performance. The data justifies the tuning method developed in Stage 2 in Chapter 4.4.

6

Optimisation of Rabi frequency

Contents

In this chapter, I present the application of the modules I developed previously within a Bayesian optimisation (BO) framework to enhance the Rabi frequency f_{Rabi} of a specified qubit. I show an autonomous algorithm that finds the optimal 3D barrier voltage settings for a given qubit.

The Rabi frequency is an interesting metric to maximise because it sets a clock rate for one of the fundamental operations of a quantum computer. Faster Rabi oscillations translate to quicker quantum computations. There is a catch though: increased Rabi oscillation speeds often lead to reduced decoherence times [80, 134, 138]. This correlation arises because a higher Rabi frequency requires stronger coupling between the qubit and external electrical signals. While this stronger coupling enhances our ability to control the qubit, it also makes the qubit more susceptible to electrical noise, leading to faster decoherence. To address this, we could instead optimise the quality factor, which is the product of Rabi frequency

and decoherence time and measures how many oscillations can be achieved before a qubit decoheres. As a proof of principle, I optimise f_{Rabi} as this is easier than, e.g., the quality factor, because optimising f_{Rabi} means I only need to infer the speed of the oscillations and not also their decay, which would need longer measurements. The framework is sufficiently general to optimise other parameters. I use the same device and measurement setup as in Chapters 4.4 and 5.2.

The optimisation process requires two main components: the sampling mechanisms for proposing new barrier voltages, and the evaluation pipeline that investigates a barrier voltage and can determine a figure of merit, in this case the Rabi frequency.

6.1 The sampling logic

The sampling mechanism is illustrated in Fig. 6.1. Using an initial barrier voltage location as a center, for example from Chapter 4.4, we can define bounds and sample quasi-random points within those bounds. In a standard BO framework, we would start with these seed points to build an initial understanding of the optimisation landscape. I also want to do this here but I want to control the order of investigation such that I can always use the information from a nearby point that has already been investigated to avoid a costly search for g and f_{Rabi}, see Chapter 5.2 for the idea behind this mechanism. Therefore, the initial seed locations are ordered by their distance from the first barrier voltage location and investigated accordingly. As in Chapter 5.2 I call the collection of information of barrier gate voltages, plunger gate voltages, g, and f_{Rabi} the "map knowledge".

Once the initial seeds are investigated, we commence the standard BO loop. This loop is subject to predefined boundaries; however, since it is unknown if the optimal solution lies inside these boundaries, we adjust the boundaries after ten new samples by determining the voltage location with highest f_{Rabi} and redefining the boundaries centered on this point. To enhance the noise resistance of this process, the best location is determined by disregarding the top 5% of the best-performing samples and calculating the mean location of the subsequent 3 best samples ranked by their associated f_{Rabi}. I made this choice to avoid the optimisation being tricked by wrong

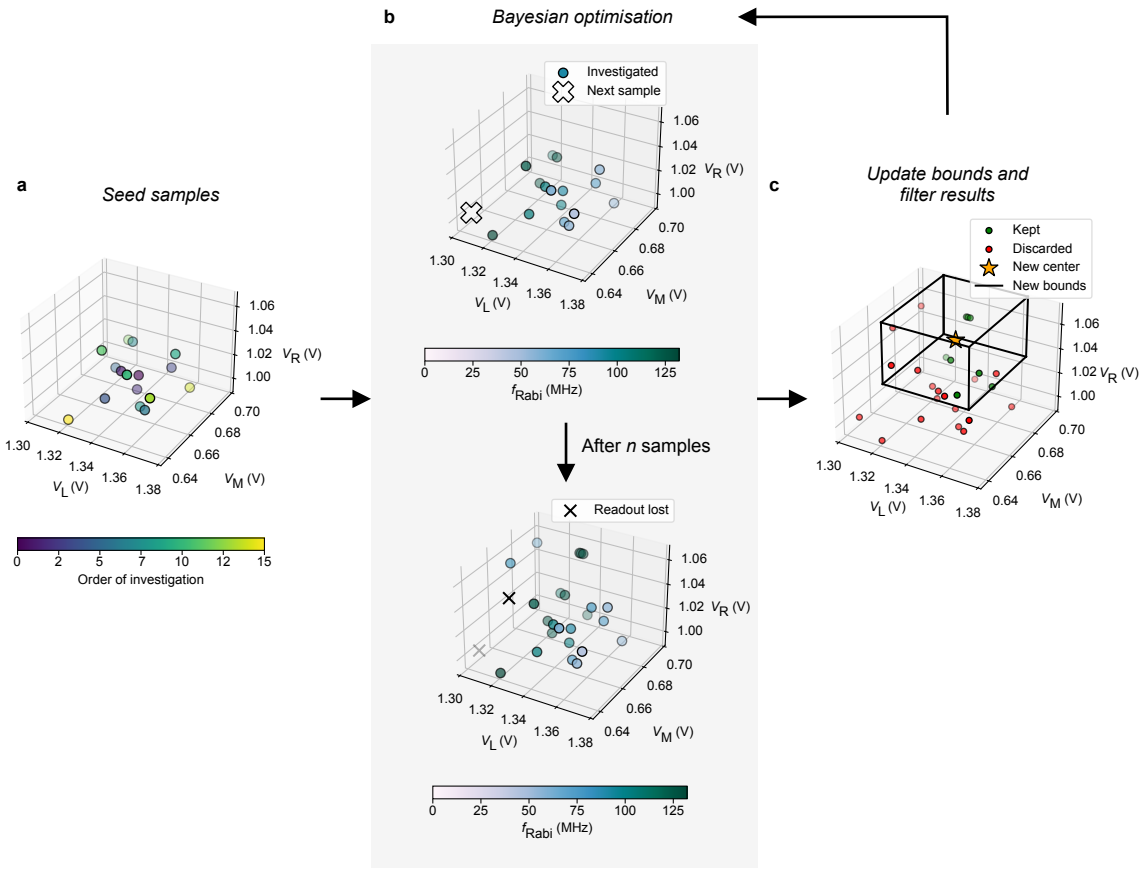


Figure 6.1: Sampling logic. a First, the seeding samples are ordered by their distance from a center to allow for a narrow search space for magnetic field resonance condition (and therefore the g-factor) and f_{Rabi}. **b** Then, a standard BO scheme is used. **c** Since it is unknown *a priori* whether the global optimum will be within bounds, we calculate the voltages of the point with highest f_{Rabi} and update the bounds accordingly as an intermediate step.

inferences of f_{Rabi} as the signal can be very noisy. This method presumes that this shifting of the boundaries will eventually enclose the global optimum. This means that the global optimum needs to have possible local optima close to it. I made this assumption after studying the qubit maps in Fig. 5.4. If the assumption is not true, the algorithm could end up in a local optimum, in which case usual strategies to avoid local optima apply, e.g., multiple re-starts of the optimisation. I use the `scikit-optimize` [139] implementation of BO to perform the optimisation.

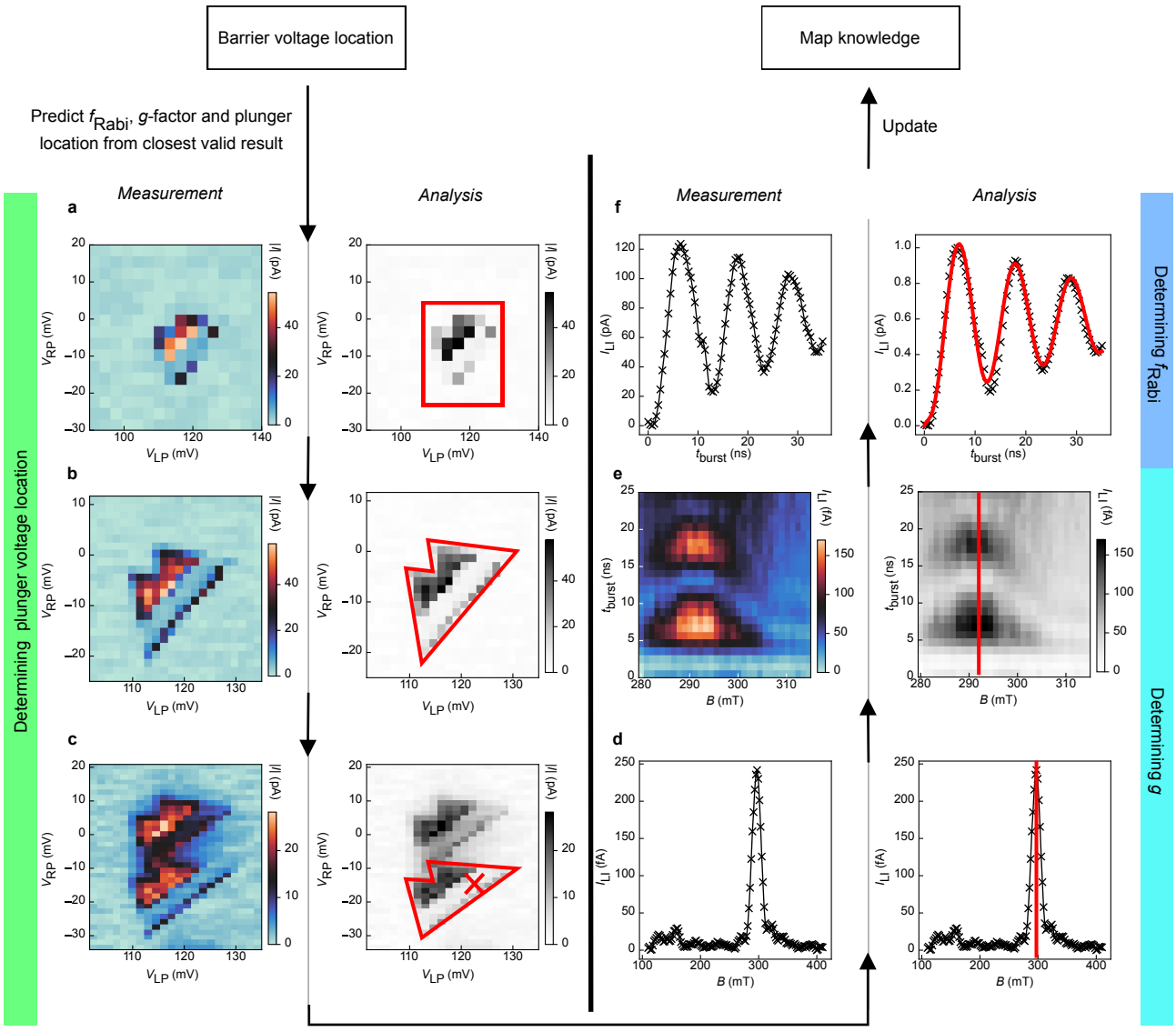


Figure 6.2: Investigation pipeline. **a** When investigating a barrier voltage location, the pipeline begins by taking a low-resolution stability diagram and updating the plunger voltage location based on that measurement. **b** A high-resolution scan is then taken to obtain a mask of the bias triangles. **c** The shape of the bias triangles in the pulsed measurement is analysed to determine the readout point relative to the mask center. **d** The resonance in the magnetic field is identified. **e** A Rabi chevron scan is conducted to finely calibrate the magnetic field position. **f** Repeated Rabi oscillation measurements and fitting are used to infer f_{Rabi}, and the map knowledge is updated accordingly.

6.2 The investigation pipeline

Next, we need a method to actually investigate the voltage locations (Fig. 6.2). The pipeline is built similarly to the one the previous chapter, with reduced measurement overhead in some parts and additional calibration steps in others.

Investigating a new barrier voltage location starts with an estimation of the plunger gate voltages via virtual gates, g, and f_{Rabi} from the closest point in the map knowledge. A centering stage corrects the plunger voltage location if necessary (Fig. 6.2**a**). A higher resolution scan of the bias triangles is then taken, first without any pulses (Fig. 6.2**b**) and then with a square pulse applied to the right plunger gate (Fig. 6.2**c**). The readout point is defined as a fixed translation from the center of the bias triangles and can therefore be chosen based on the pulsed and unpulsed stability diagrams alone (Fig. 6.2**c**). The tuning pipeline can give the information of this translation if the scheme is used as extension to the tuning pipeline, or it can be manually given. This is in contrast to Chapter 4.4 where it was determined using a search algorithm, and in contrast to Chapter 5.2 where a dedicated measurement along the detuning line was taken.

After that, the qubit resonance is found by measuring the current with varying magnetic field (Fig. 6.2**d**). If a peak is present, a Rabi chevron scan, i.e., a measurement where both the B and t_{burst} are varied, is taken. This is to finely calibrate B. The Rabi chevron measurement is analysed to find the center of the chevron (Fig. 6.2**e**). I take repeated Rabi oscillations at the magnetic field for which the resonance is present and fit a model to the data (Fig. 6.2**f**). See Appendix C.2 for details on the model used to fit Rabi oscillations.

Data is considered invalid if it falls outside a reasonable range, here 20-150MHz. This is manually set with the insight from Chapter 5.2. Further, the difference between the minimum and maximum current in the Rabi oscillation measurements (Fig. 6.2**f**) needs to be at least 10fA and the coefficient of determination R^2 of the Rabi oscillation fit needs to be at least 0.8. These two values were set after evaluating data from Chapter 5.2. If the data is considered valid, it is added to

the map knowledge as a valid point, otherwise I assign 0 MHz to the sample and do not use it to inform future investigations.

6.3 Comparison with previous chapters

Most of the pieces of the investigation pipeline come from the tuning algorithm in Chapter 4.4. However, parts of it can be skipped, e.g., PSB doesn't need to be found. Also, the readout point is not found with an expensive algorithm. The major difference between between the pipeline in this chapter and in Chapter 5.2 is that I introduced an additional step to pinpoint the correct B value by measuring a Rabi chevron. I did this to reduce mis-classifications as the algorithm now decides where to measure next based on previous investigations, unlike in Chapter 5.2 where the points where the algorithm mostly didn't take the feedback from measurements to decide on new points.

6.4 Results

I apply the Bayesian optimisation method to two specific transitions, named **A** and **C** (Fig. 5.3 from the previous chapter). Initial barrier voltages are set intentionally away from where the optimum appears to be when considering the maps in Fig. 5.4 yet within the domain where successful readout occurs. Specifically, for transition **A**, the starting voltages are $V_L = 1.350$ V, $V_M = 0.660$ V, $V_R = 1.025$ V, and for transition **C**, the voltages are $V_L = 1.322$ V, $V_M = 0.665$ V, $V_R = 1.030$ V.

The results are shown in Fig. 6.3. The sampled points in barrier voltage space are displayed in Fig. 6.3**a, d** with the first sample and the best overall marked. The Rabi frequency was improved from an initial 85 MHz to 130 MHz for transition **A**, and from 71 MHz to 127 MHz for transition **C** (see **b, e**). The progression over the number of samples is illustrated in **c, f**.

In Figure 6.4, I illustrate how the points measured by the optimisation algorithm align with those measured in Chapter 5.2. For transition **A**, the optimisation points lie well within the distribution from Chapter 5.2. The identified maximum corresponds to the direction most influential on f_{Rabi}, specifically the left barrier voltage V_{L}.

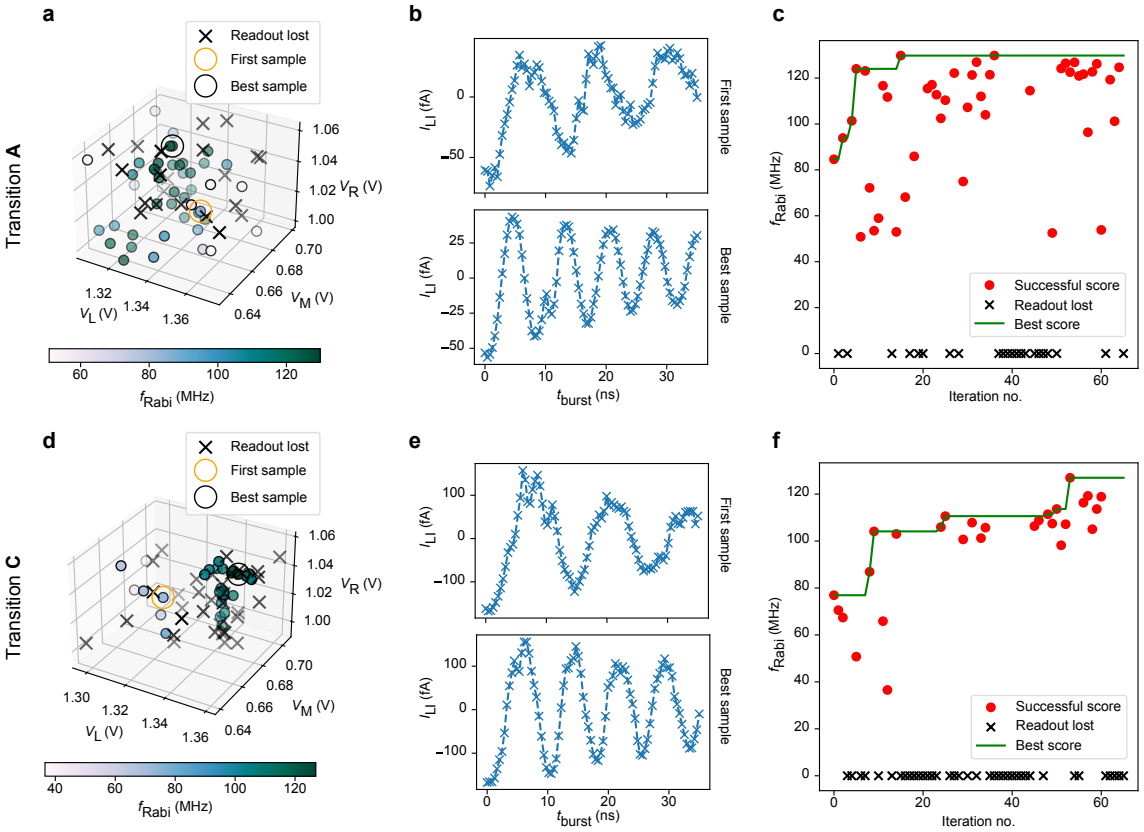


Figure 6.3: Results. Optimisation results for qubits in two different charge transitions, **A** and **C**. **a, d** show the locations in barrier voltage space that were sampled, with markers indicating the first sample and the best overall. **b, e** display these first and best samples. **c, f** show the progression of the sampled f_{Rabi} over iterations.

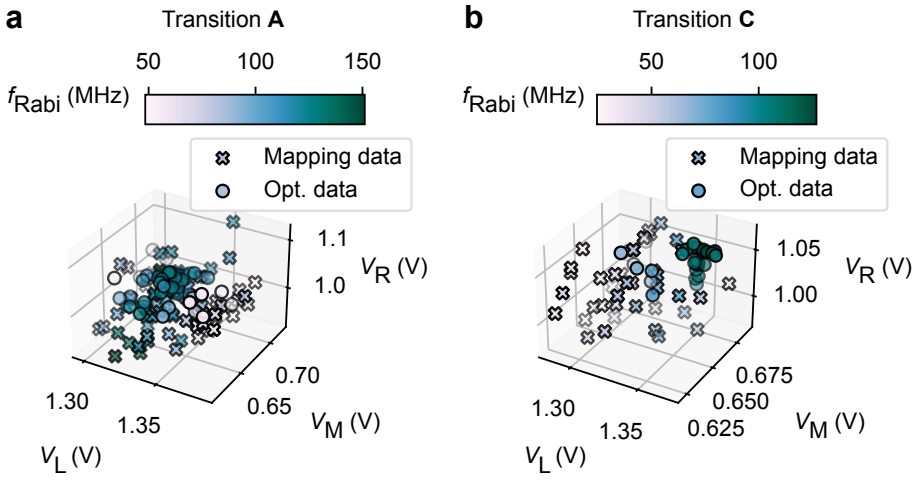


Figure 6.4: Comparison with mapping data. I show data from this chapter together with data from Chapter 5.2 for transitions **A** (panel **a**) and transition **C** (panel **b**). Data from the previous chapter is plotted as a cross, data from this chapter as a filled circle. Both share the same colorbar.

However, the mapping from the previous chapter revealed a slightly higher f_{Rabi} for lower V_{M}, which the optimisation algorithm did not identify. To improve on this, potential solutions include running the algorithm with multiple re-starts, or adjusting the exploration-exploitation trade-off to favour longer exploration periods.

In contrast, for transition **C**, the optimisation algorithm discovered a region with a maximal f_{Rabi} outside the previously measured space. The frequencies identified by the algorithm are significantly higher than those explored in the previous chapter.

In this study, I have demonstrated a Bayesian optimisation framework tailored to the enhancement of qubit performance through the targeted adjustment of barrier voltages. By applying this method to two different qubits, I observed that the Rabi frequency increased by more than 50% compared to the initial measurements. The progression of Rabi frequencies throughout the optimisation iterations illustrates the capability of the framework to navigate towards the optimal gate voltage point, thereby maximising qubit performance.

This is the first demonstration of a spin qubit that was autonomously optimised by tuning the confinement potential and aiming for highest Rabi frequency.

Future research could focus on optimising other aspects, such as decoherence times. Measuring decoherence times is typically more costly and complex than determining the Rabi frequency, because it involves more intricate experimental procedures and longer measurement durations. While Rabi frequency measurements can be obtained from straightforward Rabi oscillation experiments, assessing decoherence times requires implementing advanced pulse sequences such as Ramsey fringes or spin echo experiments. Additionally, decoherence measurements often require averaging over many repetitions to achieve reliable results, further increasing the time and resources needed. This added complexity makes these experiments ideal candidates for intelligent measurement schemes such as in Ref. [140].

Prediction is very difficult, especially if it's about the future.

— Niels Bohr

7 Conclusion & Outlook

This dissertation chronicles a transformative journey in the automation of quantum device tuning, with the qubit as our metaphorical Radcliffe Camera. We explored the necessary steps to reach this goal, addressing the major challenge of interpreting measurements in semiconductor quantum devices, i.e., to be able to "tell where we are". Initially, I demonstrated how deep learning and the careful use of experimental and simulated data is one approach to solve this problem. Next, I guided you through the process from a grounded device to achieving Rabi oscillations, confirming our arrival at the Radcliffe Camera. I further showcased the power of automation by presenting automated characterisation results. Finally, I optimised a given qubit by maximising the Rabi frequency, concluding our journey with the best possible version of our metaphorical landmark.

7.1 A look back

In Chapter 3.4, I explore the use of deep neural networks trained on both simulated and experimental data to detect bias triangles indicative of PSB in quantum dot devices. Despite the challenge of limited data, the neural network effectively solves this complex task, with cross-device validation confirming its robustness on unseen

devices. The study highlights the importance of simulated data in training due to the scarcity of experimental data.

In Chapter 4.4, I demonstrate the fully automated tuning of spin qubits, progressing from a grounded device to qubit control, achieving Rabi oscillations in 10 out of 13 trials. Most tuning processes were completed within three days, with speed mainly constrained by the integration times needed for DC transport measurements. These could be replaced by faster readout alternatives to enhance efficiency.

The algorithm's versatile and modular design allows adaptation for similar quantum devices, such as silicon FinFETs, and could be modified for other architectures with different measurement techniques.

In Chapter 5.2, I introduce a systematic approach to measure the g-factor and f_{Rabi} as a function of the three barrier gate voltages, and applied it to qubits across four different charge transitions. I analysed current signatures and the effects of varying confinement potentials.

In Chapter 6.4, I have demonstrated a BO framework specifically designed to enhance qubit performance by adjusting barrier voltages. Applying this method to two different qubits resulted in over 50% improvement in Rabi frequency from the initial points.

Let me clarify the scope of my achievements. While I believe my work makes significant strides, there are undoubtedly many opportunities to enhance the efficiency of the algorithms I propose. Additionally, there is potential for faster execution through alternative measurement techniques, which could further expedite the tuning of quantum devices.

Consider Sir Roger Bannister, who was the first (known) person to run a sub-4-minute mile in 1954 at the Iffley sports ground in Oxford. It is said that his achievement inspired many others to follow in quick succession. Much the same, I merely show that it is possible, and hope that others will follow and soon surpass me.

7.2 A look forward

The results of this dissertation open promising avenues for future research and application in the field of spin qubits. One significant direction is the utilisation of the autonomous tuning algorithm to characterise a large number of quantum devices. By systematically applying my tuning and characterisation methods across many devices, one could generate extensive datasets that will provide deeper insights into device variability and performance. This large-scale characterisation could enable the identification of optimal design parameters and operational conditions, thereby enhancing the overall reliability and efficiency of spin qubit devices.

Improvements in quantum device fabrication could also significantly impact the tuning process. Advances in fabrication techniques can lead to devices with more uniform characteristics, reducing the variability between devices and making them easier to tune [24]. As fabrication technologies evolve, we can expect more robust and reliable quantum devices that require less manual intervention and more straightforward tuning procedures.

A natural next step is to amend the methods so they can be used in devices that use charge sensors. The precise tuning of charge sensors is needed for accurate qubit operation. This adaptation will likely involve modifying the signal-processing algorithms to account for the specific characteristics and requirements of charge sensors.

One of the primary speed limitations in the current methodology is the time required to measure charge stability diagrams and qubit readout. Measuring a charge stability diagram involves sweeping gate voltages and recording the resulting current, which can be time-consuming, especially when high-resolution measurements are needed. Similarly, measuring spin qubit properties, such as Rabi frequencies, coherence times, and other critical parameters, often requires multiple averages to achieve accurate results.

A significant speedup can be achieved by transitioning from charge transport measurements to measurements using radiofrequency reflectometry. In this approach, the DQD is indirectly measured using radio frequency [141].

For an apples-to-apples comparison, consider the works of Moon *et al.* (2020) [30] and van Straaten *et al.* (2022) [32]. They address the same problem on similar device architectures, but Moon *et al.* used charge transport measurements, while van Straaten *et al.* employed fast radiofrequency reflectometry measurements. Moon *et al.* reported a tuning time of 3 hours, whereas van Straaten *et al.* achieved the same with radiofrequency measurements in just 15 minutes, representing a 10x speedup. Applying this factor, the tuning time from ground to qubit, reported as around 3 days in this dissertation, could potentially be reduced to approximately 7 hours.

Leveraging sophisticated algorithms could further reduce this time. Lennon *et al.* [142] have shown that machine learning can provide a significant reduction in measurement time. In my full tuning algorithm, I used fewer than 100 2D measurements per successful tuning run, each comprising around 20,000 data points at most. Given the bandwidth of radiofrequency measurements in the MHz range [141, 143], a tuning algorithm could potentially bring a qubit up from ground on the order of seconds. This shows that there is still plenty of room for innovation. Additional factors affecting speed include the ramp time of magnets and digital-to-analog converters, as well as the processing times of the tuning algorithm itself.

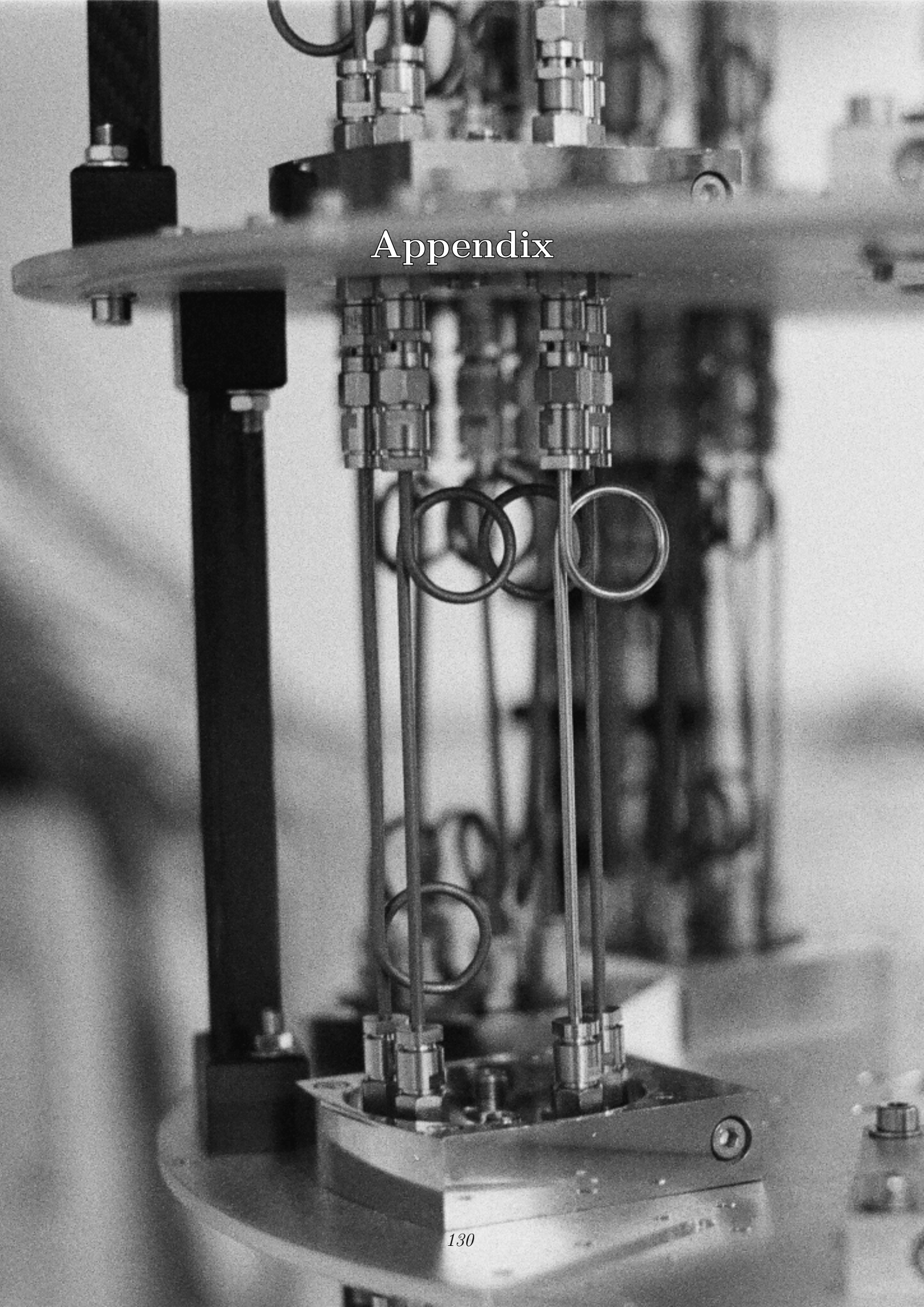
As quantum computing systems scale up, efficiently tuning large arrays of qubits becomes essential. Adapting my algorithm to manage the complexities of larger arrays—including intricate interactions among numerous qubits—is a critical next step. This may involve developing sophisticated optimisation techniques and incorporating advanced machine learning models to handle the increased dimensionality and complexity.

One promising approach to manage this complexity is parallelised tuning. By leveraging parallel processing techniques, multiple qubits or sections of a qubit array can be tuned simultaneously, significantly reducing the overall time required for system initialisation. This approach could involve deploying multiple instances of the tuning algorithm across different segments of the array, coordinating their activities to ensure optimal global performance. Parallelised tuning not only

accelerates the setup process but also allows for more scalable and flexible quantum computing architectures.

These future developments would extend the applicability of my algorithm and contribute to the advancement of quantum computing with spin qubits. By enabling the efficient characterisation and tuning of a wide range of quantum devices, my work lays a robust foundation for the scalable and reliable deployment of quantum technologies. The continued evolution of my methods will support the growing demands of the field, ultimately –hopefully– facilitating the realisation of practical and powerful quantum computers.

Appendix



A

Supporting data for Chapter 3.4

A.1 Device dimensions

We use four different silicon FinFET devices with varying dimensions in this chapter. The dimensions that are varied are the length of the plunger gates L_{P}, the length of the barrier gate L_{B}, and the width of the fin W. We show a schematic of the top view of the device with the lengths and widths that are varied in Fig. A.1. A top view SEM image of similar devices can be found in [123]. Table A.1 shows the estimated dimensions for each device.

A.2 Simulation parameters

This appendix provides a detailed list of the parameters used in the simulator. It includes descriptions, sampling ranges, comments on the dimensionality, and the physical units of each parameter. For a full list of the sampled parameters, see

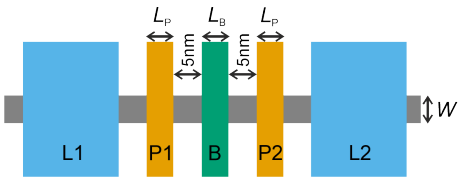


Figure A.1: Top view schematic of device layout. Colors and labels are the same as in Figs. 3.1.

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

Table A.1: Estimated device dimensions for the different devices. Illustration of the layout is given in Fig. A.1. L_B gives the length of the gap between the plunger gates for device i as it has no barrier gate.

Table A.2. The code is available at github.com/oxquantum-repo/identifying-psb.

Parameter	Description	Sampling range	Dimensionality	Physical units
$\tilde{\mu}_S$	Source chemical potential offset	$[-0.5, 0.5]$	Scalar	meV
$\tilde{\mu}_D$	Drain chemical potential offset, calculated as $\tilde{\mu}_D = \tilde{\mu}_S + V_{\text{bias}}$	Calculated, not sampled	Scalar	meV
V_{bias}	Bias voltage between source and drain	$[-2, -0.1]$	Scalar	meV
$\mathcal{T}$	Temperature in Kelvin	$[0.1, 1]$	Scalar	K
L_A	Lever arm for the left dot	$[0.5, 1.5]$	Scalar	meV/V
L_B	Lever arm for the right dot	$[0.5, 1.5]$	Scalar	meV/V
C_A	Cross-talk coefficient affecting the left dot	$[0.0, 0.5]$	Scalar	meV/V
C_B	Cross-talk coefficient affecting the right dot	$[0.0, 0.5]$	Scalar	meV/V
σ	Standard deviation for relative energy level jitter	$[0, 0.05]$	Scalar	Dimensionless
E_{C_m}	Electrostatic coupling energy between dots	$[0.01, 2]$	Scalar	meV
ΔE_A	Average energy level spacing for the left dot	$[V_{\text{bias}} - 0.2, V_{\text{bias}} + 0.2]$	Scalar	meV
ΔE_B	Average energy level spacing for the right dot	$[0.2, 0.5]$	Scalar	meV
$\delta E_A^{(i)}$	Random fluctuations in left dot energy levels	$[-\epsilon_A, \epsilon_A]$, where $\epsilon_A = \Delta E_A/4$	Vector of length $n_{\text{levels}}^{\text{left}}$	meV
$\delta E_B^{(k)}$	Random fluctuations in right dot energy levels	$[-\epsilon_B, \epsilon_B]$, where $\epsilon_B = \Delta E_B/4$	Vector of length $n_{\text{levels}}^{\text{right}}$	meV
$E_A^{(i)}$	Energy levels of the left dot, calculated using $E_A^{(i)} = i \cdot \Delta E_A + \delta E_A^{(i)}$	Calculated, not sampled	Vector of length $n_{\text{levels}}^{\text{left}}$	meV
$E_B^{(k)}$	Energy levels of the right dot, calculated using $E_B^{(k)} = k \cdot \Delta E_B + \delta E_B^{(k)}$	Calculated, not sampled	Vector of length $n_{\text{levels}}^{\text{right}}$	meV
$n_{\text{levels}}^{\text{left}}$	Number of energy levels in the left dot	Random integer from $\{2, 3\}$	Integer	Dimensionless
$n_{\text{levels}}^{\text{right}}$	Number of energy levels in the right dot	Random integer from $\{2, 3, 4, 5, 6\}$	Integer	Dimensionless
$\tilde{\Gamma}_L^{(i)}$	Source tunneling rates for left dot levels	$[0.01, 0.5]$	Vector of length $n_{\text{levels}}^{\text{left}}$	meV
$\tilde{\Gamma}_R^{(k)}$	Drain tunneling rates for right dot levels	$[0.01, 0.5]$	Vector of length $n_{\text{levels}}^{\text{right}}$	meV
$\tilde{\Gamma}_T^{(i,k)}$	Inter-dot tunneling rates between levels	$[0.01, 0.4]$	Matrix of size $n_{\text{levels}}^{\text{left}} \times n_{\text{levels}}^{\text{right}}$	meV
χ_{PSB}	Flag indicating if Pauli spin blockade is simulated	Randomly True or False	Boolean	Dimensionless
κ	Standard deviation for Gaussian blurring	$[0.8, 1.2]$	Scalar	Pixels
η	Amplitude of relative white noise added to the current	$[0.03, 0.07]$	Scalar	Dimensionless

Table A.2: Simulation parameters with descriptions, sampling ranges, dimensionality, and physical units.

A.2.1 Charge jump noise parameters and simulation

To replicate sudden shifts in the current caused by random charge rearrangements—known as charge jumps—we incorporate specific noise parameters into the simulation. The simulator randomly determines the number of charge jumps to include, typically choosing an integer between zero and two. For each charge jump, it selects an axis along which the shift will occur (either the x-axis or the y-axis) at random.

A shift magnitude is then sampled from a uniform distribution between -0.05 and 0.05, representing the fraction of the total length of the selected axis. Additionally, a point along the chosen axis is randomly selected where the shift will be applied.

In the simulation, the shift magnitude is converted into an integer number of pixels based on the size of the current data array along the chosen axis. The current data array, representing the simulated current values over the voltage grid, is then modified by shifting portions of the array by the calculated number of pixels up to the selected point along the axis. This operation is akin to cutting the data array at the specified point and moving the preceding segment by the shift amount, effectively introducing a discontinuity in the current values.

A.3 Neural Network Architectures

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

A.3.1 Modified ResNet18

The modified ResNet18 architecture is based on the original ResNet18 model [126], with changes to the input and output layers. The first convolutional layer is altered to accept two input channels instead of three, while the final fully connected layer is

changed to output two classes instead of 1000. With these changes, the model has 11,174,402 learnable parameters. We use the TorchVision [144] implementation of this model. All results in this chapter are based on models with this architecture unless otherwise explicitly stated.

A.3.2 Modified LeNet5

The ResNet18 architectures could be considered too large compared to the size of the data set. We therefore consider a much smaller neural network. Our modified LeNet5 model [145] consists of two convolutional layers, each followed by batch normalisation, a ReLU activation, and max-pooling. The feature maps are then flattened and passed through three fully connected layers, with ReLU activations after the first and second layers. This leads to 942,500 learnable parameters, which is about an order of magnitude smaller compared to the ResNet18 architecture.

A.4 Details of the training procedure

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

We use the Adam optimizer [92] with a regularisation factor of 0.001. The optimiser is initialised with a learning rate of 0.001. The learning rate is then decayed by a factor of 0.1 with a scheduler once a plateau in the training loss is reached, with a patience of 5 epochs. We used cross-entropy loss as the objective function, with balanced class weights to account for class imbalance in the training data.

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

When using only experimental data, we augment the available training data (see Table 3.1) until we have 50,000 examples. In the case of mixed data, we sample 12,500 examples from the simulator, augment them twice, and then augment the available experimental training data until we have 25,000 examples which gives us 50,000 examples in total. To counteract class imbalance, i.e. examples with and without PSB, we also weight the classes according to their prevalence in the loss function.

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

Testing data, i.e. the corresponding fold in the cross-validation procedure, is not augmented but only normalised. The neural network and its training is implemented in PyTorch [146].

B

Supporting data for Chapter 4.4

B.1 Hyperparameters

We provide a list of all hyperparameters and comments 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.

B.1.1 Define DQD

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

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.

B.1.2 Tune barriers

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.

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.

B.1.3 Find PSB

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.

Re-centering

- 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.* [135]):
  - `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.

High resolution PSB detection

- 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.

Danon gap check

- 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.

B.1.4 Find readout

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.

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.

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.

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.

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.

B.2 Efficient measurement algorithm

Taking multiple charge transition stability diagram measurements is notably time-consuming due to the predominance of featureless areas, as bias triangles are typically embedded within a skewed rectangular pattern. To address this, we developed an efficient measurement algorithm that reduces the measurement time to about a third of a brute-force scan. By rephrasing the measurement of bias triangles as an image processing task, where the goal is to determine a contour that traces the outline of a bias triangle, we significantly enhanced the efficiency of the process.

In a binary image represented by a matrix of zeros and ones, the perimeter of any grouping of non-zero elements forming a contiguous region is known as a *contour*. The Moore-Neighbour contour tracing algorithm offers a method for evaluating a complete contour given a starting point within it [147]. This algorithm works by examining only the pixels adjacent to a previously examined pixel. Since the pixel locations correspond to plunger gate voltages, measurements of well-separated pixels are both time-consuming and risk introducing noise, such as switches. Once the edge of a bias triangle is identified, its contour can be quickly measured with minimal overhead from the device. After evaluating the contour, each pixel inside 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 ones and zeros in the binary image, respectively. The threshold can either be set manually, using prior knowledge of the system, or determined dynamically.

For a fully automated system, we first performed a calibration scan using sparse sampling and evaluated the median absolute deviation threshold from these measured points. The sampling routine followed a *snake scan* pattern, where measurements are performed horizontally from left to right until reaching the boundary of the measurement region, then proceed vertically for a fixed length before continuing horizontally in the opposite direction. This process is repeated until the entire image has been covered.

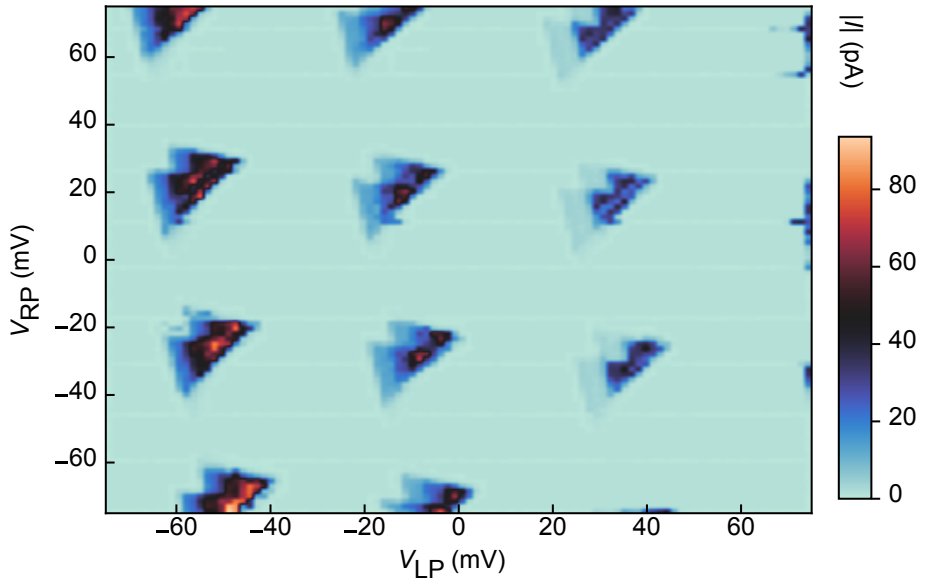


Figure B.1: Examples of measurements taken with the efficient measurement algorithm. Areas where no measurement have been taken have been filled with the threshold value.

With the threshold determined, a second sparse sampling across the measurement region is conducted. The Moore-Neighbour contour tracing routine is triggered on any measurements exceeding the threshold, followed by a routine to measure the inside of the contour. This process systematically captures a complete pair of bias triangles. Once a cluster is fully measured, the scan resumes until the next cluster triggers flood filling. To minimise wasted measurements, current values were cached. The second sampling routine also employed a snake scan, offset vertically from the original to maximize the chance of encountering a bias triangle.

Bias triangles are organised in a skewed grid pattern. The bias triangles identified 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 might 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 barycenters 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.

This method significantly streamlines the process, reducing measurement time by approximately two-thirds. For the hyperparameters reported above, the measurement of a 100 by 100 point stability diagram takes 14.5 minutes ± 3.0 minutes with this efficient measurement algorithm, compared to 43.7 minutes ± 0.1 minutes with a conventional grid scan.

B.3 Qubit gate voltage volume calculation

I present the underlying assumptions for the calculation of the haystack volume for the analogy in Chapter 4.4.

Fitting a convex hull to the 3D barrier voltage locations where qubits were found gives us $V_{\text{hull, 3D}}$:

$$V_{\text{hull, 3D}} \approx 0.00032\,\text{V}^3 = 320000\,(\text{mV})^3. \tag{B.1}$$

This can only be an estimate. We might be overestimating the volume because there might regions in the full hull without an available qubit. We might underestimate the volume because the locations are probably not at the outer edge of the possible volume.

To get an understanding how large this space is, let us consider the volume of the space in which we search. If we take the safe ranges in the given experiment, we have

$$V_{\text{safe ranges, 3D}} = 2 \times 2 \times 2\,\text{V}^3 = 8\,\text{V}^3. \tag{B.2}$$

We can compute the fraction of the space that the volume with qubit occupies compared to the full search space:

$$v_{\text{3D}} = \frac{V_{\text{hull, 3D}}}{V_{\text{safe ranges, 3D}}} \approx 0.004\%. \tag{B.3}$$

We also need to consider the restrictions imposed by the plunger gate voltages. There is only a small area where a qubit can be controlled, see Fig. 4.3. If we assume an area of 10 mV ×10 mV, we have

$$V_{\text{hull, 5D}} = V_{\text{hull, 3D}} \times 0.01\text{V} \times 0.01\text{V} \approx 3.20 \times 10^{-08}\text{V}^5 \tag{B.4}$$

Since we do not search over the full safe ranges for the plunger gates, we do not use that as a reference volume. Instead, we use the window that we search over, which is 300 mV × 300 mV. Overall this gives us a fraction of the hull volume in the 5D gate voltage space:

$$v_{\text{5D}} = \frac{V_{\text{hull, 5D}}}{V_{\text{safe ranges, 3D}} \times 0.3 \times 0.3} \approx 4.43 \times 10^{-6}\%. \tag{B.5}$$

Now to compare this to a needle in a haystack, we first need to calculate the volume of a needle. We assume the needle has a cylindrical shape, and is 5 cm long with a radius of 1.5 mm. This gives a volume of the needle:

$$V_{\text{needle}} = \pi \times 5\,\text{cm} \times (1.5\,\text{mm})^2 \approx 0.35\,\text{cm}^3. \tag{B.6}$$

A cubic haystack with side length of 2 m has a volume of:

$$V_{\text{haystack}} = (2\,\text{m})^3 = 8\,\text{m}^3 \tag{B.7}$$

The needle volume as a fraction of the haystack volume is

$$v_{\text{needle}} = \frac{V_{\text{needle}}}{V_{\text{haystack}}} \approx 4.41 \times 10^{-6}\%, \tag{B.8}$$

which is very close to v_{5D}.

Another way to compute the side length of the haystack is to compare v_{needle} to v_{5D},

$$v_{\text{needle}} = \frac{V_{\text{needle}}}{V_{\text{haystack}}} = v_{\text{5D}} \tag{B.9}$$

and solve for the V_{haystack}. This gives us a side length of a haystack of about 2 m:

$$V_{\text{haystack}} = \frac{V_{\text{needle}}}{v_{\text{5D}}} \approx (1.997\,\text{m})^3 \approx (2\,\text{m})^3. \tag{B.10}$$

B.4 Modular framework

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

B.4.1 Stage structure

1. Integration with QCoDeS [148]: 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 hyperparameters to fulfill its requirements, as detailed in the Section B.1.

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

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` function 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.

B.4.2 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.

C

Supporting methods and data for Chapters 5.2 and 6.4

Contents

C.1 Virtual gates

For small voltage ranges we can assume a linear cross talk between the gates. In particular, I am interested in how the plunger gate voltages need to be adjusted if I change the barrier gate voltage. I want to compute the compensated plunger voltage locations given their uncompensated values, the amount of change of the barrier voltage ΔV, and the coupling strength γ.

We can computed the compensated plunger voltage as

$$V_{i,\text{compensated}} = V_{i,\text{uncompensated}} - \sum_{j\in\{L,M,R\}} \Delta V_j \, \gamma_{i,j} \tag{C.1}$$

I measured the couplings once and used it throughout this work. In principle the couplings can be automatically measured using the modules presented in

	V_{L}	V_{M}	V_{R}
V_{LP}	0.5208	0.5208	0.0000
V_{RP}	0.0000	0.4166	0.6250

Table C.1: Couplings. The couplings used to compute virtual gates.

this dissertation.

The values used are show in Table C.1. For example, if the left barrier voltage V_{L} changes by 1 mV, the left plunger voltage needs to be corrected by -0.5208 mV and the right plunger voltage remains unchanged.

C.2 Model to fit Rabi oscillations

We use the following model to fit Rabi oscillation measurements.

$$I(t) = A \exp\left(\frac{-t}{\tau}\right) \cos\left(\omega t + \phi\right) + B \exp\left(\frac{-t}{\tau_2}\right) + C \tag{C.2}$$

The first summand describes Rabi oscillations with an amplitude A, angular frequency ω and decay time τ. Due to pulse distortion and heating effects, the signal may drift which is reflected by the second and third summand with an amplitude B and its own decay time τ_2 and an offset C.

The components are visualised in Fig. C.1.

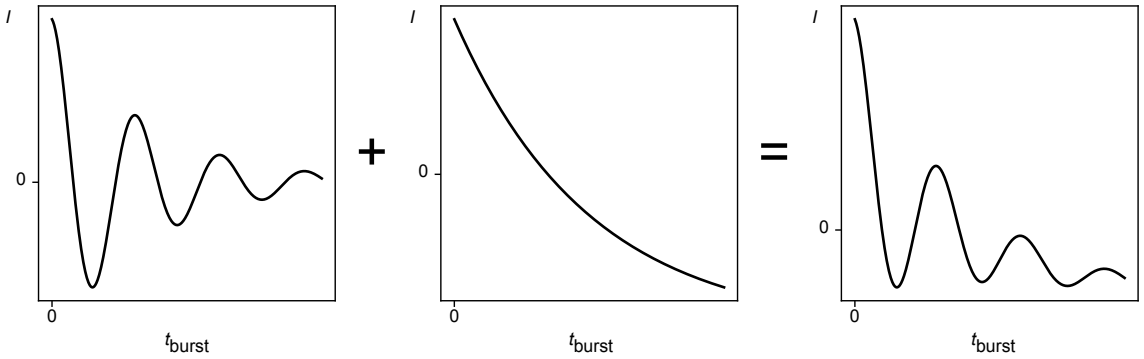


Figure C.1: Rabi model visualisation. The left panel shows a cosine with a decay envelope. The middle pane shows a decay with an offset the distortion which may be added to the signal due to pulse distortions. The right panel shows the sum of both parts.

C.3 Current dependency

The current across the DQD depends among other things on the barrier voltage setting. This relationship is mostly not linear. This is due to the exponential relationship between barrier strength and barrier gate voltage [149] and due to disorder potential that is shifted by gate voltages.

Fig. C.2 shows the distribution of currents for a relevant barrier gate voltage range for three transitions. I choose to show the current at the lowest point between the base line of the bias triangles and the first excited state line. This is marked in Fig. 5.6 with a green star. The data I show here is the same data as used for Fig. 5.6, so including measurements of barrier voltage settings where readout was lost. Similarly to Fig. 5.4, I show the mean of a slice of a GP fitted to each data set at V_R=0.99 V to aid visual interpretability.

All three maps show highly non-linear behaviours, with areas of very flat current dependency and areas of high change at the edge towards the origin. Generally, higher barrier voltages correspond to lower currents but transitions **B** and **C** show a few instances where increasing the barrier voltage actually also increases the current. This is likely due to shifts in the disorder potential.

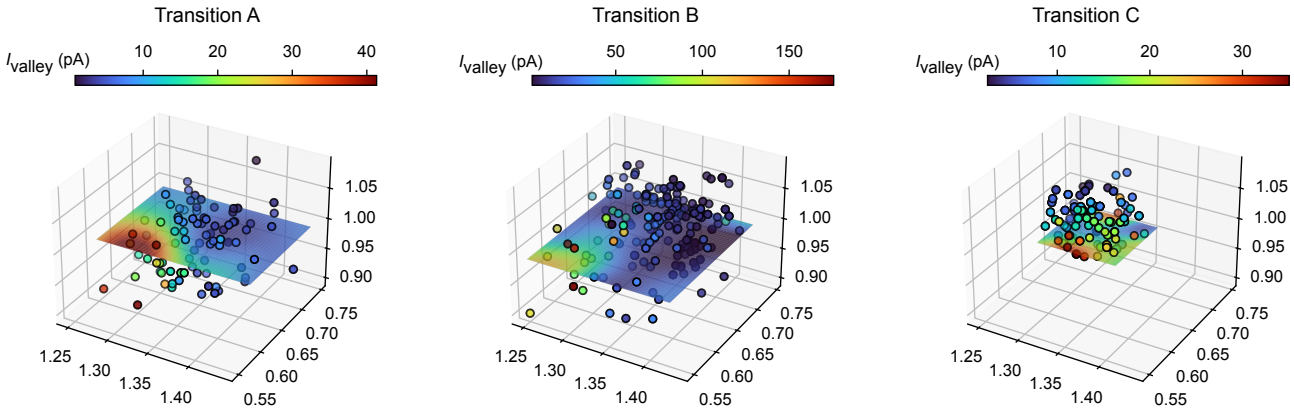


Figure C.2: Current dependency. I show the current in the valley (see the green star in Fig. 5.6) of bias triangles for three different transitions (see Fig. 5.3 for the names). For visual interpretability, I fit GPs to each dataset and plot its mean at V_R=0.99 V. All maps show strong non-linearity of the current as a dependence of the barrier gate voltages.

References

[1] Peter Brian Medawar. *The threat and the glory: reflections on science and scientists*. Oxford University Press, 1990.

[2] Arthur C Clarke. *Profiles of the Future*. Hachette UK, 2013.

[3] Richard P. Feynman. "Simulating Physics with Computers". In: *International Journal of Theoretical Physics* 21.6/7 (1982), pp. 467–488. DOI: 10.1007/BF02650179.

[4] Alán Aspuru-Guzik, Anthony D Dutoi, Peter J Love, and Martin Head-Gordon. "Simulated quantum computation of molecular energies". In: *Science* 309.5741 (2005), pp. 1704–1707. DOI: 10.1126/science.1113479.

[5] Cornelius Hempel et al. "Quantum Chemistry Calculations on a Trapped-Ion Quantum Simulator". In: *Physical Review X* 8 (3 July 2018), p. 031022. DOI: 10.1103/PhysRevX.8.031022.

[6] P.W. Shor. "Algorithms for quantum computation: discrete logarithms and factoring". In: *Proceedings 35th annual symposium on foundations of computer science*. IEEE. 1994, pp. 124–134. DOI: 10.1109/sfcs.1994.365700.

[7] Rolando D Somma, Sergio Boixo, Howard Barnum, and Emanuel Knill. "Quantum simulations of classical annealing processes". In: *Physical Review Letters* 101.13 (2008), p. 130504. DOI: 10.1103/PhysRevLett.101.130504.

[8] Sergey Bravyi, David Gosset, and Robert König. "Quantum advantage with shallow circuits". In: *Science* 362.6412 (2018), pp. 308–311. DOI: 10.1126/science.aar3106.

[9] Iris Cong, Soonwon Choi, and Mikhail D Lukin. "Quantum convolutional neural networks". In: *Nature Physics* 15.12 (2019), pp. 1273–1278. DOI: 10.1038/s41567-019-0648-8.

[10] David P. DiVincenzo. "The Physical Implementation of Quantum Computation". In: *Fortschritte der Physik* 48.9-11 (2000), pp. 771–783. DOI: 10.1002/1521-3978(200009)48:9/11<771::AID-PROP771>3.0.CO;2-E. arXiv: 0002077 [quant-ph].

[11] Daniel Loss and David P DiVincenzo. "Quantum computation with quantum dots". In: *Physical Review A* 57.1 (1998), p. 120. DOI: 10.1103/PhysRevA.57.120.

[12] Menno Veldhorst, CH Yang, JCC Hwang, W Huang, JP Dehollain, JT Muhonen, S Simmons, A Laucht, FE Hudson, Kohei M Itoh, et al. "A two-qubit logic gate in silicon". In: *Nature* 526.7573 (2015), pp. 410–414. DOI: 10.1038/nature15263.

[13] Guido Burkard, Thaddeus D Ladd, Andrew Pan, John M Nichol, and Jason R Petta. "Semiconductor spin qubits". In: *Reviews of Modern Physics* 95.2 (2023), p. 025003. DOI: 10.1103/RevModPhys.95.025003.

[14] LMK Vandersypen, H Bluhm, JS Clarke, AS Dzurak, R Ishihara, A Morello, DJ Reilly, LR Schreiber, and M Veldhorst. "Interfacing spin qubits in quantum dots and donors - hot, dense, and coherent". In: *npj Quantum Inf.* 3.1 (2017), pp. 1–10. DOI: 10.1038/s41534-017-0038-y.

[15] Toivo Hensgens, Takafumi Fujita, Laurens Janssen, Xiao Li, CJ Van Diepen, Christian Reichl, Werner Wegscheider, S Das Sarma, and Lieven MK Vandersypen. "Quantum simulation of a Fermi–Hubbard model using a semiconductor quantum dot array". In: *Nature* 548.7665 (2017), pp. 70–73. DOI: 10.1038/nature23022.

[16] Xiao Xue, Maximilian Russ, Nodar Samkharadze, Brennan Undseth, Amir Sammak, Giordano Scappucci, and Lieven MK Vandersypen. "Quantum logic with spin qubits crossing the surface code threshold". In: *Nature* 601.7893 (2022), pp. 343–347. DOI: 10.1038/s41586-021-04273-w.

[17] Akito Noiri, Kenta Takeda, Takashi Nakajima, Takashi Kobayashi, Amir Sammak, Giordano Scappucci, and Seigo Tarucha. "Fast universal quantum gate above the fault-tolerance threshold in silicon". In: *Nature* 601.7893 (2022), pp. 338–342. DOI: 10.1038/s41586-021-04182-y.

[18] Mateusz T Mądzik, Serwan Asaad, Akram Youssry, Benjamin Joecker, Kenneth M Rudinger, Erik Nielsen, Kevin C Young, Timothy J Proctor, Andrew D Baczewski, Arne Laucht, et al. "Precision tomography of a three-qubit donor quantum processor in silicon". In: *Nature* 601.7893 (2022), pp. 348–353. DOI: 10.1038/s41586-021-04292-7.

[19] Xin Zhang, Elizaveta Morozova, Maximilian Rimbach-Russ, Daniel Jirovec, Tzu-Kan Hsiao, Pablo Cova Fariña, Chien-An Wang, Stefan D Oosterhout, Amir Sammak, Giordano Scappucci, et al. "Universal control of four singlet–triplet qubits". In: *Nature Nanotechnology* 20.2 (2025), pp. 209–215. DOI: 10.1038/s41565-024-01817-9. arXiv: 2312.16101.

[20] Akito Noiri, Kenta Takeda, Takashi Nakajima, Takashi Kobayashi, Amir Sammak, Giordano Scappucci, and Seigo Tarucha. "A shuttling-based two-qubit logic gate for linking distant silicon quantum processors". In: *Nature Communications* 13.1 (2022), p. 5740. DOI: 10.1038/s41586-021-04182-y.

[21] Adam R Mills, Charles R Guinn, Michael J Gullans, Anthony J Sigillito, Mayer M Feldman, Erik Nielsen, and Jason R Petta. "Two-qubit silicon quantum processor with operation fidelity exceeding 99%". In: *Science Advances* 8.14 (2022), eabn5130. DOI: 10.1126/sciadv.abn5130.

[22] Luca Petit, HGJ Eenink, M Russ, WIL Lawrie, NW Hendrickx, SGJ Philips, JS Clarke, LMK Vandersypen, and M Veldhorst. "Universal quantum logic in hot silicon qubits". In: *Nature* 580.7803 (2020), pp. 355–359. DOI: 10.1038/s41586-020-2170-7.

[23] Leon C Camenzind, Simon Geyer, Andreas Fuhrer, Richard J Warburton, Dominik M Zumbühl, and Andreas V Kuhlmann. "A hole spin qubit in a fin field-effect transistor above 4 kelvin". In: *Nature Electronics* 5.3 (2022), pp. 178–183. DOI: 10.1038/s41928-022-00722-0.

[24] Samuel Neyens, Otto K Zietz, Thomas F Watson, Florian Luthi, Aditi Nethwewala, Hubert C George, Eric Henry, Mohammad Islam, Andrew J Wagner, Felix Borjans, et al. "Probing single electrons across 300-mm spin qubit wafers". In: *Nature* 629.8010 (2024), pp. 80–85. DOI: 10.1038/s41586-024-07275-6.

[25] AMJ Zwerver, T Krähenmann, TF Watson, Lester Lampert, Hubert C George, Ravi Pillarisetty, SA Bojarski, Payam Amin, SV Amitonov, JM Boter, et al. "Qubits made by advanced semiconductor manufacturing". In: *Nature Electronics* 5.3 (2022), pp. 184–190. DOI: 10.1038/s41928-022-00727-9.

[26] Mathieu de Kruijf, Simon Geyer, Toni Berger, Matthias Mergenthaler, Floris Braakman, Richard J Warburton, and Andreas V Kuhlmann. "A compact and versatile cryogenic probe station for quantum device testing". In: *Review of Scientific Instruments* 94.5 (2023). DOI: 10.1063/5.0139825.

[27] Edward J Thomas, Virginia N Ciriano-Tejel, David F Wise, Domenic Prete, Mathieu de Kruijf, David J Ibberson, Grayson M Noah, Alberto Gomez-Saiz, M Fernando Gonzalez-Zalba, Mark AI Johnson, et al. "Rapid cryogenic characterization of 1,024 integrated silicon quantum dot devices". In: *Nature Electronics* 8.1 (2025), pp. 75–83. DOI: 10.1038/s41928-024-01304-y. arXiv: 2310.20434.

[28] Stephan GJ Philips, Mateusz T Mądzik, Sergey V Amitonov, Sander L de Snoo, Maximilian Russ, Nima Kalhor, Christian Volk, William IL Lawrie, Delphine Brousse, Larysa Tryputen, et al. "Universal control of a six-qubit quantum processor in silicon". In: *Nature* 609.7929 (2022), pp. 919–924. DOI: 10.1038/s41586-022-05117-x.

[29] J Darulová, SJ Pauka, N Wiebe, KW Chan, GC Gardener, MJ Manfra, MC Cassidy, and Matthias Troyer. "Autonomous tuning and charge-state detection of gate-defined quantum dots". In: *Physical Review Applied* 13.5 (2020), p. 054005. DOI: 10.1103/PhysRevApplied.13.054005.

[30] H Moon, DT Lennon, J Kirkpatrick, NM van Esbroeck, LC Camenzind, Liuqi Yu, F Vigneau, DM Zumbühl, G Andrew D Briggs, MA Osborne, et al. "Machine learning enables completely automatic tuning of a quantum device faster than human experts". In: *Nature Communications* 11.1 (2020), pp. 1–10. DOI: 10.1038/s41467-020-17835-9.

[31] Brandon Severin, Dominic T Lennon, Leon C Camenzind, Florian Vigneau, Federico Fedele, Daniel Jirovec, Andrea Ballabio, Daniel Chrastina, Giovanni Isella, Mathieu de Kruijf, et al. "Cross-architecture tuning of silicon and SiGe-based quantum devices using machine learning". In: *Scientific Reports* 14.1 (2024), p. 17281. DOI: 10.1038/s41598-024-67787-z. arXiv: 2107.12975.

[32] Barnaby Van Straaten, Federico Fedele, Florian Vigneau, Joseph Hickie, Daniel Jirovec, Andrea Ballabio, Daniel Chrastina, Giovanni Isella, Georgios Katsaros, and Natalia Ares. "All-rf-based coarse-tuning algorithm for quantum devices using machine learning". In: *Physical Review Applied* 24.5 (2025), p. 054030. DOI: 10.1103/v11m-dbhm. arXiv: 2211.04504.

[33] Sandesh S Kalantre, Justyna P Zwolak, Stephen Ragole, Xingyao Wu, Neil M Zimmerman, Michael D Stewart, and Jacob M Taylor. "Machine learning techniques for state recognition and auto-tuning in quantum dots". In: *npj Quantum Information* 5.1 (2019), pp. 1–10. DOI: 10.1038/s41534-018-0118-7.

[34] Justyna P Zwolak, Thomas McJunkin, Sandesh S Kalantre, JP Dodson, ER MacQuarrie, DE Savage, MG Lagally, SN Coppersmith, Mark A Eriksson, and Jacob M Taylor. "Autotuning of double-dot devices in situ with machine learning". In: *Physical Review Applied* 13.3 (2020), p. 034075. DOI: 10.1103/PhysRevApplied.13.034075.

[35] Justyna P Zwolak, Thomas McJunkin, Sandesh S Kalantre, Samuel F Neyens, ER MacQuarrie, Mark A Eriksson, and Jacob M Taylor. "Ray-based framework for state identification in quantum dot devices". In: *PRX Quantum* 2.2 (2021), p. 020335. DOI: 10.1103/PRXQuantum.2.020335. eprint: 2102.11784.

[36] V Nguyen, SB Orbell, Dominic T Lennon, Hyungil Moon, Florian Vigneau, Leon C Camenzind, Liuqi Yu, Dominik M Zumbühl, G Andrew D Briggs, Michael A Osborne, et al. "Deep reinforcement learning for efficient measurement of quantum devices". In: *npj Quantum Information* 7.1 (2021), pp. 1–9. DOI: 10.1038/s41534-021-00434-x.

[37] Timothy A Baart, Pieter T Eendebak, Christian Reichl, Werner Wegscheider, and Lieven MK Vandersypen. "Computer-automated tuning of semiconductor double quantum dots into the single-electron regime". In: *Applied Physics Letters* 108.21 (2016), p. 213104. DOI: 10.1063/1.4952624.

[38] Stefanie Czischek, Victor Yon, Marc-Antoine Genest, Marc-Antoine Roux, Sophie Rochette, Julien Camirand Lemyre, Mathieu Moras, Michel Pioro-Ladrière, Dominique Drouin, Yann Beilliard, et al. "Miniaturizing neural networks for charge state autotuning in quantum dots". In: *Machine Learning: Science and Technology* 3.1 (2021), p. 015001. DOI: 10.1088/2632-2153/ac34db.

[39] Renato Durrer, Benedikt Kratochwil, Jonne V Koski, Andreas J Landig, Christian Reichl, Werner Wegscheider, Thomas Ihn, and Eliska Greplova. "Automated tuning of double quantum dots into specific charge states using neural networks". In: *Physical Review Applied* 13.5 (2020), p. 054019. DOI: 10.1103/PhysRevApplied.13.054019.

[40] Maxime Lapointe-Major, Olivier Germain, J Camirand Lemyre, Dany Lachance-Quirion, Sophie Rochette, F Camirand Lemyre, and Michel Pioro-Ladrière. "Algorithm for automated tuning of a quantum dot into the single-electron regime". In: *Physical Review B* 102.8 (2020), p. 085301. DOI: 10.1103/PhysRevB.102.085301.

[41] NM van Esbroeck, DT Lennon, H Moon, V Nguyen, F Vigneau, LC Camenzind, L Yu, DM Zumbühl, GAD Briggs, Dino Sejdinovic, et al. "Quantum device fine-tuning using unsupervised embedding learning". In: *New Journal of Physics* 22.9 (2020), p. 095003. DOI: 10.1088/1367-2630/abb64c.

[42] Julian D Teske, Simon Sebastian Humpohl, René Otten, Patrick Bethke, Pascal Cerfontaine, Jonas Dedden, Arne Ludwig, Andreas D Wieck, and Hendrik Bluhm. "A machine learning approach for automated fine-tuning of semiconductor spin qubits". In: *Applied Physics Letters* 114.13 (2019), p. 133102. DOI: 10.1063/1.5088412.

[43] CJ Van Diepen, Pieter T Eendebak, Bruno T Buijtendorp, Uditendu Mukhopadhyay, Takafumi Fujita, Christian Reichl, Werner Wegscheider, and Lieven MK Vandersypen. "Automated tuning of inter-dot tunnel coupling in double quantum dots". In: *Applied Physics Letters* 113.3 (2018), p. 033101. DOI: 10.1063/1.5031034.

[44] Anasua Chatterjee, Fabio Ansaloni, Torbjørn Rasmussen, Bertram Brovang, Federico Fedele, Heorhii Bohuslavskyi, Oswin Krause, and Ferdinand Kuemmeth. "Autonomous Estimation of High-Dimensional Coulomb Diamonds from Sparse Measurements". In: *Physical Review Applied* 18.6 (2022), p. 064040. DOI: 10.1103/PhysRevApplied.18.064040.

[45] D. L. Craig et al. "Bridging the Reality Gap in Quantum Devices with Physics-Aware Machine Learning". In: *Physical Review X* 14 (1 Jan. 2024), p. 011001. DOI: 10.1103/PhysRevX.14.011001.

[46] Jonas Schuff, Dominic T Lennon, Simon Geyer, David L Craig, Federico Fedele, Florian Vigneau, Leon C Camenzind, Andreas V Kuhlmann, G Andrew D Briggs, Dominik M Zumbühl, et al. "Identifying Pauli spin blockade using deep learning". In: *Quantum* 7 (2023), p. 1077. DOI: 10.22331/q-2023-08-08-1077.

[47] Ronald Hanson, Leo P Kouwenhoven, Jason R Petta, Seigo Tarucha, and Lieven MK Vandersypen. "Spins in few-electron quantum dots". In: *Reviews of Modern Physics* 79.4 (2007), p. 1217. DOI: 10.1103/RevModPhys.79.1217.

[48] Leo P Kouwenhoven, Charles M Marcus, Paul L McEuen, Seigo Tarucha, Robert M Westervelt, and Ned S Wingreen. "Electron transport in quantum dots". In: *Mesoscopic electron transport* (1997), pp. 105–214.

[49] Simon Geyer. "Spin qubits in silicon fin field-effect transistors". PhD thesis. University of Basel, 2023. URL: https://inspirehep.net/literature/2702750.

[50] Floris A Zwanenburg, Andrew S Dzurak, Andrea Morello, Michelle Y Simmons, Lloyd CL Hollenberg, Gerhard Klimeck, Sven Rogge, Susan N Coppersmith, and Mark A Eriksson. "Silicon quantum electronics". In: *Reviews of Modern Physics* 85.3 (2013), pp. 961–1019. DOI: 10.1103/RevModPhys.85.961.

[51] MF Gonzalez-Zalba, S De Franceschi, E Charbon, Tristan Meunier, M Vinet, and AS Dzurak. "Scaling silicon-based quantum computing using CMOS technology". In: *Nature Electronics* 4.12 (2021), pp. 872–884. DOI: 10.1038/s41928-021-00681-y.

[52] W. G. van der Wiel, S. De Franceschi, J. M. Elzerman, T. Fujisawa, S. Tarucha, and L. P. Kouwenhoven. "Electron transport through double quantum dots". In: *Reviews of Modern Physics* 75 (1 Dec. 2002). DOI: 10.1103/RevModPhys.75.1.

[53] C W J Beenakker. "Theory of Coulomb-blockade oscillations in the conductance of a quantum dot". In: *Physical Review B* 44.4 (1991), pp. 1646–1656. DOI: 10.1103/PhysRevB.44.1646.

[54] Wilfred G. van der Wiel, S De Franceschi, J.M. Elzerman, T Fujisawa, S Tarucha, and L.P. Kouwenhoven. "Electron transport through double quantum dots". In: *Reviews of Modern Physics* 75.1 (2002), pp. 1–22. DOI: 10.1103/RevModPhys.75.1.

[55] Florian NM Froning. "Hole spin qubits in ge/si core/shell nanowires". PhD thesis. University of Basel, 2021. URL: https://edoc.unibas.ch/83296/.

[56] K. Ono, D. G. Austing, Y. Tokura, and S. Tarucha. "Current Rectification by Pauli Exclusion in a Weakly Coupled Double Quantum Dot System". In: *Science* 297.5585 (2002), pp. 1313–1317. DOI: 10.1126/science.1070958.

[57] J. R. Petta, A. C. Johnson, A. Yacoby, C. M. Marcus, M. P. Hanson, and A. C. Gossard. "Pulsed-gate measurements of the singlet-triplet relaxation time in a two-electron double quantum dot". In: *Physical Review B* 72.16 (2005), p. 161301. DOI: 10.1103/PhysRevB.72.161301. arXiv: 0412048 [cond-mat].

[58] Yongjie Hu, Ferdinand Kuemmeth, Charles M. Lieber, and Charles M. Marcus. "Hole spin relaxation in Ge-Si core-shell nanowire qubits". In: *Nature Nanotechnology* 7.1 (2012), pp. 47–50. DOI: 10.1038/nnano.2011.234.

[59] A. C. Johnson, J. R. Petta, J. M. Taylor, A. Yacoby, M. D. Lukin, C. M. Marcus, M. P. Hanson, and A. C. Gossard. "Triplet-singlet spin relaxation via nuclei in a double quantum dot". In: *Nature* 435.7044 (2005), pp. 925–928. DOI: 10.1038/nature03815.

[60] S. Nadj-Perge, S. M. Frolov, J. W.W. Van Tilburg, J. Danon, Yu V. Nazarov, R. Algra, E. P.A.M. Bakkers, and L. P. Kouwenhoven. "Disentangling the effects of spin-orbit and hyperfine interactions on spin blockade". In: *Physical Review B* 81.20 (2010), p. 201305. DOI: 10.1103/PhysRevB.81.201305. arXiv: 1002.2120.

[61] Jason R Petta, Alexander Comstock Johnson, Jacob M Taylor, Edward A Laird, Amir Yacoby, Mikhail D Lukin, Charles M Marcus, Micah P Hanson, and Arthur C Gossard. "Coherent manipulation of coupled electron spins in semiconductor quantum dots". In: *Science* 309.5744 (2005), pp. 2180–2184. DOI: 10.1126/science.1116955.

[62] F. H.L. Koppens, C. Buizert, K. J. Tielrooij, I. T. Vink, K. C. Nowack, T. Meunier, L. P. Kouwenhoven, and L. M.K. Vandersypen. "Driven coherent oscillations of a single electron spin in a quantum dot". In: *Nature* 442.7104 (2006), pp. 766–771. DOI: 10.1038/nature05065.

[63] K. C. Nowack, F. H. L. Koppens, Yu V. Nazarov, and L. M. K. Vandersypen. "Coherent Control of a Single Electron Spin with Electric Fields". In: *Science* 318.5855 (2007), pp. 1430–1433. DOI: 10.1126/science.1148092.

[64] 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". In: *Applied Physics Letters* 113 (7 2018), p. 073102.

[65] Leon C. Camenzind, Liuqi Yu, Peter Stano, Jeramy D. Zimmerman, Arthur C. Gossard, Daniel Loss, and Dominik M. Zumbühl. "Spectroscopy of Quantum Dot Orbitals with In-Plane Magnetic Fields". In: *Physical Review Letters* 122 (20 May 2019), p. 207701. DOI: 10.1103/PhysRevLett.122.207701.

[66] A. Zarassi, Z. Su, J. Danon, J. Schwenderling, M. Hocevar, B. M. Nguyen, J. Yoo, S. A. Dayeh, and S. M. Frolov. "Magnetic field evolution of spin blockade in Ge/Si nanowire double quantum dots". In: *Physical Review B* 95.15 (2017), pp. 1–7. DOI: 10.1103/PhysRevB.95.155416.

[67] Matthias Brauns, Joost Ridderbos, Ang Li, Erik P.A.M. Bakkers, Wilfred G. Van Der Wiel, and Floris A. Zwanenburg. "Anisotropic Pauli spin blockade in hole quantum dots". In: *Physical Review B* 94.4 (2016), pp. 1–8. DOI: 10.1103/PhysRevB.94.041411. arXiv: 1608.00111.

[68] Justin K. Perron, M. D. Stewart, and Neil M. Zimmerman. "A new regime of Pauli-spin blockade". In: *Journal of Applied Physics* 119.13 (2016), p. 134307. DOI: 10.1063/1.4945393. arXiv: 1507.07515.

[69] Daisy Q. Wang, Oleh Klochan, Jo Tzu Hung, Dimitrie Culcer, Ian Farrer, David A. Ritchie, and Alex R. Hamilton. "Anisotropic Pauli Spin Blockade of Holes in a GaAs Double Quantum Dot". In: *Nano Letters* 16.12 (2016), pp. 7685–7689. DOI: 10.1021/acs.nanolett.6b03752. arXiv: 1612.01062.

[70] J Danon and Yu V Nazarov. "Pauli spin blockade in the presence of strong spin-orbit coupling". In: *Physical Review B* 80.4 (2009), p. 041301.

[71] Leon C. Camenzind. "Spins and orbits in semiconductor quantum dots". PhD thesis. University of Basel, 2019.

[72] Yinan Fang, Pericles Philippopoulos, Dimitrie Culcer, WA Coish, and Stefano Chesi. "Recent advances in hole-spin qubits". In: *Materials for Quantum Technology* 3.1 (2023), p. 012003. DOI: 10.1088/2633-4356/acb87e.

[73] Will I L Lawrie. "Spin qubits in silicon and germanium". PhD thesis. Delft University of Technology, 2022. DOI: 10.4233/uuid:97c4ea24-9672-4e0b-b7a5-e3a48258c871.

[74] Anasua Chatterjee, Paul Stevenson, Silvano De Franceschi, Andrea Morello, Nathalie P de Leon, and Ferdinand Kuemmeth. "Semiconductor qubits in practice". In: *Nature Reviews Physics* 3.3 (2021), pp. 157–177. DOI: 10.1038/s42254-021-00283-9.

[75] Giordano Scappucci, Christoph Kloeffel, Floris A Zwanenburg, Daniel Loss, Maksym Myronov, Jian-Jun Zhang, Silvano De Franceschi, Georgios Katsaros, and Menno Veldhorst. "The germanium quantum information route". In: *Nature Reviews Materials* 6.10 (2021), pp. 926–943. DOI: 10.1038/s41578-020-00262-z.

[76] Christoph Kloeffel, Mircea Trif, and Daniel Loss. "Strong spin-orbit interaction and helical hole states in Ge/Si nanowires". In: *Physical Review B* 84.19 (2011), p. 195314. DOI: 10.1103/PhysRevB.84.195314.

[77] He Liu, Ke Wang, Fei Gao, Jin Leng, Yang Liu, Yu-Chen Zhou, Gang Cao, Ting Wang, Jianjun Zhang, Peihao Huang, et al. "Ultrafast and electrically tunable rabi frequency in a germanium hut wire hole spin qubit". In: *Nano Letters* 23.9 (2023), pp. 3810–3817. DOI: 10.1021/acs.nanolett.3c00213.

[78] Jun Yoneda, Kenta Takeda, Tomohiro Otsuka, Takashi Nakajima, Matthieu R Delbecq, Giles Allison, Takumu Honda, Tetsuo Kodera, Shunri Oda, Yusuke Hoshi, et al. "A quantum-dot spin qubit with coherence limited by charge noise and fidelity higher than 99.9%". In: *Nature nanotechnology* 13.2 (2018), pp. 102–106. DOI: 10.1038/s41565-017-0014-x.

[79] Nicolas Piot, B Brun, Vivien Schmitt, Simon Zihlmann, VP Michal, A Apra, JC Abadillo-Uriel, Xavier Jehl, Benoit Bertrand, H Niebojewski, et al. "A single hole spin with enhanced coherence in natural silicon". In: *Nature Nanotechnology* 17.10 (2022), pp. 1072–1077. DOI: 10.1038/s41565-022-01196-z.

[80] Miguel J Carballido. "Fixing the Holes in Holes: On Optimizing and Scaling the Operation of Hot Hole Spin Qubits". PhD thesis. University of Basel, 2024.

[81] Miguel J Carballido, Simon Svab, Rafael S Eggli, Taras Patlatiuk, Pierre Chevalier Kwon, Jonas Schuff, Rahel M Kaiser, Leon C Camenzind, Ang Li, Natalia Ares, et al. "Compromise-free scaling of qubit speed and coherence". In: *Nature Communications* 16.1 (2025), p. 7616. DOI: 10.1038/s41467-025-62614-z. arXiv: 2402.07313.

[82] Winkler Roland. "Spin-orbit coupling effects in two-dimensional electron and hole systems". In: *Springer Tracts in Modern Physiscs: Springer, Berlin, Heidelberg* 191 (2003).

[83] Ryan M Jock, N Tobias Jacobson, Patrick Harvey-Collard, Andrew M Mounce, Vanita Srinivasa, Dan R Ward, John Anderson, Ron Manginell, Joel R Wendt, Martin Rudolph, et al. "A silicon metal-oxide-semiconductor electron spin-orbit qubit". In: *Nature Communications* 9.1 (2018), p. 1768. DOI: 10.1038/s41467-018-04200-0.

[84] Ronald Hanson and David D. Awschalom. "Coherent manipulation of single spins in semiconductors". In: *Nature* 453.7198 (June 2008), pp. 1043–1049. DOI: 10.1038/nature07129.

[85] Zhanning Wang, Elizabeth Marcellina, Alex. R. Hamilton, James H. Cullen, Sven Rogge, Joe Salfi, and Dimitrie Culcer. "Optimal operation points for ultrafast, highly coherent Ge hole spin-orbit qubits". In: *npj Quantum Information* 7.1 (Apr. 2021). DOI: 10.1038/s41534-021-00386-2.

[86] K. D. Petersson, J. R. Petta, H. Lu, and A. C. Gossard. "Quantum Coherence in a One-Electron Semiconductor Charge Qubit". In: *Physical Review Letters* 105.24 (Dec. 2010). DOI: 10.1103/physrevlett.105.246804.

[87] Christopher C Gerry and Peter L Knight. *Introductory quantum optics*. Cambridge University Press, 2023.

[88] J P Dehollain, J J Pla, E Siew, K Y Tan, A S Dzurak, and A Morello. "Nanoscale broadband transmission lines for spin qubit control". In: *Nanotechnology* 24.1 (Dec. 2012), p. 015202. DOI: 10.1088/0957-4484/24/1/015202.

[89] Vitaly N. Golovach, Massoud Borhani, and Daniel Loss. "Electric-dipole-induced spin resonance in quantum dots". In: *Physical Review B* 74.16 (Oct. 2006). DOI: 10.1103/physrevb.74.165319.

[90] James Moor. "Artificial Intelligence Conference: The Next Fifty Years". In: *AI Magazine* 27.4 (2006), pp. 87–91.

[91] G. Cybenko. "Approximation by Superpositions of a Sigmoidal Function". In: *Mathematics of control, signals and systems* 2.4 (1989), pp. 303–314. DOI: 10.1007/BF02551274.

[92] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014). DOI: 10.48550/arXiv.1412.6980.

[93] Simon Rogers and Mark Girolami. *A first course in machine learning*. Chapman and Hall/CRC, 2016.

[94] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[95] Peter I Frazier. "A tutorial on Bayesian optimization". In: *arXiv preprint arXiv:1807.02811* (2018). DOI: 10.48550/arXiv.1807.02811.

[96] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. "Practical Bayesian Optimization of Machine Learning Algorithms". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf.

[97] Joseph Hickie, Barnaby Van Straaten, Federico Fedele, Daniel Jirovec, Andrea Ballabio, Daniel Chrastina, Giovanni Isella, Georgios Katsaros, and Natalia Ares. "Automated long-range compensation of an rf quantum dot sensor". In: *Physical Review Applied* 22.6 (2024), p. 064026. DOI: 10.1103/PhysRevApplied.22.064026. arXiv: 2310.02135.

[98] Natalia Ares. "Machine learning as an enabler of qubit scalability". In: *Nature Reviews Materials* 6.10 (2021), pp. 870–871. DOI: 10.1038/s41578-021-00321-z.

[99] Justyna P Zwolak and Jacob M Taylor. "Colloquium: Advances in automation of quantum dot devices control". In: *Reviews of Modern Physics* 95.1 (2023), p. 011006. DOI: 10.1103/RevModPhys.95.011006.

[100] Hans-Christian Ruiz Euler, Marcus N Boon, Jochem T Wildeboer, Bram van de Ven, Tao Chen, Hajo Broersma, Peter A Bobbert, and Wilfred G van der Wiel. "A deep-learning approach to realizing functionality in nanoelectronic devices". In: *Nature Nanotechnology* 15.12 (2020), pp. 992–998. DOI: 10.1038/s41565-020-00779-y.

[101] AR Mills, MM Feldman, Cara Monical, PJ Lewis, KW Larson, AM Mounce, and Jason R Petta. "Computer-automated tuning procedures for semiconductor quantum dot arrays". In: *Applied Physics Letters* 115.11 (2019), p. 113501. DOI: 10.1063/1.5121444.

[102] Joshua Ziegler, Florian Luthi, Mick Ramsey, Felix Borjans, Guoji Zheng, and Justyna P Zwolak. "Automated extraction of capacitive coupling for quantum dot systems". In: *Physical Review Applied* 19.5 (2023), p. 054077. DOI: 10.1103/PhysRevApplied.19.054077.

[103] Giovanni A Oakes, Jingyu Duan, John JL Morton, Alpha Lee, Charles G Smith, and M Zalba. "Automatic virtual voltage extraction of a 2xN array of quantum dots with machine learning". In: *arXiv preprint arXiv:2012.03685* (2020). DOI: 10.48550/arXiv.2012.03685.

[104] Fabian Hader, Jan Vogelbruch, Simon Humpohl, Tobias Hangleiter, Chimezie Eguzo, Stefan Heinen, Stefanie Meyer, and Stefan van Waasen. "On Noise-Sensitive Automatic Tuning of Gate-Defined Sensor Dots". In: *IEEE Transactions on Quantum Engineering* 4 (2023), pp. 1–18. DOI: 10.1109/TQE.2023.3255743.

[105] Joshua Ziegler, Florian Luthi, Mick Ramsey, Felix Borjans, Guoji Zheng, and Justyna P Zwolak. "Tuning arrays with rays: Physics-informed tuning of quantum dot charge states". In: *Physical Review Applied* 20.3 (2023), p. 034067. DOI: 10.1103/PhysRevApplied.20.034067.

[106] Tim Botzem, Michael D Shulman, Sandra Foletti, Shannon P Harvey, Oliver E Dial, Patrick Bethke, Pascal Cerfontaine, Robert PG McNeil, Diana Mahalu, Vladimir Umansky, et al. "Tuning methods for semiconductor spin qubits". In: *Physical Review Applied* 10.5 (2018), p. 054026. DOI: 10.1103/PhysRevApplied.10.054026.

[107] Daniel Schug, Tyler J Kovach, MA Wolfe, Jared Benson, Sanghyeok Park, JP Dodson, J Corrigan, MA Eriksson, and Justyna P Zwolak. "Explainable Classification Techniques for Quantum Dot Device Measurements". In: *arXiv preprint arXiv:2402.13699* (2024). DOI: 10.48550/arXiv.2402.13699.

[108] Brian Weber and Justyna P Zwolak. "QDA^2: A principled approach to automatically annotating charge stability diagrams". In: *arXiv preprint arXiv:2312.11206* (2023). DOI: 10.48550/arXiv.2312.11206.

[109] Yui Muto, Takumi Nakaso, Motoya Shinozaki, Takumi Aizawa, Takahito Kitada, Takashi Nakajima, Matthieu R Delbecq, Jun Yoneda, Kenta Takeda, Akito Noiri, et al. "Visual explanations of machine learning model estimating charge states in quantum dots". In: *APL Machine Learning* 2.2 (2024). DOI: 10.1063/5.0193621.

[110] Jana Darulová, Matthias Troyer, and Maja C Cassidy. "Evaluation of synthetic and experimental training data in supervised machine learning applied to charge-state detection of quantum dots". In: *Machine Learning: Science and Technology* (2021). DOI: 10.1088/2632-2153/ac104c.

[111] Joshua Ziegler, Thomas McJunkin, ES Joseph, Sandesh S Kalantre, Benjamin Harpt, DE Savage, Mark G Lagally, MA Eriksson, Jacob M Taylor, and Justyna P Zwolak. "Toward robust autotuning of noisy quantum dot devices". In: *Physical Review Applied* 17.2 (2022), p. 024069. DOI: 10.1103/PhysRevApplied.17.024069.

[112] Pascal Cerfontaine, Tim Botzem, Julian Ritzmann, Simon Sebastian Humpohl, Arne Ludwig, Dieter Schuh, Dominique Bougeard, Andreas D Wieck, and Hendrik Bluhm. "Closed-loop control of a GaAs-based singlet-triplet spin qubit with 99.5% gate fidelity and low leakage". In: *Nature Communications* 11.1 (2020), pp. 1–6. DOI: 10.1038/s41467-020-17865-3.

[113] Fabrizio Berritta, Torbjørn Rasmussen, Jan A Krzywda, Joost van der Heijden, Federico Fedele, Saeed Fallahi, Geoffrey C Gardner, Michael J Manfra, Evert van Nieuwenburg, Jeroen Danon, et al. "Real-time two-axis control of a spin qubit". In: *Nature Communications* 15.1 (2024), p. 1676. DOI: 10.1038/s41467-024-45857-0.

[114] Julian Kelly, Peter O'Malley, Matthew Neeley, Hartmut Neven, and John M Martinis. "Physical qubit calibration on a directed acyclic graph". In: *arXiv preprint arXiv:1803.03226* (2018). DOI: 10.48550/arXiv.1803.03226.

[115] Christian Volk, Anne-Marije J Zwerver, Uditendu Mukhopadhyay, Pieter T Eendebak, Cornelis Jacobus van Diepen, Juan Pablo Dehollain, Toivo Hensgens, Takafumi Fujita, Christian Reichl, Werner Wegscheider, et al. "Loading a quantum-dot based "Qubyte" register". In: *npj Quantum Information* 5.1 (2019), pp. 1–8. DOI: 10.1038/s41534-019-0146-y.

[116] Hanwei Liu, Baochuan Wang, Ning Wang, Zhonghai Sun, Huili Yin, Haiou Li, Gang Cao, and Guoping Guo. "An automated approach for consecutive tuning of quantum dot arrays". In: *Applied Physics Letters* 121.8 (2022). DOI: 10.1063/5.0111128.

[117] Marcel Meyer, Corentin Déprez, Ilja N Meijer, Florian K Unseld, Saurabh Karwal, Amir Sammak, Giordano Scappucci, Lieven MK Vandersypen, and Menno Veldhorst. "Single-electron occupation in quantum dot arrays at selectable plunger gate voltage". In: *Nano Letters* 23.24 (2023), pp. 11593–11600. DOI: 10.1021/acs.nanolett.3c03349.

[118] Paul V Klimov, Julian Kelly, John M Martinis, and Hartmut Neven. "The snake optimizer for learning quantum processor control parameters". In: *arXiv preprint arXiv:2006.04594* (2020). DOI: 10.48550/arXiv.2006.04594.

[119] Paul V Klimov, Andreas Bengtsson, Chris Quintana, Alexandre Bourassa, Sabrina Hong, Andrew Dunsworth, Kevin J Satzinger, William P Livingston, Volodymyr Sivak, Murphy Yuezhen Niu, et al. "Optimizing quantum gates towards the scale of logical qubits". In: *Nature Communications* 15.1 (2024), p. 2442. DOI: 10.1038/s41467-024-46623-y.

[120] Luca Petit, Maximilian Russ, Gertjan HGJ Eenink, William IL Lawrie, James S Clarke, Lieven MK Vandersypen, and Menno Veldhorst. "Design and integration of single-qubit rotations and two-qubit gates in silicon above one Kelvin". In: *Communications Materials* 3.1 (2022), p. 82. DOI: 10.1038/s43246-022-00304-9.

[121] Chih Heng Yang, RCC Leon, JCC Hwang, Andre Saraiva, Tuomo Tanttu, Wister Huang, J Camirand Lemyre, Kok Wai Chan, KY Tan, Fay E Hudson, et al. "Operation of a silicon quantum processor unit cell above one kelvin". In: *Nature* 580.7803 (2020), pp. 350–354. DOI: 10.1038/s41586-020-2171-6.

[122] Yuta Matsumoto, Takafumi Fujita, Arne Ludwig, Andreas D Wieck, Kazunori Komatani, and Akira Oiwa. "Noise-robust classification of single-shot electron spin readouts using a deep neural network". In: *npj Quantum Information* 7.1 (2021), pp. 1–7. DOI: 10.1038/s41534-021-00470-7.

[123] Simon Geyer, Leon C Camenzind, Lukas Czornomaz, Veeresh Deshpande, Andreas Fuhrer, Richard J Warburton, Dominik M Zumbühl, and Andreas V Kuhlmann. "Self-aligned gates for scalable silicon quantum computing". In: *Applied Physics Letters* 118.10 (2021), p. 104004. DOI: 10.1063/5.0036520.

[124] FNM Froning, MJ Rančić, B Hetényi, S Bosco, MK Rehmann, Ang Li, Erik PAM Bakkers, Floris Arnoud Zwanenburg, Daniel Loss, DM Zumbühl, et al. "Strong spin-orbit interaction and g-factor renormalization of hole spins in Ge/Si nanowire quantum dots". In: *Physical Review Research* 3 (1 Jan. 2021), p. 013081. DOI: 10.1103/PhysRevResearch.3.013081.

[125] TH Stoof and Yu V Nazarov. "Time-dependent resonant tunneling via two discrete states". In: *Physical Review B* 53.3 (1996), p. 1050. DOI: 10.1103/PhysRevB.53.1050.

[126] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[127] Ross Wightman, Hugo Touvron, and Hervé Jégou. "ResNet strikes back: An improved training procedure in timm". In: *arXiv preprint arXiv:2110.00476* (2021). DOI: 10.48550/arXiv.2110.00476.

[128] Irwan Bello, William Fedus, Xianzhi Du, Ekin Dogus Cubuk, Aravind Srinivas, Tsung-Yi Lin, Jonathon Shlens, and Barret Zoph. "Revisiting ResNets: Improved Training and Scaling Strategies". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 22614–22627. DOI: 10.48550/arXiv.2103.07579.

[129] Jonas Schuff, Miguel J Carballido, Madeleine Kotzagiannidis, Juan Carlos Calvo, Marco Caselli, Jacob Rawling, David L Craig, Barnaby van Straaten, Brandon Severin, Federico Fedele, et al. "Fully autonomous tuning of a spin qubit". In: *Nature Electronics* (2026), pp. 1–10. DOI: 10.1038/s41928-025-01562-4. arXiv: 2402.03931 [cond-mat.mes-hall].

[130] JH Ungerer, P Chevalier Kwon, T Patlatiuk, J Ridderbos, A Kononov, D Sarmah, EPAM Bakkers, D Zumbühl, and C Schönenberger. "Charge-sensing of a Ge/Si core/shell nanowire double quantum dot using a high-impedance superconducting resonator". In: *Materials for Quantum Technology* 3.3 (2023), p. 031001. DOI: 10.1088/2633-4356/ace2a6.

[131] Rafael S Eggli, Simon Svab, Taras Patlatiuk, Dominique A Trüssel, Miguel J Carballido, Pierre Chevalier Kwon, Simon Geyer, Ang Li, Erik PAM Bakkers, Andreas V Kuhlmann, et al. "Cryogenic hyperabrupt strontium titanate varactors for sensitive reflectometry of quantum dots". In: *Physical Review Applied* 20.5 (2023), p. 054056. DOI: 10.1103/PhysRevApplied.20.054056.

[132] 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". In: *Nature Communications* 7.1 (2016), p. 13575. DOI: 10.1038/ncomms13575.

[133] NW Hendrickx, WIL Lawrie, L Petit, A Sammak, G Scappucci, and M Veldhorst. "A single-hole spin qubit". In: *Nature Communications* 11.1 (2020), p. 3478. DOI: 10.1038/s41467-020-17211-7.

[134] Florian NM Froning, Leon C Camenzind, Orson AH van der Molen, Ang Li, Erik PAM Bakkers, Dominik M Zumbühl, and Floris R Braakman. "Ultrafast hole spin qubit with gate-tunable spin–orbit switch functionality". In: *Nature Nanotechnology* 16.3 (2021), pp. 308–312. DOI: 10.1038/s41565-020-00828-6.

[135] Madeleine Kotzagiannidis, Jonas Schuff, and Nathan Korda. "The Automated Bias Triangle Feature Extraction Framework". In: *arXiv preprint arXiv:2312.03110* (2023). DOI: 10.48550/arXiv.2312.03110.

[136] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.

[137] Miguel J Carballido et al. "Conditional Operations of Hot Hole Spin Qubits above 1K". In: *In preparation* (2024).

[138] Jarryd J Pla, Kuan Y Tan, Juan P Dehollain, Wee H Lim, John JL Morton, Floris A Zwanenburg, David N Jamieson, Andrew S Dzurak, and Andrea Morello. "High-fidelity readout and control of a nuclear spin qubit in silicon". In: *Nature* 496.7445 (2013), pp. 334–338. DOI: /10.1038/nature12011.

[139] Tim Head, Manoj Kumar, Holger Nahrstaedt, Gilles Louppe, and Iaroslav Shcherbatyi. *scikit-optimize*. DOI: 10.5281/zenodo.10804382.

[140] Muhammad Junaid Arshad, Christiaan Bekker, Ben Haylock, Krzysztof Skrzypczak, Daniel White, Benjamin Griffiths, Joe Gore, Gavin W Morley, Patrick Salter, Jason Smith, et al. "Real-time adaptive estimation of decoherence timescales for a single qubit". In: *Physical Review Applied* 21.2 (2024), p. 024026. DOI: 10.1103/PhysRevApplied.21.024026.

[141] Florian Vigneau, Federico Fedele, Anasua Chatterjee, David Reilly, Ferdinand Kuemmeth, M Fernando Gonzalez-Zalba, Edward Laird, and Natalia Ares. "Probing quantum devices with radio-frequency reflectometry". In: *Applied Physics Reviews* 10.2 (2023). DOI: 10.1063/5.0088229.

[142] DT Lennon, H Moon, LC Camenzind, Liuqi Yu, DM Zumbühl, GAD Briggs, MA Osborne, EA Laird, and N Ares. "Efficiently measuring a quantum device using machine learning". In: *npj Quantum Information* 5.1 (2019), pp. 1–8. DOI: 10.1038/s41534-019-0193-4.

[143] RJ Schoelkopf, P Wahlgren, AA Kozhevnikov, P Delsing, and DE Prober. "The radio-frequency single-electron transistor (RF-SET): A fast and ultrasensitive electrometer". In: *Science* 280.5367 (1998), pp. 1238–1242. DOI: 10.1126/science.280.5367.1238.

[144] Adam Paszke, Sam Gross, Adam Lerer, Zeming Lin, James Bradbury, Gregory Chanan, Trevor Killeen, Natalia Gimelshein, Luca Antiga, Alban Desmaison, et al. *torchvision*. https://github.com/pytorch/vision. 2017.

[145] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[146] Adam Paszke et al. "PyTorch: An Imperative Style, High-Performance Deep Learning Library". In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[147] Theodosios Pavlidis. *Algorithms for graphics and image processing*. Springer Science & Business Media, 2012.

[148] Copenhagen / Delft / Sydney / Microsoft quantum computing consortium. *QCoDeS*. Version 0.42.1. Dec. 26, 2023. URL: https://qcodes.github.io/Qcodes/.

[149] T-K Hsiao, Cornelis J van Diepen, Uditendu Mukhopadhyay, Christian Reichl, Werner Wegscheider, and Lieven MK Vandersypen. "Efficient orthogonal control of tunnel couplings in a quantum dot array". In: *Physical Review Applied* 13.5 (2020), p. 054018. DOI: 10.1103/PhysRevApplied.13.054018.