Fast tuning, simulation and characterisation of spin qubits devices using machine learning

Barnaby van Straaten

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Hilary 2024

Abstract

Semiconducting spin quantum devices present a promising avenue for advancing noisy intermediate-scale quantum (NISQ) computers. However, the complexities involved in their tuning and characterisation currently hinder their scaling. This thesis addresses these challenges by harnessing machine-learning-based approaches.

Firstly, I used Gaussian processes to develop a fully automatic tuning algorithm that exclusively used radio-frequency measurements taken on millisecond timescales. Secondly, I employed Bayesian optimisation for charge sensor optimisation and automation. Both algorithms were deployed on depletion mode hole-based Ge/SiGe devices; however, they could easily be applied to other architectures. These automated approaches considerably outperformed manual tuning and represent vital building blocks for algorithms to tune larger quantum dot arrays.

Next, I developed algorithms to find the ground state charge configuration in the constant capacitance model, which was orders of magnitude faster than its predecessors, permitting the simulation of much larger arrays of sixteen dots. This open-source package can be used to gain insights into the charge stability diagrams of large quantum dot arrays and facilitates further tuning algorithm development.

Lastly, I present a novel application of hidden Markov models to identify and separate the state preparation, measurement and spin-flip errors presented by a pair of hot Loss Divincenzo qubits in a Si-MOS device. This approach not only unravels these error sources but also sheds light on their relationship with various parameters, such as electron temperature, thus contributing a vital understanding to the field of quantum computing.

In dedication to family and friends...

Acknowledgements

Natalia, I extend my heartfelt gratitude for your exceptional supervision. Countless achievements and results would have remained elusive without your unwavering dedication. Your contribution spans the full spectrum from nurturing a well-funded laboratory teeming with skilled individuals and steering research towards fruitful directions to even joining me in late Friday evenings to rewire fridges and meticulously catching my paper typos. Thank you so much.

Fede, thank you for turning me from a theorist into an experimentalist. I owe you everything I have learned. You were the person I ran to when things went wrong. **Joe**, thank you for initiating and consistently stimulating my newfound obsession with bikes. I have thoroughly enjoyed the bike rides and tinkering with bikes as a means of procrastination in the office. People often complained that the office was more like a bike shop than an office, which was probably true. In addition, thank you for your blazing typing speed. If I had not seen what was possible, I would have never thrown off my qwerty bad habits and embraced the Dvorak layout. After years of practice, I am about half your speed. **Seb**, thank you for coming up with the crazy idea of doing an Ironman. It took all my self-restraint to limit myself to just the half whilst you monstered the full. I look forward to the next adventure. **David**, you were the best of our five - unbroken and unwavering in your dedication to theory, whereas I was led astray. Never change; QuantrolOx is lucky to have you. **Brandon**, thank you for being the multipotentialite you are. You always told me about something new: crypto, surfing, start-up investment rounds, or just hyping people up. I wish you all the best in whatever you do next. **Kush**, you grumpy, grumpy, beautiful man. I always enjoyed chatting with you, learning from you and sometimes winding you up. I wish you well in your science to come. **Jonas**, my German friend, you were the first to develop the

full tuning algorithm despite so many members of the group racing for that accolade. Congratulations. What's next?

Sara, thank you for being my anchor through the highs and lows of this past year. Your love, patience, and constant encouragement have kept me steady. You've reminded me to laugh, to rest, and to keep things in perspective. I'm endlessly grateful for your presence in my life - and for being truly wonderful in every way.

OUSSC in particular **Jhanna, David, Manon, Dom, Del, Katy, Eva and Oliver**, thank you for putting up with my fumblings and flailings whilst I learned to be president. It was certainly not my intention to become president; however, I am glad I did, as it provided many unique experiences. From nervously checking high-altitude weather forecasts to see how badly the heat wave was melting a glacier. To the team finding out the Queen was dead whilst all naked in a steam room. The only thing I hated was giving speeches. Sacrificing my collarbone to avoid the Blues dinner speech was a fair trade in my book. **148 Walton Street**: Federio, Charlotte, Yao and Alonso, thank you for being the best housemates in Oxford. We did so much together, but then Covid hit, and we endured lockdown together. I could not have asked for a more sociable group to be isolated with.

Mother, thank you for getting me here and for the endless, generous acts that improve my life. Forever grateful. **Father**, thank you for your boundless help, stimulating conversations and rogue things you have introduced me to. **Verity**, thank you for always entertaining me, being great and keeping me on my toes. **Piers**, thank you for keeping me grounded and pushing me to go faster. **Grandfather Richard**, thank you for provoking my passion for science and engineering (and the bread). **Granny Backhouse**, thank you for being an immense source of positivity, fun and good humour (and the strawberry ice cream). **Susan Summers**, thank you for believing in me from

an early age, even when reading did not come quickly.

Contents

Chapter 1

Introduction

Quantum computing represents a paradigm shift in information processing, poised to revolutionise our approach to solving complex problems that have long stymied classical computers. Unlike classical computers, which rely on bits as the fundamental unit of information, quantum computers harness the principles of quantum mechanics to process information using quantum bits or qubits. A single qubit can be in a superposition such that its state can be written as

$$|\psi_1\rangle = \alpha_0 |0\rangle + \alpha_1 |1\rangle , \tag{1.1}$$

where α_0 and α_1 are complex numbers normalised such that $|\alpha_0|^2 + |\alpha_1|^2 = 1$. However, the true power of quantum computing comes from the phenomenon of entanglement between more than one qubit. The state of n qubits can be written as

$$|\psi_n\rangle = \sum_{x \in \{0,1\}^n} \alpha_x |x\rangle , \tag{1.2}$$

where the sum runs over all 2^n possible n-bit strings. An exponential number of complex amplitudes are required to encode the state. Quantum algorithms could leverage this exponential state space to

outperform classical algorithms across diverse fields, from cryptography and optimisation to drug discovery and materials science [1].

However, all these applications are theoretical until we can build a quantum computer. This, it turns out, is the major rub of the problem. Paul Benioff, Yuri Manin and Richard Feynman made the first proposals for quantum computers in the early 1980s [2]. However, it was only in the late 1990s that quantum computing began to transition from a purely theoretical concept to a tangible field of research and development. Recent years have witnessed remarkable progress, with advances in quantum hardware putting us in the noisy intermediate scale quantum (NISQ) era, characterised by quantum processors containing up to 1000 qubits[1], which are neither advanced enough nor large enough for fault tolerance. These processors, which are sensitive to their environment (noisy) and prone to quantum decoherence, are only just becoming capable of continuous quantum error correction [3, 4].

1.1 Types of qubits

There are many different physical implementations of qubits, each with advantages and challenges. Here are some of the main types of qubits:

- *Superconducting* qubits are one of the leading candidates for building quantum computers. They are approximately 100 µm scale circuits made from superconducting materials, which exhibit zero electrical resistance at cryogenic temperatures [5]. Superconducting qubits are relatively easy to fabricate and manipulate using microwave pulses. Examples include transmon qubits and flux qubits [6].

- *Trapped ion* qubits, where individual ions are trapped and manipulated using electromagnetic fields. The internal energy levels of ions serve as qubit states. Trapped ion qubits have

[1] At the time of writing IBM announced their Condor chip with 1121 qubits. However, in this announcement, they also said this might be their biggest chip for a while as they will now focus on qubit quality, not quantity.

exceptionally long coherence times [7].

- *Photonic qubits* where photons encode information qubits and linear optical elements and optimal instruments (such as reciprocal mirrors and waveplates) are used to enact gates. Photon detectors are used for measurement. Photonic qubits are a primary component of quantum communication systems and can be used for quantum key distribution [8].

- *Semiconductor qubits*, where qubits are implemented using electron or hole spins in semiconductor materials. Examples include quantum dots and donor qubits in silicon. Semiconductor qubits offer the advantage of compatibility with existing semiconductor fabrication techniques [9].

- *Rydberg atom* qubits, where highly-excited electronic states are used to create qubits. These qubits are manipulated using laser pulses and are prized for their long coherence times [10].

With all these implementations, what makes a "good" qubit?

1.2 The DiVincenzo criteria

According to DiVincenzo's criteria, constructing a quantum computer requires that the system meet five conditions [11]:

1. *A scalable physical system with well-characterised qubit.* A well-characterised qubit is a two-level quantum system split from other energy states by a reasonable energy gap. At its most idealistic, the scalability component of this criterion states that a unit cell should exist where scaling from one to many introduces no emergent problems. The possible difficulties encountered when trying to scale are myriad, including the explosion in the number of control lines, crosstalk, heat production and unit cell size. As such, the scalability is poorly defined, and virtually every qubit can be claimed to be scalable just as quickly as its critics argue it is not.

2. *The ability to initialise the state of the qubits to a simple fiducial state.* This fiducial state is a starting point for the gate-based qubit operations required by quantum algorithms. One convenient choice of this fiducial state is where each qubit is in the ground state. One way to obtain this state is to allow the qubits to relax. However, per the next criterion, "good" qubits should relax slowly, making this passive reset rather time-consuming. As such, active reset protocols are becoming necessary [12, 13].

3. *Long relevant decoherence times.* Qubits, coupled with the environment, naturally decohere and lose their quantum information. Two timescales can characterise qubit decoherence: the relaxation time, T_1 and the dephasing time, T_2. Both forms of decoherence are problematic for long-running; therefore, "good" qubits should have long T_1 and T_2 relative to the qubit operations[2].

4. *A "universal" set of quantum gates.* To fully access the power of quantum computers, we must be able to access the full state space alluded to in Eq. (1.2). Doing this requires a "universal" gate set, which is a small set of quantum gates which, when combined, can arbitrarily approximate any operation and thus take us to any state from the fiducial state. An example of a universal gate set is the set of single qubit rotations and the CNOT gate $\{R_x(\theta_x), R_y(\theta_y)R_z(\theta_z), \text{CNOT}\}$ [14].

5. *A qubit-specific measurement capability.* To extract information from a quantum computer, we must be able to measure the qubits, collapsing their wavefunction. Measurement is often non-trivial and carries high overhead concerning control electronics and wiring.

[2]If we are getting specific the dephasing time is parameterised by the pure dephasing time, T_ϕ. While T_2 is an aggregate dephasing time due to pure dephasing and relaxation, so that $1/T_2 = 1/T_\phi + 1/2T_1$

1.3 Semiconducting qubits and the tuning problem

In this thesis, we will focus on semiconducting qubits and the scalability component of the first criterion, particularly the tuning problem. Semiconducting spin qubit devices have a fundamental issue: they do not immediately host qubits when cooled down. There is a complex tuning process before obtaining functional qubits. Firstly, we have to *"define"* the qubit by tuning the electrostatic confinement. Then we have to *"find"* the qubit by finding the readout point and frequency. And finally, we have to *"refine"* the qubit by optimising all of the above and calibrating a gate set. And that is just one qubit...

While manual tuning may be possible for one individual unit cell, it does not scale. Therefore, unit cells can only satisfy the first of DiVincenzo's criteria if an automated algorithm exists to tune them. The work in this thesis tackles aspects of automated routines to tune and characterise semiconducting qubits as the first steps towards the automated tuning of semiconducting spin unit cells.

1.4 Thesis outline

The focus of this thesis is the automation of this extensive tune-up procedure in semiconducting quantum devices, in particular, accelerating it by incorporating high bandwidth fast radio-frequency reflectometry measurements. The structure of what is to come is as follows:

- In Chapter 2, we review the relevant literature on semiconducting devices, qubits and the efforts to tune them (automated or not).
- In Chapter 3, we will discuss experimental and computational methods used in later chapters. The experimental component of this chapter will discuss the techniques used to operate semiconducting devices, including cryogenics, radio-frequency measurements and experimen-

tal control software. Then, we will discuss the machine-learning methods and algorithms used in the tuning effort. In particular, I will introduce Gaussian processes, Bayesian optimisation, and hidden Markov models.

- Chapter 4 is our first results chapter, in which I present an algorithm to tune a semiconducting device using exclusively ohmic radio-frequency measurements.

- Chapter 5 develops an algorithm capable of fully automating the compensation of a charge sensor in a quantum dot device.

- In Chapter 6, we take a step back for physical devices and develop a constant capacitance model for large quantum dot arrays. The motivation for creating this model is to prototype tuning algorithms for larger devices without the need for an experiment. I present my algorithm for computing the ground state charge configuration, which can simulate considerably larger arrays of quantum dots than previous implementations.

- In Chapter 7, I use hidden Markov models to disentangle the state preparation, measurement and spin-flip errors present in a pair of hot Loss-DiVincenzo qubits and then explore whether hidden Markov models can be used in real-time on FPGA for qubit active reset.

- In the eighth and final chapter, I summarise and discuss the outlook for this work.

Chapter 2

Context

2.1 Semiconductor spin qubits

2.1.1 GaAs devices and the early days

The first demonstrations of semiconducting qubits were hosted in GaAs [15]. These devices were fabricated in a heterostructure composed of GaAs and AlGaAs, doped with Si to introduce free electrons. Free electrons aggregate at the AlGaAs/GaAs interface, forming a two-dimensional electron gas (2DEG) (Fig. 1 (b)). By applying negative voltages to the gate electrodes patterned on the sample, the electrons in the 2DEG can be further confined to a small island of charge called a quantum dot.

The early adoption of GaAs was because its electrons have a small effective mass, meaning quantum dots are typically formed over micron lengthscales [9, 15]. This reduces the precision requirements during fabrication, meaning fabricating these devices is possible in university cleanrooms. As a result, GaAs hosted the first demonstrations of one, two and four qubits [15–17] in semiconductor devices, as well as demonstrations of single electron confinement in 2×2, 8×1 and 3×3 arrays [17–

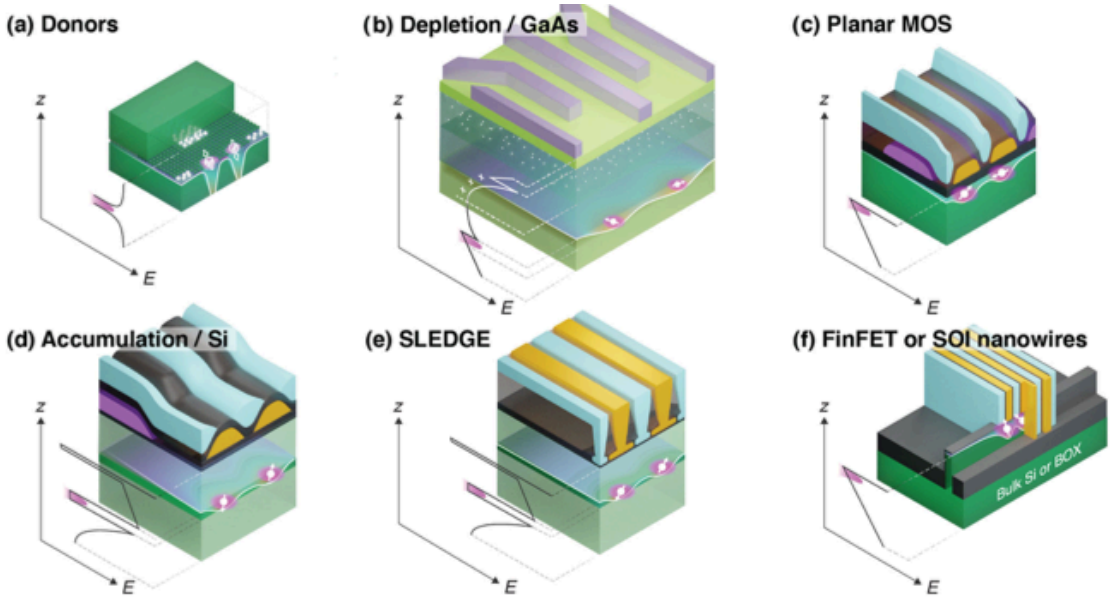


Figure 1: Device designs commonly used to confine electrons. Vertical confinement is illustrated through plots of E, the electrostatic energy of an electron. (a) Donor electrons are confined by the positive potential of a donor atom and manipulated by gates. (b) Depletion mode GaAs device, where an AlGaAs spacer layer is doped with Si to provide free electrons. These electrons populate a 2DEG formed at the AlGaAs/GaAs interface. Then, the 2DEG is depleted by applying negative voltages to gates at the top of the heterostructure. (c) A planar metal oxide semiconductor (MOS) device where electrons are accumulated on the Si/SiO_2 interface by applying positive voltages. (d) An accumulation Si/SiGe heterostructure-based device where electrons are accumulated in a quantum well, obtained by doping with Germanium. (e) Single-layer etch-defined gate electrode (SLEDGE) devices utilising a single layer of gates patterned on the top surface of a Si/SiGe heterostructure. The gates are contacted from above using vias, which allows gate wiring to fan out away from the device's active area in multiple planes. (f) A fin-field effect transistor (fin-FET) uses etching to create a quasi-1d channel and wrap-around gates to generate the electric confinement. Figure adapted from Ref. [9].

19].

However, whilst GaAs hosts qubits, it does not host "good" qubits as per DiVincenzo's third criterion. This is because both gallium and arsenic carry nuclear spins[1]. As a result, the heterostructure produces a parasitic nuclear spin bath, limiting dephasing coherence times, T_2, to the order of nanoseconds [9, 20].

Therefore, over recent years, there has been a shift to silicon-based devices. In the subsequent sections, I will briefly delve into the rationale behind this shift and explore the primary candidates for quantum computing within the realm of silicon-based qubits.

2.1.2 Silicon devices

Silicon is now the preferred material for semiconductor qubits for two primary reasons. Firstly, silicon stands out as a highly scalable material for hosting semiconducting quantum computing devices. This scalability is closely tied to the similarity between silicon qubit devices and classical transistors. Ambitious proposals have surfaced, outlining architectures capable of supporting millions of qubits [21, 22] and incorporating error-correcting codes into the design [23, 24].

Secondly, silicon offers the prospect of an exceptionally clean magnetic environment, particularly when utilising silicon isotopically purified to contain only the nuclear spinless ^{28}Si isotope[2]. With reduced hyperfine interactions, these devices boast some of the best coherence times and gate fidelities [25].

Three main device types are prominent for hosting silicon-based qubits: ^{31}P dopants in silicon[3], gate-defined dots in metal-oxide-semiconductors (MOS), and SiGe heterostructures[4] (Fig. 1 (a-d)).

[1]Two stable isotopes of gallium exist ^{69}Ga (60.1% abundance and nuclear spin 3/2) and ^{71}Ga (39.9%, 3/2) whilst only arsenic has only one stable isotope ^{75}As (100%, 3/2).

[2]Three stable isotopes of silicon exist, ^{28}Si (92.21%, 0), ^{29}Si (4.70%, 1/2) and ^{30}Si (3.90%, 0).

[3]There is only one stable isotope of phosphorus, ^{31}P, and it is spin 1/2.

[4]Four stable isotopes of germanium exist ^{70}Ge (abundance 20.4% and nuclear spin 0), ^{72}Ge (27.3%, 0), ^{73}Ge (7.78%, 9/2) and ^{74}Ge (36.7%, 0). In addition, ^{76}Ge (7.83%, 0) is only very slightly radioactive with a half-life of

Additionally, there are alternative implementations, such as (near) one-dimensional structures like Si-FinFETs and Si nanowires (Fig. 1 (e)). All these device types rely on electron or hole spins as the basis for qubits, except for donors in silicon, which frequently utilise nuclear spins [26, 27].

In addition, very recently, Ge/SiGe (Germanium doped with Silicon) heterostructures have been rapidly advancing [28–30]. This can be attributed to the charge carriers in strained germanium possessing a lower effective mass than silicon, enabling larger quantum dots and, therefore, easier fabrication. In addition, hole-based qubits in Germanium can be controlled through all electrical pulsing due to the large spin-orbit interaction relative to electrons. This negates the need for a microwave antenna or micromagnet to facilitate single qubit control (Sec. 2.2.1) [29–32], making the devices more scalable and easier to fabricate. Germanium hole spin qubits demonstrate long coherence times relative to GaAs. This is for a collection of reasons, firstly because there are fewer nuclear spin-bearing nuclei in the heterostructure contributing to hyperfine noise [28]. Secondly, the p-type character of the valence band, the hole's wavefunction, goes to zero at the nuclei of the atoms in the heterostructure, thereby suppressing the dephasing due to the hyperfine noise [28].

Hole-spin qubits may be realised in a range of devices, including Ge/SiGe heterostructures [29, 31], and fin field effect transistor (finFET) [32] and core-shell nanowires (Fig. 1 (d, f)) [33]. We use Ge/SiGe heterostructure devices in Chapters 4 and 5, and a Si-MOS device is Chapter 7. Therefore, we will now delve into each of these device types individually, analysing their distinct characteristics as foundational elements of a quantum computer.

Quantum dots in MOS

Silicon metal-oxide-semiconductor (Si-MOS) devices accumulate electrons at a Si/SiO_2 interface[5] (Fig. 1 (c)). As the effective mass of electrons in silicon is around an order of magnitude larger

1.78×10^{21} years.

[5]Oxygen has three stable isotopes ^{16}O (abundance 99.7%, nuclear spin 0), ^{17}O (0.04%, 5/2) and ^{18}O (0.19%, 0).

than in GaAs [9][6], the length-scales for the confinement are considerably smaller, on the order of hundreds of nano-meters [21, 34]. These smaller dot sizes necessitate smaller gate feature sizes, making fabrication more difficult. However, quite fortuitously, Si-MOS semiconducting quantum devices are compatible with current classical MOS technology. Therefore, it can be argued that once the recipe for Si-MOS quantum devices is perfected, the foundries could fabricate large devices with high yields.

Regarding the history of Si-MOS spin qubits, Maurand et al. [35] first demonstrated MOS devices hosting spin qubits and exhibiting spin effects, such as Pauli spin blockade. However, the observed coherence times were disappointingly short, measuring only 245 ns via Hahn echo, with the underlying cause of this decoherence being unclear. Proposed explanations point toward impurities within the material, underscoring the paramount importance of rigorous quality control in the fabrication process for these devices.

Veldhorst et al. [34] effectively dispelled doubts surrounding MOS devices by achieving significantly enhanced coherence times of 28 ms using a Carr-Purcell-Meiboom-Gill (CPMG) pulse sequence, designed to mitigate the effects of spin dephasing, improving upon the 200 µs CPMG coherence times attained in Gallium Arsenide (GaAs) [36]. Then, Veldhorst et al. [37] demonstrated two-qubit logic gates, fulfilling the universal gate set DiVincenzo criterion.

Addressing architectural considerations for scaling silicon MOS spin qubits, Veldhorst et al. [21] comprehensively explored the challenges. The study acknowledged no straightforward path to scaling Si-MOS qubits, primarily due to the minimum feature size, which dictates a gate length of approximately 7 nm. This constraint results in an area requirement of roughly 63×63 nm^2 per qubit. Moreover, to accommodate classical transistors for gate voltage control and qubit readout,

[6] Gallium arsenide has a direct band gap with the conduction band minimum at the Γ point. The effective mass of electrons in GaAs is typically isotropic and can be represented by a single value, $m^* \approx 0.0067 m_0$. Whereas, Silicon has an indirect band gap with the conduction band minimum at the six equivalent Δ valleys along the $\langle 100 \rangle directions$. The effective mass of electrons in silicon is often expressed in terms of the longitudinal and transverse effective masses, $m_l \approx 0.98$ and $m_t \approx 0.19 m_0$ [9]

further advancements in manufacturing technology will be essential to minimise thermal and volumetric impacts on the qubit module. Furthermore, as quantum dots in MOS devices are confined against the Si/SiO_2 interface, they are quite susceptible to charge noise [38]. By contrast, Ge/SiGe heterostructures historically had the quantum well 22 nm below the Oxide layer. However, recent devices have moved the dots even deeper to 55 nm to further reduce the charge noise [39]

Efforts to reduce module size are crucial for effective thermal management, as larger modules demand greater cooling power and, therefore, larger cryostats. Veldhorst et al. [21] roughly addressed this issue by estimating that the module could operate at 100 mK in a typical dilution fridge. However, it was recognised that this could potentially pose a bottleneck for scalability, suggesting that silicon spin qubits may need to operate at higher temperatures, as previously demonstrated [40], to facilitate scalability. Yang et al. [41] further contributed to this discussion by operating two qubits at 1.5 K in a Si-MOS device, achieving coherence times of 2 ms and fidelities of 98.6%. This approach demonstrated how these qubits could fit within the architectural framework envisioned by Veldhorst et al. [21] while still operating at temperatures attainable with a pumped ^{4}He system, effectively reducing costs and engineering complexities associated with thermal management in the scaling of MOS-based qubits [41].

Quantum dots in heterostructures

Silicon heterostructure-based devices define quantum dots in a quantum well approximately 50nm below the gate electrodes. This quantum well is formed by doping the Silicon with Germanium, Fig. 1.

These devices have established themselves as a contending architecture with not only the demonstrations of qubit arrays [42] but also the achievement of error-correcting threshold fidelities [43, 44] alongside their ion-implanted donor-in-silicon counterparts [45]. Devices based on heterostructures enable highly tunable quantum dots with electrons (or holes) confined in the vertical direction by

band engineering and confinement in the horizontal direction defined by gate electrodes. At the same time, SiGe devices still meet the mark of high fidelities and coherence times, with Kawakami et al. [46] demonstrating 99% gate fidelities and CPMG times of approximately 400 µs.

The high tunability of the SiGe dots is demonstrated by Lawrie et al. [47] tuning a linear array of five quantum dots. Lawrie et al. compares the manufacturing differences between Si MOS, Si/SiGe and Ge/SiGe heterostructures, making the case for all three as compatible with industrial fabrication techniques. Lawrie et al. also compares the dots' cross capacitances between the different devices. Although the devices are different in terms of layout, they have much lower cross capacitances than those seen in GaAs devices facilitating their operation.

Lawrie et al. presents the challenges of heterostructures for scaling devices, such as automated tuning and wiring logistics for each gate with supporting work that looks to help solve these problems [48, 49]. Xu et al. [50] explains that having each gate connected to a digital-to-analogue converter (DAC) is a bottleneck for scaling the number of qubits. By comparison, today's classical processor chips have only about 2000 contact pins, while billions of transistors can be integrated and operated on a single chip". Inspiration is taken from current DRAM chips, and charge locking is incorporated into SiGe devices. Charge-locking electrically detaches a line from a gate but through the use of a switching capacitor circuit, and thus, the gate of the quantum dot is floating for a period of time. When combined with demultiplexing, the number of lines to the chip can be significantly reduced. This enables one gate to float while another gate is pulsed, keeping the number of lines to the chip minimal [50]. Xu et al. verifies that the capacitor storing the dot gate potential does not affect the gate pulses. This is shown by performing electronic circuit simulations allowing pulses up to 20 GHz frequencies. Schaal et al. [51] further this by demonstrating that more than one quantum dot can be read out with a single resonator using transistors to switch between the dots. Therefore, there is a pathway path to reducing the number of input lines per qubit and enabling the addressing of large-scale device arrays.

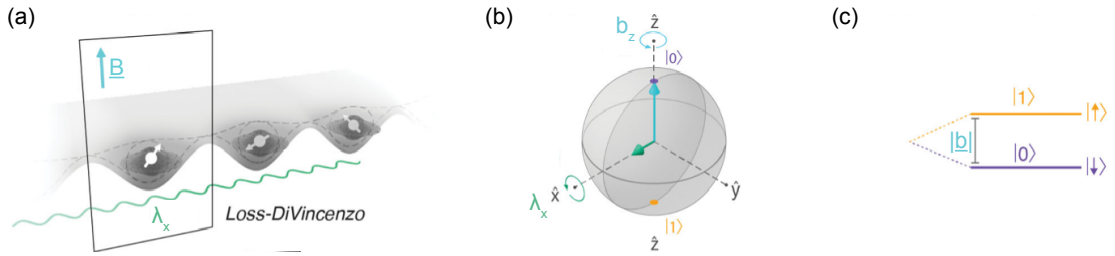


Figure 2: (a-c) Electron/hole confinement, Bloch sphere and energy level diagram of a Loss-DiVincenzo qubit. Loss-DiVincenzo qubits are encoded in a spin 1/2 electron or holes confined in a single quantum dot. A static magnetic field $\vec{B}$ defines the spins quantisation axis as parallel to $\vec{b}$ and creates the energy separation, $|\vec{b}|$ (Eq. (2.1)), between the qubits. To simplify the diagram, we have assumed that $\vec{B} \| \hat{z}$, and the g-tensor is isotropic. ESR or EDSR can be used to create an oscillating Zeeman term in the Hamiltonian equation of the form given by equation (2.2). The transverse, $\lambda_{x/y}$, components of this coupling can be used to drive coherent spin rotations between spin up and down, provided that the drive frequency is approximately correct, such that $\hbar\omega_d \approx |\vec{b}|$. Figure adapted from Ref. [9].

In addition, Ge hole spin qubits are currently enjoying intense research interest owing to their ease of operation and compatibility with existing Si technology [28]. From 2018, and within just four years, a Loss-DiVincenzo qubit [52], a single-triplet qubit [31], two-qubit devices [29] and a four-qubit Ge quantum processor [30] have been realised, demonstrating the potential of Ge for quantum information. Furthermore, Ge has also hosted record-breaking ultra-fast control frequencies, with Rabi frequencies exceeding 1.2 GHz [53, 54]. Novel, more scalable gate geometries can be developed. Borsoi et al. [55] demonstrates a 4×4 quantum dot array in a Ge/SiGe heterostructure, with a single gate electrode addressing a row or column of dots at a time.

2.2 Types of semiconducting qubit

2.2.1 Loss DiVincenzo qubits

An electron or hole spin confined in a single quantum dot under an external magnetic field $\vec{B}$ (Fig. 2 (a)) is described by the Zeeman Hamiltonian,

$$H = \vec{b} \cdot \vec{\sigma}/2 \text{ where } \vec{b} = \mu_B g \cdot \vec{B}, \tag{2.1}$$

where $\vec{b}$ is the Zeeman vector, $\vec{\sigma} := [\sigma_x, \sigma_y, \sigma_z]^T$ is a vector of the Pauli matricies and g is the g-tensor [33]. Therefore, they are a natural host for a qubit with a direct mapping between the spin and qubit states and operators, such that $|\downarrow\rangle \leftrightarrow |0\rangle$ and $|\uparrow\rangle \leftrightarrow |1\rangle$ [56]. Electrons in silicon typically have close to isotropic g tensors [57]. Holes in germanium can have extremely anisotropic g-tensors, differing by factors of up to 100 [58].

Spin control can be achieved by electron spin resonance (ESR) or electron dipole spin resonance (EDSR). ESR can be performed using an antenna to create an oscillating magnetic field near the qubit [37]. EDSR involves using an oscillating electrical field to perturb the confinement of the charge carrier and some coupling mechanism, such as strong spin-orbit coupling, a magnetic field gradient or an electrically-tunable g-tensor [9, 59–61]. Both ESR and EDSR produce a time-dependent Zeeman term in the qubit Hamiltonian. If the drive is of the form $\cos(\omega_d t)$ then the drive Hamiltonian is

$$H_d = \hbar\vec{\lambda} \cdot \vec{\sigma} \cos(\omega_d t), \tag{2.2}$$

where $\vec{\lambda}$ encodes the amplitude of modulation and the direction resulting from the coupling mechanisms [33]. For $\vec{\lambda} \perp \vec{b}$, we have Rabi driving, which drives the qubit between spin up and spin down (Fig. 2 (b)). The less common, $\vec{\lambda} \parallel \vec{b}$ produces phase driving [33].

The spin state can be measured using Elzerman readout, where spin-selective tunnelling to a

fermionic bath of electrons/holes distinguishes the spin states. The higher-energy spin state can tunnel, while the lower-energy state is forbidden from tunnelling (Fig. 2 (c)). These tunnelling events, or lack thereof, can be detected with a charge sensor [9]. If, instead, the Loss-DiVincenzo is operated inside a double quantum dot with another spin in the other dot, qubit readout can be accomplished using spin-to-charge conversion techniques, discussed in Section 2.3.

2.2.2 Single-triplet qubits

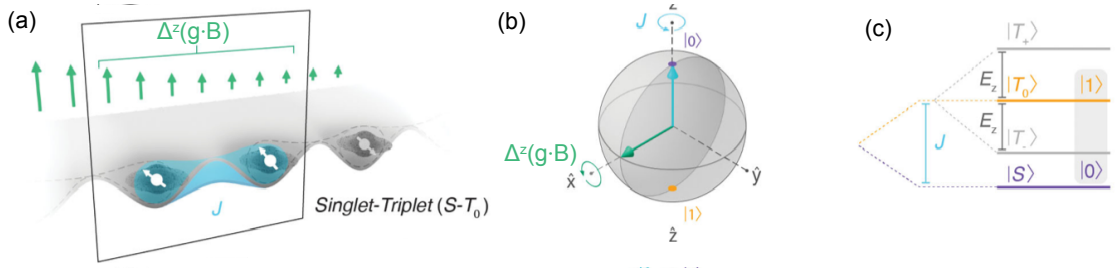


Figure 3: (a-c) Electron/hole confinement, Bloch sphere and energy level diagram of a singlet-triplet ($S-T_0$) qubit. The energy level separation is set by an electrically tunable exchange coupling J. An effective magnetic-field gradient $\Delta^z(g \cdot B)$ creates an always-on σ_x contribution to the Hamiltonian. Single qubit operations are enacted by reducing the exchange coupling, so the qubit precession axis moves away from the σ_z axis towards the σ_x. Figure adapted from Ref. [9].

Singlet-triplet qubits encode a qubit in a spin subspace of two electrons or holes in a double quantum dot in the presence of a static magnetic field gradient (Fig. 3 (a)) [15]. The most common variant is the ST_0 qubits, where the singlet state is mapped to the qubit ground state such that $|S\rangle \leftrightarrow |0\rangle$, whilst the spinless triplet state encodes the qubit excited state, $|T_0\rangle \leftrightarrow |1\rangle$ (Fig. 3 (c)). The Hamiltonian for these qubits is

$$H = \frac{1}{2}J\sigma_z + \frac{1}{2}\mu_B\Delta^z(g \cdot B)\sigma_x, \tag{2.3}$$

where J is the exchange coupling and $\Delta^z(gB)$ is the effective difference in the magnetic field between the dots. This difference in an effective magnetic field can be created by a g-factor difference or

with a magnetic field gradient [9]. The exchange coupling, J, depends on detuning (the difference in energy) between the quantum dots; therefore, it is possible to use baseband pules to perform single-qubit rotations (Fig. 3) [9]. Qubit readout is accomplished using spin-to-charge conversion, discussed next in Section 2.3.

2.3 Spin-to-charge conversion

In broad terms, spin-to-charge conversion techniques facilitate the conversion of a spin state into a charge state, enabling the differentiation of spin states through a charge sensor. One prevalent spin-to-charge conversion technique is the Pauli spin blockade (PSB) [9, 15]. In this scheme, after the qubit operations, the double quantum dot is quickly brought into a configuration where the electrons/holes should tunnel from the $(1,1)$ to the $(0,2)$ charge state. Due to the Pauli exclusion principle, the electron can only move to the other dot if the electrons form a singlet state, $|S\rangle$. The triplet states, $|T_0\rangle, |T_-\rangle, |T_+\rangle$, stay blockaded and remain in the $(1,1)$ charge state.

A variant of PSB is called parity readout; this readout technique maps odd spin states, $|\uparrow\downarrow\rangle$ and $|\downarrow\uparrow\rangle$, to the $(0,2)$ charge state, whilst even spin states, $|T_-\rangle := |\downarrow\downarrow\rangle$ and $|T_+\rangle := |\uparrow\uparrow\rangle$, are blockaded, so remain in the $(1,1)$ charge state [62]. Parity readout is performed by using PSB, then taking advantage of the $|T_0\rangle$ state decaying faster than the $|T_\pm\rangle$ states [62]. The rate of this decay depends upon the charge dephasing and relaxation time, the detuning between quantum dots at the readout configuration, the tunnelling rate between the quantum dots, and the difference between the g-factors in each of the dots [62].

2.4 Device measurement and readout

This section discusses the measurement techniques of semiconducting quantum devices: current measurements and rf reflectometry. Both methods can be used to probe quantum dots directly or

via a charge sensor.

2.4.1 Transport measurements

DC transport measurements, which involve the study of electron/hole flow through a quantum system, have been pivotal in uncovering the properties and potential applications of quantum dots [63].

Transport through quantum dots displays the phenomenon of Coulomb blockade, which manifests as a suppression of current through a dot due to the quantisation of charge and the energy cost associated with adding an electron to the QD. This effect has been meticulously studied through conductance measurements, where the discrete addition of electrons to the dot can be observed as peaks in the conductance spectrum as a function of gate voltage [64].

Transport measurements have been extended to probe the spin states of carriers in quantum dots [15, 32, 33]. However, this spin readout is possible only in the aggregate of many qubit measurements, as DC transport measurements have insufficient bandwidth for single-shot qubit measurements [65].

The bandwidth of a standard transport measurement setup is limited by the RC low-pass filter formed by the sample resistance, the amplifier's input impedance, and the electrical cables' capacitance. Quantum devices usually have resistances at least on the order of the quantum resistance $h/e^2 \approx 25.8\,\mathrm{k}\Omega$ and cables have capacitances in the range of $0.1-1\,\mathrm{nF}$ [65]. Thus, the cut-off frequency of this filter, $\omega_{RC} = 1/RC$, is usually no more than a few kilohertz.

2.4.2 Radio-frequency reflectometry

Radio-frequency reflectometry overcomes the bandwidth limitations of dc transport measurements by putting the quantum device at the end of a $Z_0 = 50\,\Omega$ transmission line with the difference

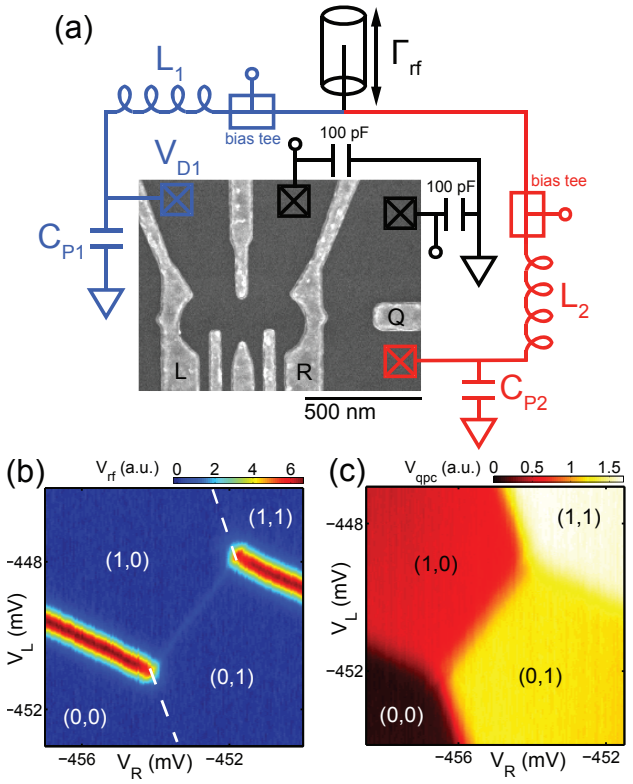


Figure 4: a) Schematic of GaAs double dot device configured to be measured by reflectometry on an ohmic and a QPC. The matching circuits for the ohmic and QPC-based reflectometry are shaded in blue and red, respectively, with the narrow restriction between gates R and Q acting as the QPC. b-c) Response of the ohmic and QPC resonant circuits, respectively, as a function of the double quantum dot's left and right gates voltages, L and R, respectively. Absolute electron numbers for the left and right dots are indicated in brackets. Figure adapted from reference [66].

in impedances intermediated by a matching circuit. Through the matching circuit, the device is illuminated by radio frequencies, and the reflected amplitude and phase encode information about resistive and reactive components of the device impedance. See Fig. 4 (b) for a measurement of a double dot using reflectometry connected to a device's ohmic contact. With every component in the amplification chain matched to Z_0, the resistive contribution to the RC filter is significantly reduced; therefore, the bandwidth is enhanced to hundreds of megahertz [65]. This has made rf reflectometry an indispensable tool for studying quantum coherence and relaxation processes in quantum dots and for developing quantum dot-based qubits for quantum computing applications. Vigneau et al., Petersson et al., Hogg et al. [65, 67, 68] exploited rf reflectometry to perform fast and sensitive charge sensing in a silicon quantum dot, paving the way for its use in quantum computing architectures [67].

Moreover, rf reflectometry is also more scalable than DC transport measurements, owing to its compatibility with multiplexing, permitting the simultaneous readout of multiple qubits [17]. Furthermore, West et al. [69] demonstrated the use of gate-based rf reflectometry for the readout of spin qubits in silicon quantum dots, further enhancing the arguments for the scalability of rf measurements. Such is the potential of radio-frequency reflectometry that it is the readout technique chosen to satisfy DiVincenzo's readout criterion in all proposed large-scale quantum dot array architectures [70–73].

2.4.3 Charge sensing

Charge sensing represents an indirect methodology for investigating the charge state of quantum devices, diverging from traditional impedance measurements to focus on the impedance of an adjacent charge sensor. This sensor's impedance is designed to depend on the device's charge state, providing a non-intrusive means to study quantum systems [74–79]. When combined with spin-to-charge conversion techniques such as Pauli spin blockade, charge sensors can also be used for single-shot readout of spin qubits [66, 69, 80]. Here, we highlight the principles of charge

sensing, emphasising its implementation via quantum point contacts (QPCs) [66] and single-electron transistors (SETs).

QPCs are narrow constrictions in a two-dimensional electron/hole gas comparable to an electron's wavelength and serve as charge sensors (Fig. 4 (a)). Their impedance is directly influenced by the surrounding electrostatic environment, making them sensitive to changes in a device's charge state. Modulations in the QPC's impedance, induced by changes in the device's charge, can be precisely measured to yield insights into charge dynamics [81]. See Fig. 4 (c) for the charge stability diagram of a double dot as measured through a radio frequency QPC (rf-QPC).

SETs use a quantum dot in place of a QPC, such that the phenomenon of the Coulomb blockade effect provides heightened sensitivity to electrostatic changes [65]. However, this sensitivity comes at a cost. By the proximity of the charge sensor to the device, there is cross-talk between the two. This cross-talk can perturb the SET charge sensor away from a Coulomb peak and thus make it insensitive. As such, it is necessary to compensate for changes in gate voltages on the device by changing the gate voltages on the SET. This requires the use of virtual gates [82] or digital feedback techniques [83, 84], which maintain a constant current flowing through the SET.

2.4.4 Device impedance and tunnel rates

As discussed later, RF measurements are sensitive only when the device impedance is near the matching impedance (Sec. 3.2.2). In charge sensing measurements (Chapter 5), the conductance G of the sensor across a Coulomb peak is determined by the tunneling rates from the source to the dot (Γ_s) and from the dot to the drain (Γ_d) according to the Breit-Wigner formula:

$$G(V_g) = \frac{e^2}{4k_BT}\left(\frac{1}{\Gamma_s}+\frac{1}{\Gamma_d}\right)^{-1}\cosh^{-2}\left(\frac{\mu(V_g)}{2k_BT}\right) \tag{2.4}$$

Here, T is the electron temperature and α represents the lever arm [85]. The conductance is significant only when the single dot energy level aligns with that of the leads ($\mu(V_g) \ll k_B T$), and the barriers are sufficiently transparent.

The ohmic reflectometry measurements, used in Chapter 4, involving two dots and an additional interdot tunnel barrier, the conductance is maximised when all three tunnel barriers are transparent and both dots' energy levels align with the leads, creating bias triangles. In both scenarios, the device's overall conductance is heavily dependent on the tunnel rates, which are exponentially sensitive to gate voltages [55]. Since RF measurements are only sensitive when the device impedance is within a specific range (Sec. 3.2.2), the thickness of the barriers is crucial for successful RF measurements.

2.5 The constant capacitance model

Quantum dots with weak inter-dot coupling and thus well-localised charges separated from the remaining electron/hole gas can be modelled by the constant capacitance model [86–90]. This model relies upon two assumptions: constant capacitances can parameterise the Coulomb interactions between electrons/holes on quantum dots (Fig. 5 (a)). And quantum effects, such as tunnelling and spin, can be neglected. This simple classical model can reproduce the charge stability diagrams of quantum dots (Fig. 5 (b)). In the following section, we will introduce the matrix formulation of the constant capacitance model.

2.5.1 The mathematical formulation

The constant capacitance model describes a quantum dot array of the quantum dots and electrostatic gates as nodes in a network connected by fixed capacitors. Each node, i, has an associated charge Q_i, an electrostatic potential V_j, and a capacitive coupling to every other node k given by c_{jk}. The capacitor C_{ij} stores a charge $q_{ij} := C_{ij}(V_i - V_j)$. The total charge on node i is therefore

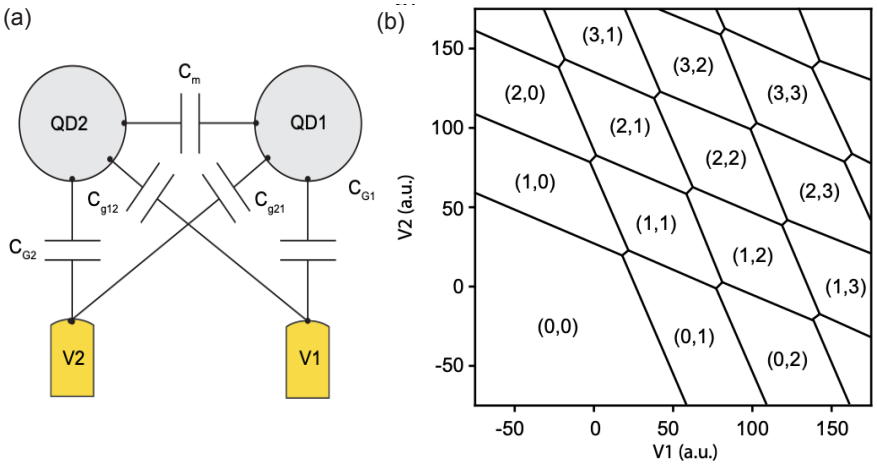


Figure 5: (a) Capacitance model of an electron double dot quantum device defined using two gates. (b) Electron charge stability diagram of the double quantum dots gates $V1$ and $V2$ are swept. The lowest energy charge states are annotated in each region. Figure adapted from Ref. [91].

related to its and other nodes' potentials according to

$$Q_i = \sum_j q_{ij} = \sum_j C_{ij}(V_i - V_j). \tag{2.5}$$

Using the Maxwell matrix formulation, we express this as $\vec{Q} = \mathbf{c}\vec{V}$, where $\mathbf{c}$ is the Maxwell capacitance matrix[7] and where the $\mathbf{c}$ matrix's diagonal elements are each node's total capacitance, and the off-diagonal elements are the negative of the capacitive coupling between the respective nodes. We now distinguish between quantum dots and electrostatic gate nodes by separating the matrix equation into

$$\begin{bmatrix} \vec{Q}_d \\ \vec{Q}_g \end{bmatrix} = \begin{bmatrix} \mathbf{c}_{dd} & \mathbf{c}_{gd} \\ \mathbf{c}_{dg} & \mathbf{c}_{gg} \end{bmatrix} \begin{bmatrix} \vec{V}_d \\ \vec{V}_g \end{bmatrix}, \tag{2.6}$$

[7] In this section, matrices with denoted with a lower case "c" are in the Maxwell form, for all subscripts.

where $\vec{Q}_{d(g)}$ represents the charge on the quantum dots (gates) and $\vec{V}_{d(g)}$ denotes the potential on the quantum dots (gates). The potential on the quantum dots can then be computed by

$$\vec{V}_d = \mathbf{c}_{dd}^{-1}(\vec{Q}_d - \mathbf{c}_{gd}\vec{V}_g). \tag{2.7}$$

The free energy of the system $F(\vec{Q}_d; \vec{V}_g) = \vec{V}_d^T \mathbf{c}_{dd} \vec{V}_d/2$ can then be written as

$$F(\vec{Q}_d; \vec{V}_g) = \frac{1}{2}\vec{Q}_d^T \mathbf{c}_{dd}^{-1} Q_d - (\mathbf{c}_{dd}^{-1}\mathbf{c}_{gd}\vec{V}_g)^T \vec{Q}_d, \tag{2.8}$$

by dropping the term without dependence on $\vec{Q}_d$. This free energy function is convex since $\mathbf{c}_{dd}$ is positive definite (for proof, see Appendix C.1). Since the charge on the gates is not of interest, in the remainder of this thesis, we drop the subscript and refer to the charge on the quantum dots as $\vec{Q}$.

The allowed charge states, $\vec{Q} = \mp e\vec{N}$ for electrons and holes, are heavily constrained in a quantum dot array. In particular, for an open quantum dot array, when the array is coupled to fermionic reservoirs from which it can draw an arbitrary number of charges, the constraints are that:

(i) The number of charges must be integers, represented as $\vec{N} \in \mathbb{Z}$, for all i;

(ii) The number of charges on any given quantum dot must be non-negative, meaning $N_i \geq 0$ for all i.

For a closed quantum dot array, which is when the leads are decoupled from the array, and the number of charges is fixed to a finite value $\hat{N}$ [92, 93], we impose the additional constraint that:

(iii) The sum of all charges in the array must equal a specific value, denoted as $\hat{N}$.

The constant capacitance model assumes that at $T = 0$ K, the system will adopt the charge state,

$\vec{N}^*$, which minimises the free energy whilst obeying the appropriate constraints, such that

$$\vec{N}^* = \underset{\vec{N} \in \mathcal{N}}{\arg\min}\, F(\vec{N}; \vec{V}_g), \tag{2.9}$$

where $\mathcal{N}$ is the set of all admissible charge configurations for the open or closed regimes of the quantum dot arrays.

2.6 Tuning quantum devices

Automating the tuning of quantum dot devices is still in its infancy, with a small number of research groups addressing the challenge and offering partial solutions. A comprehensive algorithm capable of taking a freshly fabricated device and transforming it into an array of finely tuned qubits has yet to be developed. However, existing automation algorithms, although in the early stages, are shedding light on this critical endeavour and inspiring further advancements in the field. Here, we review these algorithms.

2.6.1 Tuning without machine learning

Several algorithms [48, 94, 95] have made progress in automating the coarse definition of double quantum dots in semiconducting devices. This is a crucial initial step in the automation process and holds significant promise. These algorithms save experimentalists valuable time and are essential for achieving the scalability required to "turn on" a quantum computer.

The most common approach when designing a tuning algorithm is to mimic the workflow of a human operator [48, 95], essentially translating the manual processes into code that interacts with the necessary instrumentation. An initial algorithm developed by Baart et al. involved no machine learning but relied heavily on prior knowledge of the specific device [48]. This algorithm coarsely tuned two quantum dots within a linear array of four dots by identifying pinch-off points for

each gate through current measurements. Single dots were then fine-tuned using plunger gates (dedicated gates designed to control a dot's potential primarily), leveraging previous knowledge of appropriate voltages for the device. While not generalisable due to its reliance on device-specific knowledge, Baart et al.'s work laid the foundation for future tuning algorithms. The critical components identified in their approach included locating pinch-off points, identifying Coulomb peaks, and mapping the surface that separates regions of high and low current.

Volk et al. introduced the "$N+1$" method for tuning a linear array of eight qubits [19]. Like Baart et al., each dot was tuned sequentially using a charge sensor while considering the cross capacitances between all gates. The cross capacitances were then utilised to create "virtual gates", allowing independent manipulation of the charge state of each quantum dot without affecting neighbouring dots. These virtual gates are a linear combination of many gates such that the unwanted crosstalk is cancelled, leaving only the desired change [19]. Whilst aspects of this process were automated, the successful tuning of the array required heavy user involvement.

2.6.2 Tuning with machine learning

A different approach to automation involves leveraging mathematical and artificial intelligence (AI) tools to surpass human capabilities, as demonstrated by Moon et al. [94]. By eliminating the reliance on a typical human workflow, this approach aims to create a more general algorithm that can be applied to various devices, as it does not depend on human habits and previous knowledge of similar device tuning.

When tuning quantum dots by identifying transport features, specific gate voltage values divide the device parameter space into two regions: relatively high and relatively low (pinched-off). Typically, this hypersurface is located by conducting 1D current traces with each gate until pinch-off is achieved [48, 95]. However, Moon et al. propose a faster approach by randomly sampling points in voltage space and measuring the current in that direction from an arbitrarily chosen origin [94].

After locating multiple points (at least 30) on the hypersurface, a Gaussian process model is used to predict the hypersurface's location.

As previously observed [48], points on this hypersurface and in the vicinity of the pinch-off region often exhibit single-dot features identifiable as Coulomb peaks in 1D current traces. If Coulomb peaks are detected, a low-resolution 2d current map is generated using the plunger gates of the device, and this map is compared against the expected "honeycomb lattice" characteristic of a double dot regime. If the score exceeds a predefined threshold, this voltage regime is assumed to contain a double dot regime, and the algorithm performs a high-resolution scan for further analysis by a human operator. The process of hypersurface sampling and exploration is iteratively repeated. In each iteration, the information gathered during the investigation stage (e.g., presence of Coulomb peaks, identification of double dot regimes) is incorporated into the Gaussian process model of the hypersurface. This enhances the hypersurface prediction and guides the algorithm in selecting the most promising locations to sample the hypersurface next, based on the probability of finding Coulomb peaks.

To minimise tuning times, the algorithm focuses on targeted sampling of the hypersurface and accurate classification of Coulomb peaks in 1D traces. However, the Coulomb peak identification method's precision is not clearly explained in the article [94]. Identifying Coulomb peaks may not be as straightforward as identifying pinch-off gate voltages for various devices, which could benefit from further clarification and refinement. Nonetheless, this algorithm is already among the most general, as it does not require knowledge of the specific device gate architecture during the hypersurface sampling stage [94]. The algorithm only necessitates knowledge of the plunger gate identities to produce 2d charge stability diagrams (current maps).

The algorithm's performance is demonstrated by successfully tuning two double-dot GaAs devices. Compared to human experts (3 hours) and a purely random search algorithm (680 hours), this algorithm can tune these double dots in just 70 minutes, representing a significant improvement in

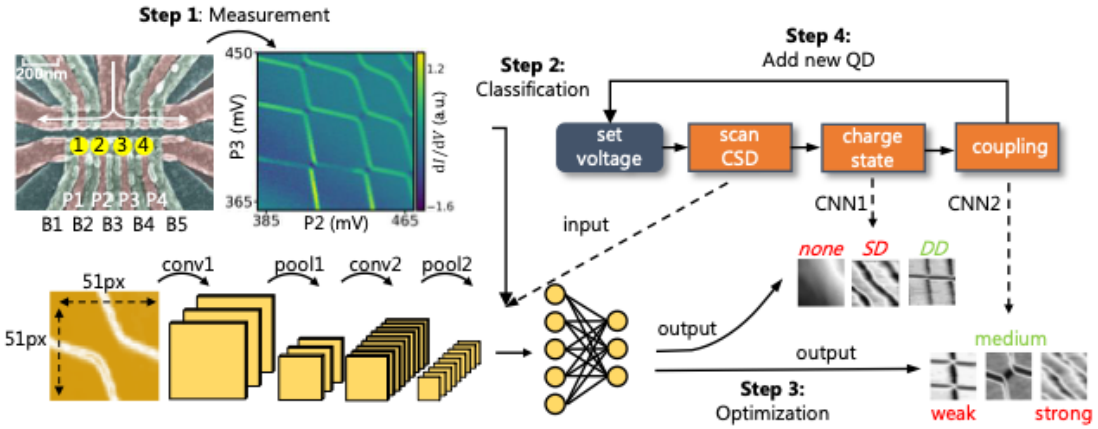


Figure 6: Autotuning process schematic. Two different CNNs analyse the obtained CSDs to determine the charge states and inter-dot tunnel coupling, and only the CSDs with QDs being in the double-dot state and with medium coupling strength are reserved. After optimising voltage configurations, new QDs are added, and a new cycle starts. Figure adapted from Ref. [96].

efficiency [94].

More recently, convolutional neural networks have gained traction and have repeatedly demonstrated the ability to act as a score function for a double dot feature quality [96–98]. With the score function defined, autotuning is a case of optimising this score function. However, pitting the score function against the neural network in this way relies upon the neural network being robust to being erroneously tricked by dot features. If the score function fails, the whole algorithm will fail. Therefore, the fundamental achievement of these works could be that they have compiled a sufficiently wide and diverse dataset so that the neural network does not get tricked [99]. I will now discuss these works in more detail.

In the work of Kalantre et al. [97], they use deep convolutional neural networks to characterise states and charge configurations of semiconductor quantum dot arrays measured through transport. They use this network to optimise the gate voltage of a simulated device. They also demonstrate that the approach could be implemented in an experimental setting by showing the network works on an experimental dataset.

Then Zwolak et al. [98] demonstrated that neural networks could be used to autotune an actual device rather than a simulated dataset. They showed the ability to fine-tune the charge state of a double quantum dot device, moving within the few electron regimes to find the $(1,1) \leftrightarrow (0,2)$ charge transition. However, whilst their double dot device is confined using five gates, three barriers and two plungers, they only give the optimiser control over the two plungers. As a result, the coupling between the dots is beyond the control of the optimiser. The barrier voltages were set to give an appropriate coupling between the dots in the last electron regime. The most remarkable aspect of their work was that the dataset, consisting of 10010 random charge sensor measurements, was entirely simulated using the constant capacitance model. And yet, it achieved an accuracy of 97.71% on experimental data.

However, the device used in Zwolak et al. [98] might be exceptional, as a later paper from the group, Ziegler et al. [99] discusses the issues introduced by poor quality noisy experimental data tricking the network. To circumvent this, they create a data quality control module to act as a "gatekeeper" so that only reliable data is processed by the state classifier. They then implement a spectrum of noise models from $1/f$ to sensor jumps to degrade the quality of the charge stability diagrams produced by the constant capacitance model. The combination of the "gatekeeper" and a new neural network trained on a simulated dataset of 1.6×10^4 simulated scans achieved an accuracy of 95.1% when tested on experimental data.

Finally, in Liu et al. [96], they combine the "N+1" method with the power of neural networks to demonstrate the tuning of a four-dot quantum dot array. In this work, they use two CNNs built of a pre-trained CNN called AlexNet. The first is trained for charge state identification. The second takes a charge stability diagram showing an interdot transition and quantifies the degree of tunnel coupling between the dots. These neural networks were trained on a dataset of 26,000 experimentally measured charge stability diagrams mixed with 3,000 simulated using the constant capacitance model. They claim that mixing in the small amount of capacitance model data avoided overfitting, as they are clean and less distorted.

Their array-tuning algorithm works by sequentially tuning double dots. For the first double dot, dots one and two, they sweep plungers $P1$ and $P2$ over a $300\,\text{mV} \times 300\,\text{mV}$ range (Fig. 6). They then apply the first CNN to a sliding window over this high-resolution scan to identify regions where double dots exist by thresholding the output of the CNN. Thereafter, then assume that the area with the least positive plunger voltages corresponds to the few-electron regime and navigate there. The appropriate barrier voltages for this charge stability seem to be set a priori. However, once a double dot charge transition is located, they use the second CNN to optimise the middle barrier voltage, in this case, $B2$. The CNN quantifies the coupling. If the coupling is too high, they increase the barrier. Too low, and they decrease it. For each adjustment, they move the window to acquire the charge stability diagram by an amount equal to 0.18 times the charge in the barrier. This value is again known a priori. The optimisation is terminated once the coupling reaches an appropriate value in the range of $0.3 - 0.7$ (a.u) as quantified by the CNN.

Following this, virtual gates are constructed using the Hough transform [90]. They then repeat the algorithm to tune dots two and three. This time, rather than $P2$, they use a virtual plunger gate, which alters the second dot's potential without distributing the first. They repeat this until the array is tuned. From the manuscript, it is unclear whether this algorithm was run more than once, its run time, how much of the dataset was measured from the device they tuned, and how robust it was to change the prior values.

Whilst I have been critical of the work of Liu et al. [96] for putting in values a priori, I hope this will become the norm. This is because when mass characterising low variability high-quality devices, we can imagine "knowing" a lot about a device even before cooling it down. Why shouldn't the tuning algorithm make use of this knowledge? However, from my experience, we are not there yet.

Returning back to Hough transforms, Oakes et al. [90], train an ensemble of regression models to extract the gradients of charge transitions from a Hough transform of a stability diagram. They demonstrate their method can generate virtual gates for a 2×2 dot array. And claim that it can

be generalised to larger $N \times 2$ bilinear arrays.

2.7 Conclusion

In conclusion, we have discussed the types of semiconducting devices and the most common qubits they host. For the qubit implementations, we have covered how the semiconducting spin state encodes the qubit, how they are manipulated, and how their state can be measured. We then introduced the measurement techniques used to probe quantum devices more generally, covering dc transport, radio-frequency reflectometry and charge sensing. Next, we introduced the constant capacitance model, a classical model that captures many of the features of the quantum dot's charge stability diagrams. Finally, we have reviewed previous work on tuning quantum devices, both manual and automated, with machine learning.

Chapter 3

Experimental setup, control software and computational methods

> Experimental investigation, to borrow a phrase employed by Kepler respecting the testing of hypotheses, is "a very great thief of time." Sometimes it costs many days to determine a fact that can be stated in a line.
>
> John William Draper

In this chapter, I describe the experimental setup. Firstly, I will go over cryogenics, particularly how the dilution refrigerator works and how the control lines are constructed to enable the application of voltages to the device with minimal temperature compromise(Section 3.1). I then detail our radio-frequency reflectometry setup (Section 3.2). Following this, I move on to the control software coordinating the instruments required to control our experiments (Section 3.3). Next, I introduce our implementation of high bandwidth measurements, which allow us to acquire two-dimensional

charge stability diagrams in milliseconds (Section 3.4). Finally, I discuss the machine learning used in this thesis, particularly Gaussian processes, Bayesian optimisation and hidden Markov models (Sections 3.5, 3.6 and 3.7, respectively).

3.1 The cryogenics

3.1.1 Principles of dilution refrigeration

The experiments in this thesis were mainly conducted in a Triton 400 dilution refrigerator from Oxford Instruments. Here, I briefly outline the basic principles of dilution refrigeration, followed by a discussion about the control lines.

The first proposal of a cryogenic ^{3}He/^{4}He dilution refrigeration dates back to 1951 by H. London, with its first experimental demonstration occurring 13 years later [100]. It relies on the phenomenon of phase separation: when a mixture of the two stable isotopes of Helium (^{3}He and ^{4}He) is cooled below the critical temperature of $T_c \sim 870$mK the mixture splits into two phases. One of these phases is a lighter "pure" phase (consisting of $> 99\%$ ^{3}He by concentration). The other is a heavier "dilute" phase (consisting of $\sim 7\%$ ^{3}He by concentration). The enthalpy of the dilute phase is higher; therefore, we can obtain cooling by pumping the ^{3}He from the dilute phase into the pure phase. This cooling power is generated at the phase interface and is directly proportional to the forced ^{3}He flow rate. See Fig. 7 for a diagram indicating how such a dilution refrigerator works.

The following describes the protocol for cooling a fridge to its base temperature, where the thermal load equals the cooling power.

1. Precool the dilution refrigerator using the pulse tube cooler to reduce the fridge temperature to approximately 10 K.

2. Pump the ^{3}He/^{4}He mixture through the impedance, causing it to condense in the mixing

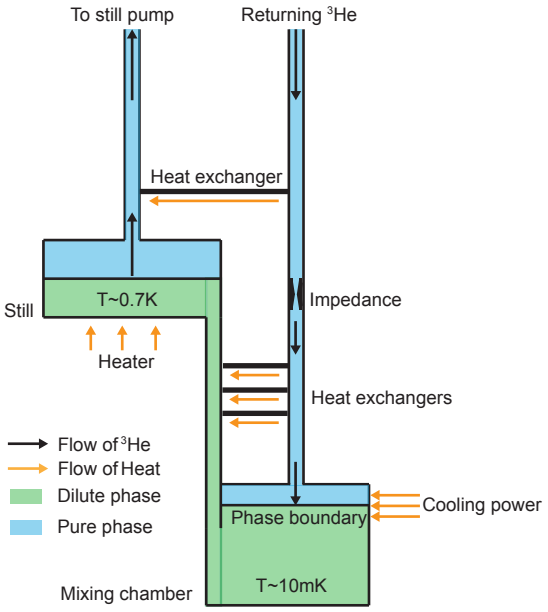


Figure 7: Diagram displaying how a dilution refrigerator works. Black and orange arrows denote the flow of ^{3}He and heat. Areas shaded in green and blue denote where the dilute and pure phases exist, respectively.

chamber and still.

3. The ^{3}He evaporates from the still further cooling it.

4. Eventually, the mixture contained within the mixing chamber and Still cools to the critical temperature, and phase separation occurs into the "pure" and "dilute" phases.

5. The mixing chamber is cooled by the method described above.

3.1.2 Cryogenic lines

Cryogenic lines are how electrical signals are communicated to and from the sample thermally anchored to the mixing chamber. These signals allow for the manipulation and readout of the quantum device. However, each line is a thermal connection between room temperature and the

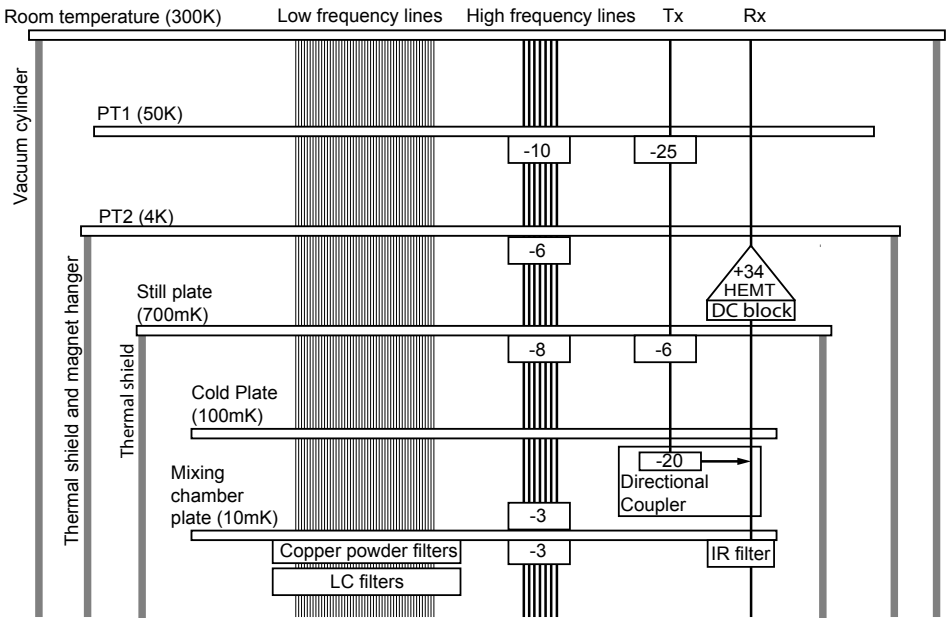


Figure 8: Diagram showing the fridge wiring. There were 48 low-frequency lines used to apply DC voltage to the device gates or ohmics. In addition, there were 7 high-frequency lines used to apply DC pulses to the gates. Finally, the Tx and Rx lines are used for radio-frequency reflectometry, with the tone being applied to Tx and returned through Rx. Attenuators are indicated by rectangles with their attenuation in dB denoted as a negative number inside

coldest part of the fridge. Therefore, they must be carefully attenuated and filtered to minimise how much they warm the sample.

This section discusses the various cryogenic lines used in a semiconducting experiment and their attenuation and filtering. Our experiments required the use of:

- *Low frequency* lines to apply dc voltages to the gates and possibly the ohmics.
- *High frequency* lines to apply fast pulses to the gates.
- *Radio frequency reflectometry lines* used to probe the device's state

See Fig. 8, a diagram of these lines. In the following sections, we will provide more details on these lines.

Low frequency lines

The low-frequency lines were used to apply DC voltages to the device's gates and ohmics. While ohmic biases are typically less than a millivolt, the gate voltages can be up to several volts.

The low-frequency lines consisted of 3 m sections of Constantan loom wires. Due to the voltage requirements of the device gates, it is impractical to attenuate the DC lines; instead, they were low-pass filtered using homemade copper powder filters in series with LC filters, thermalised to the mixing chamber (Fig. 8). These filters removed electrical noise of frequencies between a few tens of kilohertz and a few gigahertz. The resistance of these filters was 5.38 kΩ at room temperature.

High frequency lines

The high-frequency lines were used to apply control pulses and microwave tones to the device gates. The signals sent down these lines could be combined with those of the low-frequency line using a lumped-element RC bias tee on the sample board, with an RC time of 3.3 ms.

These lines consist of semi-rigid coaxial cables running between the plates of the dilution refrigerator (Fig. 8). As the required amplitude of these voltage pulses was only a few hundred millivolts, it was possible to attenuate the lines with 29 dB of attenuation. These attenuators were distributed across the various temperature stages of the dilution refrigerator; see Fig 8.

UT-085-SS-SS Coaxial cables were used for the RT-PT1, PT1-PT2 connections. Superconducting NbTi/NbTi cables were used for the PT2 - Still, Still - Cold Plate and Cold plate - Mixing chamber.

Radio-frequency reflectometry lines

The radio-frequency reflectometry lines allow us to illuminate the device with radio-frequency tones and measure the reflection. Radio-frequency reflectometry is discussed in the next section so that this section will deal only with the cryogenic lines.

There are two radio-frequency lines: the one that imparts the radio frequency and the one that returns the reflection to be measured. We refer to these lines as Tx and Rx, respectively. The imparting line, Tx, consists of an attenuated semi-rigid coaxial cable down to the mixing chamber, with 26 dB of attention. The line was connected to the device at the mixing chamber through a directional coupler, contributing another 26 dB attenuation. Below the directional coupler (Mini-circuits model ZX30-20-4) is an Eccosorb strip-line filter, which filters infrared frequencies. The reflected signal is returned to room temperature electronics via the other port of the directional coupler through the Rx line. This line is again a semi-rigid coaxial cable; however, rather than containing attenuators, it uses a +34 dB high-electron-mobility transistor (HEMT) cryogenic amplifier (The CITLF2 from Cosmic microwave components) to amplify the returning signal before it is demodulated and measured. The amplifier is preceded by a DC block (BLK-18) to protect the device.

3.2 Radio-frequency reflectometry

Here, I will discuss radio frequency reflectometry, covering the basics of high-frequency measurements, matching circuits, demodulation and how a device can be incorporated into an rf matching circuit.

3.2.1 High frequency measurements

To perform high-frequency measurements, you impart a signal and measure what is returned by either reflection or transmission. The returned signal will depend upon the impedance of the device; for example, if measuring in reflection, then the scattering matrix parameter is related to the load impedance according to

$$S_{11}(\omega) = \frac{Z_{\text{load}}(\omega) - Z_0}{Z_{\text{lead}}(\omega) + Z_0}, \tag{3.1}$$

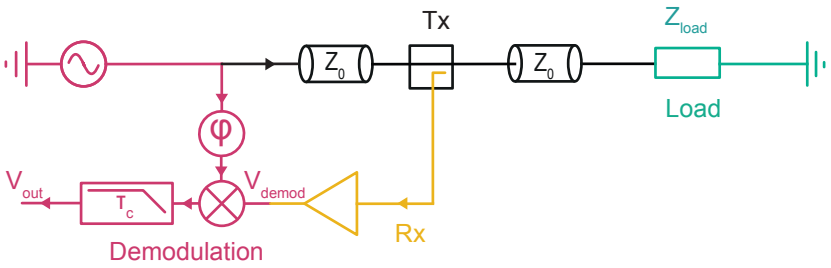


Figure 9: A reflectometry setup, where the section highlighted in pink represents the demodulation setup, where a radio frequency tone is synthesised, divided and demodulated; see section 3.2.1. Half of the tone is directed through the Tx lines depicted in black towards the sample, while the other half is reserved for demodulation at a later stage. The segment of the circuit associated with the sample and the sample matching circuit is shaded in cyan. The Tx line transmits the tone onto this circuit, where a portion is reflected. This reflected signal is retrieved from the dilution refrigerator via the Rx line, shaded in yellow and undergoes amplification along its path. Subsequently, the amplified return signal is demodulated. The Tx and Rx lines are distinguished by their shading in black and yellow, respectively. Figure adapted from Ref. [65]

.

where Z_{load} is the impedance of the device and matching circuit (Section 3.2.2), whilst Z_0 is the transmission line impedance equal to $50\,\Omega$. As the load impedance has both resistive and reactive components, the signal amplitude and phase or in-phase and out-of-phase quadratures will bear information about the device. But what are these quadratures?

Signal quadratures

A periodic signal $V(t)$ of frequency $\nu := \omega/2\pi$ can be mathematically expressed using two quadrature components. Two common choices are amplitude and phase (R, ϕ) and "in-phase" and "out-of-phase" (I, Q), such that an oscillating signal can be expressed in terms of these quadratures as:

$$V(t) = \begin{cases} R\cos(\omega t + \phi) & (R, \phi) \\ I\cos(\omega t) - Q\sin(\omega t) & (I, Q) \end{cases} \tag{3.2}$$

The relation between the two is $R = \sqrt{I^2 + Q^2}$ and $\phi = \arctan(Q/I)$. We can extract these quadratures from a signal using demodulation, which will be discussed next.

Demodulation

Demodulation is the process by which information encoded in signal quadratures is transferred to a lower carrier frequency. It requires a mixer and a low-pass filter.

In homodyne demodulation, we mix with a signal of the same frequency as the carrier signal, taking the quadratures down to zero frequency. In heterodyne demodulation, we mix with a different frequency such that the quadratures are taken to a carrier frequency equal to the frequency difference. We will restrict our discussion to only homodyne demodulation, where the mixing has the effect,

$$I\cos(\omega t) - Q\sin(\omega t) \underbrace{\longrightarrow}_{\text{mixing with } \cos\omega t} \frac{I}{2}\left[1 + \cos(2\omega t)\right] - \frac{Q}{2}\sin(2\omega t). \tag{3.3}$$

Then, the subsequent low-pass filtering extracts the in-phase, I, component alone, such that

$$\frac{I}{2}\left[1 + \cos(2\omega t)\right] - \frac{Q}{2}\sin(2\omega t) \underbrace{\longrightarrow}_{\text{low pass filter}} \frac{I}{2}. \tag{3.4}$$

Extracting the out-of-phase, Q, component is a case of mixing with $\sin(\omega t)$. Both quadratures can, therefore, be obtained by splitting the signal and demodulating each separately, one to obtain the in-phase and the other for the out-of-phase. However, these demodulated components will only contain information about the device if the impedance approximately matches the line impedance using a matching circuit, which will be discussed next. Otherwise, the information-bearing component of the signal is small and drowned out by the noise.

3.2.2 Impedance matching

The role of the impedance matching circuit is to bridge the impedance of the 50 Ω lines to that of the quantum device; otherwise, the difference in impedance would be so great that almost all the signal would be reflected irrespective of the device's impedance, as per Eq. (3.1). For example, open quantum devices typically have impedances similar to the quantum of resistance $h/e^2 \approx 25.8\text{k}\Omega$ resulting in $S_{11} \approx 0.996$.

Typical matching circuits consist of an LC resonance circuit designed so that near its resonance, the impedance is close to the line impedance [65]. For our experiments, we made use of such a circuit consisting of an inductor and a decoupling capacitor in series, along with the parasitic capacitances in parallel with the device (Fig. 10 (a)). The overall impedance of this circuit is

$$Z_{\text{load}} = i\omega L + \frac{1}{i\omega C_d} + \frac{R_s}{1 + i\omega R_s C_p}. \tag{3.5}$$

For the experiments performed in Chapters 4 and 5 we used matching circuits with $L = 2.2$ and 2.7 µH respectively, whilst $C_d = 82\text{pF}$ and $C_p \approx 1\text{pF}$. For these circuit parameters, we have $L/R_s^2 C_p \ll 1$, and this circuit has a resonance where the reactive component vanishes at $\omega_r = 1/\sqrt{LC_{||}}$ where $C_{||} = (1/C_p + 1/C_d)^{-1}$ is the parallel combination of the two capacitors. Substituting this resonant frequency into our equation for the load impedance, we find that

$$Z_{\text{load}} = \frac{L}{C_{||} R_s}\left(1 - 2\frac{\Delta\omega}{\omega_r}\right) + 2i\sqrt{\frac{L_c}{C_{||}}}\frac{\Delta\omega}{\omega_r}, \tag{3.6}$$

where $\Delta\omega$ is the difference between the probing and resonant frequencies. Perfect impedance matching occurs when the impedance on the drive resonance equals the line impedance, such that $Z_{\text{load}} = Z_0$. Under this condition, the device impedance for optimal matching is

$$R_{\text{match}} = \frac{L}{C_{||} Z_0}. \tag{3.7}$$

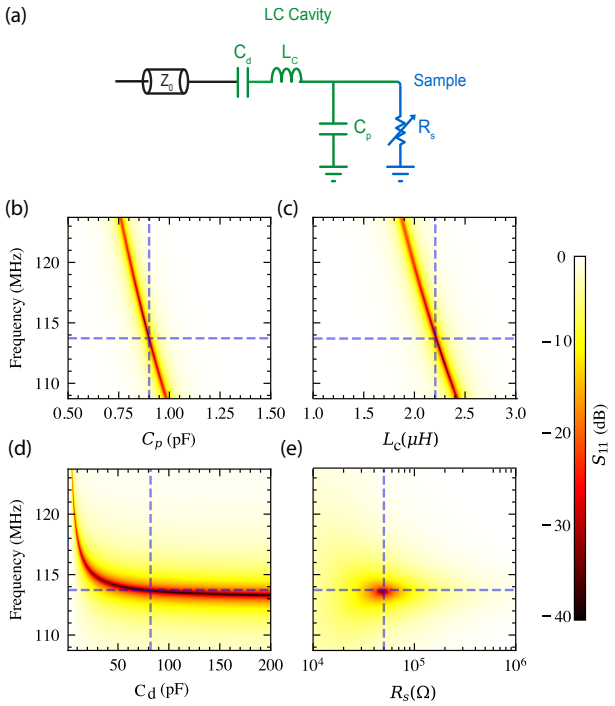


Figure 10: a) Schematic of our matching circuit consisting of an inductor, L, and a decoupling capacitor, C_d, in series along with the parasitic capacitances, C_p, in parallel with the device, which we model as a variable resistor, R_s. b-e) Plots of the dependence of the S_{11} scattering matrix parameter on the value of parasitic capacitance (C_p), the inductance (L_c), the decoupling capacitor (C_d) and the sample resistance (R_s), respectively. For the plots where the component's value is not being swept, they take $C_p = 0.9$ pF, $L = 2.2$ μH, $C_d = 82$ pF and $R_s = R_{\text{match}} = 44$ kΩ. The horizontal blue dashed lines indicate the resonance frequency of 113 MHz, and vertical lines in plots b-d indicate the value chosen for each component. While in e), the vertical line indicates the matching impedance.

The decoupling capacitor serves a dual purpose; firstly, it DC decouples the resonance circuit from other circuits connected by a common feedline. This means it is possible to use a bias tee to apply a DC bias for current measurements. Secondly, the decoupling capacitor provides a useful tuning parameter for the external quality factor Fig. 10 d), therefore making it possible to tune towards critical coupling, where an equal power is dissipated in the load and towards the line [65].

Our radio frequency measurements will be sensitive to device impedances around this matching impedance. If the device impedance is too different from the matching impedance, then the reso-

nance will be lost, and our reflectometry measurements will contain only noise (Fig. 10 (e)).

3.3 Control software

Modern quantum experiments require the intricate control of many instruments in tandem. As a result, this is achieved through a software interface. The earliest and most basic of these software interfaces merely offered the remote control of the instrument's parameters, trading a physical button/knob for a digital one. However, with time, software interfaces have evolved to provide significantly more functionality. This section outlines the various software control interfaces I have used, contributed or even built during my studies. They are presented chronologically based on when I worked with, or on, them - namely Pygor, QCoDeS and my implementation, `qgor`.

3.3.1 Pygor

Pygor was a Python-based custom measurement framework developed by the Ares group (my group); it was successfully used in our and collaborators' labs. The primary aim of the package was to separate the roles of the lab computer and the control computer, allowing the experiment to be controlled remotely. Beyond convenience, this facilitated a machine learning server to handle the experiment, yielding access to significant GPU computing.

However, I found Pygor either slow or inflexible, the two functions trading off against each other. I will now explain this limitation with an example. Suppose the user wishes to perform a gate voltage sweep measuring the current through the device for each gate voltage.

Slow: if the functions to change gate voltages and measure the currents can be exposed from the lab PC to the control server. Then, the server runs the `for` loop to change the gate voltage and measures it repeatedly. In this case, the gate voltage change and measurements pick up the network latency every iteration. Changing the voltage took a minimum of 30 ms, and measurements were also on

the order of a few tens of milliseconds. Depending on the server's location, the communication latency with the lab PC could easily be in the hundreds of milliseconds, and the latency is incurred twice every iteration. Therefore, sweeps in this configuration were considerably slower than they had to be.

Inflexible: if we shift that responsibility to the lab PC rather than the server coordinating the scan. The server specifies the scan needing to be performed and instructs the lab PC. The `for` loop takes on the lab PC, so we only incur the latency at the scan's start and at the end. However, for this, the lab PC has to "know" how to perform the scan. This meant creating the specific functionality on the server ahead of time, which led to inflexibility. The process of prototyping novel measurements was cumbersome and difficult to debug.

The final nail in the coffin for Pygor was that the drivers were custom-written. This meant that using any new instrument meant writing a new driver. This is a time-consuming and challenging task, where bugs could lead to bizarre and possibly device-breaking behaviour. Therefore, I explored an alternative software control framework for our lab, namely QCoDeS.

3.3.2 QCoDeS

QCoDeS is a Python-based data acquisition framework developed by the Copenhagen, Delft, Sydney and Microsoft quantum computing consortium. The goal is a common framework for physics experiments, so new students don't need to spend a long time learning software to participate in experiments; one has to write their code only for pieces that are very specific to their experiment. Code can and should be contributed back to the framework; moving between teams or labs and setting up a new experiment is streamlined, and physics experiments can take advantage of modern software and best practices.

QCoDeS operates on a hierarchical structure where the user interacts with a station of many instrument objects (Fig. 11). Typical instruments held by the station include digital-to-analogue

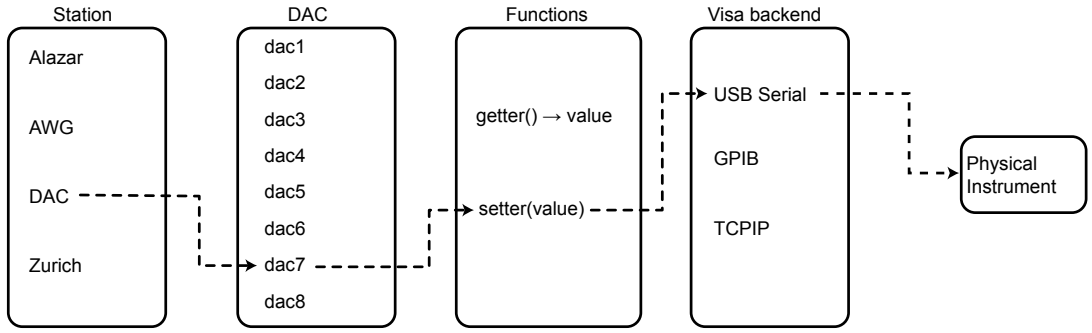


Figure 11: The station-instrument-parameter structure used by QCoDeS to encapsulate instruments and control their experimental parameters. The station contains the instruments: the Alazar, AWG, DAC and Zurich. Instruments, in turn, contain many parameters. These parameters relate to physical quantities, which can be set by the setter function or queried/measured by the getter functions. Once called, the getter and setter function interacts with the visa backend, translating commands to be sent to the instrument over the relevant computer port. Older instruments typically communicate using USB or GPIB. By my observation, instruments produced more recently prefer TCPIP connections to Ethernet cables.

converters (to set voltages), acquisition cards (to measure voltages), and arbitrary waveform generators (to apply control pulses). Each instrument object is comprised of many parameter Python object classes. These parameter objects relate to individual experimental quantities that need to be controlled, measured, or both. Parameter objects carry information such as the parameter's units and its permitted values if it is controllable. In addition, controllable parameters hold a "setter" function. This is the function the user calls to set the value to the experimental parameter in question. For example, one would call the setter function to choose a voltage or set an oscillator frequency. Such controllable parameters will also have a "getter" function, which queries the instrument asking for the current set value. By contrast, some parameters are not set to control the experiment; they are instead measured. Such parameters have no "setter" function, and their "getter" functions prompt the instrument to perform a measurement, which is then returned. In both cases, the "setter" and "getter" functions communicate with the instrument through the VISA backend, which is an abstraction layer which translates VISA commands from a human-readable form to the specific serialised form required to send it over the bus to the instrument. For example,

a very different byte string would be sent depending on whether the instrument is plugged in via USB or Ethernet.

QCoDeS' most significant strength is its repository of drivers. In total, there are over 100 different drivers immediately accessible to the user, each with a consistent structure and written to professional software standards. Furthermore, many drivers are highly documented, saving hours troubleshooting to get them to work. Finally, QCoDeS is such a software standard that some instrument manufacturers maintain drivers for their instruments.

However, QCoDeS provides more than an experiment's instrument drivers. It is a data acquisition and handling framework. Therefore, it contains functions to sweep and measure combinations of parameters. The syntax for these measurements is:

```
do2d(
  dac.dac1, dac.dac2,               # the two independent parameters to sweep
  adc.v1, adc.v2,...,               # the dependent parameters to measure
  measurement_name="dond_example",  # the measurement name
  experiment=tutorial_exp,          # the experiment to save data to
)
```

Listing 3.1: The function calls for a two-dimensional raster pattern sweep in QCoDeS.

In this measurement, the two independent parameters are swept whilst the two dependent parameters are measured. As this data is acquired, it is saved into an SQL database so that the measurement and experiment name can be used to query it. This allows the measurement to be reloaded at any point without losing data or metadata. Unfortunately, QCoDeS did not come with a live plotter to plot the data as it is acquired[1]. Instead, the user must wait until all data is acquired, and then the dataset can be plotted. Furthermore, the saving of data into the SQL

[1]Live plotting functionality has been created to work with QCoDeS, but it stands as a separate package. It is called plottr.

database happens on the main thread, meaning that it will contribute to the overall measurement time[2].

In addition, the SQL database necessitated complex data formatting, which made it time-consuming to upload and extract raw data from the database. This became problematic, for example, when I started developing high-bandwidth rf measurements. I could acquire data far faster than QCoDeS could save it. And the lack of live plotting made it challenging to benefit from the speed. I found further issues when I wanted to do live data processing pre-plotting. This led me to create `qgor`, which was directly inspired by, built off and inherited from QCoDeS.

3.3.3 Qgor

`Qgor` is my fork of QCoDeS; it inherits what I liked in QCoDeS while improving/adding functionality necessary to leverage radio-frequency measurements fully. As a result, `qgor` was designed from the ground up to support considerably higher data acquisition rates and volumes than QCoDeS. This necessitated redesigning the data processing, plotting and saving compared to that used in QCoDeS. In this section, I will describe some of the functionality I built for qgor and provide an overview of the coding principles used in making each piece.

Data flow and multithreading

When designing the data processing, plotting and saving functionality, I focused on the data flow. The measurements create data whilst it needs to be saved in a database. However, there should also be the opportunity to process and live plot the data. Data processing is a hugely encompassing term, it could be as simple as averaging a 2d dataset along an axis up to passing it through a neural network to locate triple points in a change stability diagram. The plotting is another important

[2] As of September 2022, support for multithreaded data processing was added to QCoDeS, approximately two years after I wrote qgor. Their implementation seems to work using a data pipe to push data off the main thread to a secondary thread that saves data.

component, as the plotter must support a frame rate that is high enough so that the user can observe their measurement in real-time.

As both the plotting and data processing could be computationally expensive, it became apparent that these tasks should happen on a different thread than the one coordinating and performing the measurements. This avoids the plotting and processing compute time, contributing to the overall measurement time. However, it was not clear where to put the data saving. There were three options:

1. On the main thread, the disadvantage is that data-saving times will contribute to the overall measurement time.

2. On the data processing and plotting thread. On the face of it, this initially seemed like the best option; the data was already being passed to this thread, so why not save it as well? However, data processing and plotting are rather error-prone operations. With the data-saving on the same thread, the plotter or processor failing risks losing experimental data.

3. On its own thread. The cons of this method arise from counterintuitive Python subtleties. Python global interpreter lock means spawning new threads requires cloning the entire interpreter. This is not a cheap operation. So spawning two threads rather than just one is wasteful, but as this happens just once, it is manageable. However, each piece of measured data must be sent to both threads. In Python, a pipe is recommended for sending data between threads. Everything going through this pipe must be pickled (serialised), which is annoying because it means that one cannot send NumPy arrays and is computationally expensive. Benchmarks I ran showed that pickling an object, then sending it through the pipe and pickling an object to save it took approximately the same time - making having a separate thread to do the data saving pointless.

Ultimately, I picked option one with a twist; I discovered a niche piece of `NumPy` functionality

called memory-maps, which are `NumPy`-like arrays where the data is stored in a binary file on disk rather than in active memory. When the user changes a value in this array, the change is immediately flushed to disk, and when a value is queried, it is loaded. As with everything `NumPy`, this functionality was heavily optimised (well beyond what I would write), which, when paired with SSDs' write speeds, made saving data to this format so fast there was no reason not to do it on the main thread.

There were, however, advantages beyond speed; memory maps also elegantly solved the problem of moving data between threads. If the main thread was coordinating the measurement and writing data to the memory map, I found that I could create a read-only memory map on the data processing and plotting (DPP) thread, which pointed to the same location, thereby making most current data accessible, see (Fig. 12). The `NumPy` backend handled keeping this data up to date. The program running on the DPP thread was simply a while loop; if the data had changed, it processed and plotted the data. If the processing creates new data, this is easily stored on disk by a separate memory map with write permission. There is no risk of data races because only one thread can have a memory map with write permissions.

Furthermore, by constructing the dataflow in this manner, the user has the guarantee that any data they have seen plotted has to be saved in the database[3]. Fig. 12 shows a data flow schematic.

3.4 Fast two-dimensional scans

The bandwidth of rf measurements permitted measuring charge stability diagrams in milliseconds, approximately four orders of magnitude faster than an equivalent DAC scan used in the lab. However, accessing this speed necessitated using an AWG, rather than the DAC, to sweep gate voltages, introducing significant experimental control complexity. These scans represent a signif-

[3]There is the possibility of the main thread overwriting data; however, it falls to the user to write measurement functions which do not. Defensive design can only go so far.

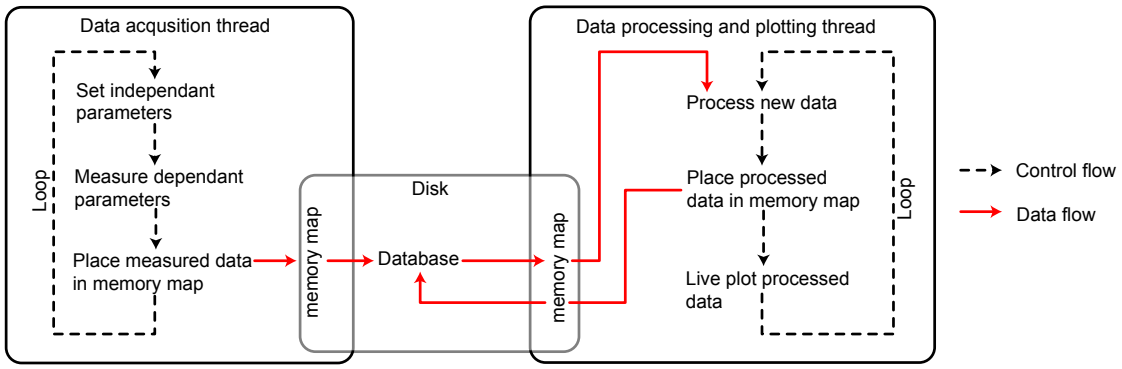


Figure 12: The data flow, indicated by red arrows, occurs during a typical qgor measurement. Data is acquired on the data acquisition thread, typically the main thread, where it is placed into a NumPy memory map. The NumPy backend immediately saves it to disk as a binary file; thus, the data is saved to the database. On the data processing and plotting thread, a memory map with read-only permission makes the same data accessible, where it can be processed and plotted. Another memory map with write permissions saves any new data created in the data processing stage. The grey arrows indicate the flow of Python programs running as part of the qgor data acquisition, saving, processing and plotting.

icant change in how data is acquired, array by array rather than point by point. This change is what prompted the creation of `qgor`. I could not adequately get QCoDeS to support this new measurement paradigm.

This section outlines how I implement these fast scans. I will present two iterations; the first is a reimplementation of raster-pattern scans presented in the literature [101, 102]. These scans could be acquired so quickly that when coupled with `qgor`'s live plotter they resembled video. Therefore, they have the nickname video mode [101]. With these scans, I demonstrated an automated tuning algorithm using rf exclusively and the automation of a charge sensor (see Chapters 4 and 5 respectively).

However, the duration and resolution of these raster scans were limited by the bias tee's time constant. This limitation was keenly exposed when using these scans to tune a charge transition capable of hosting a singlet-triplet qubit. This led me to create a second iteration of the fast scans, which swept gate voltages in a spiral pattern, thus circumventing the limitations of the raster

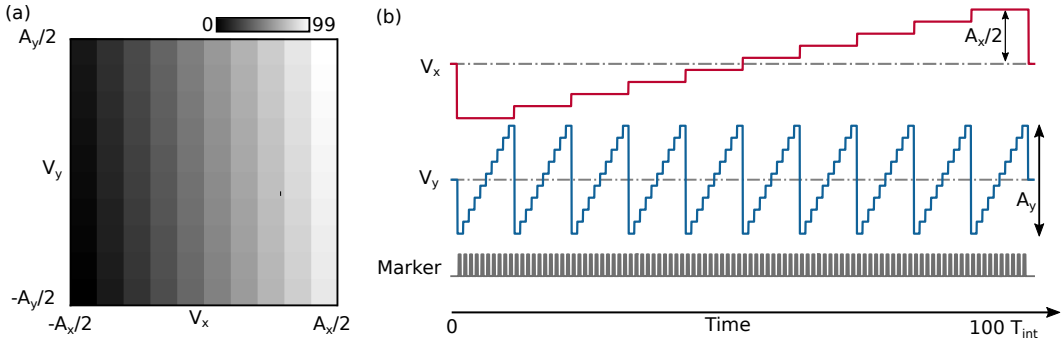


Figure 13: a) A diagram showing how a 10×10 raster measurement navigates through gate voltage space, where the pixel value indicates the order in which the pixel is measured. b) The red and blue curve voltages to be applied to the V_x and V_y gates must navigate through the gate voltage space in the raster pattern. The grey line is a rising-edge marker required to trigger an rf measurement at the correct time.

scan. Without these limitations, I could interleave a $S - T_-$ pulse sequence with the spiral scan. Searching for transitions showing spin physics through the artefact associated with the $S - T_-$ avoided crossing with these measurements (Section 3.4.2).

The original raster scans are presented in Section 3.4.1. Then, the improved spiral scans in Section 3.4.2, with the interleaving of the pulses in the following subsection.

3.4.1 Raster pattern

My first iteration of the fast two-dimensional scans followed a raster pattern, as demonstrated in Refs. [101–103]. This was achieved using a pair of stepped-sawtooth waveforms applied to the sample through the high-frequency lines. Radio-frequency reflectometry was used to measure the device. The acquisition of the demodulated rf signal was synchronised by a marker.

To perform a fast measurement of $N \times N$ pixels, I used an AWG to synthesise two waveforms, labelled V_x and V_y (Fig. 13) where the integration time per pixel is τ_{int}. The V_y waveform consisted of a "fast" stepped sawtooth of period $\tau_y = N\tau_{\text{int}}$. The "slow" V_x period was N times the "fast" V_y's equalling $\tau_x = N^2\tau_{\text{int}}$. However, for modest resolutions, say 100×100, this "slow" waveform

was susceptible to distortion as the period of the waveform was comparable to the time constant of the bias-tees ($\tau_{RC} \sim 3.3\,\text{ms}$). I was able to pre-compensate the waveform for the distortion introduced by the bias-tee, such that the correct waveform arrived at the device gate, discussed in the next paragraph. However, even with this pre-compensation, I was limited in the integration time and resolution of the scans I could perform due to the finite output voltage range of the AWG. To mitigate the short integration times, I could average multiple passes across the two-dimensional scan together such that I measured each pixel multiple times.

To perform the pulse pre-compensation, I modelled the pulse distortion introduced by the bias-tee as a high-pass filter. The action of a first-order high-pass filter with a time constant τ on a signal $s(t)$ is $s_{\text{filtered}}(t) = \mathcal{F}^{-1}[\mathcal{F}[s(t)](\omega) \cdot H_{HP}(\omega)]$ where $H_{HP}(\omega) = i\omega\tau/(1 + i\omega\tau)$, and $\mathcal{F}[\cdot]$ denotes a Fourier transform. To correct for this effect, we pre-compensated the waveforms according to $s_{\text{comp}}(t) = \mathcal{F}^{-1}[\mathcal{F}[s(t)](\omega) \cdot H_{HP}^{-1}(\omega)]$, such that the final shape reaching the device gates would be exactly $s(t)$.

3.4.2 Spiral pattern

The distortion issue faced by the raster pattern prompted us to consider paths which traverse every pixel in a two-dimensional grid in a way which is less affected by the high-pass filter. We found a spiral pattern highly effective, with the longest "periods" being $\tau_{x/y} = 2N\tau_{\text{int}}$ (Fig. 14). So, by switching to a spiral pattern, we could perform fast two-dimensional scans without filter compensation. Another advantage of the spiral pattern over the raster was that it did not require instantaneous jumps, which could damage the device if they were too large.

The spiral pattern also permitted measuring with higher resolution or spending more time at each pixel. When tuning a qubit, we found that the extra time at each pixel could play a pulse sequence designed to mix (if possible) the S ground state with the T_- excited state before measuring. This T_- mixing introduced an artefact into the charge stability diagram, which strongly hints that a

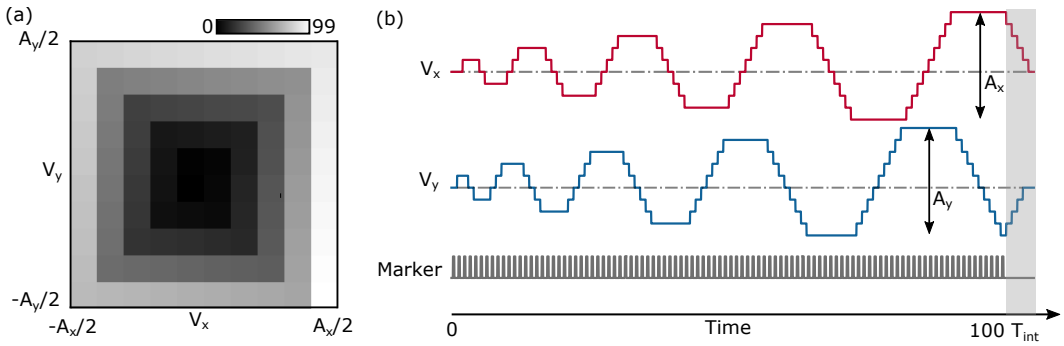


Figure 14: a) A diagram showing how a 10×10 pixel spiral measurement navigates through gate voltage space, where the pixel value indicates the order in which the pixel is measured. b) The voltages which we apply to the V_x (red) and V_y (blue) in order to navigate through the gate voltage space in a spiral pattern. The grey line is a rising-edge marker required to trigger an rf measurement at the correct time. The region shaded in grey indicates the times where V_x and V_y are ramped to zero to prepare for the next measurement.

qubit could be formed at that transition (Fig. 15 (a-d)). The ability to perform these scans quickly helped enormously when looking for elusive qubit features; however, this work is not presented as part of the current thesis.

3.4.3 Video and video game mode

The combination of the fast two-dimensional scans and `qgor`'s efficient data pipelines made it possible to acquire, process and plot many times a second, meaning that the user is shown a video of the device's current state. However, if the fast scan is taking place on the main thread, the device features shown in the video will be static and unchanging up to the shaking caused by charge noise.

As a result, I decided to further enhance the functionality by building video-game mode, where the user could alter the device's state whilst the continuous video measurements were displayed. Building this necessitated moving the measurement and acquisition off the main thread. This new thread continuously performed the last 2d scans and then passed the data through the memory map to the data processing and plotting thread (Fig. 12). The main thread retained control of all

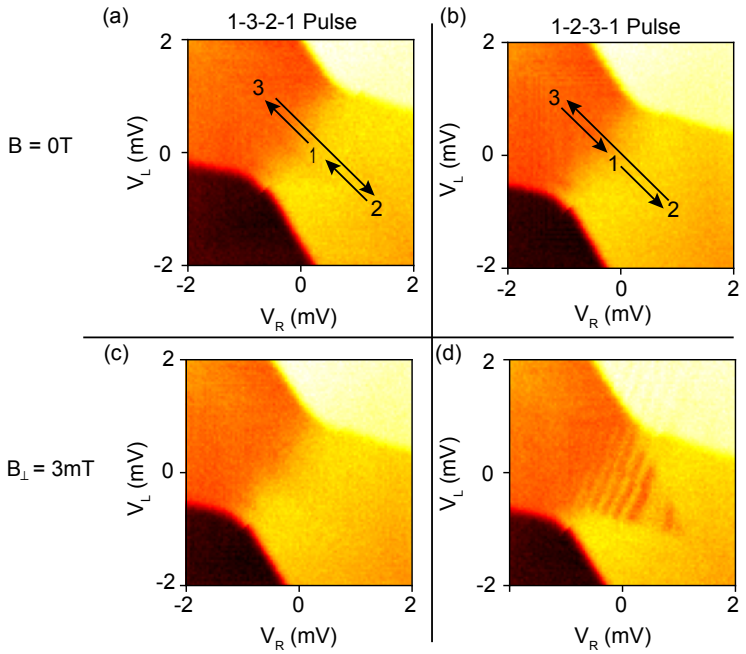


Figure 15: Measurements of the (1,1)-(2,0) interdot transition in a hole-based Ge/SiGe depletion device as measured with fast radio-frequency measurements on an rf-set using a fast two-dimensional scan. The panels show the outcome of this measurement for different pulse sequences and under different magnetic field conditions. The 1-2-3-1 pulse sequence pulses from one to two, waits for 100ns, then pulses to three, again waits for 100ns. And finally, pulses to four, where the measurement is performed. Whilst the 1-3-2-1 pulse goes to three first and then to two. (a) Measurement performing the 1-3-2-1 pulse under no magnetic field. (b) Measurement performing the 1-2-3-1 pulse under no magnetic field. (c) Measurement performing the 1-3-2-1 pulse under a magnetic field of 3mT perpendicular to the sample. (d) Measurement performing the 1-2-3-1 pulse under a magnetic field of 3mT perpendicular to the sample. This combination of pulse and field features associated with the Pauli spin blockade of the T_- spin state are present.

the instruments not involved in the 2d scans, and the user could alter any of their parameters to see the effect immediately. Naturally, parameters such as DAC voltages were bound to keys on the keyboard to make quantum device tuning like a video game.

3.5 Gaussian processes

Gaussian processes (GPs) are a powerful framework for statistical modelling and predicting data [104, 105]. They are a non-parametric approach that provides a flexible way to model complex, non-linear relationships between variables. They provide a flexible and probabilistic approach to regression, interpolation, and uncertainty quantification. GPs will be used in Chapter 4 to create an algorithm which automatically tunes quantum dots using exclusively radio-frequency measurements. Here the intention is to simply introduce GPs mathematically and discuss their key properties.

In formal terms, a Gaussian process represents a collection of random variables, where any finite subset follows a joint Gaussian distribution [104]. When employing Gaussian processes for regression tasks, the distribution is characterised by a mean function $\mu(x)$ and a covariance function (often termed as the kernel) $k(x, x')$. Thus, we define it as:

$$\vec{f} \sim \mathcal{GP}(\vec{\mu}, k(\vec{x}, \vec{x}^{\mathsf{T}})), \tag{3.8}$$

Here, f denotes the function we intend to model, with $\vec{f} = [f(x_1), f(x_2), \ldots, f(x_n)]$ representing samples of this function at coordinates $\vec{x} = [x_1, x_2, \ldots, x_n]$. Where mean and covariance functions

are evaluated element-wise, thus

$$\vec{\mu} = [\mu(x_1), \mu(x_2), \ldots, \mu(x_n)] \text{ and } k(\vec{x}, \vec{x}^{\intercal}) = \begin{bmatrix} k(x_1, x_1) & k(x_1, x_2) & \ldots & k(x_1, x_n) \\ k(x_2, x_1) & k(x_2, x_2) & \ldots & k(x_2, x_n) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_n, x_1) & k(x_n, x_2) & \ldots & k(x_n, x_n) \end{bmatrix}. \tag{3.9}$$

Mean and covariance functions

The mean function $\mu(x)$ represents the expected value of the GP at any given input point x. It encodes our prior beliefs about the function. Common choices for $\mu(x)$ include constant values or linear functions.

The covariance function $k(x, x')$ specifies how the values of the GP are correlated as a function of the input locations x and x'. This function determines the smoothness and shape of the sampled functions. A popular choice for $k(x, x')$ is the Radial Basis Function (RBF) or Gaussian kernel [105]:

$$k(x, x') = \exp\left(-\frac{1}{2\ell^2}\|x - x'\|^2\right) \tag{3.10}$$

Here, ℓ determines the length scale, influencing the smoothness of the functions generated by the GP.

Predictions and inference

Gaussian processes can make predictions for input points $\vec{x}_*$ by calculating the conditional distribution of $\vec{f_*} = [f(x_{*1}, f(x_{*2}, ..., f(x_{*n})]$ given the observed data. This conditional distribution is also Gaussian,

$$\vec{f_*}|\vec{x}, \vec{f}, \vec{x}_* \sim \mathcal{N}(\vec{\mu}_*, \Sigma_*) \tag{3.11}$$

where posterior mean $\vec{\mu}_*$ and covariance Σ_* of this distribution incorporation the samples in $\vec{f}$ [105], such that

$$\vec{\mu}_* = \vec{\mu} + k(\vec{x}_*, \vec{x})k(\vec{x}_*, \vec{x}_*)^{-1}\vec{f}, \tag{3.12}$$

$$\Sigma_* = k(\vec{x}_*, \vec{x}_*) - k(\vec{x}_*, \vec{x})k(\vec{x}_*, \vec{x}_*)^{-1}k(\vec{x}, \vec{x}_*). \tag{3.13}$$

This posterior distribution interpolates the samples of f, quantifying its uncertainty. As shown in Fig. 16, in regions where there are many points (such as around $\theta = 1$), the Gaussian process is confident in its predictions. Whilst around $\theta = 6$, the nearest points are far afield, resulting in the Gaussian process quantifying less confidence in its predictions.

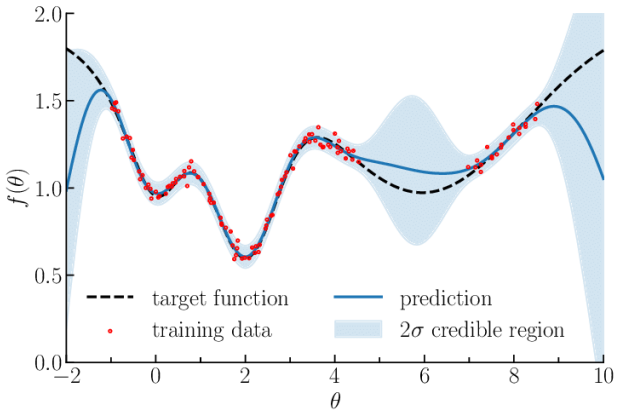


Figure 16: Illustration of Gaussian process regression on the test function $f(\theta) = 2 - \exp(-(\theta-2)^2) - \exp(-(\theta-6)^2/10)$. Samples from the test function, subject to a Gaussian observation noise with standard deviation $\sigma = 0.03$, are shown as red points. The blue line and shaded regions show the Gaussian prediction's predicted mean and two standard deviation confidence intervals. Figure adapted from Ref. [106]

Accounting for noisy observations

Another key benefit of GPs is their ability to accept and account for the noise in observations. This feature is particularly advantageous when observations are costly, making it impractical to average

multiple observations at a single point. Those observations would be better spent by distributing them across the functions landscape. In addition, by incorporating noise directly into the model, GPs offer a more realistic representation of the underlying processes, leading to more robust and reliable predictions, even in the presence of uncertainty.

For example, suppose that we are unable to sample our function f perfectly and instead obtain noisy observations of the form $\vec{t} = \vec{f} + \vec{n}$ where $\vec{n} \sim \mathcal{N}(0, \sigma^2 I_n)$ where I_n is the n-d identity matrix. In this case, the distribution of the noisy observations is conditioned upon the distribution of f, such that $p(\vec{t}|\vec{f}, \sigma^2) = \mathcal{N}(\vec{f}, \sigma^2 I_n)$. We can integrate out the dependence on f to obtain a distribution for just t

$$p(\vec{t}|\sigma^2) = \int p(\vec{t}|\vec{f}, \sigma^2) p(\vec{f}) d\vec{f} = \mathcal{N}(\vec{\mu}, k(\vec{x}, \vec{x}^T) + \sigma^2 I_n). \tag{3.14}$$

Therefore, the noisy samples of f also can be described by a Gaussian process, with a modified kernel to quantify the degree of noise. This is evident in Fig. 21, where there is some residual uncertainty in the GP's predictions even for angles θ, which exactly match that of a sample.

Hyperparameters

Gaussian Processes have hyperparameters such as σ^2 and ℓ in the covariance function. These hyperparameters control the behaviour of the GP and are typically optimised through maximum likelihood estimation. Analytical formulas exist for the log-likelihood gradient with respect to the hyperparameters so that numerical gradient-based optimisation routines can be used [105].

3.6 Bayesian optimisation

Gaussian processes form the basis for Bayesian Optimisation (BO), a powerful technique for optimising expensive and noisy objective functions [106]. This section will discuss the connection

between GPs and BO, highlighting how GPs serve as a fundamental tool for the optimisation paradigm. Bayesian optimisation will be used in Chapter 5 to automate a charge sensor.

Gaussian processes as surrogate models

In Bayesian optimisation, the goal is to find the global optimum of an unknown, expensive-to-evaluate objective function $f(x)$. Instead of directly querying $f(x)$, which can be time-consuming or costly, Bayesian optimisation employs a surrogate model to approximate the true function (Fig. 17). GPs are a natural choice for constructing these surrogate models due to their probabilistic nature and ability to capture uncertainty [106].

The GP surrogate model provides a probabilistic estimate of the objective function at any given input point x, as previously defined in equation (3.8). The surrogate GP can predict the mean $\mu(x)$ of the objective function and the uncertainty associated with this prediction, captured by the variance $\sigma^2(x)$ (represented by solid blue line and blue shaded region in Fig. 17). This uncertainty is a crucial aspect of Bayesian optimisation, as it guides the exploration of the input space.

Acquisition functions

To decide where to sample the objective function next, Bayesian optimisation puts the surrogate GP model into an acquisition function, denoted as $\alpha(x)$ and as a green line in Fig. 17. The acquisition function balances exploring uncertain regions with exploiting promising areas to find the global optimum efficiently. Several commonly used acquisition functions include Probability of Improvement (PI), Expected Improvement (EI), and Upper Confidence Bound (UCB).

The choice of acquisition function depends on the specific optimisation problem and the trade-off between exploration and exploitation. These acquisition functions leverage the GP surrogate model's predictions and uncertainties to make informed decisions about where to evaluate the true

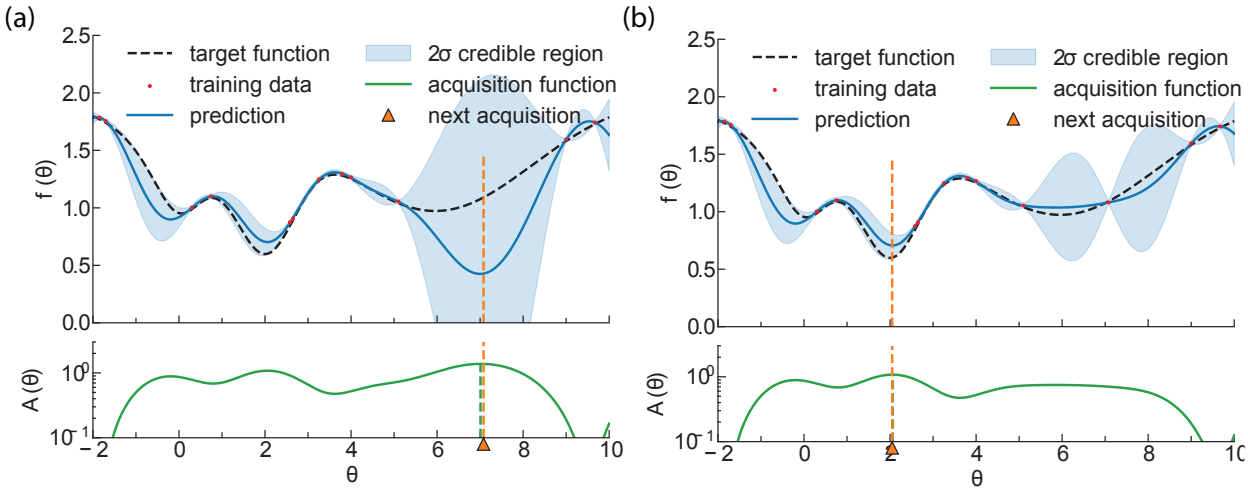


Figure 17: a-b) Illustration of two consecutive steps of a Bayesian optimiser which is attempting to minimise the test function of Figure 16. For each step, the top panel shows the training data points (red points) and the regression (blue line and shaded region). The bottom panel shows the acquisition function (the expected improvement, solid green line) with its maximiser (dashed green line). The next acquisition point, i.e. where to run a simulation to be added to the training set, is shown in orange; it differs from the maximiser of the acquisition function by a small random number. The acquisition function used is the expected improvement, which aims to find the minimum of f. Hyperparameters of the regression kernel are optimised after each acquisition. Figure adapted from Ref. [106].

objective function.

Sequential optimisation

Bayesian optimisation proceeds iteratively by selecting the next input point to evaluate based on the acquisition function, observing the true objective function at that point, and updating the GP surrogate model. Fig. 17 shows iterations 11-14 of a Bayesian optimiser minimising the test function from Fig. 16. The iterative process continues until a stopping criterion is met.

By sequentially updating the surrogate model and selecting input points with the highest expected improvement, Bayesian optimisation efficiently explores the input space and converges towards the global optimum while minimising the number of function evaluations.

Hyperparameter tuning

Gaussian Processes within Bayesian optimisation also offer a convenient framework for hyperparameter tuning in machine learning models. Parameters such as the length scale ℓ and noise level σ^2 of the GP covariance function can be optimised using Bayesian methods. This allows for automated and data-driven selection of hyperparameters, improving model performance.

3.7 Hidden Markov models

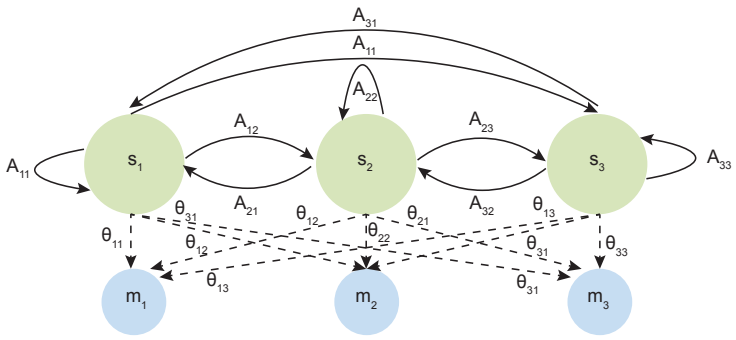


Figure 18: Diagram depicting hidden Markov model with three hidden states $\{s_1, s_2, s_3\}$ and three observable states $\{m_1, m_2, m_3\}$. The elements of the A matrix encode probabilities of transition from state i to j state, denoted by solid arrows. The elements of the emission matrix θ encode the probability of measuring the ith observable state when in the jth hidden state, denoted by dashed arrows.

Hidden Markov models (HMM), are one of the most popular methods in machine learning and statistics for modelling sequences such as speech [107] and proteins [108, 109]. An HMM defines a probability distribution over sequences of observations $\vec{m} = \{m_0, m_1, ..., m_{N-1}\}$ by using another sequence of unobserved hidden discrete states $\vec{s} = \{s_0, s_1, ..., s_{N-1}\}$. These hidden states evolve according to a Markov chain, defined by a transition matrix such that $A_{ij} := P(s_{n+1} = i \mid s_n = j)$.

In a categorical HMM, a special case of a HMM, the probability of an observation is conditioned on the hidden state according to an emission matrix defined as $\theta_{ij} := P(m_n = i \mid s_n = j)$. One final

parameter is required to define the HMM, a probability vector encoding the probability of starting in a hidden state $\vec{\pi}_i = P(s_0 = i)$ [110].

For HMM models, there exist three important algorithms:

1. The *Baum-Welch* algorithm [111], which, given a set of observations, $\vec{m}$, performs expectation maximisation to obtain the most likely set of HMM parameters specified by $(\vec{\pi}_i, A_{ij}, \theta_{ij})$.
2. The *Forward-Backward* algorithm, which computes the likelihood of a set of observations, $\vec{m}$, given some HMM parameters, $(\vec{\pi}_i, A_{ij}, \theta_{ij})$.
3. The *Viterbi* algorithm [112], which given a set of observations, $\vec{m}$, and a set of HMM parameters $(\vec{\pi}_i, A_{ij}, \theta_{ij})$ it finds the most likely set of hidden states, $\vec{s}$.

Another flavour of HHM is a Gaussian HHM, where there is a continuum of possible emission states; the emission matrix is instead a Gaussian probability density function. The algorithms listed above can be also be applied to Gaussian HHMs.

We will apply HHMs to SPAM and spin-flip error analysis and to qubit active reset in Chapter 7.

Chapter 4

All rf-based tuning algorithm for quantum devices using machine learning

> Never spend 6 minutes doing
> something by hand when you can
> spend 6 hours failing to automate it.
>
> Zhuowei Zhang

With previous tuning algorithms, the slow current measurements mean the tuning algorithms were measurement-limited, meaning they had to make intelligent decisions on where to measure next simply because measurements were so time-consuming. This chapter explores tuning using fast RF measurements, lifting this bottleneck. The work presented in this chapter resulted in the publication:

Barnaby, v. S., Fedele, F., Vigneau, F., Hickie, J., Jirovec, D., Ballabio, A., Chrastina, D., Isella, G., Katsaros, G., & Ares, N. *All rf-based tuning algorithm for quantum devices using machine*

learning. arXiv.org. `https://arxiv.org/abs/2211.04504`

In this chapter, we demonstrate an algorithm that tunes a quantum device using exclusively rf measurements, eliminating the need for DC transport measurements. This should allow for more scalable device architectures. The algorithm aimed to define double quantum dots, a cornerstone of the solid-state qubit effort, from scratch. These double dots can host either a singlet-triple qubit or a pair of Loss-Divincenzo spin qubits [113, 114].

Our algorithm exploits the bandwidth of rf measurements to acquire high-resolution two-dimensional charge stability diagrams in milliseconds, as presented in [101, 102], the implementation of which we discussed previously in Sec. 3.4. Acquiring and processing the data from these scans was what `qgor` was built for (Sec. 3.3.3). The speed of these scans was a paradigm shift compared to current-based tuning algorithms, where similar resolution transport measurements would take minutes. However, this speed came at a cost; rf measurements are only sensitive when the device impedance is close to the rf matching circuit (Sec. 3.2.2). This is true in a small portion of the device gate voltage space, outside which the rf signal corresponding to device features falls below the noise floor.

4.1 Experimental setup

I will discuss our experimental setup to demonstrate our rf tuning algorithm here.

4.1.1 The device

We demonstrate our tuning algorithm in a hole-based quantum dot array hosted in a depletion mode Ge/SiGe heterostructure, which can be operated as two distinct double dots devices, labelled A and B, where five voltage gates shape the electrostatic potential (Fig. 19 (a)). The fabrication process of this device is similar to that described in [31]. AC and DC signals were applied to the plunger gates for each quantum dot, V_2, V_4 and V_6, via 3.3 kHz bias-tees. Measurements were

performed at 50 mK.

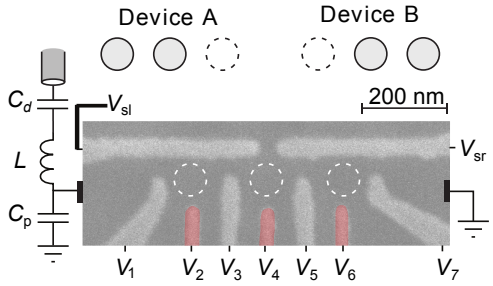


Figure 19: A scanning electron micrograph of the Ge/SiGe quantum dot array, with our matching circuit connected to the device ohmic. Bias tees are used on gates shaded in red to apply both ac and dc voltages. Devices A and B demonstrate how two different double dots can be defined in the array.

4.1.2 The radio frequency setup

For our rf setup, we opted to connect the matching circuit to the device ohmic, thereby creating an rf setup most closely analogous to that of transport measurements through the device. The idea was to bootstrap knowledge from previous algorithms. Our rf circuit board was nominally identical to the one demonstrated in [115]. It had spaces for an inductor and decoupling capacitor in series with the parallel combination of the device and another capacitor. To match as large device impedances as possible, we choose the inductor to be $L = 2.2\,\mu\text{H}$ inductor (one of the largest we had available) whilst leaving the slot for the parallel capacitor empty, thereby minimising the parasitic capacitance. We engineered the matching circuit using these circuit elements to meet the matching condition at the highest device impedance possible, according to Eq. (3.7). We chose the decoupling capacitor $C_d = 92\,\text{pF}$ based on room temperature experimentation without the device bonded, manually optimising the depth of the resonance.

The fast scans

The device measurements consisted of fast raster two-dimensional gate voltage scans over the device plunger gates, as discussed in Sec. 3.4.1. The waveforms were synthesised by a Tektronix 5014c AWG, with an amplitude of 100 mV × 100 mV and 100 by 100 pixels, respectively. We used an integration time of 1 µs per pixel, enabling an entire scan to be completed in 10 ms. We employed an Alazar ATS9440 acquisition card, which was hosted in a PCIe port of the lab PC. This card was capable of using eight PCIe lanes for data transfer rates up to 1.6 GB/s, meaning that the data transfer time for these scans was 0.4 ms.

4.2 The tuning algorithm

Our rf tuning algorithm comprises two key components: a navigation strategy for the device's gate voltage space and a scoring function to evaluate the presence and quality of double dot features. Navigating the device gate voltage space is somewhat more complicated than using transport measurements because the rf measurements are only sensitive when the device impedance is approximately compatible with the matching impedance. This rules out the traditional transport algorithm's approach of looking for the marked increase in impedance when the device pinches off, delineating the device's transition from an "open" to a "closed" state [116, 117]. We cannot search for this feature through rf measurements as, if we were to design our matching circuit for these relatively open dot features; we would be unable to measure more depleted dots with a higher impedance. As discussed in the previous Section 4.1.2, we matched as large device impedances as possible, thereby sacrificing the ability to measure the pinch-off hypersurface.

Accordingly, instead of searching for the pinch-off hypersurface, we defined an rf hypervolume, which is the volume of gate voltage space where the rf matching network is sufficiently sensitive to impedance changes in the quantum device, due to the tunnel barriers being of appropriate thickness

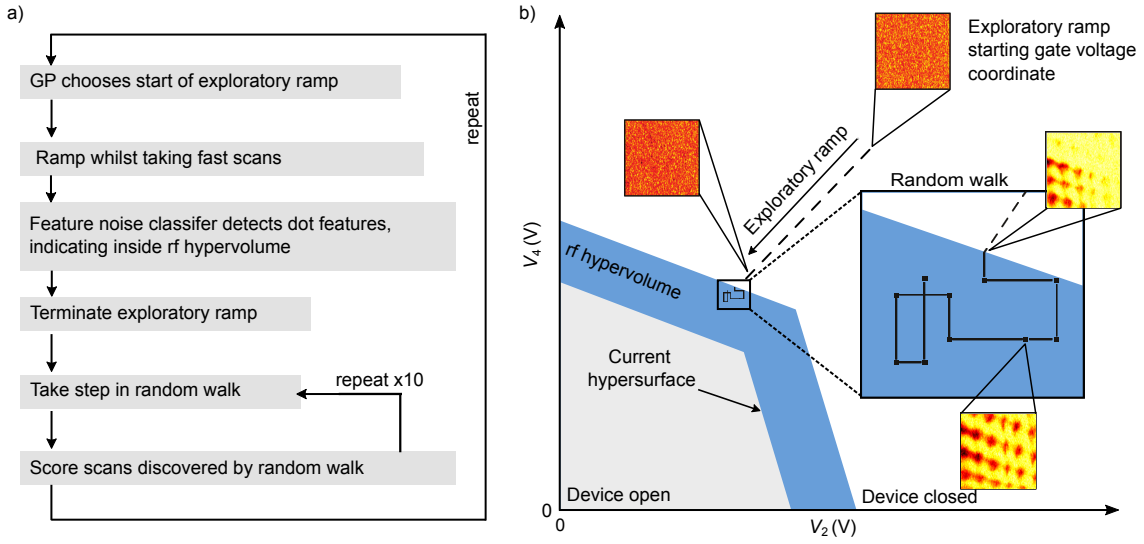


Figure 20: a) A flow diagram indicating how the rf tuning algorithm works. b) An illustrative diagram showing an iteration of the algorithm. Firstly, the algorithm performs an exploratory ramp along a randomly chosen direction, where a Gaussian process chooses the starting gate voltage coordinate to lie close to, but still outside, the rf hypervolume. Along the length of the exploratory ramp, fast two-dimensional scans are performed. The rf classifier evaluates these scans to determine whether they show rf features or just noise. The first instance of the rf classifier classifying features, rather than noise, is made when the exploratory ramp has just entered the rf hypervolume, where the rf is sensitive. Next, the exploratory ramp is terminated, and the algorithm then explores the local gate voltage region by random walk while taking two-dimensional scans. The score function evaluates these scans to determine the quality of the double-dot the algorithm has found.

(Sec. 2.4.3). The matching circuit is insensitive outside the hypervolume, so noise drowns out the in-phase and out-phase demodulated signals. The current hypersurface and the rf hypervolume may not necessarily overlap. With our matching network components, the rf hypervolume existed at larger gate voltages than the pinch-off hypersurface.

Using exploratory gate voltage ramps, our algorithm navigates to the rf hypervolume. These ramps move gate voltages towards 0 V, at approximately $1\,\mathrm{V\,s^{-1}}$, along a randomly chosen radial direction in the space defined by voltage gates $V_1 - V_5$ for device A, or $V_3 - V_7$ for device B (Fig. 20). The first of these ramps start from gate voltage coordinates bounded by the gate voltage upper limits, which was 4V for all gates (Fig. 21 (a)). This gate voltage limit was chosen to allow the maximum operational range within which the device was safe from leakage. Along the length of a ramp, the algorithm performs fast 2d gate voltage scans (Sec. 4.1.2) every 50 mV. The algorithm evaluates each scan using a classifier to determine whether it shows just noise or dot features. When the classifier first detects dot features, the ramp has entered the rf hypervolume, and the ramp is terminated.

The algorithm then searches for double-dot features near the gate voltages where the rf hypervolume was found by searching around the area using a random walk. Double dot features are identified and quantified using a score function based on the discrete-time two-dimensional Fourier transform (Sec. 4.2.3). The random walk is performed by perturbing a randomly chosen gate voltage by an amount sampled from a Gaussian distribution with a standard deviation of 50 mV.

The algorithm efficiently explores the gate voltage space by repeatedly undertaking these exploratory ramps. With each ramp, it might find new double-dot features that score better than any previously found. Over time, with repeated exploratory ramps, the algorithm explores an increasing fraction of gate voltage space; therefore, the score associated with the highest-scoring double-dot features should improve over the algorithm's run time. Concurrently, each ramp's termination gate voltage coordinate updates a Gaussian process. These observations of the hypervolume location

inform the starting gate voltage coordinate of subsequent ramps, allowing them to start closer to the hypervolume and waste less time measuring regions of gate voltage space where only noise can be observed (Fig. 21). It follows that as the algorithm improves its model for the location of the hypervolume, it becomes more efficient at exploring the gate voltage space.

The following sections describe the Gaussian process (GP), the rf classifier and the score function used by the algorithm.

4.2.1 The Gaussian process

The Gaussian process (GP) is used to increase the efficiency of the exploratory ramps in navigating to the rf hypervolume by providing them with better starting points. It does this by predicting the radius $r(\hat{x})$ of the outermost boundary of the rf hypervolume in a given direction, denoted by the unit vector $\hat{x}$. Given observed data points $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$ and their corresponding function values $\vec{r} = \{r(x_1), r(x_2), \ldots, r(x_n)\}$, we can make predictions for new input points x_* by calculating the conditional distribution of $r(x_*)$ given the observed data. This conditional distribution is also Gaussian:

$$r(x_*)|\mathbf{X}, \vec{r}, x_*, \text{mean}_{\vec{\theta}}, \text{kernel}_{\vec{\phi}} \sim \mathcal{N}(\mu_*, \sigma_*^2) \tag{4.1}$$

This distribution's mean μ_* and variance σ_*^2 can be computed using the GP mean and covariance functions.

Initially, we set the mean function such that $\mu_*(\hat{x})$ equals the mid-point of the voltage range, with the RBF kernel (Sec. 3.5) function designed such that the entire voltage range falls within three standard deviations, $\sigma_*(\hat{x})$. This prior reflects our initial uncertainty about the hypersurface's location.

As the algorithm progresses, each observation of the rf hypersurface's location updates the GP, refining the posterior distribution (Fig. 21). Continuous updates allow the algorithm to make in-

creasingly accurate predictions about the hypersurface's boundary, guiding the exploratory ramps more efficiently. We begin the exploratory ramps three standard deviations outside the rf hypersurface.

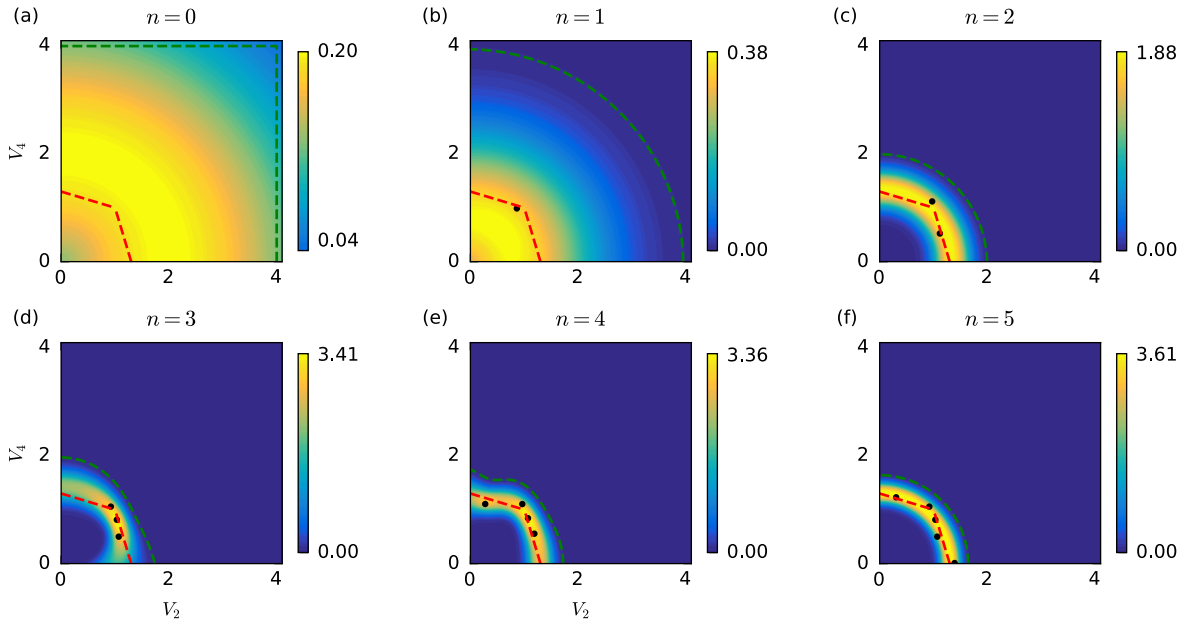


Figure 21: Visualisation of Gaussian process posterior distribution updates in two dimensions. This series of plots illustrates the dynamic evolution of the Gaussian Process (GP) posterior probability distribution as it models the outer boundary of the rf hypervolume. Each plot captures a different stage by progressively incorporating increasing noisy observations. The true position of the rf hypervolume is marked in red, providing a reference point against which the GP's performance can be evaluated. The noisy observations are depicted as black dots. The colourmap indicates the probability density of the Gaussian processes posterior distribution for the location of the rf hypervolume's outside edge. The dashed green line indicates three standard deviations outside the mean, which is where we choose to start exploratory ramps.

4.2.2 The feature/noise classifier

The role of the feature/noise classifier is to evaluate whether one of our fast 2d scans shows noise or quantum dot features. If a scan at a given set of gate voltages shows quantum dot features, that coordinate must lie within the rf hypervolume. However, in principle, both in-phase (I) and

out-of-phase (Q) components of the rf signal could contain useful information about these features (Fig. 22 (a)). Therefore, we perform blind signal separation to extract any information-bearing signal from the quadratures. We then perform a hypothesis test to see whether the extracted signal differs from the noise.

We use the popular method of principal component analysis (PCA) to extract any signal from the quadratures. PCA considers the signal's covariance matrix to optimally find the direction in the $I-Q$ plane with the largest variance and projects the data onto it [118]. Our algorithm uses PCA to optimally project a pair of 2d scans measured in both quadratures into one (Fig. 22 (b)).

To test whether this projected data is dominated by noise, we then perform a two-sample Kolmogorov-Smirnov hypothesis test, as implemented in `scipy` [119, 120], to compare the pixel value distribution of the projected scan against a reference noise measurement. The null hypothesis is that the two distributions are identical; therefore, the 2d scan contains just noise. The alternative hypothesis is that the distributions differ, so the scan shows quantum dot features instead. Our critical p-value is 1/1000, so p-values below this value lead to rejecting the null hypothesis in favour of the alternative. This choice of p-value means that, on average, the null hypothesis is falsely rejected only once in every thousand scans.

The noise was characterised by taking ten 2d scans with all gate voltages set to their maximum value. As a way to illustrate this test, a histogram of the PCA -projected pixel values of the scan in Fig. 22 (a) is overlaid on the measured reference noise distribution (Fig. 22 (c)). Quantum dot features can be observed in Fig. 22 (a), and a long-tailed distribution captures the presence of these features when compared to the noise distribution in Fig. 22 (c). The 2d scan would thus be classified as containing features, and the exploratory gate voltage ramp would terminate.

Mathematically, to perform projection via principal component analysis, we subtract the mean from data before projecting onto the eigenvector corresponding to the largest eigenvalue of the

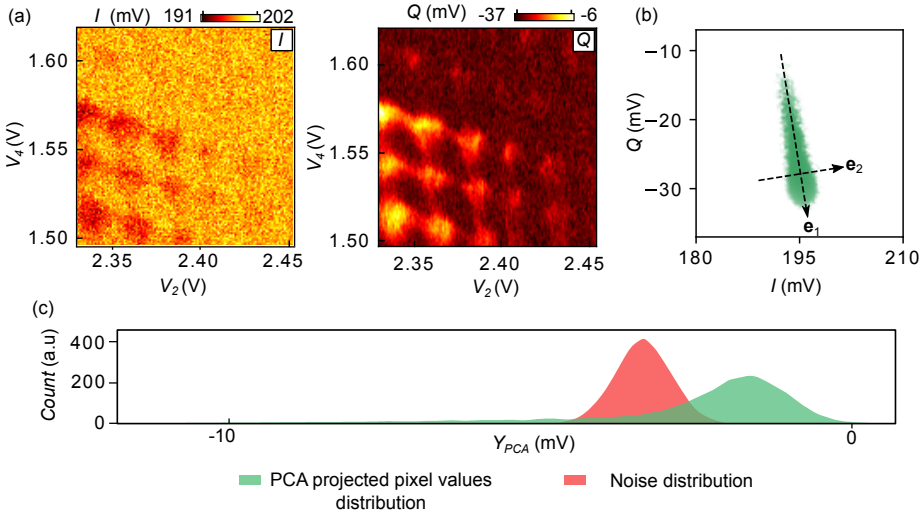


Figure 22: (a) The demodulated in-phase (I) and out-of-phase (Q) quadratures of a 2d scan, showing double quantum dot features on the edge of the rf hypervolume. (b) A scatter graph of the demodulated I and Q quadratures of the 2d scans in (a). We observe two principal directions of variance. The information-bearing component can be found along $\vec{e}_1$, while the less informative component is along $\vec{e}_2$. (c) A histogram of the I and Q data is shown in green once projected onto $\vec{e}_1$. For comparison, we show a histogram of pre-measured noise projected similarly in grey. The noise has a Gaussian distribution, while the double dot features produce a distinctly different distribution, providing a means to distinguish them.

covariance matrix,

$$\begin{bmatrix} \mathrm{cov}(\vec{I},\vec{I}) & \mathrm{cov}(\vec{I},\vec{Q}) \\ \mathrm{cov}(\vec{Q},\vec{I}) & \mathrm{cov}(\vec{Q},\vec{Q}) \end{bmatrix}. \tag{4.2}$$

4.2.3 The score function

The score function in our rf tuning algorithm is designed to distinguish scans exhibiting double dot features by assigning them high scores and lower scores to scans depicting single dots or other features. Traditional score functions which utilised current measurements have been based on neural networks [97–99], Hough transforms [121, 122], custom fitting models [123], handcrafted heuristics [116], and even ray-based classification [124]. We found that all these approaches could not be

directly used in our case, as the neural network-based score functions are explicitly trained upon transport data, and neural networks cannot generalise beyond their training data. The custom-fitting model and handcrafted heuristics are particular to the device. We had little success extending and extrapolating these methods to our use case. We could not easily generate representative training data for a neural network. We had some success with the handcrafted heuristics and fitting models, however, as we found that the open dot regime, when measured in rf measurements, was a consistent source of edge cases, which tricked our heuristics. Finally, we found the Hough transform sensitive to noise, making it incompatible with our fast scans, which contain moderate amounts of noise.

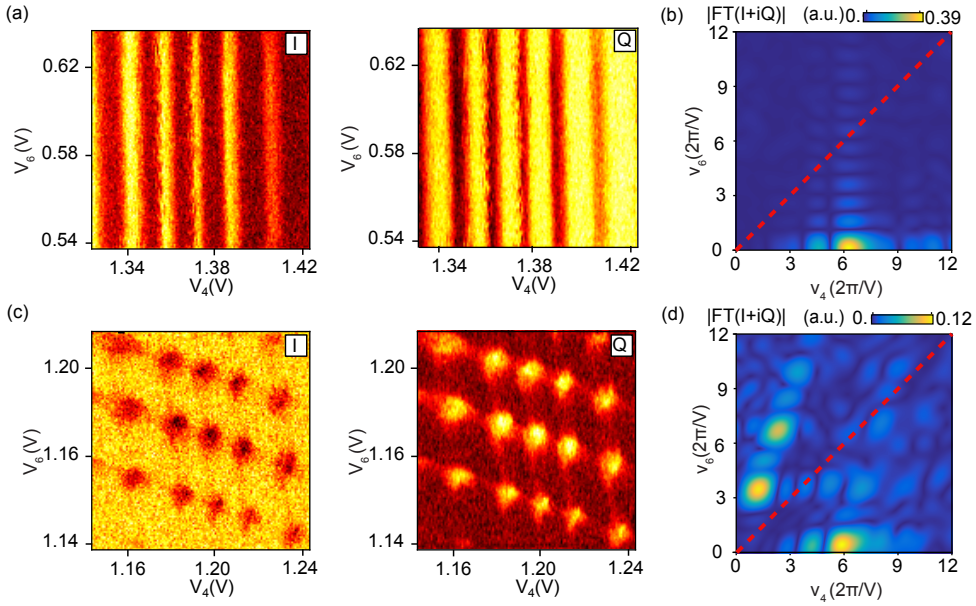


Figure 23: a-b) A 2d scan showing single dot features and corresponding time-discrete Fourier transform of the standardised I and Q data. The single dot features show one distinct periodicity, resulting in a single strong maximum in the Fourier transform (with fringes resulting from the finite window size). c-d) A 2d scan shows double-dot features and the corresponding time-discrete Fourier transform of the standardised I and Q data. The double dot features show two distinct periodicities (and associated harmonics), resulting in two strong maxima in the Fourier transform on either side of the diagonal $\nu_x = \nu_y$ marked as a dashed red line.

To address these challenges, we developed a scoring function rooted in the discrete-time Fourier

transform (DTFT) of the 2d scans. This approach is inherently phase-independent, enabling it to assess dot features in any combination of the in-phase and out-of-phase scans. In addition, the Fourier transform is resistant to noise, inherited by our score function.

The DTFT of these scans identifies periodic patterns, with each distinct periodicity manifesting as a prominent peak in the Fourier spectrum. The stronger the periodic nature of the features, the more pronounced these peaks. Our scoring function pinpoints the two strongest maxima in the Fourier transform characteristic of double quantum dot features. We restrict the search to periodicities that align with the behaviour of two quantum dots, each sufficiently coupled to their respective plunger gates to exhibit a few charge transitions over a 100 mV plunger gate sweep. In practical terms, for plunger gates V_x and V_y, we examine Fourier frequencies $\nu_x, \nu_y \in [0, 12]$, focusing only on maxima situated on opposite sides of the $\nu_x = \nu_y$ line. The score is then determined by the intensity of the weaker two maxima. Consequently, scans showing single-dot features typically yield only one peak, while other features produce none, resulting in lower scores.

A comparative illustration of double and single dot features, alongside their respective Fourier transforms, is displayed in Fig. 23 (a-d).

To apply the DTFT-based scoring function to the I and Q data obtained from a 2d scan of resolution $N \times M$, we first standardize the complex dataset $\vec{Z} := \vec{I} + i\vec{Q} \in \mathbb{C}^{N\times M}$ as $\vec{\mathcal{Z}} := (\vec{Z} - \bar{Z})/\sigma_Z$, where $\bar{Z}$ and σ_Z represent the complex mean and standard deviation, respectively. This standardisation ensures that the score reflects the quality of the double dot features rather than their alignment with the matching conditions. Then the DTFT of $\vec{\mathcal{Z}}$ is computed as follows:

$$\mathrm{DTFT}[\vec{\mathcal{Z}}]_{\delta\gamma} = \frac{1}{NM} \sum_{a=1}^{N} \sum_{b=1}^{M} \mathcal{Z}_{ab} \exp\left\{-2\pi i(\nu_\delta a + \nu_\gamma b)\right\}, \tag{4.3}$$

where $\vec{\nu}$ is a high-resolution linearly spaced frequency array between 0 and 12. This frequency range encoded in the $\vec{\nu}$ vector was chosen based on prior knowledge about the device, that the typical lever arm should result in a small number of charge transitions in our 100 mV sweep. Finally,

the score is taken as $s = \min(\max_<, \max_>)$ where $\max_<$ and $\max_>$ denote maxima in $|\text{DTFT}[\vec{\mathcal{Z}}]|$ constrained to line on either side of the line $\nu_x = \nu_y$.

4.3 Results

To assess our algorithm's performance on devices A and B, we executed 24 tuning sessions, each lasting one hour. During these sessions, the algorithm attempted to tune devices A and B 13 and 11 times, respectively, generating approximately 20,000 scans with corresponding scores. We will seek to validate the effectiveness of our algorithm in two stages: first, we will establish the score function as a reliable indicator of double quantum dot regimes, and second, we will demonstrate our algorithm improves this score faster and more efficiently compared to more straightforward search strategies.

4.3.1 Evaluating the score function

In this section, we will discuss how effective our score function was. To evaluate the correlation between our score function and expert assessment of double dot quality, we categorised the scans into three groups: non-double-dot, imperfect double-dot, and satisfactory double-dot. Two independent experts classified these scans into these categories, blind to the scores assigned by the algorithm. Figure 4 (a) presents a stacked histogram of the scores, segmented by the experts' classifications. The visualisation indicates that scans with non-double and satisfactory quantum dot features consistently receive low and high scores, respectively.

We will now employ the error matrix formulation from statistical classification. Under its typical formulation, each matrix column represents the ground truth, while each row represents predicted classification (or vice versa). Unfortunately, we do not have ground truth values; human labelling has biases and inaccuracies. In addition, our score function is not a classifier; therefore, we will

need to assign classifications based on thresholds for this analysis.

For the sake of this analysis, we assigned thresholds optimally to minimise the number of disagreements with the human labels. These thresholds were 0.04 and 0.08 (to one significant figure), resulting in the confusion matrix, shown in Table 1, illustrating the score function's accuracy in differentiating between double dot regimes. The score function demonstrated a high accuracy rate, with minimal false positives even in numerous scans lacking double quantum dot features, and effectively distinguished between satisfactory and imperfect double quantum dots. These thresholds exist solely for this analysis and do not affect the algorithm. See appendix B.2 for the confusion matrices separated by device.

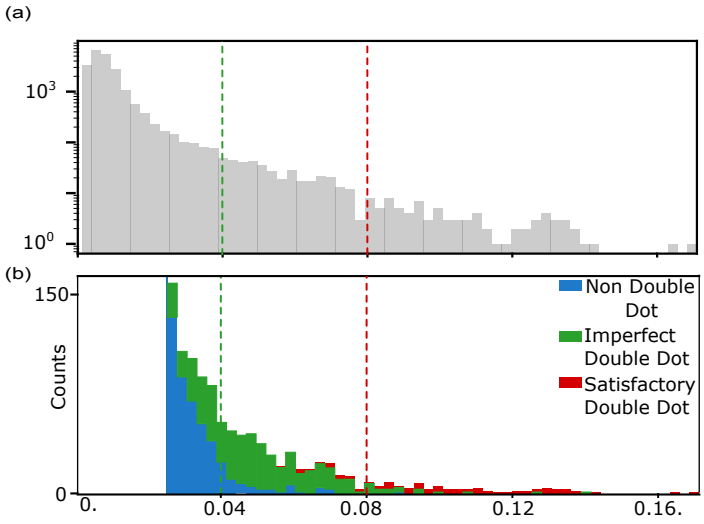


Figure 24: a) A histogram on a log scale of all corresponding scores of all the acquired scans. The approximate scores denoting the transition between non-double and imperfect double dot features are marked as a vertical green dashed line, as judged using manual human labelling. Likewise, a red dashed line denotes the score above which the dot features are said to correspond to satisfactory double dots. b) A histogram scores on a linear scale, where the colour coding of each bin indicates relative proportions of the scans judged by human labelling to be non-double dots (blue), imperfect double-dot (green) and satisfactory-double dots (red). For the most part, the red double dot region lies above the blue non-double dot region, whilst the green region corresponding to satisfactory double dots lies above both the orange and the blue. The scores below 0.025 have been omitted for clarity.

Table 1: Confusion matrix for devices A and B.

Device A & B (%)	Expert Classification		
	Non-double dot	Imperfect double dot	Satisfactory double dot
score < 0.04	96.95	0.91	0.00
$0.04 \leq$ score < 0.08	0.25	1.42	0.08
score ≥ 0.08	0.01	0.11	0.29

4.3.2 Evaluating the search algorithm

Now that we have convinced ourselves that the score function quantifies the quality of double dots, we will assess the algorithm's effectiveness, which aims to improve the score with time.

Over its run time, the algorithm explores a greater fraction of the rf hypervolume and discovers more dot features. Each newly-discovered dot feature has a chance of scoring higher than anything found before; therefore, on average, the score associated with the highest-scoring features should improve over time. The purple curves in Fig. 25 (a-b) show the evolution of the best score found by the algorithm as a function of run time for devices A and B, respectively. For a baseline comparison, we ran a reduced version of our algorithm using the exploratory ramps but did not benefit from any of the Gaussian processes, the rf classifier, or the subsequent random walk, shown in orange. We also ran a random search algorithm that randomly chose a coordinate in gate voltage space, set the gate voltages, and then performed a scan, shown in black (Fig. 25 (a-b)).

We found that our algorithm outperforms these more simplistic search strategies. This is likely because our algorithm spends a more significant fraction of its time exploring regions that have the potential to correspond to double dot regimes.

Overall, the full algorithm found and identified 16.5 and 16.8 scans an hour on average, corresponding to double dots (satisfactory and non-ideal) in devices A and B, respectively. If we restrict our attention to satisfactory double dots, the rates are 2.4 and 2.7 scans an hour. These tuning times are the fastest recorded in laterally-defined quantum dots [116, 117]. See Appendix B.3 for plots

of satisfactory double dots tuned by the algorithm in each hour-long run.

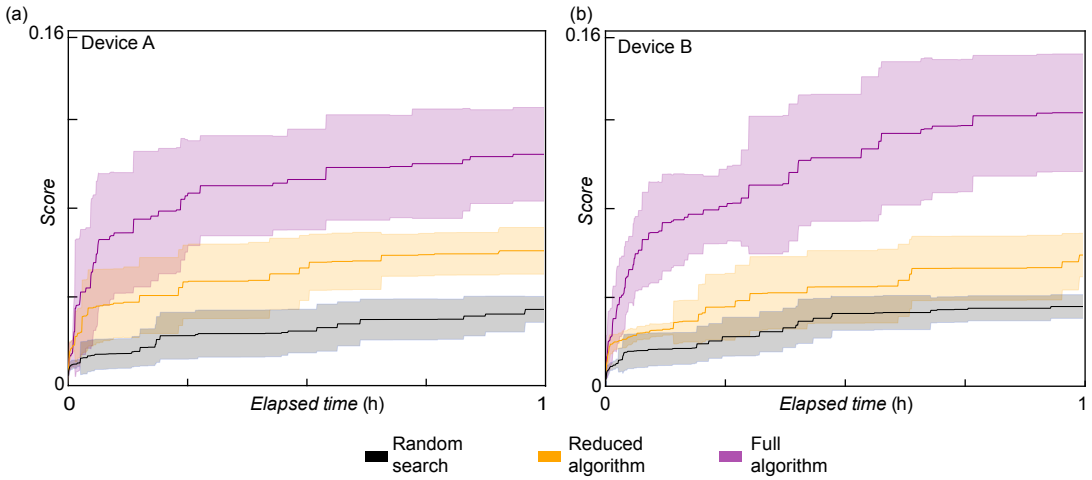


Figure 25: a-b) The graph or data shows the highest scores achieved within a specified tuning time for different algorithms, namely random search (shown in grey), a reduced version of the algorithm (shown in orange), and the full algorithm (shown in pink) when applied on devices A and B. The curves on the graph represent the average performance obtained from multiple runs, which helps to account for the statistical variations observed between these runs.

4.4 Conclusion

In conclusion, we have demonstrated the first algorithm for tuning double dots from scratch, which exclusively used radio frequency measurements. Such a result is an essential first step toward the scalability of quantum devices. Our algorithm reliably tuned satisfactory double quantum dots in approximately 15 minutes, the fastest recorded time in laterally-defined quantum dots. In creating this algorithm, we developed a robust method for distinguishing noise from rf features and proposed a score function based on the Fourier transform with significant noise tolerance.

Faster device tuning times could be achieved using a more informative Gaussian prior to the Gaussian process, particularly one which encodes knowledge of the gate architecture - possibly using an electrostatic model of the device [125]. Or use a Bayesian optimiser to choose the exploratory ramp directions to maximise the score.

The fast 2d scans could be generalised to many radial rays to probe the charge states of large quantum dot arrays, as performed slowly in [126, 127]. Along with frequency multiplexing [128], our algorithm could allow for scalable tuning and, thus, fully automatic tuning of multiple small quantum dot arrays.

Chapter 5

Automated long-range compensation of an rf quantum dot sensor

The previous chapter presented an algorithm that uses ohmic radio-frequency reflectometry measurements to tune double quantum dots. The next step was to deplete those dots so that one hole was in each dot. This necessitated moving to charge sensing measurements, which have their own unique complexities when keeping them tuned. This chapter outlines an automated protocol to maintain the charge sensor's tuning and retune it when necessary.

The work in this chapter resulting in the publication: Hickie, J., **Barnaby, V. S.**, Fedele, F., Jirovec, D., Ballabio, A., Chrastina, D., Isella, G., Katsaros, G., Ares, N.. *Automated long-range compensation of an rf quantum dot sensor*. `https://arxiv.org/abs/2310.02135`.

In this work, J.H. and I performed the experiment and developed the algorithm. G.K., A.B., and D.C. fabricated the sample. G.I. grew the heterostructure. N.A. supervised and conceived the project.

In this chapter, we introduce an automated approach to finding and maintaining the high sensitivity

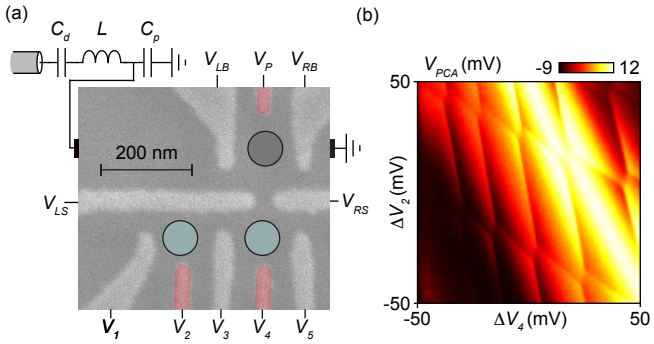


Figure 26: a) A scanning electron microscopy image of a device nominally identical to the one used in this experiment. The gate electrodes connected to microwave lines are colour-coded in pink. A bias tee combines a dc offset with the ac signal for the gates in red. b) Demodulated signal from the rf-QD obtained by PCA when the device is in a double dot regime. We perform this 2d measurement by ramping gates V_2 and V_4 in a raster pattern with an arbitrary waveform generator for a given point in gate voltage space.

of an rf sensor dot while sweeping across a large range of a device's gate voltage space. The algorithm was demonstrated in the same hole-based Ge/SiGe heterostructure quantum dot array as in the previous chapter but is again designed to be agnostic as to the device material and geometry.

Our algorithm employs Bayesian optimisation to optimally tune the rf sensor dot to a regime where it is highly sensitive. Once the optimal readout contrast is found, virtual gates [19, 129–131] are used to maintain the readout sensitivity in fast $100\,\mathrm{mV} \times 100\,\mathrm{mV}$ measurement windows. This ac-based compensation is combined with a dc-based compensation algorithm to allow for the rf sensing of a stability diagram over $1\,\mathrm{V} \times 1\,\mathrm{V}$, in which we can distinguish more than 25×25 charge transitions, using a single sensing peak. Our compensation algorithm can be repeatedly recalibrated to arbitrarily extend this range, thus overcoming the limitation of the linear approximation required by the constant interaction model.

5.1 Experimental setup

These experiments were performed in an electrostatically-defined Ge/SiGe quantum dot (QD) fabricated similarly to that described in [132], the same device used in the previous chapter. Gates $V_1 - V_5$ are used to define quantum dots in the main device, while gates $V_{\mathrm{LB}}, V_{\mathrm{P}}$, and V_{RB} define the charge sensor. Gates V_{LS} and V_{RS} separate the sensing dot from the device dots and were both set to 1.8 V for the experiment. For readout, we connected the ohmic of the charge sensor to an L-matching network with a 92 pF decoupling capacitor C_{d}, 2.7 µH inductor L, and a parasitic capacitance C_{p} (Sec. 26) [76]. We used a Zurich Instruments UHFLI to drive the circuit at a power of −90 dBm at the sample and demodulate the reflected signal.

The presented algorithm relies on the same fast 2d measurements as before; however, in this case, we measured the demodulated in-phase (I) and out-of-phase (Q) components of the reflectometry signal from the rf-QD sensor for each pixel with an integration time of 1 µs. To optimally capture both the I and Q components of the rf-QD signal, we use principal component analysis (PCA). The signal of the rf-QD demodulated in this way is labelled V_{PCA} and corresponds to the first principal component of the measured data.

5.2 The algorithm

Our algorithm consists of two stages. The first stage optimally tunes the charge sensor barriers ($V_{\mathrm{LB}}, V_{\mathrm{RB}}$) to achieve maximum charge noise sensitivity (Sec. 5.2.1), defined as the change of the rf signal when a charge transition occurs in the device. Optimal barriers are found when the rf-QD's impedance falls within a specific range such that the readout matching circuit has an impedance close to the $Z_0 = 50\,\Omega$ line impedance (Sec. 3.2.2). We use a direct approach by defining a score function for the charge sensor's visibility and using Bayesian optimisation.

The second stage is designed to maintain the tuning of the charge sensor while the device gate

electrodes (V_{1-5}) are swept (Sec. 5.2.2). We compensate for this cross-talk by constructing virtual gates. These virtual gates include the charge sensor's plunger gate (V_{P}) to compensate for the capacitive coupling between the device's gate electrodes and the rf-QD sensor.

5.2.1 Optimally tuning charge sensor tunnel barriers using Bayesian optimisation

In this subsection, we discuss optimally tuning the change sensor tunnel barriers, so they are compatible with the matching circuit 2.4.3. To do this, we need an optimiser and a score function to optimise. In the following, we will introduce both.

To optimise the rf-QD sensor's sensitivity, we define a score function. With optimally tuned barriers, the difference between the reflected signal on and off a Coulomb peak is maximised, maximising the height of the peaks and, thus, the readout sensitivity. We have found that maximising the variance of a charge sensing measurement is a fast and noise-resistant method for optimising the visibility of the charge sensor (Figure 27 a-c)).

To evaluate the score function, we take a fast 2d measurement (as described previously in Sec. 3.4.1) and determine the variance of the data after projection onto the first principal component, $\mathrm{Var}(V_{\mathrm{PCA}})$. The variance is a natural choice for a score function. It can be used to optimise not just the position of the Coulomb peak but also many other experimental parameters such as rf frequency and power. The variance of V_{PCA} is conceptually the same as considering the variance of the demodulated I component in a measurement where the phase is perfectly tuned. If we only had access to the I component of the demodulated data, the score function could be evaluated by taking the variance in the I component. In this case, the phase would need to be optimised as a parameter in the subsequent Bayesian optimisation.

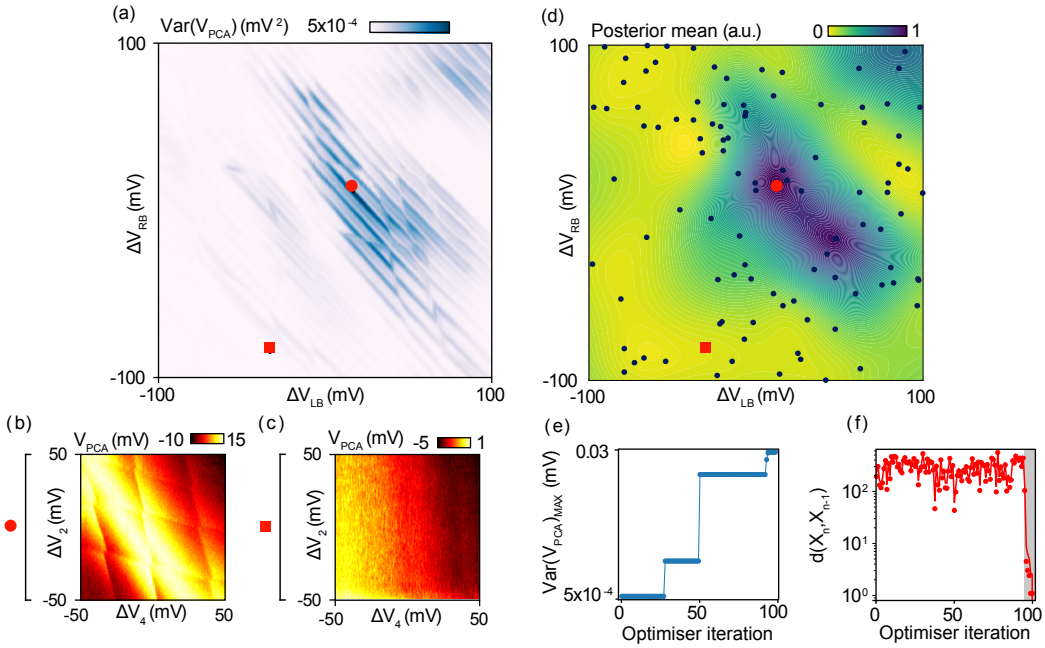


Figure 27: a) An example of the underlying score function over which we run the barrier optimisation routine, representing the variance of V_{PCA} for a 200 mV $\times$ 200 mV window in the gate voltage space given by the V_{LB} and V_{RB}. Each pixel in this scan represents the variance of V_{PCA} in a fast 2d measurement. b,c) Example fast 2d measurements taken at the points depicted by the red circle and square, respectively, with (b) corresponding to a sensor configuration with good matching. d) The posterior of a Gaussian process over a set of values of $\mathrm{Var}(V_{\mathrm{PCA}})$ from (a), marked by black points and chosen by the optimiser. The optimiser aims to find the maximum value of $\mathrm{Var}(V_{\mathrm{PCA}})$ in the gate voltage window considered. e) The optimiser found the maximum value of $\mathrm{Var}(V_{\mathrm{PCA}})$ as a function of the number of iterations. As the optimisation progresses, the optimal value of $\mathrm{Var}(V_{\mathrm{PCA}})$ increases. f) Distance between sequential gate voltage coordinates chosen by the optimiser. This distance decreases significantly for the last few iterations of the optimiser (a grey area), corresponding to exploitation rather than exploration.

Now that we have defined a score function to optimise, we must choose an optimiser. However,

first, let's better understand the score function landscape.

In Fig. 27 (a), we show an example of the score function landscape over which we optimise the barriers, V_{LB} and V_{RB}. Each pixel in this figure panel is the variance of V_{PCA} for a fast 2d measurement. The fast oscillations in this landscape are due to the charge sensor Coulomb peaks; only when the peak is entirely in the window will the variance be large. Overlaid on top of these Coulomb peak oscillations is the matching of the charge sensor impedance to the rf matching circuit. The quality of the matching modulates the height of Coulomb peak oscillations. When the matching is optimal, the height of the Coulomb peaks is maximal and thus the score, such that $\max \mathrm{Var}(V_{\mathrm{PCA}})$. This is where we wish to tune the to.

To find this optimum, we could perform the dense scan in Fig. 27 (a) and take the values of V_{LB} and V_{RB}, which maximise the score function. However, this is a very time-consuming approach, considering the measurement comprises 500×500 fast 2d measurements. Instead, we wish to choose an optimiser that can get there in fewer measurements.

We tried the "standard" optimisers, such as Nelder-Mead, Powell, CG, and BFGS and found that these optimisers performed poorly. They got stuck in the local optima created by the Coulomb peak oscillations. This was obvious in hindsight as the `scipy` documentation describes them as local optimisers. Therefore, we tried global optimisers such as basin-hopping and dual annealing. These optimisers could find the global optimum but were slow to do so.

We tried Bayesian optimisation as an alternative, as introduced in Section 3.6. Initially, it performed poorly because Coulomb oscillations cause length-scale hyper-parameters to be short. This meant the underlying Gaussian process model's predictions quickly became of low confidence away from observations. Therefore, if we weighted the BO for exploration, it effectively performed a random search. If we weighted it for exploitation, it was unwilling to leave its local optima.

Therefore, we attempted kernel engineering, where you design the GP's kernel to be of the same form as the score function landscape. We did this by adding an oscillating term to the kernel.

When this worked, it did improve things compared to the vanilla BO kernel. However, the fitting process for the kernel hyperparameters became very unreliable - often failing to find an appropriate length scale.

Surprisingly, the BO performed well when the fitting failed, yielding a far too large length scale. With this large length scale, the GP model effectively ignored the Coulomb oscillations and focused on the matching features.

To exploit this, we stripped out the oscillating component, using just the Matérn 5/2 kernel and constrained the length-scale hyperparameter to be larger than the scale of the coulomb oscillations. This forces the GP model to capture only the large-scale variations in the score function without the short-scale periodic variation (Fig. 27 a)). The BO optimiser chooses V_{LB} and V_{RB} in our optimisation approach. Initially, the first ninety voltages are chosen to explore the domain of the score function, building the Gaussian process model (Fig. 27 d)). The last ten iterations exploit the information in the model to converge on the global maximum (Fig. 27 d-f)). A sharp Coulomb peak, optimal for sensing, will be thus found at the coordinates in gate voltage space, corresponding to the global maximum of $\mathrm{Var}(V_{\mathrm{PCA}})$. Therefore, optimising the barriers in this way requires approximately 100 fast 2d measurements.

5.2.2 Compensating for cross-talk

In the algorithm's previous stage, we identified an optimal Coulomb peak and barrier voltages for sensing. This Coulomb peak in the rf-QD sensor is affected by the device gate voltages due to cross-talk. We aim to create virtual gates for the quantum dot device that allow us to move through the gate voltage space while keeping the rf-QD Coulomb peak fixed in the 2d measurement window. We achieve this by compensating each device gate with a contribution from V_{P}. The resulting virtual gates V'_{1-5} would necessarily require V_{P} to act in the opposite direction to the device gate.

Speaking mathematically, suppose we work in the gate voltage space spanned by $\{V_{1-5}, V_P\}$, and the first five unit vectors $\hat{e}_i$ for $i \in [1,5]$ are directions of changes in device gate voltages and $\hat{e}_P$ is parallel to a change in the charge sensor's voltage. To compensate for a change in one of the gate voltages $\Delta\vec{V}$, we must increment the voltage on the sensor plunger gate V_P by a value opposite in sign and scaled by the relative strength of the compensating contribution, such that $\Delta\vec{V}' = \Delta\vec{V} - \gamma_i\hat{e}_P$ where γ_i denotes the ratio of the coupling of the ith device gate and the sensor plunger to the sensor dot.

To compensate for an arbitrary change in device voltages, such that the direction of change is not necessarily parallel to one of the gate directions but has no component parallel to the charge sensor plunger, such that $\Delta\vec{V} \perp \hat{e}_P$ but otherwise unconstrained. We must increment the voltage on the sensor plunger gate V_P by a value opposite in sign and scaled by the relative strength of the compensating contribution

$$\Delta\vec{V}' = \Delta\vec{V} - (\vec{\gamma} \cdot \Delta\vec{V})\hat{e}_P. \tag{5.1}$$

To determine the values in $\vec{\gamma}$, we measure the effect of each gate V_{1-5} and V_P on the position of the rf-QD Coulomb peak in the 2d measurement window (Fig. 28 (a, b)). This position, d, is the shortest distance from the lower left corner of the 2d measurement to a linear regression fit of the rf-QD Coulomb peak. We quantify the strength of the effect of the gate voltages on d as $\alpha_i = \Delta d_i/\Delta V_i$, where Δd_i is the change in d when a given ΔV changes the device gate voltage V_i. We can choose a gate voltage range ΔV_i within which the values of α_i are constant, i.e. there is a linear dependence between d and V_i. If the entire width of the rf-QD Coulomb peak is not visible in the 2d measurement window, the estimation of d becomes less reliable.

The strength of the compensating contribution from V_P in each virtual gate $V_i^{'}$ is given by a coefficient γ_i, given by $\gamma_i = \alpha_i/\alpha_P$, where α_P is the strength of the effect of V_P on d. The values of γ_i quantify the relative strength of each device gate compared to the strength of V_P on d. We would expect that γ_i is thus less than one since the plunger gate has a greater effect on the rf-QD

Coulomb peak position than any of the device gates.

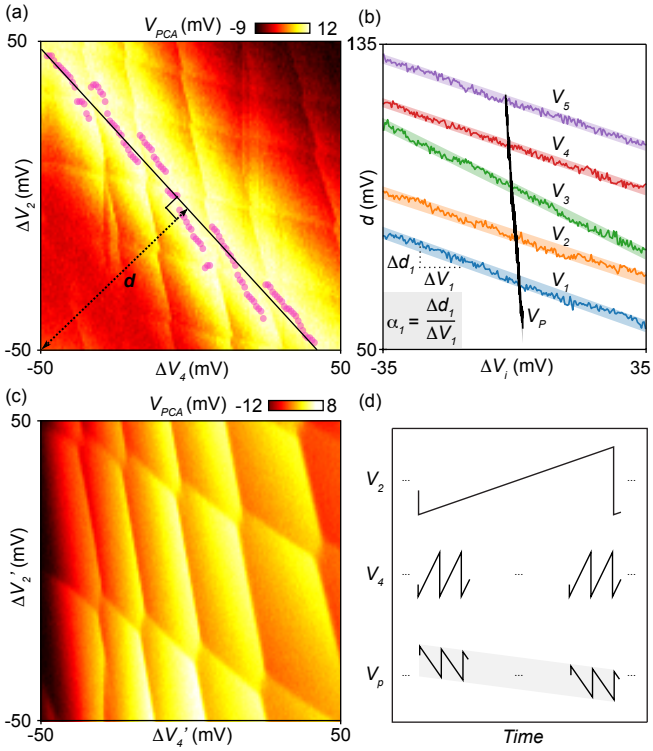


Figure 28: a) A 2d measurement window. Purple points identify the rf-QD Coulomb peak maximum. The solid line is a regression fit through these points. The dashed line indicates d, the perpendicular distance of the rf-QD Coulomb peak to the lower left corner of the 2d measurement window. b) Values of the Coulomb peak offset, d, as a function of changes in gate voltage for each of the gates of interest. Note the curves have been vertically offset to aid the reader. As expected, the sensor's plunger gate, V_{P}, has the strongest gradient. c) An Example of a 2d measurement window after AC compensation. The uncompensated version is in Fig. 1 (b). d) The top two waveforms are schematics of the voltages applied to device gates to perform a 2d measurement window. The bottom waveform is the ac compensation voltage applied to V_{P}.

The time to estimate γ_i depends on several factors but is entirely deterministic. These factors include the resolution of the 2d measurement windows, the pixel integration time, the time it takes to sweep a gate voltage over ΔV_i, the computer's processing speed, the number of device gates, and the number of sensors being compensated. In this setup, with five device gates and a single sensor plunger, the time taken to generate all the values of γ_i is approximately 10 seconds. This

time could be reduced at the expense of noise robustness.

Our cross-talk calibration and compensation algorithm extracts the values of γ_i to achieve compensation over long-range device gate sweeps and for the gate sweeps within a 2d measurement window. We will refer to the former (latter) as dc (ac) compensation.

5.2.3 Dc compensation

With the $\vec{\gamma}$ values, performing dc compensation was a matter of setting DAC values. We defined a virtual DAC voltage class, which handled everything about the behind-the-scenes virtual gates.

5.2.4 Ac compensation

We can observe the rf-QD Coulomb peak within a measurement window due to cross-talk. To compensate for this cross-talk, we create a waveform to compensate the fast sweeps that define the 2d measurement window. This third waveform is a linear combination of the two waveforms being used to generate the plunger fast sweeps (Fig. 28 (c, d)):

$$V_{\mathrm{P}}(t) = -\begin{bmatrix}\gamma_2 \\ \gamma_4\end{bmatrix} \cdot \begin{bmatrix}V_2(t) \\ V_4(t)\end{bmatrix}, \tag{5.2}$$

where $V_{\mathrm{P}}(t), V_2(t), V_4(t)$ are the ac components of the voltages applied to each of the three gates, V_{P} compensates for the cross-capacitive coupling between the device gates and the rf-QD. In this way, we remove the gradients corresponding to the rf-QD Coulomb peak from the 2d measurement window background (Fig. 26 (d)). We found that the same $\gamma_{2/4}$ were appropriate for use in the dc and ac compensation. However, this might not be true if the fast lines to the device and charge sensor gates have different attenuation.

This ac compensation approach readily applies to arbitrary pulses involving any number of device

gates. Equation (5.2) is thus generalised by including additional time-dependent components scaled by the relevant γ value.

5.3 Results

This section discusses the results and performance of our compensation algorithm. If we had successfully constructed a perfectly-compensated gate, the Coulomb peak offset d will not change as the compensated gate voltage $V^{'}$ is modified. We, therefore, measure the performance of our DC compensation algorithm by calculating the standard deviation of the peak offset, σ_d, as a function of gate voltage sweeps for three cases:

1. *Uncompensated arbitrary*, these virtual gate sweep directions are constructed as an arbitrary linear combination of gates V_1 to V_5, with no compensation applied to the charge sensor plunger V_P.

2. *Compensated arbitrary*; the virtual gate sweep directions are constructed as in the uncompensated case. However, a compensation term is included to the device plunger according to equation (5.1).

3. *Compensated true*; one of the device gates is chosen and compensated accordingly.

For this performance test, a Coulomb peak was first selected by the Bayesian optimisation routine described in Section 5.2.1. The same peak was used for each of the benchmarking runs. To produce the arbitrary virtual gate directions, we generated 100 random vectors from the standard-normal distribution $\vec{\epsilon}_1, \ldots, \vec{\epsilon}_{100} \in \mathbb{R}^5$. These vectors were then used to create uncompensated and compensated virtual gates.

Figure. 29 (a) shows σ_d as a function of the length of the gate voltage sweep, ΔV_{g}, for the three different sets of gate sweeps described above. Without dc compensation, at $\Delta V_g \approx 50\,\mathrm{mV}$ the Coulomb peak is no longer visible within the measurement window, and σ_d saturates at 35 mV.

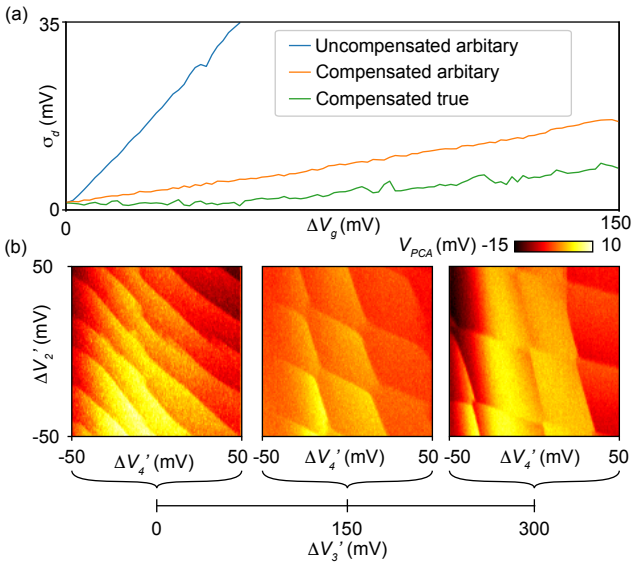


Figure 29: a) The standard deviation of a Coulomb peak's position quantified by d as a function of the length of the gate voltage sweep, ΔV_g, for uncompensated arbitrary, compensated arbitrary and true gate voltage sweeps. b) Three 2d measurement windows for different gate voltage configurations. These configurations are accessed by changing the compensated middle barrier gate voltage value, V_3', by $\Delta V_3'$. Our compensation algorithm allows us to choose a Coulomb peak of our sensor dot and use it to explore the device gate voltage space going from a single dot to a double dot regime. Ac compensation is applied for each of these 2d measurement windows.

Also, as we further increase ΔV_g, a different Coulomb peak becomes visible in the measurement window. With dc compensation, the algorithm drastically reduces σ_d over a gate voltage range of 150 mV for both compensated device gates and compensated virtual gates. Ramping compensated device gates has a smaller effect on σ_d than ramping compensated virtual gates. This might have to do with non-linearities in the effect of gate voltages that are accentuated in their combination. Our compensation algorithm applied to V_3 allows us to measure device operation regimes from single to double dot for a given Coulomb peak of the sensor dot (Fig. 29 (b)).

Combining both ac and dc compensation algorithms, we can produce large stability diagrams with clear charge transitions by maintaining the chosen Coulomb peak's offset for the sensor dot fixed. We demonstrate this by measuring 1000 mV $\times$ 1000 mV stability diagrams made up of 10×10 2d

measurement windows. In Fig. 30 ((a)), it is evident that when no compensation is applied, the visibility of charge transitions in the stability diagram varies significantly. The effect of cross-talk manifests in seven Coulomb peak transitions of the sensor dot. With dc compensation applied, a Coulomb peak chosen by the Bayesian optimisation routine is used, and its offset is fixed. The sensor's sensitivity remains thus constant across different 2d measurement windows (Fig. 30 (b)). Still, we observe an apparent change in the sensor's sensitivity within individual 2d measurement windows. Using both dc and ac compensation, the sensor's sensitivity is kept mostly constant across the full stability diagram (Fig. 30 (c)). This shows the extent of our compensation algorithm's capabilities, with charge transitions visible for almost a volt in each device plunger direction. This range encompasses the largest scans performed on current devices. In future devices, the hope is that the fabrication improves so that we have stronger prior knowledge about the device and, therefore, will not have to perform such large scans to tune the device into the correct state.

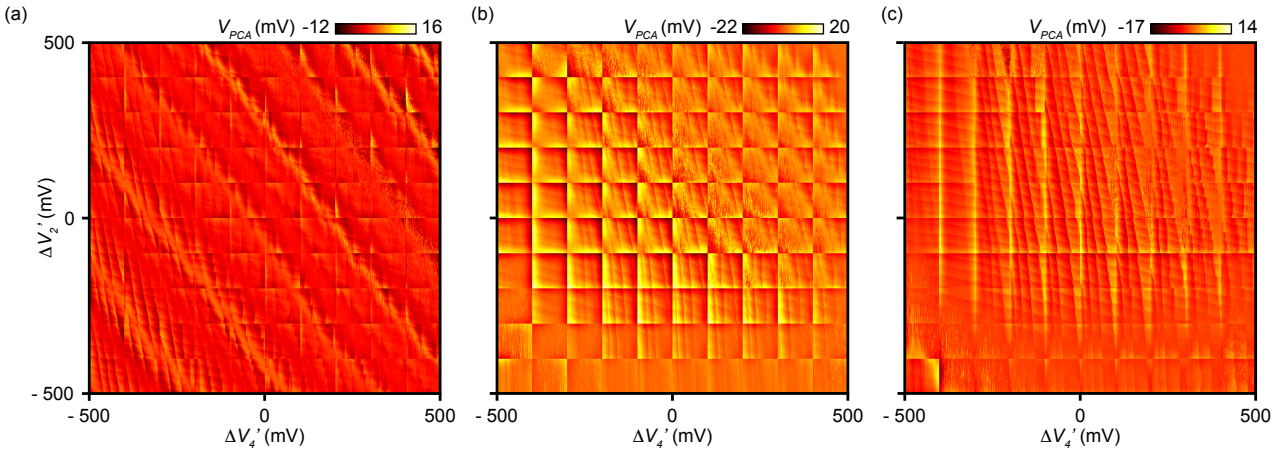


Figure 30: 1000 mV x 1000 mV mosaic of 2d measurement windows for three different cases: a) without ac or dc compensation, b) just dc compensation for V_2 and V_4, and c) both dc and ac compensation for the same gates. The dc compensation maintains the sensor Coulomb peak offset, d, fixed and the ac compensation makes charge transitions significantly clearer. The edges of the 2d measurement windows are still noticeable in panel c due to imperfections in the waveform compensation used to correct the distortion introduced by filters.

5.4 Conclusion

We demonstrate how a quantum dot charge sensor can be quickly tuned and compensated for both ac and dc sweeps of a device's gate voltages. The combination of the algorithms presented results in a $1\,\mathrm{V} \times 1\,\mathrm{V}$ stability diagram measured while maintaining an optimal transconductance of the sensor dot found by a Bayesian optimiser.

The gate voltage range within which this compensation approach can be applied is limited by the failure of the linearity assumption underlying the definition of virtual gates. Therefore, the compensation's effectiveness reduces as the device gate voltages are swept far away from the gate voltage location at which the virtual gates are calibrated. In our approach, recalibrating the virtual gates takes approximately ten seconds, allowing for the expansion of the gate voltage range in which compensation can be applied. The effect of charge switches and gate voltage drifts that change the sensitivity of the rf-QD can be overcome by integrating a closed-loop feedback system for small measurement windows, such as those demonstrated in Refs. [83, 84].

Our approach in this work naturally scales to device architectures with multiple sensors. The ac compensation can also be used for any fast 2d measurement, such as those involving interlaced pulse sequences. The algorithm is fully automated, requires no human intervention, and is suitable for use with other automated tuning procedures.

Chapter 6

QArray: a GPU-accelerated constant capacitance model simulator for large quantum dot arrays

With the previous two algorithms developed, it was conceivable to combine them, such that the rf tuning algorithm (with minor modifications) tuned the sensor into a single dot. They tuned the device dots into the double dot regime, where we could then use the charge sensor compensation to keep the sensor tuned while we searched for the last hole regime. However, doing so necessitated developing a programmatic understanding of the charge stability diagram to identify lead, inter-dot and perhaps even spurious dot transitions. To this end, I developed a constant capacitance model simulator of the charge stability diagram, with the view of using it to generate training data for a neural network and for offline (away from the experiment) testing. Along the way, I discovered software optimisations which made my implementation considerably faster than previous implementations, making the simulator useful in its own right and resulting in the publication:

Barnaby, v. S., Hickie, J., Schorling, L., Schuff, J., Fedele, F., & Ares, N. *QArray: a GPU-*

accelerated constant capacitance model simulator for large quantum dot arrays. arXiv.org. `https://arxiv.org/abs/2404.04994`

This chapter introduces `QArray`, an open-source software package for simulating quantum dot charge stability diagrams using the constant capacitance model. The package contains a suite of useful functionalities, most notably new algorithms for computing the lowest energy charge configuration at its core. In existing software packages, this is done using a brute-force algorithm, which iterates over every possible charge state where each dot can contain between zero and $n_{\max}$ charges[1] [87, 133].

Here we introduce two new algorithms, which we call `QArray`-default and `QArray`-thresholded. These algorithms greatly reduce the number of charge states needing to be iterated over to find the lowest energy state. Table 2 displays the number of charge states needing to be considered by the traditional brute-force, as opposed to the `Qarray`-default and `Qarray`-thresholded algorithms[2] algorithms applied to array containing n_{dot} quantum dots.

As presented in the table, the number of charge states needing to be iterated over by the `QArray` algorithms scale exponentially with the number of quantum dots but in a lower base than the brute-force algorithm. This results in a considerable reduction in runtime. For example, simulating an eight-dot array containing four charges with the `QArray`-default rather than the brute-force algorithm necessitates iterating over $[(4+1)/2]^8 \approx 1526$ times fewer charge states. Further improvements in compute time could be obtained at the price of accuracy by using `QArray`-thresholded (see later sections). This speedup in computation time makes my approach a promising solution for practical applications requiring the constant capacitance model, such as generating diverse datasets for training neural networks [96–98].

[1] The value of $n_{\max}$ must be set ahead of time by the user. If the set value is too small, the set of charge states iterated over by the brute-force algorithm might not contain the lowest energy charge state. The brute-force algorithm will return a charge state that is not necessarily the lowest energy.

[2] Strictly speaking, the value presented in the table is the expected number of charge configurations. In the worst case, every charge state might need to consider $2^{n_{\text{dot}}}$ charge states.

Algorithm	Number of charge states to be considered	Software implementations
Brute-force	$(n_{\max}+1)^{n_{\text{dot}}}$ where $n_{\max} \in \{1, 2, ..., \infty\}$	JAX, Python
`Qarray`-default	$2^{n_{\text{dot}}}$	Rust, JAX, Python
`Qarray`-thresholded	$(t+1)^{n_{\text{dot}}}$ where $t \in [0, 1]$	Rust, Python

Table 2: A table showing the number of charge states needing to be considered by the software implementations of the brute-force, `QArray`-default and `QArray`-thresholded algorithms. Where n_{dot} refers to the number of quantum dots in the array being simulated, is $n_{\max}$ maximum number of charges considered in any dot by the brute-force algorithm and t is the threshold used in the `QArray`-thresholded algorithms, which quantifies the degree of approximation.

The `Qarray` software package contains multiple implementations of the brute-force, default and thresholded algorithms in Python, JAX or Rust (See Table 2), any of which are callable from Python. The Rust implementations harness the full potential of multi-core processors [134, 135], while a JAX implementation provides GPU acceleration. The pure Python implementations are maintained for comparison testing with other implementations. The Rust and JAX implementations of the `QArray` algorithms can model arrays with 16 or more dots and simulate smaller arrays faster than they can be measured experimentally, even in rf [136–138].

In the following sections, we will discuss the default and thresholded algorithms. However, first, we will restate the problem of finding the lowest energy charge configuration and determine that this problem is, in general, NP-hard.

6.1 Problem of computing the lowest energy charge state

As discussed in Section 2.5, the constant capacitance model gives a formula for computing the electrostatic energy of a charge configuration, namely

$$F(\vec{Q}; \vec{V}_g) = \frac{1}{2}\vec{Q}^T \mathbf{c}_{dd}^{-1} Q - (\mathbf{c}_{dd}^{-1}\mathbf{c}_{gd}\vec{V}_g)^T \vec{Q}. \tag{6.1}$$

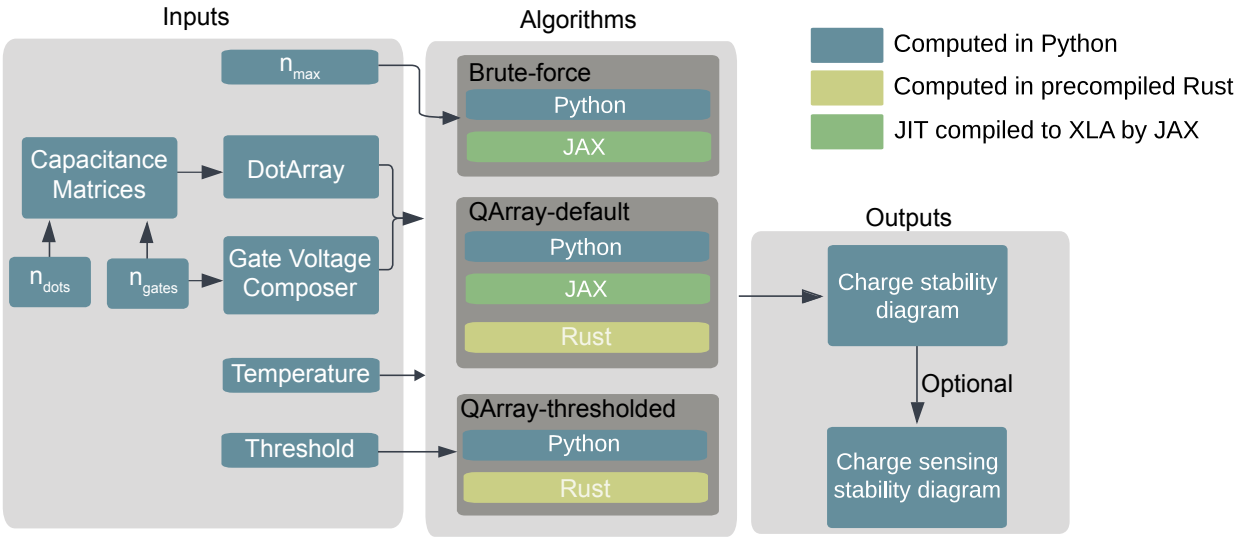


Figure 31: Organisation of the `QArray` Package: The structure of the Qarray package features multiple software implementations of the brute-force, default and thresholded algorithms as colour-coded boxes denoting that it was programmed in Python (blue), Rust (orange), and JAX (green). The dot array class carries the capacitance matrices and is the means by which the user calls the various algorithms to compute the ground set. The gate voltage composer class generates voltage vectors for sweeps or can be used to define arbitrary virtual gate sweeps. DotArray offers methods for computing charge stability diagrams for open or closed regimes with any algorithm of choice. Optionally, the charge stability diagram can be coupled with simulated charge sensors to model measuring the charge stability diagram using one or more charge sensing dots. If the brute-force algorithm is used, $n_{\max}$ or the threshold, t, must be passed respectively. Finally, a temperature parameter can be passed to the algorithm to simulate thermal broadening.

But how does one compute which charge state minimises this free energy? The allowed charge configurations are heavily constrained, which counterintuitively makes finding the ground state more difficult. For an open quantum dot array, which is when the array is coupled to fermionic reservoirs from which it can draw an arbitrary number of charges, the constraints are:

(i) The number of charges must be integers, represented as $\vec{N} \in \mathbb{Z}^{n_{\text{dot}}}$;

(ii) The number of charges in any given quantum dot must be non-negative, meaning $N_i \geq 0$ for all i.

For a closed quantum dot array, which is when the leads are decoupled from the array [92, 93], and

the number of charges is fixed to a finite value $\hat{N}$, there is the additional constraint that:

(iii) The sum of all charges in the array must equal a specific value, denoted as $\hat{N}$.

These constraints make the problem of finding the lowest energy charge configuration an instance of a class of optimisation problems called constrained convex quadratic integer problems, which are NP-hard [139, 140]. If P≠NP, as is suspected, then no algorithm can find the ground state deterministically in polynomial time. However, this does not mean that algorithms better than brute force do not exist, as demonstrated by the `QArray` algorithms discussed in the next section.

6.2 The QArray algorithms

The `QArray` algorithms can be broken down into two steps. First, we neglect constraint (i), that the number of charges on a quantum dot must take an integer value, and compute the continuously-valued charge state that minimises the free energy. We will refer to this as the continuous minimum (Fig. 32 (a)).

In the second step, we apply integer charge constraint (i) by evaluating the energy of each discrete charge state that neighbours the continuous minimum and take the state with the lowest energy (Fig. 32 (b)). The `QArray`-default and `QArray`-thresholded algorithms differ in which charge states they define as neighbouring the continuous minimum. The thresholded algorithm is much more selective (Fig. 32 (b)). We formally justify considering only the neighbouring discrete charge states in Appendix C.3.1. For an intuitive explanation, consider that the free energy is a convex function minimised at the continuous minimum. The lowest-energy discrete charge states must lie close to the continuous minimum. Charge states further from the continuous minimum must be higher in energy.

In the next section, we will discuss how the continuous minimum is computed for both open and closed arrays, then how the default and thresholded algorithms find the discrete charge state lowest

in energy from the nearest neighbours to the continuous minimum (Sec. 6.2.2).

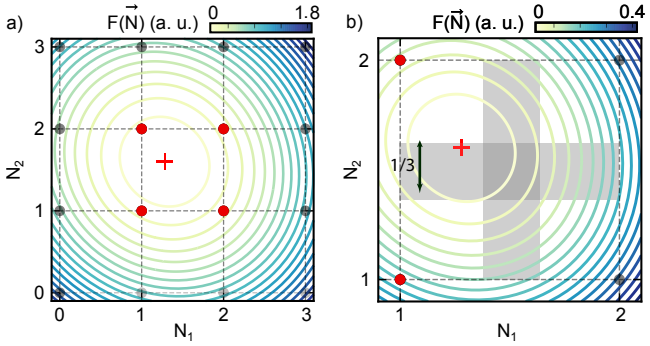


Figure 32: a) Diagram depicting the default algorithm applied to a double dot. The algorithm first finds the continuous minimum (marked with a red cross), which minimises the free energy. The free energy landscape, as a function of the charge on each dot, is shown with the contour lines. After computing the continuous minimum, the algorithm evaluates the nearest neighbour discrete charge states (marked as red dots) to determine which is lowest in energy. By comparison, the brute force implementation considers all charge states, leading to it also having to consider the charge states marked as black dots. In this case, (1, 2), the charge state is the discrete charge state lowest in energy. b) Diagram depicting the thresholded algorithm applied to a double dot. The algorithm computes the continuous minimum (red cross) as in the default algorithm. However, rather than just considering all nearest neighbour discrete charge states, thresholding is used to reduce further the number of charge states needing to be evaluated compared to the default. As the continuous minimum does not lie within the vertical grey rectangle, where the width equals the threshold $t = 1/3$, it is unnecessary to consider both the floor and ceiling values for N_1. Instead, it can be rounded down. By contrast, the continuous minimum does lie in the horizontal rectangle, so both floor and ceiling values for N_2 must be considered. The discrete charge states, considered by the thresholded algorithm, are shown as red dots.

6.2.1 Computing the continuous minimum

To compute the continuous minimum charge configuration for both open and closed quantum dot arrays, we initially try the analytical solution for the charge configuration derived by neglecting constraints that the charge state must take integer, non-negative values, constraints (i) and (ii) respectively. If we are lucky and this solution satisfies constraint (ii) such that no dot contains a negative number of charges, we accept it and use it. Otherwise, we compute a solution using a numerical solver, which explicitly applies constraint (ii). We try the analytical solution before

resorting to the numerical solver because it is inexpensive to compute compared to the solver, and trying it first offers an overall time advantage.

The analytical solutions

The continuous minimum for an open quantum dot array is

$$\vec{Q}^{*\text{cont open}} = \mathbf{c}_{gd}\vec{V}_g, \tag{6.2}$$

see Appendix C.2.1 for a formal derivation.

For closed quantum dot arrays, we use Lagrangian multipliers to account for the fixed number of charges in the array (imposed by constraint (iii)). As derived in Appendix C.2.2, the continuous solution can therefore be written as

$$\vec{Q}^{*\text{cont closed}} = \mathbf{c}_{gd}\vec{V}_g + \left(\hat{Q} - \vec{1}^T\mathbf{c}_{gd}\vec{V}_g\right)\frac{\mathbf{c}_{dd}\vec{1}}{\vec{1}^T\mathbf{c}_{dd}\vec{1}}, \tag{6.3}$$

where $\vec{1} = (1, 1, \ldots, 1)^T \in \mathbb{R}^{n_{\text{dot}}}$ is the one vector and $\hat{Q}$ is the confined charge.

In both the open and closed cases, one or more of the elements of $\vec{Q}^{*\text{cont open/closed}}$ may correspond to a negative number of electrons/holes. If this occurs, we must fall back on the numerical solver to implicitly apply the constraint. The numerical solver is discussed next.

The numerical solver

The numerical solver computes the continuous minimum for open and closed dot arrays as a constrained optimisation problem. This allows us to minimise $F(\vec{N}; \vec{V}_g)$, with the optimisation variable being $\vec{N} := \mp\vec{Q}/e$ for electrons and holes respectively, with $\vec{V}_g$ fixed, subject to the linear constraints of the form $\vec{l} \leq \mathbf{A}\vec{N} \leq \vec{u}$, where $\vec{l}$ and $\vec{u}$ are bounds vectors. Constraint (ii) can be encoded in this

form by setting the constraint matrix $\mathbf{A}$ to the identity, and the lower and upper bound vectors ($\vec{l}$ and $\vec{u}$) to vectors of all zeros and infinities, respectively. For a closed array, constraint (iii) can be encoded by setting the constraint matrix as a row matrix with all terms equal to one and upper and lower bound vectors as single value elements equal to $\hat{N}$. Combining constraints for open and closed quantum dot arrays involves concatenating the associated constraint matrix and bound vectors.

6.2.2 Evaluation of the nearest neighbour discrete charge states

With the continuous minimum in hand, the `QArray` algorithms iterate over the nearest neighbouring discrete charge states to find which one is the lowest in energy. The algorithms differ in how they define the nearest neighbouring discrete charge states. In the following subsection, I will discuss precisely which charge states default and thresholded algorithms iterate over.

The default algorithm

The default algorithm will consider all the different charge configurations obtained by the element-wise flooring and ceiling of the values in the continuous minimum. Therefore, the default algorithm iterates over all $2^{n_{\text{dot}}}$ possible ways of rounding the elements of the continuous minimum up or down to the nearest integer. The charge states considered by the default algorithm when computing the lowest energy charge state of a double dot are highlighted in red in Fig. 32 (a)). This considerably reduces the number of charge states needing to be considered compared to brute-force (Tab. 2). We formally justify considering only the neighbouring discrete charge states in Appendix C.3.1.

If the quantum dot array is closed, we eliminate the configurations that lead to the wrong number of charges in the array.

The thresholded algorithm

The motivation behind the thresholded algorithm was the observation that it was sometimes unnecessary to consider every one of the $2^{n_{\text{dot}}}$ possible charge combinations. Only if the decimal part of the element continuous minimum charge state, $\vec{N}^{*\text{cont}}$, is close to 1/2 should we consider both charge configurations obtained by rounding up and down to the nearest integer. Otherwise, we can round to the closest charge configuration. The charge states considered by the thresholding algorithm when computing the lowest energy charge state of a double dot, with a threshold of 1/3, are highlighted in red in Fig. 32 (b))

If we define the threshold as $t \in [0, 1]$, then the condition for having to consider both the floor and ceiling charge configurations on the ith quantum dot is $\left|\text{frac}[N_i^{*\text{cont}}] - 1/2\right| \leq t/2$ (Fig. 32 (b)). The total number of charge states needed to evaluate after the thresholding scales with $(t+1)^{n_{\text{dot}}}$ (see Appendix C.4.1 for deviation).

When considering closed arrays, we found that applying the threshold may produce no charge states with the correct number of charges. In this case, we double the threshold and recompute. This threshold is left to the user's discretion; however, in Appendix C.3.2, we try to motivate a good choice of threshold.

6.3 Miscellaneous functionality and results

This section discusses miscellaneous aspects of `QArray` that are adjoint to the main algorithms. In particular, how we simulated charge sensing measurements (Sec. 6.3.1) and thermal broadening (Sec. 6.3.2). Finally, we state some analytical results which we derived and found useful (Sec. 6.3.3).

6.3.1 Charge sensors

Experimentally, the charge stability diagram is not directly available. Instead, it is measured indirectly using a charge sensor. Therefore, in order to train neural networks for automated tuning, it is necessary to simulate the signature of a charge sensor.

To simulate charge-sensing measurements, we distinguish between charge-sensing dots and regular dots. We first compute the charge state that minimises the free energy of the regular dots. With the charges on the regular dots fixed, we then determine the charge configuration on the sensor dots that minimises the overall free energy. This method ensures that the charge state of the regular dots is not influenced by the charge state of the sensor dots, although interdot transitions between a sensor dot and a regular dot can still occur.

Next, we evaluate the difference in free energy between the minimum free energy charge state, and the charge states with one fewer and one more charge on each charge sensor, which we denote as $\Delta F_{\pm}(\vec{V}_g)$. We assume that the charge sensor dots are in the strongly coupled regime, so the conductance follows the Breit-Wigner formula [85]. Consequently, the charge sensor's response is Lorentzian, and we express the sensor response as:

$$s(\vec{V}_g) = \sum_{\pm} \frac{1}{[\Delta F_{\pm}(\vec{V}_g)/\Gamma]^2 + 1} \tag{6.4}$$

where Γ is the sum of the tunnel rates to the source and drain, $\Gamma = \Gamma_S + \Gamma_D$. This parameter should be configured by the user to match their experimental setup. We sum the responses for both $\Delta F_{\pm}(\vec{V}_g)$, rather than just using the smaller of the two, to avoid discontinuities in the charge sensor response when the lowest energy charge state changes. Near a Lorentzian peak, only one component will contribute significantly. Additionally, we include various noise models to simulate white noise, $1/f$ noise, and sensor switches due to fluctuating charge traps, among others.

6.3.2 Thermal Broadening

For a finite temperature, the quantum dot system will not adopt the lowest energy charge state but instead adopts a thermally mixed state. This can be accounted for by replacing the arg min function in $\vec{N}^* = \arg\min_{\vec{N}\in\mathcal{N}} F(\vec{N};\vec{V}_g)$, equation (2.9), with a soft arg min function, which computes a Boltzmann weighted sum of the applicable charge states, such that

$$\begin{aligned}\vec{N}^* &= \underset{\vec{N}\in\mathcal{N}}{\text{soft arg min}}\, F(\vec{N};\vec{V}_g) \text{ where} \\ \underset{\vec{N}}{\text{soft arg min}}\, F(\vec{N}) &= \frac{\sum_{\vec{N}} \vec{N} \exp\Big[-F(\vec{N})/k_BT\Big]}{\sum_{\vec{N}} \exp\Big[-F(\vec{N})/k_BT\Big]},\end{aligned} \tag{6.5}$$

where the summation runs over all the allowed charge configurations. For a closed array, this means eliminating the charge states with the total number of charges different than $\hat{N}$.

6.3.3 Other analytical results

In developing `QArray`, I derived some useful analytical results associated with the constant capacitance model. This section lists them.

Optimal gate voltages: the gate voltages that minimise the charge state $\vec{Q}_d$'s free energy, $F(\vec{V}_g,\vec{Q}_d)$ is

$$\vec{V}^* = (\mathbf{R}\mathbf{c}_{gd})^+ \mathbf{R}\vec{Q}_d, \tag{6.6}$$

where $\mathbf{R}^T\mathbf{R} = \mathbf{c}_{dd}^{-1}$ is the Cholesky factorisation and $+$ denotes the Moore-Penrose pseudo inverse. The proof of this result is presented in Appendix C.4.2

Optimal virtual gate matrix: the optimal virtual gate matrix, α, used to construct virtual gates is

$$\alpha = (\mathbf{c}_{gd}\mathbf{c}_{dd}^{-1})^+ \tag{6.7}$$

where + denotes the Moore-Penrose pseudoinverse [19]. The ith row of this matrix encodes the electrostatic gate voltages required to change the ith dot's potential. The proof of this result is presented in Appendix C.4.3

6.4 Implementation

Within `QArray`, I have implemented the brute-force, default and thresholded algorithms to find the lowest energy state in Python, Rust or JAX. The Rust and JAX implementations fulfil different use cases. The Rust implementation is optimised for computation on a CPU, while a GPU can accelerate the JAX implementation. The Python implementation offers no practical advantages over the Rust and JAX cores but is maintained for testing.

Rust is a statically typed, compiled systems-level programming language on par with `C` in speed, [135, 141, 142]. Traditionally, high-level languages such as Python provide a Garbage collector to reduce your control over memory management. Whereas languages like `c` provide functions like `free` and `allocate` "to shoot yourself in the foot". Rust introduces a new approach known as ownership and borrowing.

The Rust implementations of the default and thresholded algorithm take advantage of parallelism when sensible, based on workload at runtime, thanks to the functionality provided by Rayon [143]. In addition, we used caching and function memorisation to avoid re-evaluating the discrete charge configurations that need to be iterated over. However, appreciating that Rust is a relatively niche language compared to the ubiquitousness of Python, we interloped the Rust core with Python. This allows the user to gain all the benefits of Rust in their familiar Python coding environment.

JAX is a Python-based machine learning framework that combines autograd and XLA (accelerated linear algebra). Its structure and workflow mirror that of `numpy`; however, at run time, the code can be just in time (`jit`) compiled and auto-vectorised to primitive operations for significant

performance improvements. Through the `vmap` function, JAX will compile the functions to XLA and execute them in parallel with a GPU. Unfortunately, the discrete nature of the constant capacitance model negates the possibility of using JAX's autograd capabilities.

The brute-force and default algorithms were implemented in JAX. Sadly, the thresholded algorithm was incompatible with JAX as the sizes of all arrays must be known at compile time, and the JAX core cannot perform the thresholding discussed previously.

For the numerical solver to compute the continuous minimum, we used the OSQP (Operator Splitting Quadratic Program) solver (over those implemented by `scipy.optimize` [120]), which is optimised explicitly for constrained convex quadratic problems [144]. The solver requires only a single matrix factorisation for set-up, while all the other operations are computationally cheap, with no divisions required after the initialisation. As such, we found the OSQP solver converges more quickly than the scipy implementations - from a wall clock perspective.

Finally, we expressed the elements of the capacitance matrix in Farad per Coulomb ($\mathrm{F\,C^{-1}}$) rather than in pure SI units where the quantities are typically of the order of atto-Farad and sub-atto-Joules.

6.5 Performance

Here, we discuss the performance of the different implementations and algorithms. The evaluation focuses on the average computation time required to determine the ground state charge configuration for a 100×100 grid of randomly selected gate voltages, each associated with a randomly generated set of capacitance matrices. We do this for quantum dot arrays with up to 16 quantum dots. In Figure 33 (a-b), we benchmark the implementations of the brute-force and traditional algorithms executed on both the CPU within the Apple M1 Pro System on Chip (SOC) and an NVIDIA GTX 1080TI GPU.

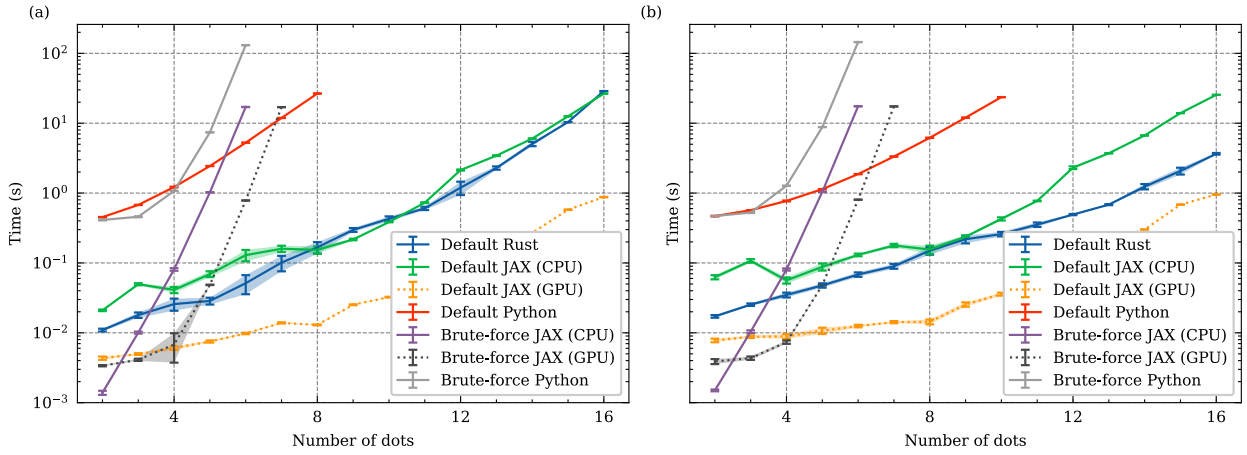


Figure 33: (a-b) Benchmarks measuring the average computation time required by each implementation of the brute-force and default algorithm to generate 100×100 pixels charge stability diagrams with gate voltage ranges covering many charge transitions. The computations involve randomly chosen capacitance matrices and gate voltages and are performed on increasing sizes of both open (left) and closed quantum dot arrays (right). The shaded region around the line indicates the two standard deviation confidence intervals such that 95% of the compute times fell within this region. We ran the JAX algoritms on both CPU and GPU and provided results for both configurations. For the closed quantum dot arrays, the number of confined electrons/holes matches the number of quantum dots within the array. For the brute-force algorithm, we set $n_{\max}$ to equal the number of dots in the array.

As shown in Figure 33 for arrays with more than five dots, the default algorithm is at least one order of magnitude faster than the brute-force algorithm. It gains a more significant advantage as the number of dots increases. Even the Python implementation of the default algorithm becomes faster than the JAX brute force for arrays with more than six quantum dots (on a CPU). However, the Rust and JAX implementations of the default algorithm are still faster due to the parallelisation and GPU usage offered by these cores. The GPU-accelerated JAX implementation can simulate the charge stability diagram of a 16-dot array in less than a second.

In Figure 34, we benchmark the Rust implementation of the thresholded algorithm for different threshold values. As expected, smaller threshold values result in faster computation. We also note larger time gains in the open dots regime compared to the closed dots regime. This discrepancy arises because when simulating quantum dot arrays in the closed regime, the threshold strategy may fail if no charge states with the correct number of charges are found. When this happens, the algorithm doubles the threshold value and recomputes the stability diagram. While this method ensures a correct stability diagram is generated, the additional computational overhead leads to smaller time gains than the open dot regime.

However, this extra performance might not necessarily come "for free". If the threshold value is too small, using the threshold algorithm will distort the charge stability diagram, as discussed in Section 6.7. To summarise that discussion, the critical threshold value depends upon the c_{dd} capacitance matrix, particularly the ratio of the on and off-diagonal elements. For example, in a double dot where the dots are very weakly capacitively coupled to each other compared to the coupling to the gates, the ratio will be small and small threshold values will be permissible - leading to significant speedups. By contrast, when the dots are strongly coupled, only thresholds close to one will not distort the charge stability diagram. Fortuitously, strongly coupled dots are poorly described by the constant capacitance model, as effects such as tunnel coupling become important. We motivate a method for determining the threshold in Appendices C.3.1 and C.3.2.

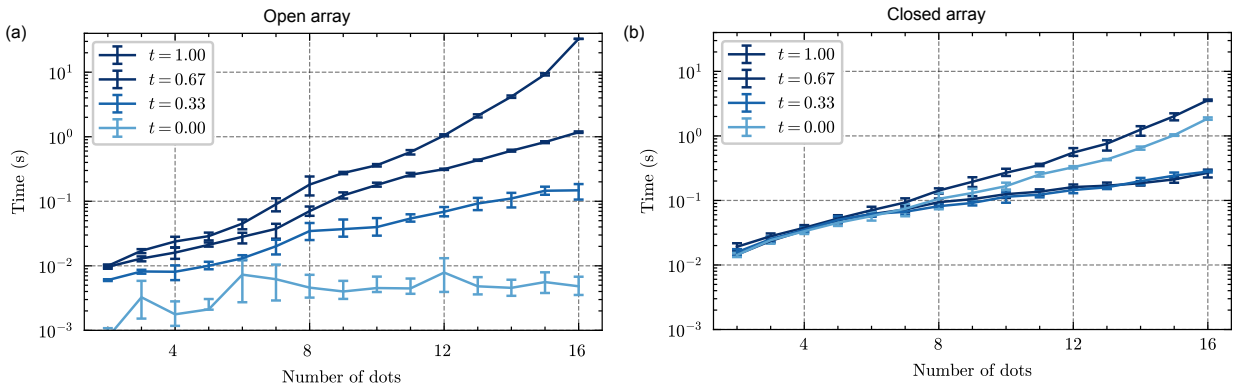


Figure 34: (a-b) Benchmarks measuring the average computation time required by the rust core to generate a 100×100 pixels charge stability diagram, with different thresholds, for increasing numbers of quantum dots in both the open and closed regimes. The number of charges enclosed for the closed array was set to the number of dots. The error bars denote two standard deviations.

6.6 Using QArray

To demonstrate the capabilities of `QArray`, I simulated the 20 charge stability diagrams, sweeping different gate pairs for both an open and closed quadruple quantum dot array (Fig. 35). We simulate the closed array with one, two, three and four holes confined within it. These charge stability diagrams demonstrate the complexity presented by an array containing even a moderate number of dots. Each panel was computed with a resolution of 150×150 pixels and took 10 ms to compute using the Rust implementation of the default algorithm.

Beyond simply computing idealised charge stability diagrams `QArray` can simulate the response from a charge sensor and account for the thermal broadening of transitions. Combined, these two capabilities make it possible to generate simulations that look considerably closer to experimental data.

Figure 36 (a) shows a measurement of the charge stability diagram of an open quadruple dot, using a charge sensor performed by Delbecq et al. [145]. In Figure 36 (b), we attempt to reconstruct this measurement in simulation using `QArray` with a resolution of 200×200 pixels.

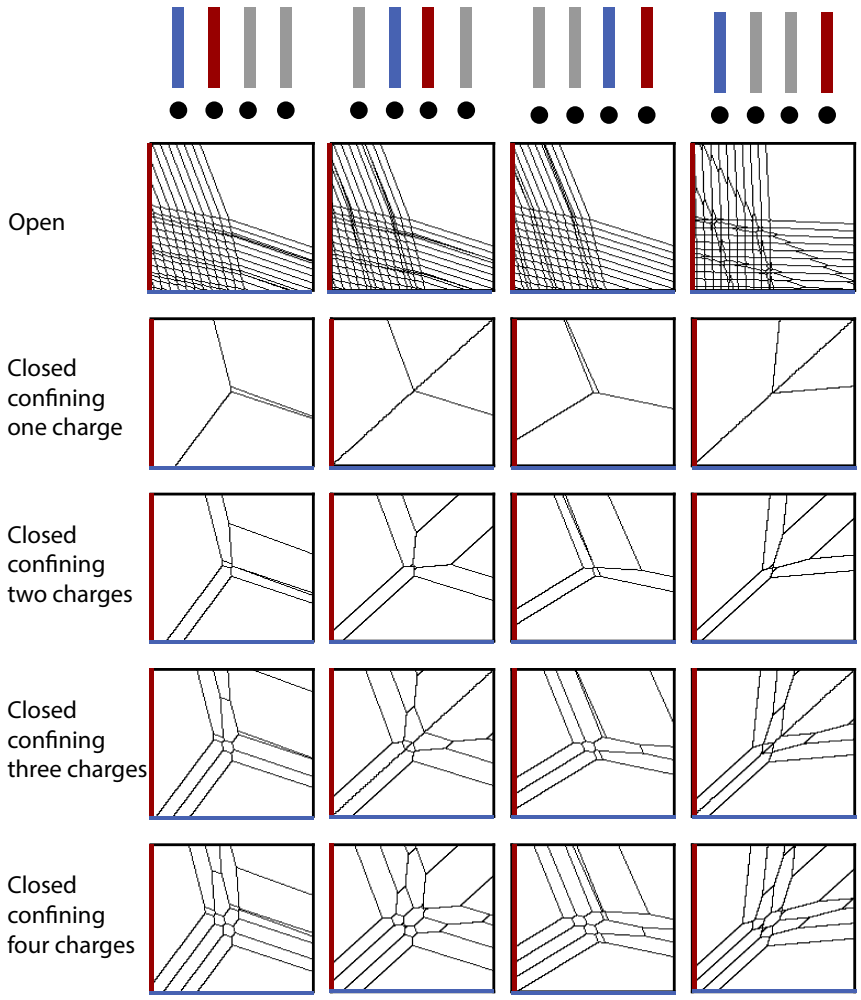


Figure 35: Charge stability diagrams of a quadruple dot device, as different combinations of gates (indicated by red and blue gates in the diagram at the top of each column) are swept. We plot these charge stability diagrams for open and closed quantum dot arrays, confining one, two, three and four holes. The black lines indicate wherever the charge state changes.

Algorithm	Implementation	Simulation time Figure 36 b) (s)	Simulation time Figure 36 d) (s)
Brute-force	Python	4.6	137.6
Brute-force	JAX	0.05 + 0.32 jit	21.1 + 0.4 jit
Default	Python	4.92	18.1
Default	JAX	0.09 + 0.8 jit	2.1+ 0.8 jit
Default	Rust	0.10	0.71
Thresholded $t = 1/2$	Python	2.81	10.8
Thresholded $t = 1/2$	Rust	0.08	0.66

Table 3: The compute times for all the different implementations of the brute-force, default and thresholded algorithms, to recreate plots b) and d) in Figure . For the JAX-based implementations, we include the one-off jit compile time. For the brute-force algorithm, we used $n_{\max} = 2$ and 5 for the open four-dot array and the closed five-dot array in b) and d), respectively.

Likewise, Figure 36 (c) shows the measurement from Mortemousque et al. [18] of the stability diagram of a five-electron closed configuration within a five-dot cross-geometry array. It displays a plot of a linear combination of charge sensor current derivatives for virtual gates δV_X^- (affecting left-right detuning) and δV_Y^- (affecting top-bottom detuning). In Figure 36 (d), we again recreate it in simulation with a resolution of 400×400 pixels. In both cases, we achieved good agreement between the simulation and the measurement visually. Of course, there are some discrepancies: the constant capacitance model does not capture the curvature of transitions due to tunnel coupling nor the possibility of the capacitive couplings between dots changing within the charge stability diagram. The code to recreate these simulations is provided as examples within the `QArray` package. Only some of the required capacitive couplings were reported in the manuscripts; the remaining ones we estimated based on geometric considerations of the device layout and then adjusted them manually to get the charge stability diagrams to match.

For both reconstructions, we used the Rust implementation of the default algorithm. However, we could have used any implementation of any algorithm; the compute times are listed in Table 3. All implementations of all algorithms produced virtually identical plots.

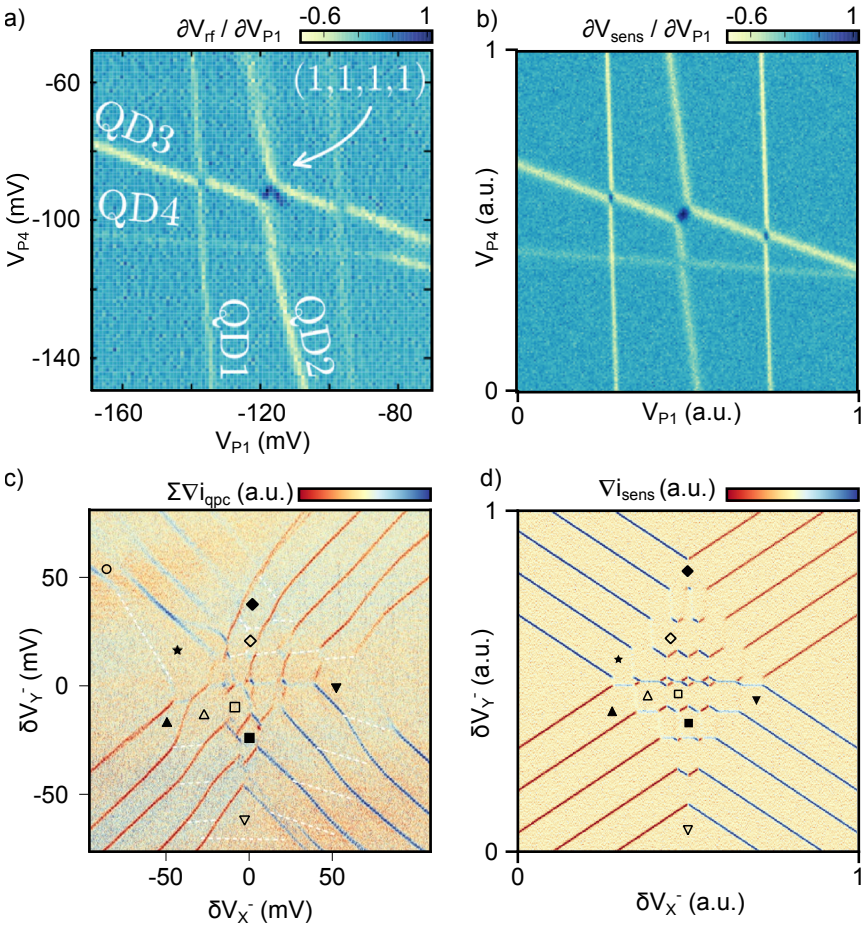


Figure 36: (a) Adaptation of figure 2b from Delbecq et al. [145] showing the stability diagram of a linear quadruple quantum dot array in the open regime as a function of the outermost pair of plunger gates. (b) A recreation/extension of Figure 2c from Delbecq et al. [145], simulated using `QArray`. Going beyond their original simulation, we can model the interdot transitions, thermal broadening and the effect of charge sensing for closer correspondence between the simulated and experimental data. (c) Adaptation of figure 3a from Mortemousque et al. [18] showing the stability diagram of a five-electron configuration within a five-dot cross-geometry array. It displays a plot of a linear combination of charge sensor current derivatives for virtual gates δV_X^- (affecting left-right detuning) and δV_Y^- (affecting top-bottom detuning). (d) A recreation of Figure 3a from Mortemousque et al. [18] using the `QArray` package.

6.7 Effect of the threshold

As evidenced in Figure 34, the thresholding strategy can yield significant reductions in the compute time by reducing the number of charge states evaluated to determine whether they are the lowest in energy. However, using values that are too small for the threshold will introduce artefacts in the charge stability diagram. Decreasing the threshold further will exaggerate these artefacts to the degree that for $t = 0$, all interdot transitions will be lost. We demonstrate this in Fig. 37 a), where we simulate a double quantum dot with dot-dot capacitance matrix

$$c_{dd} = \begin{bmatrix} 1.4 & -0.2 \\ -0.2 & 1.4 \end{bmatrix}. \tag{6.8}$$

In this case, our empirical understanding of the threshold suggests the minimum value should be the ratio of the off-diagonal elements to the on so $t_{\min} = 0.2/1.4 = 1/7$, this is confirmed by plots e) and f) in Figure 37 being identical. For smaller but non-zero values of the threshold, the charge stability diagram is distorted with the size of the interdot transition being suppressed with decreasing threshold (Fig. 37 b-d)). In Appendix C.3.2, we motivate a formula for the minimum value of the threshold t. In this case, the value suggested is more conservative at 0.202. While not wrong, this value is quite a bit larger than what empirical intuition suggests; more work is required to understand the thresholded algorithm fully.

In addition, even when the threshold is well above the minimum, the thresholding strategy appears to introduce almost imperceptible distortions into the charge stability diagram. Figure 38 compares the charge stability diagram produced by the Rust implementation threshold set to 1/3 with that produced by the brute-force implemented in JAX for a quadruple dot. For the capacitance matrix used in this simulation, our empirical understanding of the threshold value required to capture the interdot transitions accurately is $t_{\min} = 0.08$, whilst the analytical formula gives 0.19. We hypothesise that the thresholding exacerbates the error due finite precision of the OSQP solver.

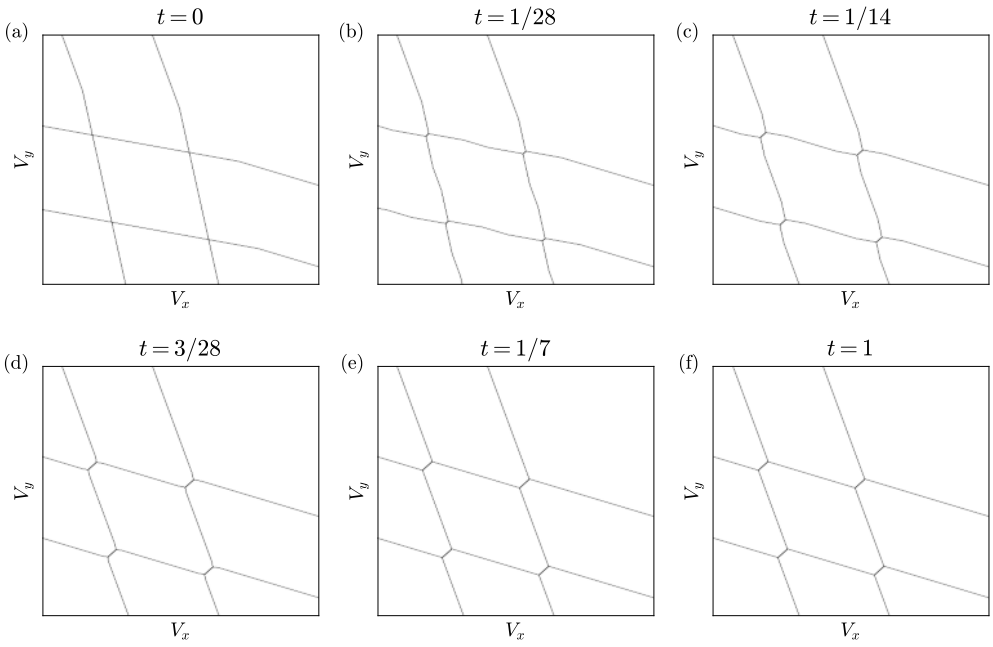


Figure 37: (a-e) Charge stability diagrams of an open double quantum dot produced by the thresholded algorithm for threshold values smaller than or equal to the empirical minimum value of 1/7 based on the capacitance matrix. The value of the threshold used in the simulation is stated above in the plot. f) Simulated charge stability diagram with the thresholded algorithm, with the threshold set to one, so it performs identically to the default algorithm.

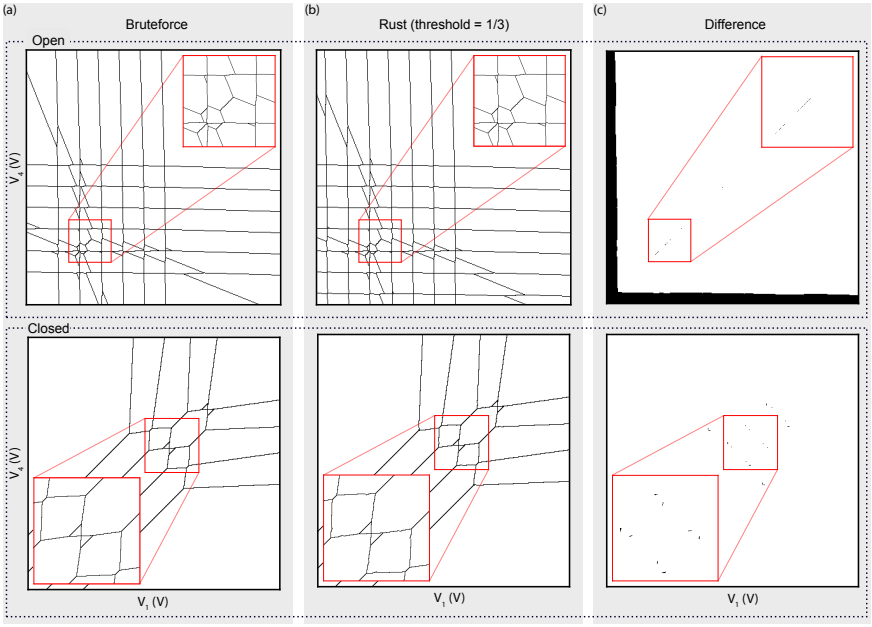


Figure 38: Diagrams illustrating the errors introduced by the thresholding strategy. (a) The charge stability diagram of a four-gate quadrupled quantum dot device as a function of the left and rightmost gates, in both the open and closed regimes, where four holes are confined. The brute-force core was used to compute the charge stability diagrams. (b) Identical charge stability diagram, but computed using the Rust core with a threshold of 1/3. (c) The pixel-wise difference between the brute force and the Rust core. Black pixels indicate that the predictions differ.

For example, if a threshold of $t = 2/3$ is used, the charge stability diagram is identical to brute force.

In summary, more work is required to understand the implications of the thresholding strategy fully. However, it is worth noting that these distortions in the charge stability diagram are next to irrelevant when using the constant capacitance model to generate training data for neural networks such as in references [96–98]. As effects we manually add to make the charge stability diagram look more "realistic", such as thermal broadening, measuring the charge stability diagram through a charge sensor and noise, will entirely swap the distortions. And the speed of the thresholded algorithm will allow for much larger more diverse training datasets.

6.8 Summary and outlook

In summary, I presented `qarray`, a software package that implements novel, better scaling algorithms to find the ground state charge configuration dramatically faster than previous versions of the constant capacitance model. With this speed comes the ability to simulate considerably larger quantum dot arrays of the size which could considered scalable unit cells, as per DiVincenzo's first criterion. On top of this, we built the functionality to include the effects of charge sensors and thermal broadening, allowing for the simulated scans to visually match those seen experimentally.

This model is freely available to download and use via the Python package index. I hope that members of the community use it to better understand their charge stability diagrams so that it is of use for their experimental effort. In addition, given that the model can reproduce experimental data, it is possible to use it for offline testing of tuning algorithms. This will become particularly vital for automating the operation of larger arrays, interpreting measurements, and navigating through the high dimensional charge stability diagram. In addition, the speed and GPU acceleration aid the creation of larger, more diverse training datasets for neural networks such as those used in Liu et al., Kalantre et al., Zwolak et al. [96–98]. These datasets could lead to higher accuracy charge state classifiers and, therefore, more successful automated device tuning.

Chapter 7

Qubits and hidden Markov models

> The hidden harmony is better than the obvious.
>
> Heraclitus

The work in the first half of this chapter was born out of a collaboration between the Australian full-stack quantum computing company, Diraq, a colleague of mine, Brandon Severin, who was based in Australia at the time, and myself. And it contributed to the publication:

Huang, J. Y., Su, R. Y., Lim, W. H., Feng, M., **Van Straaten, B.**, Severin, B., Gilbert, W., Stuyck, N. D., Tanttu, T., Serrano, S., Cifuentes, J. D., Hansen, I., Seedhouse, A. E., Vahapoglu, E., Leon, R. C. C., Abrosimov, N. V., Pohl, H., Thewalt, M. L. W., Hudson, F. E., Yang, C. H. *High-fidelity spin qubit operation and algorithmic initialization above 1 K*. Nature, 627(8005), 772-777. https://doi.org/10.1038/s41586-024-07160-2

7.1 Introduction

Assuming we could fabricate a large-scale semiconducting qubit chip, how would we control it? The two options are to fit many control lines in the dilution fridge and control the chip using classical control electronics located at hotter stages of the fridge (or even room temperature) or fabricate on-chip control electronics.

Both these approaches have the same fundamental issue: heat. Classical control electronics produce heat, whilst each line is a thermal connection to the hotter stages of the fridge. At milli-kelvin temperatures, dilution refrigerators only have a few tens of micro-watts of cooling power to remove this heat. Therefore, if you wish to operate your qubits at milli-kelvin temperatures, the thermal budget severely limits the number of fridge lines and/or complexity of the in-situ control.

So why not operate your qubits at kelvin temperatures where the fridge's cooling power is orders of magnitudes greater? For superconducting qubits, the answer is that you cannot. A host of physical effects, such as quasiparticle poisoning and excitation to the transmon's higher energy levels, make it impractical [146]. By contrast, for semiconducting qubits, it is possible, because quantum dot charging energies are large, typically on the order of 10 meV and the qubits couple weakly to the thermally activated charge noise. Over the last year, several papers have been published suggesting that semiconducting qubits are not significantly degraded at higher temperatures. So-called 'hot qubits' operate above 1 K, orders of magnitude hotter than the 10 mK base temperature of dilution refrigerators [41, 147–151].

Leading up to this work, Diraq were performing a comprehensive study on the temperature tolerance of two SiMOS electron-based Loss-DiVincenzo qubits hosted in a double quantum dot. Their qubits showed remarkable tolerance to these temperatures, with single and two-qubit gate fidelities above 99%. However, as temperatures increased, qubit initialisation became non-trivial as there ceased to be a cold bath for passive qubit reset. To circumvent this, Diraq developed an active reset protocol. However, evaluating the fidelity of this protocol was proving difficult. They could not

identify what was an initialisation error and what was a readout error. The problem was made even more difficult by their observation that the measurement sequence had a finite probability of causing a spin/charge state flip.

My contribution was to develop a hidden Markov model that could untangle SPAM errors and measurement-induced spin flips. I thereby obtained the fidelity of Diraq's initialisation technique - which Diraq has since patented. This is discussed in the first half of the coming chapter (Sec. 7.2).

The second half of the chapter then explores whether hidden Markov models can be implemented on FGPA to be used in real-time in the active initialisation process (Sec. 7.3). And whether this approach has advantages over the method Diraq developed.

7.2 Hidden Markov models for fidelity estimation

Here, we present a machine learning-based method we named *error-causation*, which analyses a series of repeated qubit measurements to extract the SPAM errors along with the probabilities that the qubit state changes during or as a result of the measurement. Our method is based on a HMMs [152]. However, this is not all our model can do; it can infer the true underlying spin state of the system based on observed measurements.

We investigate the effectiveness of our method *error-causation* on simulated data. From this, we conclude that it works and that the uncertainties in predicting initialisation, readout and spin-flip fidelities are dominated by variance in the underlying data rather than the estimations performed by the HMM. Then, we apply it to the experimental data from a Si-MOS device, obtaining initialisation fidelities up to 99.34% at temperatures of 1 K [153].

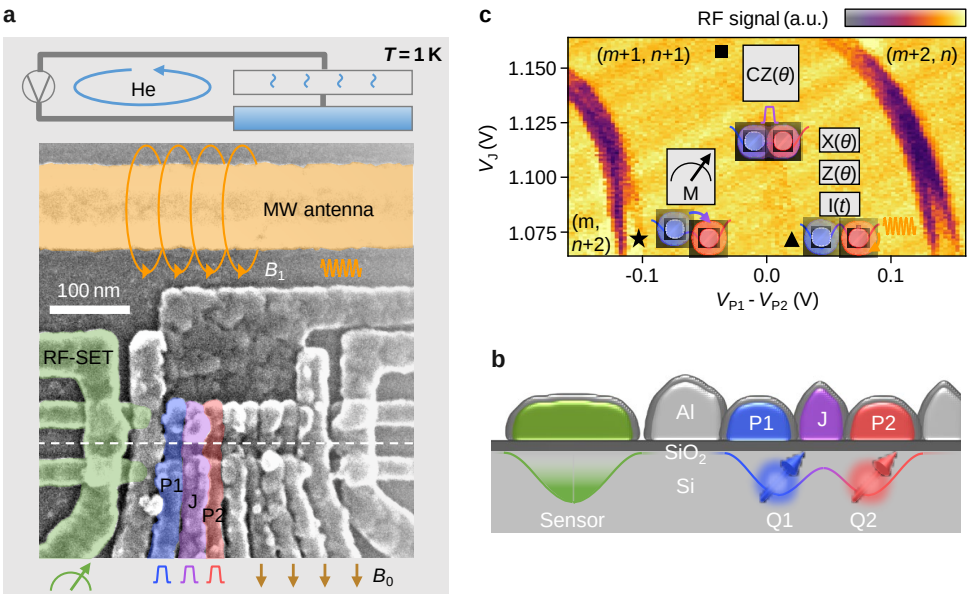


Figure 39: a) An SEM image of a Si-MOS device similar to that used in this work. Active gate electrodes and microwave antennae are highlighted in colour. B_0 and B_1 with their corresponding arrows are the external DC and the antenna-induced AC magnetic fields, respectively. The system operates at a temperature of 1 K. b) A cross-sectional schematic of the device architecture (along the dashed line in (a)) shows the material stack and the RF-SET sensor used to detect the corresponding state of the double quantum dot defined by dots Q1 and Q2. c) Charge stability diagram as a function of P1, P2 voltage detuning, $V_{P1} - V_{P2}$ and the J gate voltage V_J, showing the operation regime. The number of electrons in the left and right dot are given by m and n, respectively. The operation points for readout (M), single-qubit (X, Z, I) and two-qubit controlled phase (CZ) operation are labelled as star ($\star$), triangle ($\blacktriangle$) and square ($\blacksquare$), respectively. The insets schematically show the operations that are performed at each position. Reproduced from Ref. [153].

7.2.1 Methods

The device

Experimental data was obtained from two qubits realised in a Si-MOS gate-defined double quantum dot device (Fig. 39 a)). The plunger gates, P1 & P2, form the quantum dots beneath them, the J gate-electrode controls the coupling between the quantum dots. An odd number of electrons were loaded into each dot. The unpaired electrons on each dot are operated in the two-qubit basis of

$|\downarrow\downarrow\rangle, |\downarrow\uparrow\rangle, |\uparrow\downarrow\rangle, |\uparrow\uparrow\rangle$, with $\uparrow$ and $\downarrow$ signifying spin up and spin down respectively. States where the spins of the electrons are parallel are referred to as 'even', for example, $|\downarrow\downarrow\rangle$ and $|\uparrow\uparrow\rangle$. States where the spins of the electrons are antiparallel are referred to as 'odd', for example, $|\downarrow\uparrow\rangle$ and $|\uparrow\downarrow\rangle$.

The states were measured via parity readout, a method based on PSB (Sec. 2.3) [62] . A rf-SET operating at 210 MHz was used for readout of the spin states. Even states correspond to a blockade, resulting in a relatively low signal on the rf-SET, whereas odd states correspond to an unblockaded portion of the PSB region and, therefore, a relatively higher signal on the rf-SET. By setting a threshold for the rf-SET signal, even and odd states are classified and assigned binary values of 1 and 0, respectively. The classification data is then analysed a posteriori by a HMM, from the Python Package `hmmlearn` [154], to extract the state preparation and measurement errors.

Applying HMMs to our problem

To estimate the SPAM fidelities, we utilise a categorical HMM where the hidden state space is the qubit parity states, such that $s_0 \leftrightarrow |\downarrow\downarrow\rangle$ or $|\uparrow\uparrow\rangle$ whilst $s_1 \leftrightarrow |\downarrow\uparrow\rangle$ or $|\uparrow\downarrow\rangle$. The observation space is the measurement outcomes (after a threshold has been applied to the RF-SET data) such that m_0 and m_1 represent even and odd parity measurements, respectively. The initialisation fidelity is encoded in the initial probability vector, so if $P_{\text{init even}}$ ($P_{\text{init odd}}$) denotes the initialisation fidelity into an even (odd) state, then

$$\vec{\pi} = \begin{bmatrix} P_{\text{init even}} \\ P_{\text{init odd}} \end{bmatrix}. \tag{7.1}$$

where the probabilities of initialising into an even or odd state are complementary such that $P_{\text{init odd}} = \bar{P}_{\text{init even}} := 1 - P_{\text{init even}}$. Furthermore, if the fidelity of reading out the even and odd states are $P_{\text{read, even}}$ and $P_{\text{read, odd}}$, then the emission matrix is

$$\theta = \begin{bmatrix} P_{\text{read, even}} & \bar{P}_{\text{read, even}} \\ \bar{P}_{\text{read, odd}} & P_{\text{read, odd}} \end{bmatrix}. \tag{7.2}$$

And finally, if $P_{\text{even}\to\text{odd}}$ and $P_{\text{odd}\to\text{even}}$ denote the probability of a measurement causing a transition from the even to odd state and vice versa, then the transition matrix is

$$\Pi = \begin{bmatrix} P_{\text{even}\to\text{even}} & P_{\text{even}\to\text{odd}} \\ P_{\text{odd}\to\text{even}} & P_{\text{odd}\to\text{odd}} \end{bmatrix}. \tag{7.3}$$

Fitting the HMM

With this model defined, the *Baum-Welch* algorithm was used to fit it to the experimental observations, yielding best fitting the initialisation probability vector, $\vec{\pi}$, transition matrix, A, and emission matrix, θ.

To quantify the uncertainty in these parameters, we used the Cramér-Rao bound [155], which states that if $\text{est}_{\vec{\phi}}(\vec{y})$ is an unbiased estimate of the parameters $\vec{\phi}$ given the data $\vec{y}$, then

$$\text{cov}_{\vec{\phi}}\left(\text{est}_{\vec{\phi}}(\vec{y})\right) \geq I\left(\vec{\phi};\vec{y}\right)^{-1} \tag{7.4}$$

where $I(\vec{y}|\vec{\phi})_{ij} = -\partial^2 \log L(\vec{\phi}|\vec{y})/\partial\phi_i\partial\phi_j$ is the Fisher information matrix, whilst $L(\vec{y}|\vec{\phi})$ is the likelihood. Therefore, we can obtain lower bounds on each parameter's uncertainty from the diagonal elements of the inverse of the Fisher information matrix. This means that the uncertainty in the estimate of a particular parameter is inversely related to how sharp the maximum of the log-likelihood is. In our case, the parameters $\vec{\phi} = [P_{\text{init, even}}, P_{\text{read, even}}, P_{\text{read, odd}}, P_{\text{even}\to\text{odd}}, P_{\text{odd}\to\text{even}}]^T$ and the data is the measurement observations, such that $\vec{y} = \vec{m}$. These values were chosen to be approximately the values observed experimentally. The *Forward-Backward* algorithm is used to compute the likelihood of a set of measurements given a set of parameters, $L(\vec{\phi};\vec{m})$, and the gradients are calculated numerically using finite differences.

This method for quantifying uncertainty, in effect, uses a Laplace approximation around the maximum likelihood solution. In the field of machine learning, this can be regarded as rudimentary

compared to modern methods like MCMC and variational Bayes, which produce full posteriors, including uncertainty, as part of their operation. That said, the rudimentary method worked.

Once the model is fitted, we can use the *Viterbi* algorithm to find the most likely sequence of hidden qubit states based on a sequence of measurements.

7.2.2 Results

In the following sections, we describe the results of the simulated and experimental data.

Simulated results

To test and verify the performance of our method, we generated simulated parity readout measurement outcomes, with the aim of the simulated data set possessing similar qualitative characteristics to that of the real data from the Si-MOS device (Fig. 43). The simulated data was created using a Markov process with $P_{\text{init,even}} = 99.00\,\%$. The spin-flip and readout probabilities were set to $1.00\,\%$, $2.00\,\%$, $99.50\,\%$, and $99.00\,\%$, for $P_{\text{even}\to\text{odd}}$, $P_{\text{odd}\to\text{even}}$, $P_{\text{read,even}}$, and $P_{\text{read,odd}}$ respectively. The (predicted true) underlying states by the HMM from the simulated data are compared to the ground truth simulated states; the HMM demonstrates high accuracy as shown by the small number of incorrect predictions by the HMM in Fig. 43 (f)).

	Parameter				
	$P_{\text{init, even}}$	$P_{\text{even}\to\text{odd}}$	$P_{\text{odd}\to\text{even}}$	$P_{\text{read, even}}$	$P_{\text{read, odd}}$
Ground truth value	0.9900	0.0100	0.0200	0.9950	0.9900
Starting guess value	0.9000	0.1000	0.1000	0.9000	0.9000
Baum-Welch fitted value	0.9899	0.0097	0.0217	0.9949	0.9899
Uncertainty	0.0033	0.0008	0.0040	0.0006	0.0029

Table 4: A table containing the ground truth values of the HMM used to generate the data-set of 1000 repeats of 20 measurement sequences. The table includes the starting guesses used when the Baum-Welch algorithm is used to fit to this data-set, the values the Baum-Welch algorithm converged to and the corresponding uncertainties as per the Cramér-Rao bound.

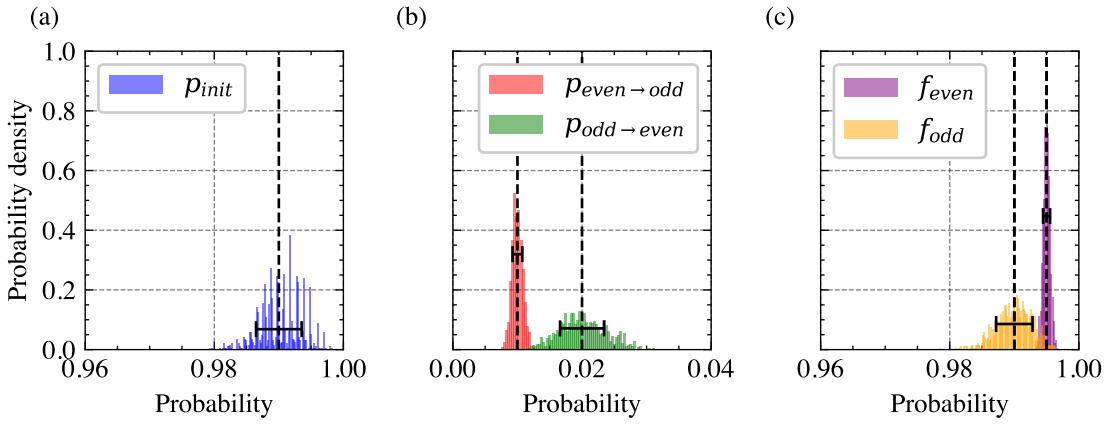


Figure 40: Histograms of the values fitted by the Baum-Welch when supplied 1000 randomly generated datasets, each consisting of 1000 repeats of 20 measurement sequences. In addition, the ground truth value used to generate the dataset and the expected uncertainty, as per the Cramér-Rao bound, are overlaid as a vertical line and error bars, respectively. The distributions of $P_{\text{init, even}}$, $P_{\text{even}\to\text{odd}}$ & $P_{\text{odd}\to\text{even}}$ and $P_{\text{read,even}}$ & $P_{\text{read, odd}}$ are shown in (a,b,c) respectively.

The performance of the *Baum-Welch* algorithm of the HMM and the parameters used to generate 1000 repeats of 20 PSB measurements are shown in Table 4. The Baum-Welch algorithm requires prior guess values to the respective probabilities before performing a fit, all of which were on the order of 10 % away from the ground truth values. The fitted values output by the Baum-Welch algorithm and their corresponding Cramér-Rao bound uncertainties are consistent with the ground truth probabilities, such that the ground truth value fell within the one standard deviation uncertainty 2/3 of the time.

We checked the reliability of our measure of uncertainty by generating 1000 data sets of 1000 repeats of 20 measurement sequences. We plotted histograms of the initialisation, spin-flip and readout probabilities output by the Baum-Welch algorithm in Figure 40. The same ground truth values (Tab. 4) were used for each data-set and are shown by the dashed vertical lines in Fig. 40. The horizontal error bars correspond to one standard deviation computed by the Cramér-Rao bound.

We investigated the dependence of the Cramér-Rao uncertainty bounds on data-set size (Fig. 41). We computed the log-likelihood of each state probability parameter from increasing fractions of

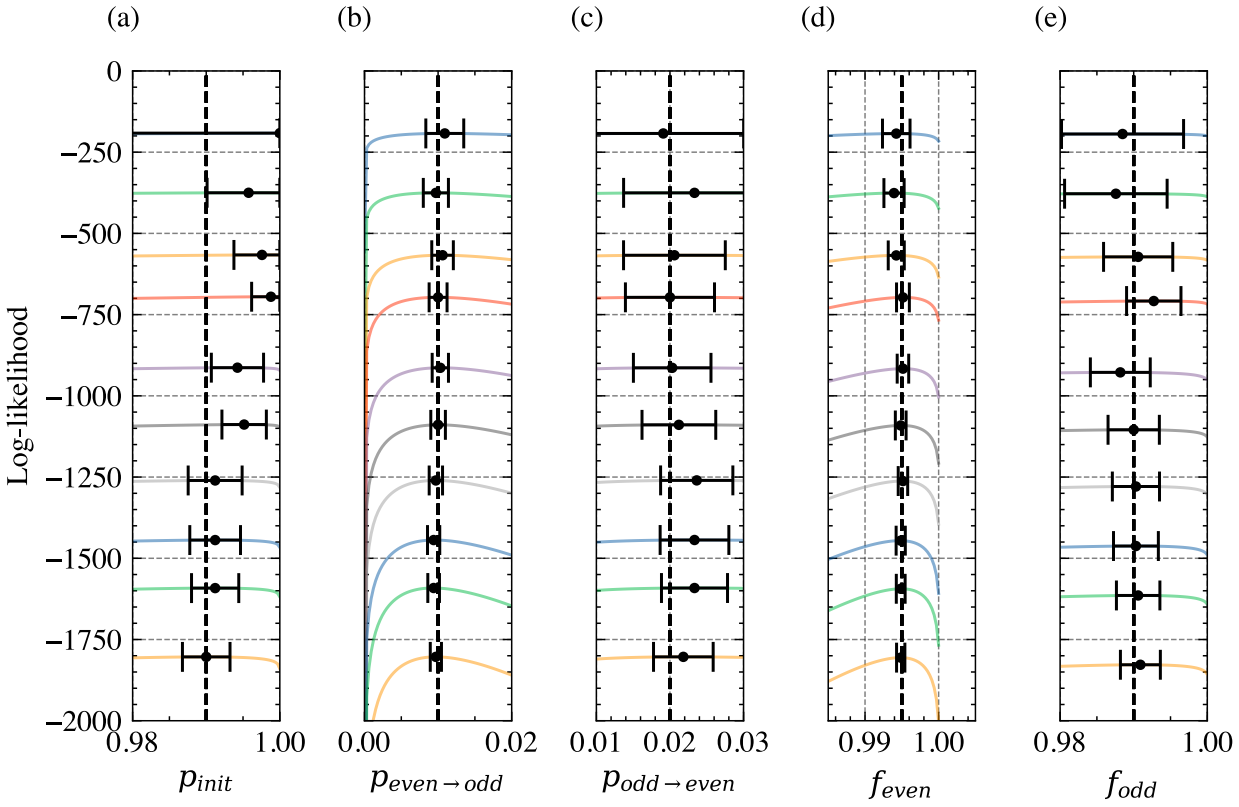
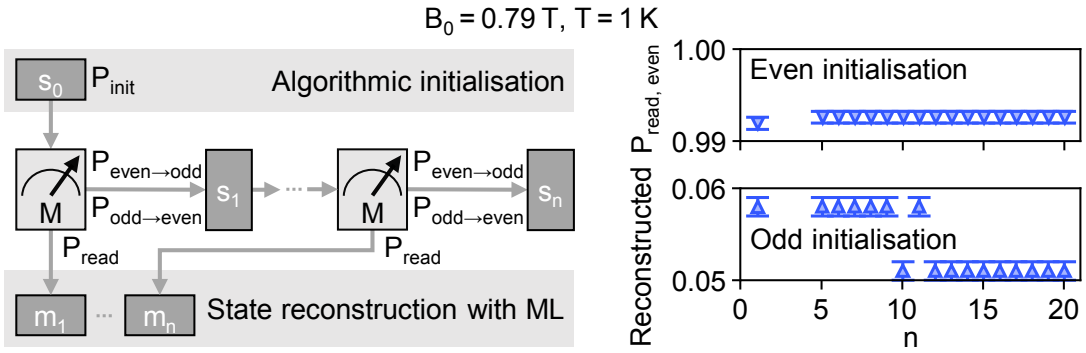


Figure 41: (a-e) Slices through the log-likelihood landscape for the Hidden Markov model detailed in table 4 generating a data-set of increasing fractions of a data-set of 1000 repeats of 20 measurement sequences. From top to bottom, the curves correspond to $10\%, 20\%, ..., 100\%$. The ground truth value used to generate the data-set for each parameter slice is overlaid as a dashed line. And for each data-set size, the most likely parameter value and its corresponding uncertainty are plotted as a point with error bars.



$P_{\text{init, even}}$ (%)	$P_{\text{even}\rightarrow\text{odd}}$ (%)	$P_{\text{odd}\rightarrow\text{even}}$ (%)	$P_{\text{read, even}}$ (%)	$P_{\text{read, odd}}$ (%)
99.34 ± 0.27	2.59 ± 0.13	1.97 ± 0.30	99.34 ± 0.08	96.15 ± 0.44

Figure 42: Schematic representing the measurement sequence and state reconstruction. The initial state s_0 (even or odd) is prepared via algorithmic initialisation and then n PSB readouts are performed throughout which the state evolves to s_n. SPAM error analysis is performed on the PSB readout measurements $m_1 ... m_n$, using a Categorical Hidden Markov model enabling predictions of initialisation, P_{init}, readout, P_{read}, and state change (or spin-flip), $P_{\text{even}\rightarrow\text{odd}}$ and $P_{\text{odd}\rightarrow\text{even}}$, fidelities. Using the model, the underlying state and corresponding PSB measurement probability can be reconstructed, P_{blockade}. Reproduced from reference [153].

a simulated data-set of measurement traces, 1000 repeats of 20 measurement sequences. The most likely parameter values (corresponding to the log-likelihood maxima) and their respective Cramér-Rao uncertainties were calculated for each corresponding sub-sample of the data-set. The Cramér-Rao uncertainties decrease with increasing data-set sizes across all SPAM parameters.

Experimental results

Repeated parity readout was performed on the Si-MOS device at a temperature of 1 K and a B_0 of 0.79 T for twenty iterations, $n = 20$, and the Hidden Markov model was used to reconstruct the states and perform state preparation and measurement error analysis (Fig. 42). Using the algorithmic initialisation developed by Diraq and presented in reference [153] the qubit was initialised into the even state ($|\downarrow\downarrow\rangle$) or the odd state ($|\uparrow\downarrow\rangle$), and 20 repeated readout cycles were performed. Using

the Hidden Markov model we can infer the state preparation, P_{init}, measurement, P_{read} fidelities as well as reconstruct the underlying qubit states based on the measurement outcomes $m_1...m_n$. Additionally the probability of respective state changes, $P_{\text{even}\to\text{odd}}$ and $P_{\text{odd}\to\text{even}}$. The initialisation fidelities were inferred as $99.34 \pm 0.27\,\%$, $94.67 \pm 0.73\,\%$ for $P_{\text{init,even}}$ and $P_{\text{init,odd}}$ respectively. The readout fidelities were inferred as $99.34 \pm 0.08\,\%$, $96.15 \pm 0.44\,\%$ for $P_{\text{read,even}}$ and $P_{\text{read,odd}}$ respectively. The probability for a spin flip occurring, $P_{\text{even}\to\text{odd}}$ and $P_{\text{odd}\to\text{even}}$, were inferred to be $2.59 \pm 0.13\,\%$ and $1.97 \pm 0.30\,\%$ respectively. The probability of PSB occurring, $P_{\text{read,even}}$, based on state reconstruction increases from $99.2\,\%$ to $99.3\,\%$ when $n = 5$ and initialised into the even state. Conversely, $P_{\text{read,even}}$ decreases from $5.8\,\%$ to $5.1\,\%$ at $n = 12$ when the system is initialised into the odd state.

The presence of these spin flips can be attributed to the finite T_1 time at the PSB point of 1.36 ± 0.06 ms and the small diabacitiy of the readout pulses.

7.2.3 Discussion

We can be confident in the SPAM parameters achieved in the Si-MOS device, Fig. 42, given our relatively low levels of uncertainty on simulated data where we can check the predicted states against the ground truth (Fig. 43). The use of HMMs for SPAM error analysis relies on the assumption that the qubit spin-flip behaviour is Markovian. This appears to be true and a reasonable assumption to make up until the limitation of qubit T_1 times in the device, which likely takes effect in the later stages of the PSB measurement sequences from $n > 20$ (Fig. 43).

In both the simulated and real data-set SPAM analysis, we see relatively high values of uncertainty for P_{init} (Fig. 42 & Tab. 4). This is because, relative to the other events, such as spin-flips or readout, initialisation only happens a limited number of times. Therefore, fewer statistics result in relatively high uncertainty. Whereas, $P_{\text{read,even}}$ has the lowest uncertainty out of all the parameters as there as so many readout examples to observe.

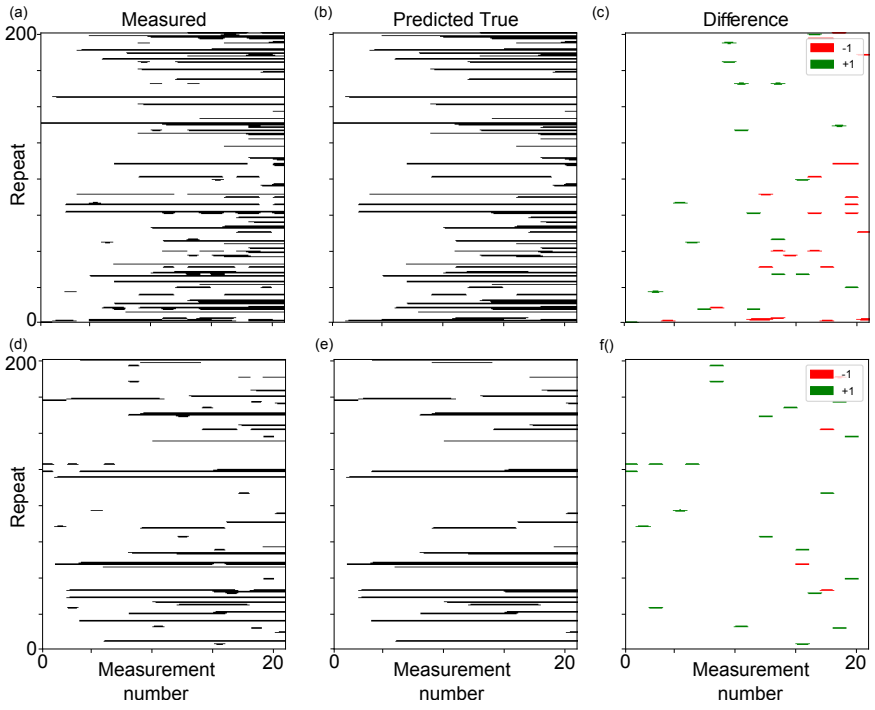


Figure 43: Experimental and simulated parity readout measurement outcomes and predicted states. a) Real parity readout measurements from the Si-MOS device. b) HMM Predictions of the true underlying hidden state. c) The difference between the measured state and the predicted state. d) Simulated parity readout measurements generated by the Hidden Markov model for even initialisation. e) The predicted true underlying hidden states of the simulated data. f) The difference between the measured state and the predicted state.

Approximately two-thirds of the predicted SPAM parameters for the simulated data, by the Baum-Welch algorithm, fall within a single standard deviation of the ground truth set by the Cramér-Rao bound (Fig. 40). This confirms that the Cramér-Rao bound is a suitable measure of uncertainty and hints that the source of uncertainty is not due to the Baum-Welch fitting process but to the underlying randomness of the generated data. Such that the variance in the parameters if we were to use the Baum Welch algorithm to fit the same data, repeatedly with different starting conditions, would be considerably smaller.

Additionally, we see the Cramér-Rao bound decrease with increasing data set size, Figure 41, providing further support for its use as a suitable measure of uncertainty. Empirically, for large

data sets, a parameter's uncertainty ϵ scales with the data set size, N, such that $\epsilon \propto N^{-1/2}$. Moreover, the decrease in uncertainty is reflected in the log-likelihood maxima becoming sharper with increasing data set size. As demonstrated in both the real and simulated data sets (Figure 42 & Tab. 4), parameters for which there are not abundant examples to train from such as P_{init} and $P_{\text{odd}\to\text{even}}$ have relatively wider Cramér-Rao bounds than other parameters even with larger data sets (Fig. 41).

7.3 Hidden Markov models for active reset

If your qubit is in a cold bath, reset is trivial: wait. After a few multiples of the qubit's T_1 lifetime, the qubit will have relaxed to its thermal ground state, which, to a good approximation, is the ground state. This is known as passive reset.

However, for Diraq's hot qubits, there was no cold bath, and their T_1 time was milliseconds, meaning passive reset was inappropriate. Therefore, they needed to perform an active reset - where the qubit is forced into the ground state by a sequence of measurements and pulses.

The active reset sequence for the Diraq's pair of Loss-Divinchezo qubits discussed previously is outlined in Fig. 44 a). In steps 1 - 3, the qubits are repeatedly loaded in a thermally mixed state; then their parity is measured. This is repeated until they are determined to be in an even parity state. In steps 4 - 6, a CNOT gate is applied to the even parity state; then the parity is measured again. If the parity is again even, the initialisation is complete. Otherwise, it has to be restarted. The time taken for each of the operations in the initialisation protocol is listed in Fig. 44 b) - with the integration time for parity measurements dwarfing the time cost of all other operations.

This reset protocol is, in effect, a filter. If and only if both qubits have their spin down at the end of step 1, will the algorithm reach step 7. Otherwise, one of the parity measurements along the way will yield an odd parity, and the algorithm will be restarted at step 1. Therefore, the expected

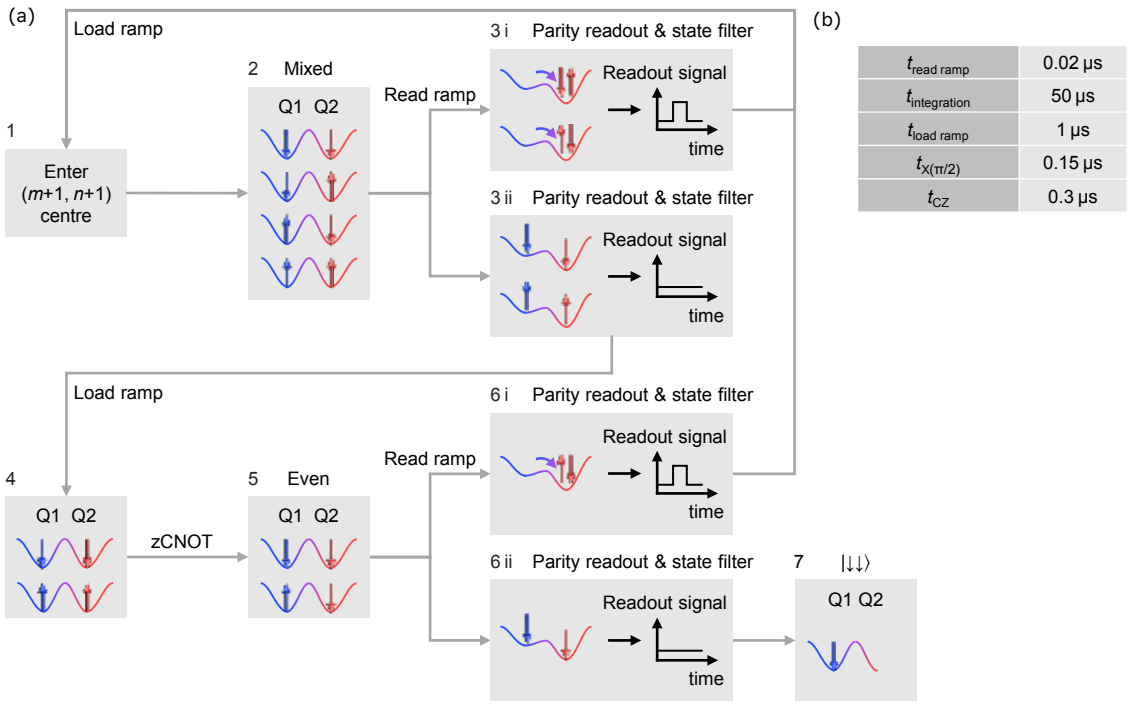


Figure 44: a) The full protocol of the algorithmic initialisation of a two-qubit system based on parity readout. In steps 1-3, the qubits are repeatedly loaded in a thermally mixed state; then their parity is measured. This is repeated until they are determined to be in an even parity state. In steps 4-6, a CNOT gate is applied to the even parity state; then the parity is measured again. If the parity is again even, the initialisation is complete. Otherwise, it has to be restarted. b) A table indicating the time required to perform each qubit operation. The ramp times to perform a load and readout ramp are $t_{\text{load ramp}}$ and $t_{\text{read ramp}}$. The integration time required to distinguish the parity states is $t_{\text{integration}}$. The time to perform a $\pi/2$ X single qubit rotation or a CZ two-qubit gate is is $t_{X(\pi/2)}$ and t_{CZ}

number of iterations is given by

$$\mathbb{E}[N_{\text{iter}}] = \sum_{n=1}^{\infty} p_{\downarrow\downarrow}(1-p_{\downarrow\downarrow})^{n-1} n = 1/p_{\downarrow\downarrow}. \tag{7.5}$$

where $p_{\downarrow\downarrow}$ denotes the probability both qubits are spin down following step 1.

In the next section, we develop a "greedy" algorithm based on a hidden Markov model to perform active reset faster than the Diraq algorithm.

7.3.1 Methods

Our protocol uses a hidden Markov model to track and update a posterior probability distribution for which spin state the qubits are in. This probability distribution is updated based on the outcome of parity-based measurements and on which qubit operations the algorithm has decided to enact. Based on this posterior, the algorithm decides the next qubit operation based on the most likely spin state after the action. This repeats until the posterior probability that the qubit is in the ground state is sufficient. The algorithm is terminated at this point, and the qubit is initialised. The protocol is discussed in the next section.

The protocol

Here, we define the HMM model required to track the posterior probabilities and detail the protocol. For two qubits, there are four possible spin states, $|\downarrow\downarrow\rangle, |\uparrow\downarrow\rangle, |\downarrow\uparrow\rangle, |\uparrow\uparrow\rangle$, therefore, we use a hidden Markov model with four hidden states. To capture the essence of parity readout, we use only two observation states corresponding to spin parity: even and odd. So, our emission matrix takes the form

$$\theta = \begin{bmatrix} p(\text{even}|\,|\downarrow\downarrow\rangle) & p(\text{even}|\,|\downarrow\uparrow\rangle) & p(\text{even}|\,|\uparrow\downarrow\rangle) & p(\text{even}|\,|\uparrow\uparrow\rangle) \\ p(\text{odd}|\,|\downarrow\downarrow\rangle) & p(\text{odd}|\,|\downarrow\uparrow\rangle) & p(\text{odd}|\,|\uparrow\downarrow\rangle) & p(\text{odd}|\,|\uparrow\uparrow\rangle) \end{bmatrix}. \tag{7.6}$$

The transition matrix is where we depart from a traditional HMM implementation - because we need more than one. We need one for each of the possible qubit operations allowed to be chosen by the protocol, which will be:

1. a CNOT gate,
2. an X gate on either qubit, followed by a CNOT,
3. An X gate on both qubits, followed by a CNOT.

However, it is important to distinguish that the transition matrix we attribute to each operation encodes the transformation of probabilities caused by the operation followed by a parity measurement. This is important as parity measurements are demolition measurements, as after the measurement, the unblocked odd parity state can be reloaded in either of the odd spin states. Notationally, we will denote each of these transition matrices as Π^a where $a \in \{I, X_1, X_2, X_1 \otimes X_2\} := \mathcal{A}$, dropping the CNOT operation common in all cases.

This posterior probability distribution can be obtained iteratively according to the update rule

$$\pi_i^{n+1} \propto \theta_{pi} \left(\sum_{j=1}^{4} \Pi_{ij}^a \pi_j^n \right). \tag{7.7}$$

where $\vec{\pi}^0$ encodes the prior knowledge about the qubit's spin state before the protocol starts. If we do not know anything about the spin state, we can set the prior probability vector $\vec{\pi}$ to have an equal probability in every state (in this case, 1/4). However, having so little knowledge would be strange, as typically, qubits are measured and then reinitialised, so we should know the parity of the qubits. So, in principle, we could encode this in the prior probability vector to save the protocol the cost of the first measurement.

Now that we have defined the hidden Markov model, we can define the rule to choose the next action. The strategy is simple: perform the necessary single qubit operations to turn the most likely qubit state in the posterior probability distribution into the $|\downarrow\downarrow\rangle$ spin state and then perform

a CNOT operation, see Tab. 5.

Most likely state	Chosen action
$\downarrow\downarrow$	CNOT
$\downarrow\uparrow$	CNOT $\otimes X_2$
$\uparrow\downarrow$	CNOT $\otimes X_1$
$\uparrow\uparrow$	CNOT $\otimes X_2 \otimes X_1$

Table 5: A table summarising the action chosen by the HMM-based protocol for each spin state.

The algorithm chooses to terminate once a sufficient posterior probability that the qubits are in the $|\downarrow\downarrow\rangle$ spin state. We will denote this threshold $t_{\text{term}} \in [0, 1]$, so that the algorithm terminates when

$$\pi^n_{\downarrow\downarrow} \geq t_{\text{term}}. \tag{7.8}$$

Gaussian HMMs

In the protocol described above, we used a categorical HMM, which necessitated us classifying the data beforehand. We can adapt our algorithm to take the unclassified raw demodulated in-phase voltage for the rf-set by using a Gaussian HMM. This is an example of soft-decoding, which has been shown to be beneficial [156].

Gaussian HMMs do away with the emission matrix and instead use a Gaussian distribution where the mean and standard deviation are conditioned upon the hidden state, such that $P(I|h) = \mathcal{N}(I; \mu_h, \sigma_h)$. We capture how parity measurements cannot distinguish the even spin states from one another by setting $\mu_{\uparrow\uparrow} = \mu_{\downarrow\downarrow}$. Likewise, for the odd parity states and the standard deviations. For the Gaussian Hidden Markov models, the update rule, based on Bayes rule, is

$$\pi^{n+1}_i \propto \mathcal{N}(I; \mu_h, \sigma_h) \left(\sum_{j=1}^{4} \mathbf{\Pi}^a_{ij} \pi^n_j \right). \tag{7.9}$$

Everything else to do with the protocol remains unchanged.

FPGA implementation

This section discusses implementing the protocol on the Quantum Machines OPX. Speaking reductively, the protocol consisted of a few matrix multiplications, array summations and one argmax. This functionality is shipped as standard as part of Quantum Machines `qua` or can be easily implemented in macros. As a result, implementing the algorithm was a simple matter of translating the formula discussed above into `qua`. So, for brevity, we will cover only the problems encountered along the way - rather than the implementation details.

FPGAs generally do not support floating point numbers; the more rudimentary fixed point format represents non-integer numbers. In particular, `qua` uses the `4.28` format, where 32 bits represent the number with one sign bit, three integer bits and 28 fractional bits. As such, these fixed-point numbers can represent the range $(-8, 8]$ in steps of $2^{-28} \approx 4 \times 10^{-9}$. The advantage of this format is that addition and subtraction can be performed in one clock cycle or 4 ns. In contrast, multiplication takes four clock cycles or 16 ns. And finally, the division is expensive, taking 96 clock cycles or 384 ns.

Unfortunately, our protocol necessitated a division, and the clock cycle overhead was incurred. This made it necessary to rescale the posterior probability vector $\vec{\pi}$ as calculated according to Eqs. (7.7) and (7.9). We avoided this overhead by computing the most likely spin state on the unscaled probabilities. Then, the renormalisation was performed concurrently with the next pulse sequence.

The fixed-point number format introduced another issue in computing our Gaussian probabilities in equation (7.9). This computation necessitated evaluating $(I-\mu)^2/2\sigma^2$; however, for $|I-\mu| > 4\sigma$ this will cause an overflow, as assigning a value will be larger than eight. If this were to happen, the value would be modulo 16 unsigned, such that $9.0 \to -7.0$. To avoid this, we clipped the I values so they could not exceed four standard deviations from the mean. Such events only occur 0.1% of the time, so this carried a negligible performance penalty.

Operation	Fidelity (%)
X_1, X_2	99.85
CNOT	98.82
Readout even	99.34
Readout odd	96.15

Table 6: A table summarising the fidelity of different operations used to simulate the active reset.

7.3.2 Results

Here, we describe the results associated with the active reset protocol. Unfortunately, the results in this section are only simulated - however, the `qua` code was written and working (up to unforeseen experimental issues).

For the simulation, we use the single X gate fidelities, CNOT fidelities and readout fidelities presented by Huang et al. [153] at 1 K. They are summarised in the Table 6. The achieved fidelity and the average number of required measurements are presented in Fig. 45. The categorical and Gaussian HMM-based protocols achieved higher fidelities than the Diraq method - in some cases using fewer measurements. The steps in the categorical hidden Markov model approach initialisation fidelity and an average number of parity measurements are due to the increasing threshold requiring a greater number of consecutive measurements of even parity before it is sufficiently confident to terminate.

However, evaluating the effectiveness of the HMM protocol with these relatively good readout fidelities hides its true merit. It should be able to combine the knowledge of the outcomes of multiple measurements to compensate for poor readout - but only when needed. Therefore, to investigate this, we evaluated its performance as a function of the simulated integration time, assuming the signal-to-noise ratio was proportional to $t_{\text{int}}^{-1/2}$, where t_{int} is the simulated integration time (Fig. 46).

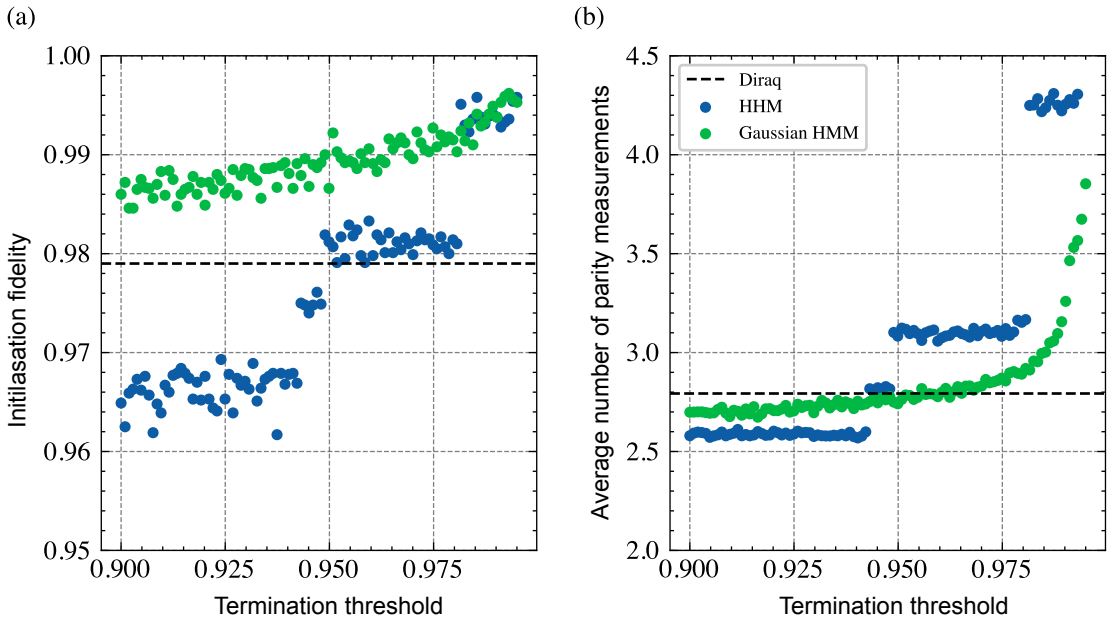


Figure 45: a) A plot showing the initialisation fidelity achieved by the categorical and Gaussian hidden Markov model-based reset protocols as a function of the termination threshold, t_{term}. Overlaid as a dashed line is the fidelity the Diraq protocol achieves for active reset. b) The average number of measurements required by the protocols. The average number of measurements the Diraq method needed is overlaid as a horizontal line. The protocol's wall clock time is simply the product of the average number of parity measurements multiplied by the integration time, to a good approximation.

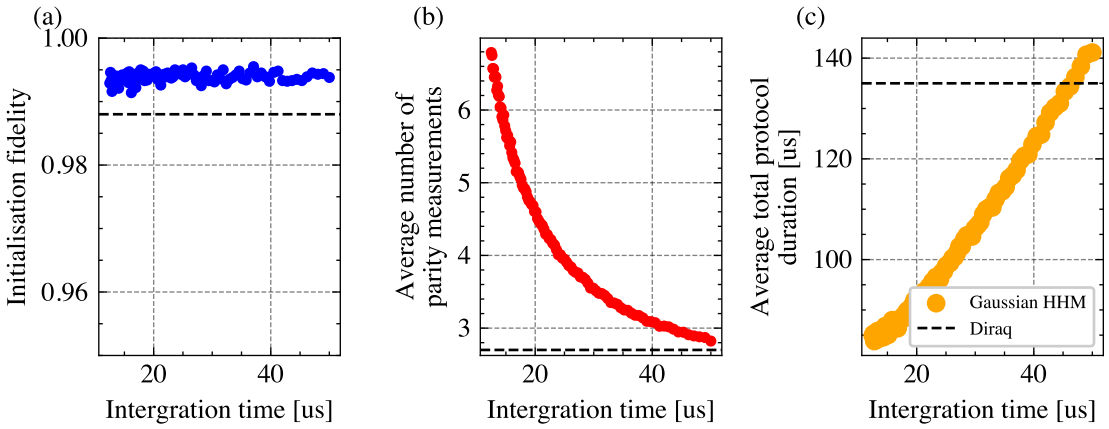


Figure 46: a) The initialisation fidelity achieved by the Gaussian HMM protocol for reduced parity measurement integration time. b) The average number of parity measurements required as a function of the measurement integration time. c) The average total protocol duration as a function of integration time. In plots a-c), the fidelity, average number of measurements and total protocol duration are presented for the Diraq method with an integration time of 50 µs is shown by the horizontal dashed line.

7.3.3 Discussion

Both the Categorical and Gaussian hidden Markov models achieved higher initialisation fidelities and often in fewer measurements than the Diraq protocol (Fig. 45). The advantage of the HMM-based protocols becomes more apparent when we reduce the integration time (Fig. 46). The protocol can compensate for the reduced readout fidelity by combining the outcomes of multiple measurements; therefore, the initialisation fidelity is not reduced, Fig. 46 a-b). However, the increase in the number of measurements is overshadowed by the reduction in the integration time, such that the total duration of the protocol decreases. Therefore, the Gaussian HMM-based protocol can initialise the qubits in the ground state with greater fidelity and in less time than the Diraq protocol - achieving a 99.3% initialisation fidelity compared to the Diraq's simulated 98.8%. This was possible in as little as two-thirds of the time.

7.4 Conclusion

We successfully demonstrate using a HMM to perform SPAM error analysis on Pauli spin blockade measurements from a Si-MOS device at a temperature of 1 kelvin, achieving initialisation and readout fidelities of 99.34 %. We verified the use and performance of our HMM for SPAM error analysis on simulated data. We extracted the ground truth simulated states while achieving SPAM probabilities and respective uncertainties that provide confidence in our method.

The lightweight nature of HMMs lent itself to being implemented on field-programmable arrays, enabling fast active reset of qubits at elevated temperatures based on online measurements and live signal processing. Finally, we presented simulated results that suggest HMMs could find applications in this area. Such developments are vital contributions to the growing drive towards rapid cryogenic stage on-chip multiplexing, signal processing and control of quantum devices [157–163], an inevitable evolution to achieve a universal fault-tolerant quantum computer based on spins in semiconductors.

Chapter 8

Conclusion

> To succeed, jump as quickly at opportunities as you do at conclusions.
>
> Benjamin Franklin

8.1 Looking back

In the earlier sections of this thesis, I emphasised the criticality of scalability within DiVincenzo's framework, highlighting the necessity for a scalable unit cell that could be seamlessly tuned. Concurrently, I underscored the inadequacy of transport measurements in meeting the readout criterion, thus advocating the adoption of radio-frequency measurements. Relying on transport measurements for tuning only to render them obsolete post-tuning is wasteful and hinders chip scaling, given the inherently limited scalability of transport measurements compared to radio-frequency techniques.

I introduced the first automated tuning algorithm to address this challenge, exclusively utilising high-bandwidth radio-frequency measurements. This algorithm represents a paradigm shift in tun-

ing methodologies, significantly reducing measurement costs while leveraging swift measurements alongside Gaussian processes, hypothesis tests, and a custom score function. Next, I developed another algorithm on this path that completely automates a charge sensor's operation. So that charge-sensing measurements can be incorporated with tuning algorithms without the issues associated with having them fall out of tune. I hope my work contributes to the transition towards tuning exclusively with rapid radio-frequency measurements. This advancement can stimulate exploration into innovative and more scalable architectures within the community, further advancing the field of quantum information processing.

Then, I introduced an implementation of the constant capacitance model, which can scale to simulate considerably larger devices. This tool will aid in developing new devices where the charge stability diagrams become increasingly complex. This model's utility increases as device quality improves and variability decreases. This will reduce the difficulty of the coarse tuning problem, such that defining quantum dots might be a case of setting predetermined gate voltages. However, this is unlikely to take the device to immediately correct the charge state - fine tuning will be required. My capacitance model is helpful in developing these tuning strategies and offline testing.

Finally, I used hidden Markov models to untangle SPAM errors from spin flip errors in hot qubits. The ability to do so will help us better understand the sources of these errors and ultimately eliminate them. These hidden Markov models were then applied to the active reset problems, showing promise according to simulations. However, the proof is in the experimental demonstration.

8.2 Looking forward

If I had another year, I would like to use QArray for its intended purpose: to generate a vast dataset to train a neural network to interpret charge stability diagrams to identify charge transitions, particularly the interdot transitions. This capability would immediately open up many possibilities.

In particular, we could use it to calibrate virtual plunger and barrier gates automatically. Virtual plunger gates are designed only to alter a single dot's potential. Virtual barrier gates change a barrier's potential without thickness without changing any of the dot's potentials. Such a virtual gate calibration is the natural successor of the charge sensor compensation work, where we, in effect, defined virtual gates that change the device dots' potential without affecting the charge sensor potential.

With the ability to calibrate virtual gates, I would automate moving through the change stability diagram to find a particular charge regime and then search for Pauli spin Blockade. I envision an algorithm that would move through the charge stability diagram until it finds the last transitions. It would then step back one charge transition and focus on the $(1, 1) \rightarrow (0, 2)$ interdot transition. With the calibrated virtual gates, it could then adjust tunnel barriers and sensor dots until it found PSB, where PSB could be identified by looking for latching at the interdot, which is sufficiently long-lived to be associated with charge. QArray already has the functionality to generate the latched charge stability diagrams required to train a neural network to identify PSB in this manner.

I would also like to explore tuning quantum dots in the isolated or closed regime [18]. In this regime, there are no transitions to the leads, which greatly reduces the number of charge transitions in the charge stability diagram. This might make it drastically easier to develop an automated tuning algorithm to navigate the charge stability diagram to find the $(1, 1) \rightarrow (0, 2)$ transition, where PSB could be found. However, in the isolated regime, it is less clear how to define useful virtual gates. A virtual gate, which raises the potential of a single dot, will cause charges to leave that dot to any of the other dots. It may be the case the most useful virtual gates are "detuning" gates, which raise the potential of one dot and lower another, such that charge transitions are guaranteed to be from one to the other. Is it possible to automatically calibrate these gates? Are they the most useful? I hope QArrays ability to simulate large dot systems in the isolated regime will help answer these questions.

A final avenue I would like to consider is how one would go about tuning larger arrays, in particular, tuning subsections in parallel. This work is the natural extension of the RF tuning work, where I tuned as fast as possible by reducing the measurement cost by exclusively using fast measurements. However, such a method will not scale to larger arrays if it has to be done sequentially.

Looking forward, I am hopeful that semiconducting chips will close the gap with the other qubit implementations. The automated tuning effort will continue to grow bigger, better, and faster.

Thank you for reading.

Bibliography

[1] Francesco Bova, Avi Goldfarb, and Roger G. Melko. Commercial applications of quantum computing. 8(1):2. ISSN 2196-0763. doi: 10.1140/epjqt/s40507-021-00091-1. URL https://doi.org/10.1140/epjqt/s40507-021-00091-1.

[2] Paul Benioff. The computer as a physical system: A microscopic quantum mechanical hamiltonian model of computers as represented by turing machines. *Journal of Statistical Physics*, 22:563–591, 05 1980. doi: 10.1007/BF01011339.

[3] Rajeev Acharya, Igor Aleiner, Richard Allen, Trond I. Andersen, Markus Ansmann, Frank Arute, Kunal Arya, Abraham Asfaw, Juan Atalaya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Joao Basso, Andreas Bengtsson, Sergio Boixo, Gina Bortoli, Alexandre Bourassa, Jenna Bovaird, and et al. Suppressing quantum errors by scaling a surface code logical qubit, 2022.

[4] Neereja Sundaresan, Theodore J Yoder, Youngseok Kim, Muyuan Li, Edward H Chen, Grace Harper, Ted Thorbeck, Andrew W Cross, Antonio D Córcoles, and Maika Takita. Matching and maximum likelihood decoding of a multi-round subsystem quantum error correction experiment. *arXiv preprint arXiv:2203.07205*, 2022.

[5] Chenlu Wang, Xuegang Li, Huikai Xu, Zhiyuan Li, Junhua Wang, Zhen Yang, Zhenyu Mi, Xuehui Liang, Tang Su, Chuhong Yang, Guangyue Wang, Wenyan Wang, Yongchao Li, Mo Chen, Chengyao Li, Kehuan Linghu, Jiaxiu Han, Yingshan Zhang, Yulong Feng, Yu Song, Teng Ma, Jingning Zhang, Ruixia Wang, Peng Zhao, Weiyang Liu, Guangming Xue, Yirong Jin, and Haifeng Yu. Towards practical quantum computers: transmon qubit with a lifetime approaching 0.5 milliseconds. *npj Quantum Information*, 8(1), January 2022. ISSN 2056-6387. doi: 10.1038/s41534-021-00510-2. URL http://dx.doi.org/10.1038/s41534-021-00510-2.

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

[7] Colin D. Bruzewicz, John Chiaverini, Robert McConnell, and Jeremy M. Sage. Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, 6(2), May 2019. ISSN 1931-9401. doi: 10.1063/1.5088164. URL http://dx.doi.org/10.1063/1.5088164.

[8] Sergei Slussarenko and Geoff J. Pryde. Photonic quantum information processing: A concise review. *Applied Physics Reviews*, 6(4):041303, 10 2019. ISSN 1931-9401. doi: 10.1063/1.5115814. URL https://doi.org/10.1063/1.5115814.

[9] Guido Burkard, Thaddeus D. Ladd, Andrew Pan, John M. Nichol, and Jason R. Petta. Semiconductor spin qubits. *Rev. Mod. Phys.*, 95:025003, Jun 2023. doi: 10.1103/RevModPhys.95.025003. URL https://link.aps.org/doi/10.1103/RevModPhys.95.025003.

[10] Loïc Henriet, Lucas Beguin, Adrien Signoles, Thierry Lahaye, Antoine Browaeys, Georges-Olivier Reymond, and Christophe Jurczak. Quantum computing with neutral atoms. *Quantum*, 4:327, September 2020. ISSN 2521-327X. doi: 10.22331/q-2020-09-21-327. URL http://dx.doi.org/10.22331/q-2020-09-21-327.

[11] David P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48(9-11):771–783, September 2000. ISSN 1521-3978. URL https://arxiv.org/abs/quant-ph/0002077.

[12] T. Kobayashi, T. Nakajima, K. Takeda, A. Noiri, J. Yoneda, and S. Tarucha. Feedback-based active reset of a spin qubit in silicon. *npj Quantum Information*, 9(1), June 2023. ISSN 2056-6387. doi: 10.1038/s41534-023-00719-3. URL http://dx.doi.org/10.1038/s41534-023-00719-3.

[13] C. H. Yang, R. C. C. Leon, J. C. C. Hwang, A. Saraiva, T. Tanttu, W. Huang, J. Camirand Lemyre, K. W. Chan, K. Y. Tan, F. E. Hudson, K. M. Itoh, A. Morello, M. Pioro-Ladrière, A. Laucht, and A. S. Dzurak. Operation of a silicon quantum processor unit cell above one kelvin. *Nature*, 580(7803):350–354, April 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2171-6. URL https://doi.org/10.1038/s41586-020-2171-6.

[14] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2011. ISBN 9781107002173. URL https://www.amazon.com/Quantum-Computation-Information-10th-Anniversary/dp/1107002176?SubscriptionId=AKIAIOBINVZYXZQZ2U3A&tag=chimbori05-20&linkCode=xm2&camp=2025&creative=165953&creativeASIN=1107002176.

[15] J. R. Petta, A. C. Johnson, J. M. Taylor, E. A. Laird, A. Yacoby, M. D. Lukin, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Coherent manipulation of coupled electron spins in semiconductor quantum dots. *Science*, 309(5744):2180–2184, 2005. doi: 10.1126/science.1116955. URL https://www.science.org/doi/abs/10.1126/science.1116955.

[16] John M. Nichol, Lucas A. Orona, Shannon P. Harvey, Saeed Fallahi, Geoffrey C. Gardner, Michael J. Manfra, and Amir Yacoby. High-fidelity entangling gate for double-quantum-dot spin qubits. 3(1):3. ISSN 2056-6387. doi: 10.1038/s41534-016-0003-1. URL https://doi.org/10.1038/s41534-016-0003-1.

[17] Federico Fedele, Anasua Chatterjee, Saeed Fallahi, Geoffrey C. Gardner, Michael J. Manfra, and Ferdinand Kuemmeth. Simultaneous operations in a two-dimensional array of

singlet-triplet qubits. *PRX Quantum*, 2(4), October 2021. ISSN 2691-3399. doi: 10.1103/prxquantum.2.040306. URL http://dx.doi.org/10.1103/PRXQuantum.2.040306.

[18] Pierre-André Mortemousque, Emmanuel Chanrion, Baptiste Jadot, Hanno Flentje, Arne Ludwig, Andreas D Wieck, Matias Urdampilleta, Christopher Bäuerle, and Tristan Meunier. Coherent control of individual electron spins in a two-dimensional quantum dot array. *Nature Nanotechnology*, 16(3):296–301, 2021.

[19] C. Volk, A. M.J. Zwerver, U. Mukhopadhyay, P. T. Eendebak, C. J. van Diepen, J. P. Dehollain, T. Hensgens, T. Fujita, C. Reichl, W. Wegscheider, and L. M.K. Vandersypen. Loading a quantum-dot based QbyteĂİ register. *npj Quantum Information*, 5(1):1–12, 2019. ISSN 20566387.

[20] Federico Fedele, Anasua Chatterjee, Saeed Fallahi, Geoffrey C. Gardner, Michael J. Manfra, and Ferdinand Kuemmeth. Simultaneous operations in a two-dimensional array of singlet-triplet qubits. *PRX Quantum*, 2:040306, Oct 2021.

[21] M. Veldhorst, H. G.J. J Eenink, C. H. Yang, and A. S. Dzurak. Silicon CMOS architecture for a spin-based quantum computer. *Nature Communications*, 8(1), 12 2017. ISSN 2041-1723. doi: 10.1038/s41467-017-01905-6. URL http://dx.doi.org/10.1038/s41467-017-01905-6.

[22] Brandon Buonacorsi, Zhenyu Cai, Eduardo B. Ramirez, Kyle S. Willick, Sean M. Walker, Jiahao Li, Benjamin D. Shaw, Xiaosi Xu, Simon C. Benjamin, and Jonathan Baugh. Network architecture for a topological quantum computer in silicon. *Quantum Science and Technology*, 4(2), 2019. ISSN 20589565. doi: 10.1088/2058-9565/aaf3c4. URL https://iopscience.iop.org/article/10.1088/2058-9565/aaf3c4/meta.

[23] Zhenyu Cai, Michael A. Fogarty, Simon Schaal, Sofia Patomäki, Simon C. Benjamin, and John J. L. Morton. A Silicon Surface Code Architecture Resilient Against Leakage Errors. *Quantum*, 3:212, 12 2019. ISSN 2521-327X. doi: 10.22331/q-2019-12-09-212.

[24] Joe O'gorman, Naomi H. Nickerson, Philipp Ross, John J.L. Morton, and Simon C. Benjamin. A silicon-based surface code quantum computer. *npj Quantum Information*, 2(1):1–14, 2 2016. ISSN 20566387. doi: 10.1038/npjqi.2015.19.

[25] Peter Stano and Daniel Loss. Review of performance metrics of spin qubits in gated semiconducting nanostructures. *Nature Reviews Physics*, 4(10):672–688, August 2022. ISSN 2522-5820. doi: 10.1038/s42254-022-00484-w. URL http://dx.doi.org/10.1038/s42254-022-00484-w.

[26] Jarryd J. Pla, Kuan Y. Tan, Juan P. Dehollain, Wee H. Lim, John J.L. 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. *Nature*, 496(7445):334–338, 4 2013. ISSN 00280836. doi: 10.1038/nature12011. URL http://www.ncbi.nlm.nih.gov/pubmed/23598342https://www.nature.com/articles/nature12011.

[27] Juha T. Muhonen, Juan P. Dehollain, Arne Laucht, Fay E. Hudson, Rachpon Kalra, Takeharu

Sekiguchi, Kohei M. Itoh, David N. Jamieson, Jeffrey C. McCallum, Andrew S. Dzurak, and Andrea Morello. Storing quantum information for 30 seconds in a nanoelectronic device. *Nature Nanotechnology*, 9(12):986–991, 2014. ISSN 17483395. doi: 10.1038/nnano.2014.211.

[28] 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. *Nat Rev Mater*, 6(10):926–943, October 2021. ISSN 2058-8437. doi: 10.1038/s41578-020-00262-z. URL https://www.nature.com/articles/s41578-020-00262-z.

[29] N. W. Hendrickx, D. P. Franke, A. Sammak, G. Scappucci, and M. Veldhorst. Fast two-qubit logic with holes in germanium. *Nature*, 577(7791):487–491, January 2020. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-019-1919-3. URL http://www.nature.com/articles/s41586-019-1919-3.

[30] Nico W. Hendrickx, William I. L. Lawrie, Maximilian Russ, Floor van Riggelen, Sander L. de Snoo, Raymond N. Schouten, Amir Sammak, Giordano Scappucci, and Menno Veldhorst. A four-qubit germanium quantum processor. *Nature*, 591(7851):580–585, March 2021. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-021-03332-6. URL http://www.nature.com/articles/s41586-021-03332-6.

[31] Daniel Jirovec, Andrea Hofmann, Andrea Ballabio, Philipp M. Mutter, Giulio Tavani, Marc Botifoll, Alessandro Crippa, Josip Kukucka, Oliver Sagi, Frederico Martins, Jaime Saez-Mollejo, Ivan Prieto, Maksim Borovkov, Jordi Arbiol, Daniel Chrastina, Giovanni Isella, and Georgios Katsaros. A singlet-triplet hole spin qubit in planar Ge. *Nature Materials*, 20(8):1106–1112, August 2021. ISSN 1476-1122, 1476-4660. doi: 10.1038/s41563-021-01022-2. URL http://www.nature.com/articles/s41563-021-01022-2.

[32] 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. *Nat Electron*, 5(3):178–183, March 2022. ISSN 2520-1131. doi: 10.1038/s41928-022-00722-0. URL https://www.nature.com/articles/s41928-022-00722-0.

[33] 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, Erik P. A. M Bakkers, Stefano Bosco, J. Carlos Egues, Daniel Loss, and Dominik M. Zumbühl. A qubit with simultaneously maximized speed and coherence, 2024.

[34] M Veldhorst, J C C Hwang, C H Yang, A W Leenstra, B de Ronde, J P Dehollain, J T Muhonen, F E Hudson, K M Itoh, A Morello, and A S Dzurak. An addressable quantum dot qubit with fault-tolerant control-fidelity. *Nature Nanotechnology*, 9(12):981–985, 2014. ISSN 1748-3395. doi: 10.1038/nnano.2014.216. URL https://doi.org/10.1038/nnano.2014.216.

[35] R. Maurand, X. Jehl, D. Kotekar-Patil, A. Corna, H. Bohuslavskyi, R. Laviéville, L. Hutin,

S. Barraud, M. Vinet, M. Sanquer, and S. De Franceschi. A CMOS silicon spin qubit. *Nature Communications*, 7(1):3–8, 11 2016. ISSN 20411723. doi: 10.1038/ncomms13575.

[36] Hendrik Bluhm, Sandra Foletti, Izhar Neder, Mark Rudner, Diana Mahalu, Vladimir Umansky, and Amir Yacoby. Dephasing time of GaAs electron-spin qubits coupled to a nuclear bath exceeding 200μs. *Nature Physics*, 7(2):109–113, 12 2011. ISSN 17452481. doi: 10.1038/nphys1856.

[37] M. Veldhorst, C. H. Yang, J. C.C. Hwang, W. Huang, J. P. Dehollain, J. T. Muhonen, S. Simmons, A. Laucht, F. E. Hudson, K. M. Itoh, A. Morello, and A. S. Dzurak. A two-qubit logic gate in silicon. *Nature*, 526(7573):410–414, 10 2015. ISSN 14764687. doi: 10.1038/nature15263.

[38] Tong Wu and Jing Guo. Variability and fidelity limits of silicon quantum gates due to random interface charge traps. *IEEE Electron Device Letters*, 41(7):1078–1081, 2020. doi: 10.1109/LED.2020.2997009.

[39] Mario Lodari, Nico W Hendrickx, William I L Lawrie, Tzu-Kan Hsiao, Lieven M K Vandersypen, Amir Sammak, Menno Veldhorst, and Giordano Scappucci. Low percolation density and charge noise with holes in germanium. *Materials for Quantum Technology*, 1(1):011002, jan 2021. doi: 10.1088/2633-4356/abcd82. URL `https://dx.doi.org/10.1088/2633-4356/abcd82`.

[40] C. H. Yang, A. Rossi, R. Ruskov, N. S. Lai, F. A. Mohiyaddin, S. Lee, C. Tahan, G. Klimeck, A. Morello, and A. S. Dzurak. Spin-valley lifetimes in a silicon quantum dot with tunable valley splitting. *Nature Communications*, 4(1):1–8, 6 2013. ISSN 20411723. doi: 10.1038/ncomms3069.

[41] C. H. Yang, R. C.C. Leon, J. C.C. Hwang, A. Saraiva, T. Tanttu, W. Huang, J. Camirand Lemyre, K. W. Chan, K. Y. Tan, F. E. Hudson, K. M. Itoh, A. Morello, M. Pioro-Ladrière, A. Laucht, and A. S. Dzurak. Operation of a silicon quantum processor unit cell above one kelvin. *Nature*, 580(7803):350–354, 4 2020. ISSN 14764687. doi: 10.1038/s41586-020-2171-6.

[42] Nico W. Hendrickx, William I.L. Lawrie, Maximilian Russ, Floor van Riggelen, Sander L. de Snoo, Raymond N. Schouten, Amir Sammak, Giordano Scappucci, and Menno Veldhorst. A four-qubit germanium quantum processor. *Nature*, 591(7851):580–585, 3 2021. ISSN 14764687. doi: 10.1038/s41586-021-03332-6. URL `https://doi.org/10.1038/s41586-021-03332-6`.

[43] Akito Noiri, Kenta Takeda, Takashi Nakajima, Takashi Kobayashi, Amir Sammak, Giordano Scappucci, and Seigo Tarucha. Fast universal quantum gate above the fault-tolerance threshold in silicon. *Nature 2022 601:7893*, 601:338–342, 1 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04182-y. URL `https://www.nature.com/articles/s41586-021-04182-y`.

[44] Xiao Xue, Maximilian Russ, Nodar Samkharadze, Brennan Undseth, Amir Sammak, Giordano Scappucci, and Lieven M.K. Vandersypen. Quantum logic with spin qubits crossing the surface code threshold. *Nature 2022 601:7893*, 601:343–347, 1 2022. ISSN

1476-4687. doi: 10.1038/s41586-021-04273-w. URL https://www.nature.com/articles/s41586-021-04273-w.

[45] Mateusz T. Madzik, Serwan Asaad, Akram Youssry, Benjamin Joecker, Kenneth M. Rudinger, Erik Nielsen, Kevin C. Young, Timothy J. Proctor, Andrew D. Baczewski, Arne Laucht, Vivien Schmitt, Fay E. Hudson, Kohei M. Itoh, Alexander M. Jakob, Brett C. Johnson, David N. Jamieson, Andrew S. Dzurak, Christopher Ferrie, Robin Blume-Kohout, and Andrea Morello. Precision tomography of a three-qubit donor quantum processor in silicon. *Nature 2022 601:7893*, 601(7893):348–353, 1 2022. ISSN 1476-4687. doi: 10.1038/s41586-021-04292-7. URL https://doi.org/10.1038/s41586-021-04292-7.

[46] Erika Kawakami, Thibaut Jullien, Pasquale Scarlino, Daniel R. Ward, Donald E. Savage, Max G. Lagally, Viatcheslav V. Dobrovitski, Mark Friesen, Susan N. Coppersmith, Mark A. Eriksson, and Lieven M.K. Vandersypen. Gate fidelity and coherence of an electron spin in an Si/SiGe quantum dot with micromagnet. *Proceedings of the National Academy of Sciences of the United States of America*, 113(42):11738–11743, 10 2016. ISSN 10916490. doi: 10.1073/pnas.1603251113.

[47] W. I.L. Lawrie, H. G.J. Eenink, N. W. Hendrickx, J. M. Boter, L. Petit, S. V. Amitonov, M. Lodari, B. Paquelet Wuetz, C. Volk, S. G.J. Philips, G. Droulers, N. Kalhor, F. Van Riggelen, D. Brousse, A. Sammak, L. M.K. Vandersypen, G. Scappucci, and M. Veldhorst. Quantum dot arrays in silicon and germanium. *Applied Physics Letters*, 116(8):080501, 2 2020. ISSN 0003-6951. doi: 10.1063/5.0002013. URL https://aip.scitation.org/doi/abs/10.1063/5.0002013http://aip.scitation.org/doi/10.1063/5.0002013.

[48] T. A. Baart, P. T. Eendebak, C. Reichl, W. Wegscheider, and L. M.K. Vandersypen. Computer-automated tuning of semiconductor double quantum dots into the single-electron regime. *Applied Physics Letters*, 108(21), 2016. ISSN 00036951. doi: 10.1063/1.4952624. URL http://dx.doi.org/10.1063/1.4952624.

[49] Sandesh S. Kalantre, Justyna P. Zwolak, Stephen Ragole, Xingyao Wu, Neil M. Zimmerman, M. D. Stewart, and Jacob M. Taylor. Machine learning techniques for state recognition and auto-tuning in quantum dots. *npj Quantum Information*, 5(1):1–10, 2019. ISSN 20566387. doi: 10.1038/s41534-018-0118-7. URL http://dx.doi.org/10.1038/s41534-018-0118-7.

[50] Y. Xu, F. K. Unseld, A. Corna, A. M. J. Zwerver, A. Sammak, D. Brousse, N. Samkharadze, S. V. Amitonov, M. Veldhorst, G. Scappucci, R. Ishihara, and L. M. K. Vandersypen. On-chip Integration of Si/SiGe-based Quantum Dots and Switched-capacitor Circuits. 5 2020. URL http://arxiv.org/abs/2005.03851.

[51] Simon Schaal, Alessandro Rossi, Virginia N. Ciriano-Tejel, Tsung-Yeh Yang, Sylvain Barraud, John J. L. Morton, and M. Fernando Gonzalez-Zalba. A cmos dynamic random access architecture for radio-frequency readout of quantum devices. *Nature Electronics*, 2 (6):236–242, June 2019. ISSN 2520-1131. doi: 10.1038/s41928-019-0259-5. URL http://dx.doi.org/10.1038/s41928-019-0259-5.

[52] Hannes Watzinger, Josip Kukučka, Lada Vukušić, Fei Gao, Ting Wang, Friedrich Schäffler, Jian-Jun Zhang, and Georgios Katsaros. A germanium hole spin qubit. *Nat Commun*, 9 (1):3902, December 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-06418-4. URL http://www.nature.com/articles/s41467-018-06418-4.

[53] Ke Wang, Gang Xu, Fei Gao, He Liu, Rong-Long Ma, Xin Zhang, Zhanning Wang, Gang Cao, Ting Wang, Jian-Jun Zhang, Dimitrie Culcer, Xuedong Hu, Hong-Wen Jiang, Hai-Ou Li, Guang-Can Guo, and Guo-Ping Guo. Ultrafast coherent control of a hole spin qubit in a germanium quantum dot. *Nat Commun*, 13(1):206, December 2022. ISSN 2041-1723. doi: 10.1038/s41467-021-27880-7. URL https://www.nature.com/articles/s41467-021-27880-7.

[54] He Liu, Ke Wang, Fei Gao, Jin Leng, Yang Liu, Yu-Chen Zhou, Gang Cao, Ting Wang, Jianjun Zhang, Peihao Huang, Hai-Ou Li, and Guo-Ping Guo. Ultrafast and electrically tunable rabi frequency in a germanium hut wire hole spin qubit. *Nano Letters*, 23(9):3810–3817, April 2023. ISSN 1530-6992. doi: 10.1021/acs.nanolett.3c00213. URL http://dx.doi.org/10.1021/acs.nanolett.3c00213.

[55] Francesco Borsoi, Nico W. Hendrickx, Valentin John, Marcel Meyer, Sayr Motz, Floor van Riggelen, Amir Sammak, Sander L. de Snoo, Giordano Scappucci, and Menno Veldhorst. Shared control of a 16 semiconductor quantum dot crossbar array. *Nature Nanotechnology 2023*, pages 1–7, 8 2023. ISSN 1748-3395. doi: 10.1038/s41565-023-01491-3. URL https://www.nature.com/articles/s41565-023-01491-3.

[56] Daniel Loss, David P DiVincenzo, and P DiVincenzo. Quantum computation with quantum dots. *Phys.\ Rev.˜A*, 57(1):120–126, 1997. URL https://journals.aps.org/pra/pdf/10.1103/PhysRevA.57.120.

[57] Tuomo Tanttu, Bas Hensen, Kok Wai Chan, Chih Hwan Yang, Wister Wei Huang, Michael Fogarty, Fay Hudson, Kohei Itoh, Dimitrie Culcer, Arne Laucht, Andrea Morello, and Andrew Dzurak. Controlling spin-orbit interactions in silicon quantum dots using magnetic field direction. *Physical Review X*, 9(2), May 2019. ISSN 2160-3308. doi: 10.1103/physrevx.9.021028. URL http://dx.doi.org/10.1103/PhysRevX.9.021028.

[58] N. W. Hendrickx, L. Massai, M. Mergenthaler, F. Schupp, S. Paredes, S. W. Bedell, G. Salis, and A. Fuhrer. Sweet-spot operation of a germanium hole spin qubit with highly anisotropic noise sensitivity, 2023.

[59] N. Ares, G. Katsaros, V. N. Golovach, J. J. Zhang, A. Prager, L. I. Glazman, O. G. Schmidt, and S. De Franceschi. Sige quantum dots for fast hole spin rabi oscillations. *Applied Physics Letters*, 103(26), December 2013. ISSN 1077-3118. doi: 10.1063/1.4858959. URL http://dx.doi.org/10.1063/1.4858959.

[60] Alessandro Crippa, Romain Maurand, Léo Bourdet, Dharmraj Kotekar-Patil, Anthony Amisse, Xavier Jehl, Marc Sanquer, Romain Laviéville, Heorhii Bohuslavskyi, Louis Hutin, Sylvain Barraud, Maud Vinet, Yann-Michel Niquet, and Silvano De Franceschi. Electrical

spin driving by g-matrix modulation in spin-orbit qubits. *Phys. Rev. Lett.*, 120:137702, Mar 2018. doi: 10.1103/PhysRevLett.120.137702. URL https://link.aps.org/doi/10.1103/PhysRevLett.120.137702.

[61] Benjamin Venitucci, Léo Bourdet, Daniel Pouzada, and Yann-Michel Niquet. Electrical manipulation of semiconductor spin qubits within the g-matrix formalism. *Phys. Rev. B*, 98: 155319, Oct 2018. doi: 10.1103/PhysRevB.98.155319. URL https://link.aps.org/doi/10.1103/PhysRevB.98.155319.

[62] Amanda E. Seedhouse, Tuomo Tanttu, Ross C.C. Leon, Ruichen Zhao, Kuan Yen Tan, Bas Hensen, Fay E. Hudson, Kohei M. Itoh, Jun Yoneda, Chih Hwan Yang, Andrea Morello, Arne Laucht, Susan N. Coppersmith, Andre Saraiva, and Andrew S. Dzurak. Pauli blockade in silicon quantum dots with spin-orbit control. *PRX Quantum*, 2:010303, Jan 2021. doi: 10.1103/PRXQuantum.2.010303. URL https://link.aps.org/doi/10.1103/PRXQuantum.2.010303.

[63] R. Hanson, L. P. Kouwenhoven, J. R. Petta, S. Tarucha, and L. M. K. Vandersypen. Spins in few-electron quantum dots. *Rev. Mod. Phys.*, 79:1217–1265, Oct 2007. doi: 10.1103/RevModPhys.79.1217. URL https://link.aps.org/doi/10.1103/RevModPhys.79.1217.

[64] Leo P. Kouwenhoven, Charles M. Marcus, Paul L. McEuen, Seigo Tarucha, Robert M. Westervelt, and Ned S. Wingreen. *Electron Transport in Quantum Dots*, pages 105–214. Springer Netherlands, Dordrecht, 1997. ISBN 978-94-015-8839-3. doi: 10.1007/978-94-015-8839-3_4. URL https://doi.org/10.1007/978-94-015-8839-3_4.

[65] Florian Vigneau, Federico Fedele, Anasua Chatterjee, David Reilly, Ferdinand Kuemmeth, Fernando Gonzalez-Zalba, Edward Laird, and Natalia Ares. Probing quantum devices with radio-frequency reflectometry. *arXiv:2202.10516*, 2022. URL https://arxiv.org/abs/2202.10516.

[66] K. D. Petersson, C. G. Smith, D. Anderson, P. Atkinson, G. A. C. Jones, and D. A. Ritchie. Charge and spin state readout of a double quantum dot coupled to a resonator. *Nano Letters*, 10(8):27892793, July 2010. ISSN 1530-6992. doi: 10.1021/nl100663w. URL http://dx.doi.org/10.1021/nl100663w.

[67] K. D. Petersson, C. G. Smith, D. Anderson, P. Atkinson, G. A. C. Jones, and D. A. Ritchie. Charge and Spin State Readout of a Double Quantum Dot Coupled to a Resonator. *Nano Lett.*, 10(8):2789–2793, August 2010. ISSN 1530-6984, 1530-6992. doi: 10.1021/nl100663w. URL https://pubs.acs.org/doi/10.1021/nl100663w.

[68] Mark R Hogg, Prasanna Pakkiam, Samuel K Gorman, Andrey V Timofeev, Yousun Chung, Gurpreet K Gulati, Matthew G House, and Michelle Y Simmons. Single-shot readout of multiple donor electron spins with a gate-based sensor. *arXiv:2203.09248*, 2022. URL https://arxiv.org/abs/2203.09248.

[69] Anderson West, Bas Hensen, Alexis Jouan, Tuomo Tanttu, Chih Hwan Yang, Alessandro Rossi, M. Fernando Gonzalez-Zalba, Fay Hudson, Andrea Morello, David J. Reilly, and An-

drew S. Dzurak. Gate-based single-shot readout of spins in silicon. *Nature Nanotechnology*, 14(5):437–441, 2019. ISSN 17483395.

[70] Zhenyu Cai, Michael A Fogarty, Simon Schaal, Sofia Patomäki, Simon C Benjamin, and John JL Morton. A silicon surface code architecture resilient against leakage errors. *Quantum*, 3:212, 2019. URL https://quantum-journal.org/papers/q-2019-12-09-212/.

[71] Ruoyu Li, Luca Petit, David P. Franke, Juan Pablo Dehollain, Jonas Helsen, Mark Steudtner, Nicole K. Thomas, Zachary R. Yoscovits, Kanwal J. Singh, Stephanie Wehner, Lieven M. K. Vandersypen, James S. Clarke, and Menno Veldhorst. A crossbar network for silicon quantum dot qubits. *Science Advances*, 4(7):3960, 2018. doi: 10.1126/sciadv.aar3960. URL https://www.science.org/doi/abs/10.1126/sciadv.aar3960.

[72] M. Veldhorst, H. G. J. Eenink, C. H. Yang, and A. S. Dzurak. Silicon CMOS architecture for a spin-based quantum computer. *Nat Commun*, 8(1):1766, December 2017. ISSN 2041-1723. doi: 10.1038/s41467-017-01905-6. URL http://www.nature.com/articles/s41467-017-01905-6.

[73] M. F. Gonzalez-Zalba, S. de Franceschi, E. Charbon, T. Meunier, M. Vinet, and A. S. Dzurak. Scaling silicon-based quantum computing using CMOS technology. *Nat Electron*, 4(12):872–884, December 2021. ISSN 2520-1131. doi: 10.1038/s41928-021-00681-y. URL https://www.nature.com/articles/s41928-021-00681-y.

[74] R. J. Schoelkopf, P. Wahlgren, A. A. Kozhevnikov, P. Delsing, and D. E. Prober. The radio-frequency single-electron transistor (RF-SET): A fast and ultrasensitive electrometer. *Science*, 280(5367):1238–1242, 1998. ISSN 00368075. doi: 10.1126/science.280.5367.1238.

[75] C Barthel, M Kjærgaard, J Medford, M Stopa, C M Marcus, M P Hanson, and A C Gossard. Fast sensing of double-dot charge arrangement and spin state with a radio-frequency sensor quantum dot. *Physical Review B*, 81(16):161308, 4 2010.

[76] N. Ares, F. J. Schupp, A. Mavalankar, G. Rogers, J. Griffiths, G. A.C. Jones, I. Farrer, D. A. Ritchie, C. G. Smith, A. Cottet, G. A.D. Briggs, and E. A. Laird. Sensitive Radio-Frequency Measurements of a Quantum Dot by Tuning to Perfect Impedance Matching. *Physical Review Applied*, 5(3):1–6, 2016. ISSN 23317019. doi: 10.1103/PhysRevApplied.5.034011.

[77] T Müller, B Küng, S Hellmüller, P Studerus, K Ensslin, T Ihn, M Reinwald, and W Wegscheider. An in situ tunable radio-frequency quantum point contact. *Applied Physics Letters*, 97 (20):202104, 11 2010. ISSN 0003-6951.

[78] Akito Noiri, Kenta Takeda, Jun Yoneda, Takashi Nakajima, Tetsuo Kodera, and Seigo Tarucha. Radio-Frequency-Detected Fast Charge Sensing in Undoped Silicon Quantum Dots. *Nano Letters*, 20(2):947–952, 2 2020. ISSN 15306992.

[79] 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. *Applied Physics Reviews*, 10(2):021305, 02 2023. ISSN 1931-9401. doi: 10.1063/5.0088229.

[80] M.R. Hogg, P. Pakkiam, S.K. Gorman, A.V. Timofeev, Y. Chung, G.K. Gulati, M.G. House, and M.Y. Simmons. Single-shot readout of multiple donor electron spins with a gate-based sensor. *PRX Quantum*, 4:010319, Feb 2023.

[81] D. J. Reilly, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Fast single-charge sensing with a rf quantum point contact. *Applied Physics Letters*, 91(16), October 2007. ISSN 1077-3118. doi: 10.1063/1.2794995. URL `http://dx.doi.org/10.1063/1.2794995`.

[82] T.-K. Hsiao, C.J. van Diepen, U. Mukhopadhyay, C. Reichl, W. Wegscheider, and L.M.K. Vandersypen. Efficient orthogonal control of tunnel couplings in a quantum dot array. *Physical Review Applied*, 13(5), May 2020. ISSN 2331-7019. doi: 10.1103/physrevapplied.13.054018. URL `http://dx.doi.org/10.1103/PhysRevApplied.13.054018`.

[83] C. H. Yang, W. H. Lim, F. A. Zwanenburg, and A. S. Dzurak. Dynamically controlled charge sensing of a few-electron silicon quantum dot. *AIP Advances*, 1(4), 2011. ISSN 21583226.

[84] Takashi Nakajima, Yohei Kojima, Yoshihiro Uehara, Akito Noiri, Kenta Takeda, Takashi Kobayashi, and Seigo Tarucha. Real-Time Feedback Control of Charge Sensing for Quantum Dot Qubits. *Physical Review Applied*, 15(3):1, 2021. ISSN 23317019.

[85] Andreas Fuhrer. *Phase coherence, orbital and spin states in quantum rings*. Doctoral thesis, ETH Zurich, ZÃijrich, 2003. Diss., Naturwissenschaften ETH ZÃijrich, Nr. 15094, 2003.

[86] W. G. van der Wiel, S. De Franceschi, J. M. Elzerman, T. Fujisawa, S. Tarucha, and L. P. Kouwenhoven. Electron transport through double quantum dots. *Rev. Mod. Phys.*, 75:1–22, Dec 2002. doi: 10.1103/RevModPhys.75.1. URL `https://link.aps.org/doi/10.1103/RevModPhys.75.1`.

[87] Justyna P. Zwolak, Sandesh S. Kalantre, Xingyao Wu, Stephen Ragole, and Jacob M. Taylor. Qflow lite dataset: A machine-learning approach to the charge states in quantum dot experiments. *PLOS ONE*, 13(10):1–17, 10 2018. doi: 10.1371/journal.pone.0205844. URL `https://doi.org/10.1371/journal.pone.0205844`.

[88] D. Schröer, A. D. Greentree, L. Gaudreau, K. Eberl, L. C. L. Hollenberg, J. P. Kotthaus, and S. Ludwig. Electrostatically defined serial triple quantum dot charged with few electrons. *Phys. Rev. B*, 76:075306, Aug 2007. doi: 10.1103/PhysRevB.76.075306. URL `https://link.aps.org/doi/10.1103/PhysRevB.76.075306`.

[89] H Van Houten, CWJ Beenakker, and AAM Staring. Coulomb-blockade oscillations in semiconductor nanostructures. *arXiv preprint cond-mat/0508454*, 2005.

[90] Giovanni A. Oakes, Jingyu Duan, John J. L. Morton, Alpha Lee, Charles G. Smith, and M. Fernando Gonzalez Zalba. Automatic virtual voltage extraction of a 2x2 array of quantum dots with machine learning, 2021.

[91] Federico Fedele. Spin interactions within a two-dimensional array of gaas double dots.

[92] Hanno Flentje, Benoit Bertrand, Pierre-AndrÃľ Mortemousque, Vivien Thiney, Arne Ludwig, Andreas D. Wieck, Christopher BÃďuerle, and Tristan Meunier. A linear triple quantum dot system in isolated configuration. *Applied Physics Letters*, 110(23):233101, 06 2017. ISSN 0003-6951. doi: 10.1063/1.4984745. URL https://doi.org/10.1063/1.4984745.

[93] Benoit Bertrand, Hanno Flentje, Shintaro Takada, Michihisa Yamamoto, Seigo Tarucha, Arne Ludwig, Andreas D. Wieck, Christopher Bäuerle, and Tristan Meunier. Quantum manipulation of two-electron spin states in isolated double quantum dots. *Phys. Rev. Lett.*, 115:096801, Aug 2015. doi: 10.1103/PhysRevLett.115.096801. URL https://link.aps.org/doi/10.1103/PhysRevLett.115.096801.

[94] H. Moon, D. T. Lennon, J. Kirkpatrick, N. M. van Esbroeck, L. C. Camenzind, Liuqi Yu, F. Vigneau, D. M. Zumbühl, G. A.D. D. Briggs, M. A. Osborne, D. Sejdinovic, E. A. Laird, and N. Ares. Machine learning enables completely automatic tuning of a quantum device faster than human experts. *Nature Communications*, 11(1):1–18, 12 2020. ISSN 20411723. doi: 10.1038/s41467-020-17835-9. URL https://doi.org/10.1038/s41467-020-17835-9.

[95] J Darulová, S J Pauka, N Wiebe, K W Chan, G C Gardener, M J Manfra, M C Cassidy, and M Troyer. Autonomous Tuning and Charge-State Detection of Gate-Defined Quantum Dots. *Physical Review Applied*, 13(5):54005, 5 2020. doi: 10.1103/PhysRevApplied.13.054005. URL https://link.aps.org/doi/10.1103/PhysRevApplied.13.054005.

[96] 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. *Applied Physics Letters*, 121(8), August 2022. ISSN 1077-3118. doi: 10.1063/5.0111128. URL http://dx.doi.org/10.1063/5.0111128.

[97] Sandesh S Kalantre, Justyna P Zwolak, Stephen Ragole, Xingyao Wu, Neil M Zimmerman, Michael D Stewart, and Jacob M Taylor. Machine learning techniques for state recognition and auto-tuning in quantum dots. *npj Quantum Information*, 5(1):1–10, 2019. URL https://www.nature.com/articles/s41534-018-0118-7.

[98] Justyna P. Zwolak, Thomas McJunkin, Sandesh S. Kalantre, J.P. Dodson, E.R. MacQuarrie, D.E. Savage, M.G. Lagally, S.N. Coppersmith, Mark A. Eriksson, and Jacob M. Taylor. Autotuning of Double-Dot Devices In Situ with Machine Learning. *Physical Review Applied*, 13(3), March 2020. ISSN 2331-7019. URL http://dx.doi.org/10.1103/PhysRevApplied.13.034075.

[99] Joshua Ziegler, Thomas McJunkin, E. S. Joseph, Sandesh S. Kalantre, Benjamin Harpt, D. E. Savage, M. G. Lagally, M. A. Eriksson, Jacob M. Taylor, and Justyna P. Zwolak. Toward Robust Autotuning of Noisy Quantum Dot Devices. *Physical Review Applied*, 17 (2):024069, February 2022. ISSN 2331-7019. doi: 10.1103/PhysRevApplied.17.024069. URL http://arxiv.org/abs/2108.00043.

[100] P. Das, R. Bruyn de Ouboter, and K. W. Taconis. A realization of a london-clarke-mendoza

type refrigerator. In J. G. Daunt, D. O. Edwards, F. J. Milford, and M. Yaqub, editors, *Low Temperature Physics LT9*, pages 1253–1255, Boston, MA, 1965. Springer US. ISBN 978-1-4899-6443-4.

[101] J. Stehlik, Y.-Y. Liu, C. M. Quintana, C. Eichler, T. R. Hartke, and J. R. Petta. Fast charge sensing of a cavity-coupled double quantum dot using a Josephson parametric amplifier. *Physical Review Applied*, 4(1):014018, July 2015. ISSN 2331-7019. doi: 10.1103/PhysRevApplied.4.014018. URL http://arxiv.org/abs/1502.01283.

[102] F. J. Schupp, F. Vigneau, Y. Wen, A. Mavalankar, J. Griffiths, G. A. C. Jones, I. Farrer, D. A. Ritchie, C. G. Smith, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. D. Briggs, N. Ares, and E. A. Laird. Sensitive radiofrequency readout of quantum dots using an ultra-low-noise SQUID amplifier. *Journal of Applied Physics*, 127(24):244503, June 2020. ISSN 0021-8979, 1089-7550. doi: 10.1063/5.0005886. URL http://aip.scitation.org/doi/10.1063/5.0005886.

[103] Barnaby van Straaten, Federico Fedele, Florian Vigneau, Joseph Hickie, and Natalia Ares. All rf-based tuning algorithm for quantum devices using machine learning, 2021. URL https://arxiv.org/abs/2111.11285.

[104] Simon Rogers and Mark A. Girolami. *A First Course in Machine Learning*. Chapman and Hall / CRC machine learning and pattern recognition series. CRC Press, 2011. ISBN 978-1-43-982414-6.

[105] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press. ISBN 978-0-262-25683-4. doi: 10.7551/mitpress/3206.001.0001. URL https://doi.org/10.7551/mitpress/3206.001.0001.

[106] Florent Leclercq. Bayesian optimization for likelihood-free cosmological inference. *Phys. Rev. D*, 98:063511, Sep 2018. doi: 10.1103/PhysRevD.98.063511. URL https://link.aps.org/doi/10.1103/PhysRevD.98.063511.

[107] B. H. Juang and L. R. Rabiner. Hidden markov models for speech recognition. *Technometrics*, 33:251–272, 1991. ISSN 15372723. doi: 10.1080/00401706.1991.10484833.

[108] Anders Krogh, Michael Brown, I. Saira Mian, Kimmen SjÃűlander, and David Haussler. Hidden markov models in computational biology: Applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531, 2 1994. ISSN 0022-2836. doi: 10.1006/JMBI.1994.1104.

[109] Sean R Eddy. What is a hidden markov model? *Nature Biotechnology*, 22(10):13151316, October 2004. ISSN 1546-1696. doi: 10.1038/nbt1004-1315. URL http://dx.doi.org/10.1038/nbt1004-1315.

[110] Matthew J Beal, Zoubin Ghahramani, and Carl Edward Rasmussen. The Infinite Hidden Markov Model. *Advances in Neural Information Processing Systems*, 14, 2001. URL http://www.gatsby.ucl.ac.uk.

[111] L. R. Rabiner and B. H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4–16, 1986. ISSN 07407467. doi: 10.1109/MASSP.1986.1165342.

[112] Andrew Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967. URL `httpss://doi.org/10.1109/TIT.1967.1054010`.

[113] Anasua Chatterjee, Paul Stevenson, Silvano De Franceschi, Andrea Morello, Nathalie P. de Leon, and Ferdinand Kuemmeth. Semiconductor qubits in practice. *Nat Rev Phys*, 3 (3):157–177, March 2021. ISSN 2522-5820. doi: 10.1038/s42254-021-00283-9. URL `http://www.nature.com/articles/s42254-021-00283-9`.

[114] Guido Burkard, Thaddeus D Ladd, John M Nichol, Andrew Pan, and Jason R Petta. Semiconductor spin qubits. *arXiv:2112.08863*, 2021. URL `https://arxiv.org/abs/2112.08863`.

[115] N Ares, FJ Schupp, A Mavalankar, G Rogers, J Griffiths, GAC Jones, I Farrer, DA Ritchie, CG Smith, A Cottet, et al. Sensitive radio-frequency measurements of a quantum dot by tuning to perfect impedance matching. *Physical Review Applied*, 5(3):034011, 2016. URL `https://arxiv.org/abs/1510.06944`.

[116] H. Moon, D. T. Lennon, J. Kirkpatrick, N. M. van Esbroeck, L. C. Camenzind, Liuqi Yu, F. Vigneau, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, D. Sejdinovic, E. A. Laird, and N. Ares. Machine learning enables completely automatic tuning of a quantum device faster than human experts. *Nature Communications*, 11(1):4161, December 2020. ISSN 2041-1723. doi: 10.1038/s41467-020-17835-9. URL `http://www.nature.com/articles/s41467-020-17835-9`.

[117] B Severin, Dominic T Lennon, Leon C Camenzind, Florian Vigneau, F Fedele, D Jirovec, A Ballabio, D Chrastina, G Isella, M de Kruijf, et al. Cross-architecture tuning of silicon and sige-based quantum devices using machine learning. *arXiv:2107.12975*, 2021. URL `https://arxiv.org/abs/2107.12975`.

[118] Alaa Tharwat. Independent component analysis: An introduction. 17(2):222–249. ISSN 2634-1964, 2210-8327. doi: 10.1016/j.aci.2018.08.006. URL `https://doi.org/10.1016/j.aci.2018.08.006`. Publisher: Emerald Publishing Limited.

[119] J. L. Hodges. The significance probability of the smirnov two-sample test. *Ark. Mat.*, 3 (5):469–486, January 1958. ISSN 0004-2080. doi: 10.1007/BF02589501. URL `http://projecteuclid.org/euclid.afm/1485893310`.

[120] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian

Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[121] A. R. Mills, M. M. Feldman, C. Monical, P. J. Lewis, K. W. Larson, A. M. Mounce, and J. R. Petta. Computer-automated tuning procedures for semiconductor quantum dot arrays. *Applied Physics Letters*, 115(11):113501, 2019. doi: 10.1063/1.5121444. URL `https://doi.org/10.1063/1.5121444`.

[122] Maxime Lapointe-Major, Olivier Germain, J Camirand Lemyre, Dany Lachance-Quirion, Sophie Rochette, F Camirand Lemyre, and Michel Pioro-Ladrière. Algorithm for automated tuning of a quantum dot into the single-electron regime. *Physical Review B*, 102(8):085301, 2020. URL `https://journals.aps.org/prb/abstract/10.1103/PhysRevB.102.085301`.

[123] T. A. Baart, P. T. Eendebak, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen. Computer-automated tuning of semiconductor double quantum dots into the single-electron regime. *Applied Physics Letters*, 108(21):213104, May 2016. ISSN 0003-6951, 1077-3118. doi: 10.1063/1.4952624. URL `http://aip.scitation.org/doi/10.1063/1.4952624`.

[124] Justyna P. Zwolak, Thomas McJunkin, Sandesh S. Kalantre, Samuel F. Neyens, E.R. MacQuarrie, Mark A. Eriksson, and Jacob M. Taylor. Ray-based framework for state identification in quantum dot devices. *PRX Quantum*, 2(2), June 2021. URL `https://journals.aps.org/prxquantum/abstract/10.1103/PRXQuantum.2.020335`.

[125] David L Craig, H Moon, Federico Fedele, Dominic T Lennon, B Van Straaten, Florian Vigneau, Leon C Camenzind, Dominik M Zumbühl, G Andrew D Briggs, Michael A Osborne, et al. Bridging the reality gap in quantum devices with physics-aware machine learning. *arXiv:2111.11285*, 2021. URL `https://arxiv.org/abs/2111.11285`.

[126] 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. *arXiv:2108.10656*, 2021. URL `https://arxiv.org/abs/2108.10656`.

[127] Justyna P. Zwolak, Sandesh S. Kalantre, Thomas McJunkin, Brian Weber, and Jacob M. Taylor. Ray-based classification framework for high-dimensional data. *ArXiv*, abs/2010.00500, 2020. URL `https://api.semanticscholar.org/CorpusID:222090994`.

[128] J. M. Hornibrook, J. I. Colless, A. C. Mahoney, X. G. Croot, S. Blanvillain, H. Lu, A. C. Gossard, and D. J. Reilly. Frequency multiplexing for readout of spin qubits. *Appl. Phys. Lett.*, 104(10):103108, March 2014. ISSN 0003-6951, 1077-3118. doi: 10.1063/1.4868107. URL `http://aip.scitation.org/doi/10.1063/1.4868107`.

[129] T. Hensgens, T. Fujita, L. Janssen, Xiao Li, C. J. Van Diepen, C. Reichl, W. Wegscheider, S. Das Sarma, and L. M.K. Vandersypen. Quantum simulation of a Fermi-Hubbard model using a semiconductor quantum dot array. *Nature*, 548(7665):70–73, 2017. ISSN 14764687.

[130] Tim Botzem, Michael D. Shulman, Sandra Foletti, Shannon P. Harvey, Oliver E. Dial, Patrick Bethke, Pascal Cerfontaine, Robert P.G. McNeil, Diana Mahalu, Vladimir Umansky, Arne Ludwig, Andreas Wieck, Dieter Schuh, Dominique Bougeard, Amir Yacoby, and Hendrik Bluhm. Tuning Methods for Semiconductor Spin Qubits. *Physical Review Applied*, 10(5):1, 2018. ISSN 23317019.

[131] A. R. Mills, D. M. Zajac, M. J. Gullans, F. J. Schupp, T. M. Hazard, and J. R. Petta. Shuttling a single charge across a one-dimensional array of silicon quantum dots. *Nature Communications*, 10(1), 2019. ISSN 20411723.

[132] Daniel Jirovec, Andrea Hofmann, Andrea Ballabio, Philipp M. Mutter, Giulio Tavani, Marc Botifoll, Alessandro Crippa, Josip Kukucka, Oliver Sagi, Frederico Martins, Jaime Saez-Mollejo, Ivan Prieto, Maksim Borovkov, Jordi Arbiol, Daniel Chrastina, Giovanni Isella, and Georgios Katsaros. A singlet-triplet hole spin qubit in planar Ge. *Nature Materials*, 20(8): 1106–1112, 2021. ISSN 14764660.

[133] URL `https://qtt.readthedocs.io/en/latest/`.

[134] Nicholas D Matsakis and Felix S Klock II. The rust language. In *ACM SIGAda Ada Letters*, volume 34, pages 103–104. ACM, 2014.

[135] Manuel Costanzo, Enzo Rucci, Marcelo Naiouf, and Armando De Giusti. Performance vs programming effort between rust and c on multicore architectures: Case study in n-body. 2021.

[136] J. Stehlik, Y.-Y. Liu, C. M. Quintana, C. Eichler, T. R. Hartke, and J. R. Petta. Fast charge sensing of a cavity-coupled double quantum dot using a josephson parametric amplifier. *Phys. Rev. Appl.*, 4:014018, Jul 2015. doi: 10.1103/PhysRevApplied.4.014018. URL `https://link.aps.org/doi/10.1103/PhysRevApplied.4.014018`.

[137] F. J. Schupp, F. Vigneau, Y. Wen, A. Mavalankar, J. Griffiths, G. A. C. Jones, I. Farrer, D. A. Ritchie, C. G. Smith, L. C. Camenzind, L. Yu, D. M. ZumbAhl, G. A. D. Briggs, N. Ares, and E. A. Laird. Sensitive radiofrequency readout of quantum dots using an ultra-low-noise SQUID amplifier. *Journal of Applied Physics*, 127(24):244503, 06 2020. ISSN 0021-8979. doi: 10.1063/5.0005886. URL `https://doi.org/10.1063/5.0005886`.

[138] 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 tuning algorithm for quantum devices using machine learning, 2022.

[139] Jaehyun Park and Stephen Boyd. A semidefinite programming method for integer convex quadratic minimization. *Optimization Letters*, 12(3):499–518, mar 2017. doi: 10.1007/s11590-017-1132-y. URL `https://doi.org/10.1007%2Fs11590-017-1132-y`.

[140] Christoph Buchheim, Alberto Caprara, and Andrea Lodi. An effective branch-and-bound algorithm for convex quadratic integer programming. volume 135, pages 285–298, 06 2010. ISBN 978-3-642-13035-9. doi: 10.1007/978-3-642-13036-6_22.

[141] Vincent Ng. Rust vs c++, a battle of speed and efficiency. 05 2023. doi: 10.36227/techrxiv.22792553.v1.

[142] Nikolay Ivanov. Is rust c++-fast? benchmarking system languages on everyday routines, 2022.

[143] Rayon-Rs. Rayon-rs/rayon: Rayon: A data parallelism library for rust. URL https://github.com/rayon-rs/rayon.

[144] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: an operator splitting solver for quadratic programs. *Mathematical Programming Computation*, 12(4):637–672, 2020. doi: 10.1007/s12532-020-00179-2. URL https://doi.org/10.1007/s12532-020-00179-2.

[145] M. R. Delbecq, T. Nakajima, T. Otsuka, S. Amaha, J. D. Watson, M. J. Manfra, and S. Tarucha. Full control of quadruple quantum dot circuit charge states in the single electron regime. *Applied Physics Letters*, 104(18):183111, 05 2014. ISSN 0003-6951. doi: 10.1063/1.4875909. URL https://doi.org/10.1063/1.4875909.

[146] Jose Aumentado, Gianluigi Catelani, and Kyle Serniak. Quasiparticle poisoning in superconducting quantum computers. *Physics Today*, 76(8):34–39, 08 2023. ISSN 0031-9228. doi: 10.1063/PT.3.5291. URL https://doi.org/10.1063/PT.3.5291.

[147] L. M. K. Vandersypen, H. Bluhm, J. S. Clarke, A. S. Dzurak, R. Ishihara, A. Morello, D. J. Reilly, L. R. Schreiber, and M. Veldhorst. Interfacing spin qubits in quantum dots and donorsâĂŤhot, dense, and coherent. *npj Quantum Information*, 3(1):1–10, 12 2017. ISSN 2056-6387. doi: 10.1038/s41534-017-0038-y.

[148] L. Petit, J. M. Boter, H. G.J. Eenink, G. Droulers, M. L.V. Tagliaferri, R. Li, D. P. Franke, K. J. Singh, J. S. Clarke, R. N. Schouten, V. V. Dobrovitski, L. M.K. Vandersypen, and M. Veldhorst. Spin Lifetime and Charge Noise in Hot Silicon Quantum Dot Qubits. *Physical Review Letters*, 121(7):076801, 8 2018. ISSN 10797114. doi: 10.1103/PHYSREVLETT.121.076801/FIGURES/4/MEDIUM. URL https://journals.aps.org/prl/abstract/10.1103/PhysRevLett.121.076801.

[149] L. Petit, H. G.J. Eenink, M. Russ, W. I.L. Lawrie, N. W. Hendrickx, S. G.J. Philips, J. S. Clarke, L. M.K. Vandersypen, and M. Veldhorst. Universal quantum logic in hot silicon qubits. *Nature 2020 580:7803*, 580(7803):355–359, 4 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-2170-7. URL https://www.nature.com/articles/s41586-020-2170-7.

[150] Luca Petit, Maximilian Russ, Gertjan H.G.J. Eenink, William I.L. Lawrie, James S. Clarke, Lieven M.K. Vandersypen, and Menno Veldhorst. Design and integration of single-qubit rotations and two-qubit gates in silicon above one Kelvin. *Communications Materials 2022 3:1*, 3(1):1–7, 11 2022. ISSN 2662-4443. doi: 10.1038/s43246-022-00304-9. URL https://www.nature.com/articles/s43246-022-00304-9.

[151] Leon C. Camenzind, Simon Geyer, Andreas Fuhrer, Richard J. Warburton, Dominik M. Zumbühl, and Andreas V. Kuhlmann. A hole spin qubit in a fin field-effect transistor above

4 kelvin. *Nature Electronics 2022 5:3*, 5(3):178–183, 3 2022. ISSN 2520-1131. doi: 10.1038/s41928-022-00722-0. URL https://www.nature.com/articles/s41928-022-00722-0.

[152] Leonard E. Baum and Ted Petrie. Statistical Inference for Probabilistic Functions of Finite State Markov Chains. *https://doi.org/10.1214/aoms/1177699147*, 37(6):1554–1563, 12 1966. ISSN 0003-4851. doi: 10.1214/AOMS/1177699147. URL https://projecteuclid.org/journals/annals-of-mathematical-statistics/volume-37/issue-6/Statistical-Inference-for-Probabilistic-Functions-of-Finite-State-Markov-Chains/10.1214/aoms/1177699147.fullhttps://projecteuclid.org/journals/annals-of-mathematical-s.

[153] Jonathan Y. Huang, Rocky Y. Su, Wee Han Lim, MengKe Feng, Barnaby van Straaten, Brandon Severin, Will Gilbert, Nard Dumoulin Stuyck, Tuomo Tanttu, Santiago Serrano, Jesus D. Cifuentes, Ingvild Hansen, Amanda E. Seedhouse, Ensar Vahapoglu, Nikolay V. Abrosimov, Hans-Joachim Pohl, Michael L. W. Thewalt, Fay E. Hudson, Christopher C. Escott, Natalia Ares, Stephen D. Bartlett, Andrea Morello, Andre Saraiva, Arne Laucht, Andrew S. Dzurak, and Chih Hwan Yang. High-fidelity operation and algorithmic initialisation of spin qubits above one kelvin. 2023. URL https://arxiv.org/abs/2308.02111v2.

[154] Gael Varoquaux, Sergei Lebedev, Antony Lee, Chris Farrow, Matthew Danielson, and Anmol. hmmlearn, 2023. URL https://github.com/hmmlearn/hmmlearn.

[155] Harald Cramér. *Mathematical methods of statistics*. 1946.

[156] B. D'Anjou and W. A. Coish. Soft decoding of a qubit readout apparatus. *Phys. Rev. Lett.*, 113:230402, Dec 2014. doi: 10.1103/PhysRevLett.113.230402. URL https://link.aps.org/doi/10.1103/PhysRevLett.113.230402.

[157] J. M. Hornibrook, J. I. Colless, A. C. Mahoney, X. G. Croot, S. Blanvillain, H. Lu, A. C. Gossard, and D. J. Reilly. Frequency multiplexing for readout of spin qubits. *Applied Physics Letters*, 104(10), 3 2014. ISSN 00036951. doi: 10.1063/1.4868107/130884. URL /aip/apl/article/104/10/103108/130884/Frequency-multiplexing-for-readout-of-spin-qubits.

[158] J. M. Hornibrook, J. I. Colless, I. D. Conway Lamb, S. J. Pauka, H. Lu, A. C. Gossard, J. D. Watson, G. C. Gardner, S. Fallahi, M. J. Manfra, and D. J. Reilly. Cryogenic control architecture for large-scale quantum computing. *Physical Review Applied*, 3(2):024010, 2 2015. ISSN 23317019. doi: 10.1103/PHYSREVAPPLIED.3.024010/FIGURES/6/MEDIUM. URL https://journals.aps.org/prapplied/abstract/10.1103/PhysRevApplied.3.024010.

[159] I. D. Conway Lamb, J. I. Colless, J. M. Hornibrook, S. J. Pauka, S. J. Waddy, M. K. Frechtling, and D. J. Reilly. A FPGA-based instrumentation platform for use at deep cryogenic temperatures. *Review of Scientific Instruments*, 87(1):14701, 1 2016. ISSN 10897623. doi: 10.1063/1.4939094/367382. URL /aip/rsi/article/87/1/014701/367382/An-FPGA-based-instrumentation-platform-for-use-at.

[160] Jeroen Petrus Gerardus Van Dijk, Bishnu Patra, Sushil Subramanian, Xiao Xue, Nodar

Samkharadze, Andrea Corna, Charles Jeon, Farhana Sheikh, Esdras Juarez-Hernandez, Brando Perez Esparza, Huzaifa Rampurawala, Brent R. Carlton, Surej Ravikumar, Carlos Nieva, Sungwon Kim, Hyung Jin Lee, Amir Sammak, Giordano Scappucci, Menno Veldhorst, Lieven M.K. Vandersypen, Edoardo Charbon, Stefano Pellerano, Masoud Babaie, and Fabio Sebastiano. A Scalable Cryo-CMOS Controller for the Wideband Frequency-Multiplexed Control of Spin Qubits and Transmons. *IEEE Journal of Solid-State Circuits*, 55(11):2930–2946, 11 2020. ISSN 1558173X. doi: 10.1109/JSSC.2020.3024678.

[161] Andrea Ruffino, Tsung Yeh Yang, John Michniewicz, Yatao Peng, Edoardo Charbon, and Miguel Fernando Gonzalez-Zalba. A cryo-CMOS chip that integrates silicon quantum dots and multiplexed dispersive readout electronics. *Nature Electronics 2021 5:1*, 5(1):53–59, 12 2021. ISSN 2520-1131. doi: 10.1038/s41928-021-00687-6. URL `https://www.nature.com/articles/s41928-021-00687-6`.

[162] Yatao Peng, Andrea Ruffino, Tsung Yeh Yang, John Michniewicz, Miguel Fernando Gonzalez-Zalba, and Edoardo Charbon. A Cryo-CMOS Wideband Quadrature Receiver With Frequency Synthesizer for Scalable Multiplexed Readout of Silicon Spin Qubits. *IEEE Journal of Solid-State Circuits*, 57(8):2374–2389, 8 2022. ISSN 1558173X. doi: 10.1109/JSSC.2022.3174605.

[163] T. Kobayashi, T. Nakajima, K. Takeda, A. Noiri, J. Yoneda, and S. Tarucha. Feedback-based active reset of a spin qubit in silicon. *npj Quantum Information 2023 9:1*, 9(1):1–8, 6 2023. ISSN 2056-6387. doi: 10.1038/s41534-023-00719-3. URL `https://www.nature.com/articles/s41534-023-00719-3`.

[164] Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990. ISBN 0521386322.

Appendices

Appendix A

Experimental setup

A.1 Impedance matching

To match the device impedance to the line impedance of 50 Ω, we build it into a matching circuit. The circuit constitutes an LC resonance circuit made from an inductor, L and a capacitor C_d. However, parasitic capacitances to the ground are present in the sample board; we model these as a capacitance C_p in parallel with the device impedance, see Fig. 10. The impedance of this circuit is

$$Z_{\text{load}} = i\omega L + \frac{1}{i\omega C_d} + \frac{R_{\text{dev}}}{1 + i\omega C_p R_{\text{dev}}}, \tag{A.1}$$

where R_{dev} is the device resistance. This simplified model includes just the idealised versions of the circuit elements, removing, for instance, the capacitance and resistance of the inductor and the parallel resistances of the capacitors. The exclusion of these parameters simplifies the analysis but does not significantly impact its conclusion. To simplify further, since $R_{\text{dev}} \gg 1$ and $C_p \gg C_d$, we can remove the term including C_d to return

$$Z_{\text{load}} = i\omega L + \frac{R_{\text{dev}}}{1 + i\omega C_p R_{\text{dev}}}, \tag{A.2}$$

At the resonant frequency, the reactive component of this circuit's impedance vanishes, such that $\mathrm{Im}(Z_{\mathrm{load}}) = 0$, we will derive this frequency. The reactive component of the impedance is

$$\mathrm{Im}(Z_{\mathrm{load}}) = \omega L - \frac{1}{\omega C_d} - \frac{\omega R_{\mathrm{dev}}^2 C_p}{1 + \omega^2 R_{\mathrm{dev}}^2 C_p^2} = 0 \tag{A.3}$$

Rearranging and dividing through by ωL yields

$$1 = \frac{1}{\omega^2 L C_d} - \frac{1}{(L/R_{dev}^2 C_p) + \omega^2 L C_p} \tag{A.4}$$

Now we typically operate in a regime where $L/(R_{dev}^2 C_p) \ll 1$ allowing us the approximate solution of

$$\omega_{\mathrm{res}} \approx \sqrt{\frac{1}{L}\left(\frac{1}{C_d} + \frac{1}{C_p}\right)} := \sqrt{\frac{1}{LC_{||}}}. \tag{A.5}$$

where $C_{||}$ indicates the parallel combination of the decoupling capacitor and the parasitic capacitances. In most cases, we would expect the paralytic capacitance to be much smaller than that of the decoupling capacitor, such that $C_p \ll C_d$, meaning that the parasitic capacitance sets the resonant frequency and

$$\omega_{res} \approx 1/\sqrt{LC_p} \tag{A.6}$$

If we consider small deviations away from this resonant frequency, such that $\omega = \omega_r + \Delta\omega$, the real part of this equation simplifies to

$$\mathrm{Re}(Z_{\mathrm{load}}) = \frac{R_{\mathrm{dev}}}{1 + \omega^2 C_P^2 R_{\mathrm{dev}}^2} \tag{A.7}$$

$$\approx \frac{R_{\mathrm{dev}}}{\omega^2 C_P^2 R_{\mathrm{dev}}^2} \tag{A.8}$$

$$= \frac{L}{R_{\mathrm{dev}} C_P}\left(1 - 2\frac{\Delta\omega}{\omega_r}\right) \tag{A.9}$$

where we have used the $\omega \approx \omega_r$, which when combined with A.6 allows us to deduce that $\omega^2 C_P^2 R_{\text{dev}}^2 \gg 1$. By similar analysis, the imaginary part of Eq. A.2 becomes

$$\begin{aligned}
\text{Im}(Z_{\text{load}}) &= \omega L - \frac{1}{\omega C_d} - \frac{\omega C_p R_{\text{dev}}^2}{1 + \omega^2 C_p^2 R_{\text{dev}}^2} && \text{(A.10)} \\
&\approx \omega L - \frac{\omega C_p R_{\text{dev}}^2}{\omega^2 C_P^2 R_{\text{dev}}^2} && \text{(A.11)} \\
&= \omega L - \frac{1}{\omega}\left(\frac{1}{C_P} - \frac{1}{C_d}\right) && \text{(A.12)} \\
&= L(\omega_r + \Delta\omega) - \frac{1}{C_{||}(\omega_r + \Delta\omega)}. && \text{(A.13)}
\end{aligned}$$

With rearrangement and approximation based on $\Delta\omega/\omega_r \ll 1$, we can express

$$\text{Im}(Z_{\text{load}}) = 2\sqrt{\frac{L}{C_{||}}}\frac{\Delta\omega}{\omega_r}. \tag{A.14}$$

Combining the real and imaginary parts, we reach the equation

$$Z_{\text{load}} = R_{\text{eff}}\left(1 - 2\frac{\Delta\omega}{\omega_r}\right) + 2i\sqrt{\frac{L}{C_{||}}}\frac{\Delta\omega}{\omega_r} \quad \text{where} \quad R_{\text{eff}} = \frac{L}{R_{\text{dev}} C_P} \tag{A.15}$$

Therefore, at resonance, the reactive component of the matching circuit vanishes, and the remaining resistive component is given by R_{eff}. We use this formula to match the impedance of the circuit to the line impedance of $Z_0 = 50\,\Omega$. Considering Eq. A.1, we can rearrange to show that the device resistance at which the matching circuit's impedance is $Z_0 = 50\,\Omega$ is

$$R_{\text{matching}} = \frac{L}{C_P Z_0}. \tag{A.16}$$

By using our knowledge of a device's resistance in a given regime of interest, we can chose the matching circuit's components that result in the matching condition being met in the regime of interest.

Changes in the device's capacitance (modelled by C_P) and its resistance (R_{dev}) perturb the load's impedance from the approximate $50\,\Omega$ that is set by the choice of circuit parameters. As a result, we are in the regime where $|d\Gamma/dZ_{\text{load}}| > 0$, and we can obtain information about the device's resistance by measuring the change in reflected signal power and phase.

Appendix B

All rf-based tuning

B.1 The discrete-time Fourier transform's noise characteristics

If $\vec{Z} \in \mathbb{C}^{N\times M}$ is a complex dataset of NM elements and $\vec{N} \in \mathbb{C}^{N\times M}$ is the corresponding gaussian noise, where each element is complex-normally distributed such that $N_{ij} \sim \mathcal{CN}(0, \sigma_N)$ (meaning that the real and imaginary parts of elements of $\vec{N}$ are independently normally distributed with a mean of zero and variance of $\sigma_N^2/2$). Then

$$\mathrm{DTFT}[\vec{Z} + \vec{N}]_{\delta\gamma} \sim \mathcal{CN}\left(\mathrm{DTFT}[\vec{Z}]_{\delta\gamma}, \sigma_N/\sqrt{NM}\right), \tag{B.1}$$

where DTFT[$\cdot$] is the discrete-time Fourier transform. Therefore, the noise is suppressed by a factor of $1/\sqrt{NM}$ in the discrete-time Fourier transform. This can be shown by using the additive nature of the discrete-time Fourier transform. We can write $\mathrm{DTFT}[\vec{Z}+\vec{N}]_{\delta\gamma} = \mathrm{DTFT}[\vec{Z}]_{\delta\gamma} + \mathrm{DTFT}[\vec{N}]_{\delta\gamma}$, where the time-discrete Fourier transform of the noise $\vec{N}$ is

$$\mathrm{DTFT}[\vec{N}]_{\delta\gamma} = \frac{1}{NM}\sum_{a=1}^{N}\sum_{b=1}^{M} e^{-2\pi i(\nu_\delta^x a + \nu_\gamma^y b)} N_{ab}. \tag{B.2}$$

As the complex-normal distribution is circularly symmetric the multiplication by the complex phase factor, $\exp\Big(-2\pi i(\nu_\delta^x a + \nu_\gamma^y b)\Big)$, leaves the distribution unchanged, thus $\exp\Big(-2\pi i(\nu_\delta^x a + \nu_\gamma^y b)\Big)N_{ab} \sim \mathcal{CN}(0, \sigma_N)$, which means that $\mathrm{DTFT}[\vec{N}]_{\delta\gamma}$ can be regarded as the mean of NM samples from a complex-normal distribution. The central limit theorem predicts that this mean will be distributed according to $\mathrm{DTFT}[\vec{N}]_{\delta\gamma} \sim \mathcal{CN}(0, \sigma_N/\sqrt{NM})$. As there is no uncertainty in the Fourier transform of the noiseless signal, $\vec{N}_{\delta\gamma}$ the sum of the Fourier transform of the signal and noise will be distributed according to equation B.1.

B.2 Separated confusion matrices for Devices A and B

This appendix section presents the confusion matrices for the score function applied to scans from devices A and B.

Table 7: Confusion matrix for device A.

Device A (%)	Expert Classification		
	Non-double dot	Imperfect double dot	Satisfactory double dot
score < 0.04	96.96	0.93	0.00
$0.04 \leq$ score < 0.08	0.19	1.43	0.06
score ≥ 0.08	0.00	0.11	0.30

Table 8: Confusion matrix for device B.

Device B (%)	Expert Classification		
	Non-double dot	Imperfect double dot	Satisfactory double dot
score < 0.04	96.93	0.89	0.00
$0.04 \leq$ score < 0.08	0.30	1.41	0.09
score ≥ 0.08	0.01	0.10	0.27

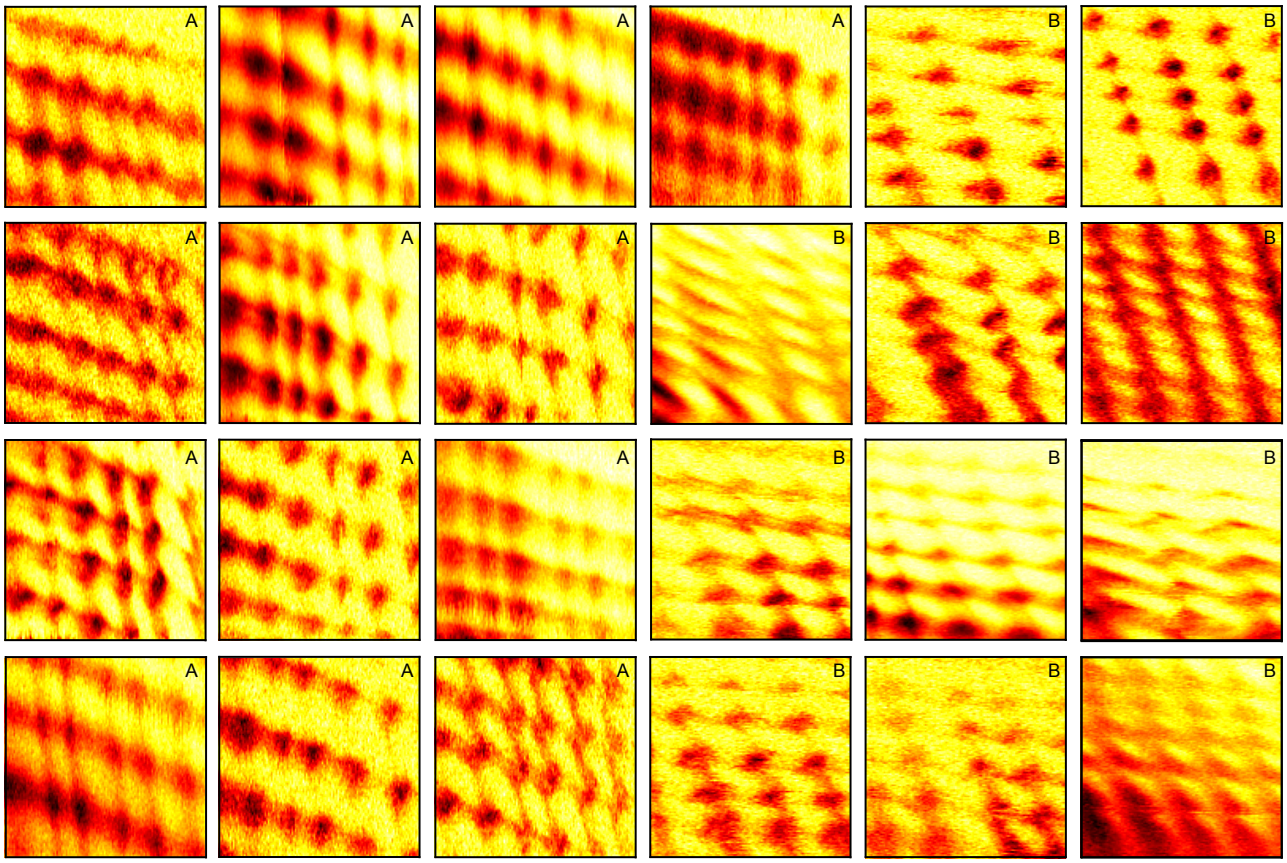


Figure 47: The highest-scored double dot features found by the algorithm by the end of each hour-long tuning session performed on devices A or B, labelled in the top corner. For scans taken in device A (B), the horizontal and vertical axis are gates V_2 (V_4) and V_4 (V_6) respectively, both of these gate voltages are swept over a 100 mV range.

B.3 Outcomes from the hour-long tuning sessions

The highest scoring of the double dot features found by the algorithm in each hour-long tuning session (Fig. B.3).

Appendix C

QArray

C.1 Positive Definite Proof

This section proves that the $\mathbf{c}_{dd}$ is positive definite, making the optimisation problem convex.

Theorem C.1.1. *For any set of capacitances between nodes such that $c_{ij}\,\forall i,j$ are real, non-negative and $c_{ij} = c_{ji}$ the Maxwell matrix $\mathbf{C}^M \in \mathbb{R}^{(n_{dot}+n_{gate})\times(n_{dot}+n_{gate})}$ is defined as*

$$C_{ij}^M = \left(\sum_k c_{ik}\right)\delta_{ij} - c_{ij}(1-\delta_{ij}). \tag{C.1}$$

Let $\mathbf{C} \in \mathbb{R}^{n_{dot}\times n_{dot}}$ be the upper left block of $\mathbf{C}^M$. $\mathbf{C}$ is positive definite assuming the physical case of every dot being coupled to at least one gate, namely $\forall i\,\exists j > n_{dot}$ such that $c_{ij} > 0$.

Proof. A matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is called strictly diagonally dominant if for all n diagonal entries $|A_{ii}| > \sum_{k\neq i}|A_{ik}|$ holds. We start proving that $\mathbf{C}$ is strictly diagonally dominant by observing

$$C_{ii} = \sum_{k=1}^{n_{\mathrm{dot}}+n_{\mathrm{gate}}} c_{ik} > \sum_{k\neq i}^{n_{\mathrm{dot}}} |C_{ik}| = \sum_{k\neq i}^{n_{\mathrm{dot}}} c_{ik}.$$

This holds due to the non-negative elements c_{ij} and can be simplified to the true statement $\sum_{k=n_{dot}+1}^{n_{dot}+n_{gate}} c_{ik} > 0$. Strict diagonal dominance implies positive definiteness for symmetric matrices with non-negative elements by [164, Theorem 6.1.10]. □

C.2 Continumum minima derivations

C.2.1 Open quantum dot arrays

In this section, we derive the continuous minimum charge state for open quantum dot array - the charge state that minimises (2.8) neglecting constraints (i) and (ii). The free energy is

$$F(\vec{V}_g, \vec{Q}_d) := \frac{1}{2}(\mathbf{c}_{dg}\vec{V}_g - \vec{Q}_d)^T \mathbf{c}_{dd}^{-1}(C_{dg}\vec{V}_g - \vec{Q}_d) \tag{C.2}$$

For ease of notation going forward, we will write this as

$$F(\vec{V}, \vec{Q}) = \frac{1}{2}(V - Q)^T C^{-1}(V - Q) \tag{C.3}$$

where $\vec{V} := \mathbf{c}_{gd}\vec{V}_d$. Differentiating L with respect to λ yields

$$\frac{\partial L}{\partial \lambda} = -Q^T\vec{1} + \hat{Q} \stackrel{!}{=} 0 \rightarrow \hat{Q} = Q^T\vec{1}. \tag{C.4}$$

Differentiating F for Q_i gives

$$\frac{\partial F}{\partial Q_i} = -\left[C^{-1}(V - Q))\right]_i \stackrel{!}{=} 0 \longrightarrow Q_i = V_i \tag{C.5}$$

Therefore, the continuous minimum for an open quantum dot array is

$$\vec{Q}^{*\text{cont open}} = \vec{V} \tag{C.6}$$

C.2.2 Closed quantum dot arrays

In this section, we derive the continuous minimum charge state for closed quantum dot arrays - the charge state that minimises (2.8) neglecting constraints (i) and (ii). If the array contains $\hat{Q}$ charges, we can incorporate (iii) through the method of Lagrangian multipliers with the Lagrangian

$$L(\vec{V}_g, \vec{Q}_d, \hat{N}) := \frac{1}{2}(\mathbf{c}_{dg}\vec{V}_g - \vec{Q}_d)^T \mathbf{c}_{dd}^{-1}(C_{dg}\vec{V}_g - \vec{Q}_d) - \lambda\left(\sum_i Q_{di} - \hat{Q}\right). \tag{C.7}$$

For ease of notation going forward, we will write this as

$$L(\vec{V}, \vec{Q}, \hat{Q}) = \frac{1}{2}(V - Q)^T C^{-1}(V - Q) - \lambda\left(Q^T\vec{1} - \hat{Q}\right) \tag{C.8}$$

where $\vec{V} := \mathbf{c}_{gd}\vec{V}_d$, whilst $\vec{1}$ is the one vector with all entries one, and otherwise subscripts have been dropped. Differentiating L with respect to λ yields

$$\frac{\partial L}{\partial \lambda} = -Q^T\vec{1} + \hat{Q} \stackrel{!}{=} 0 \to \hat{Q} = Q^T\vec{1}. \tag{C.9}$$

Differentiating L for Q_i gives

$$\frac{\partial L}{\partial Q_i} = -\left[C^{-1}(V - Q))\right]_i - \lambda \stackrel{!}{=} 0 \longrightarrow Q_i = V_i + \lambda C\vec{1}, \tag{C.10}$$

Plugging (C.10) into (C.9) and solving for λ yields

$$\lambda^* = \frac{\hat{Q} - \vec{1}^T V}{\vec{1}^T C\vec{1}}. \tag{C.11}$$

The Lagrangian in (C.8) can be rewritten as

$$\begin{aligned} L(\vec{V},\vec{Q},\hat{Q}) &= \frac{1}{2}Q^T C^{-1} Q - V^T C^{-1} Q + \frac{1}{2} V^T C^{-1} V - \lambda^* \vec{1}^T Q + \lambda^* \hat{Q} \\ &= \frac{1}{2}\left(V + C\vec{1}\lambda^* - Q\right)^T C^{-1} \left(V + C\vec{1}\lambda^* - Q\right) + \lambda^* \hat{Q} - \lambda^* V^T \vec{1} - \frac{1}{2}\lambda^{*2} \vec{1}^T C \vec{1} \end{aligned} \tag{C.12}$$

by using the matrix generalisation of completing the square. Since the last three terms do not depend on $\vec{Q}$, this new Lagrangian is again quadratic in $\vec{Q}$ with the positive definite matrix C. Therefore, the continuous minimum is

$$\begin{aligned} Q^{*\text{cont min}} &= V + C\vec{1}\lambda^* \\ &= V + \left(\hat{Q} - \vec{1}^T \vec{V}\right) \frac{C\vec{1}}{\vec{1}^T C \vec{1}}. \end{aligned} \tag{C.13}$$

C.3 The thresholding strategy

C.3.1 Optimality criteria for limit cases

This section discusses whether only considering the nearest charge states to the continuous minimum is sufficient. We consider two limiting cases and prove that in these cases, it is sufficient. The optimisation problem set out in equation (2.9) can be written in the form

$$\arg\min_{\vec{x}\in\mathbb{R}^n} \frac{1}{2}\vec{x}^T \mathbf{M} \vec{x} + \vec{b}^T \vec{x} \tag{C.14}$$

$$\text{subject to } x_i \geq 0 \ \forall i \tag{C.15}$$

$$\text{and } x_i \in \mathbb{Z} \ \forall i, \tag{C.16}$$

where $\mathbf{M}$ is positive definite matrix. In the following, we prove that if M were diagonal, it would be sufficient to always round the continuous minimum to the nearest discrete change state.

Theorem C.3.1. *The global minimiser $\vec{x}^*_{discret}$ of the optimisation problem* (C.14),(C.15)*, and* (C.16) *can be obtained by rounding each component of the minimiser $\vec{x}^*_{cont}$ to the continuous optimisation problem consisting of* (C.14) *and* (C.15)*.*

Proof. Since $\mathbf{M}$ is diagonal, the optimisation problem uncouples to n independent one-dimensional optimisation problems of the form $\min_{x_i} 1/2 M_{ii} x_i^2 + b_i x_i$ with the non-negativity constraint $x_i \geq 0$ and the integer constraint $x_i \in \mathbb{Z}$. Now we show that rounding the solution $x^*_{i,cont}$ from the one-dimensional continuous optimisation problem to the closest integer yields the global minimiser $x^*_{i,discret}$. If $x^*_{i,cont} = 0$, the integer constraint is already fulfilled. If $x^*_{i,cont} > 0$, then the parabola is symmetric around $x^*_{i,cont}$. Therefore, rounding to the closest integer minimises the one-dimensional integer-constrained optimisation problem. □

With small perturbations away from this diagonal case, we cannot always round to the nearest charge to reliably find the lowest energy configuration. Nevertheless, we can guarantee that the minimiser is one of the 2^{n_n} surrounding neighbours around the continuous solution as long as the condition number $\kappa(\mathbf{M}) = \lambda_{max}/\lambda_{min}$ is sufficiently low. Intuitively, a low condition number corresponds to an almost spherical symmetric potential.

Theorem C.3.2. *If $\kappa(\mathbf{M}) \leq 1 + 4/n$, then one of the 2^n surrounding neighbours of the continuous minimiser x^*_{cont} is the global minimiser $x^*_{discret}$ of the integer optimisation problem.*

Proof. We proceed by showing that even the closest integer point apart from the surrounding neighbours of the continuous minimiser has a larger objective function value than one of the surrounding neighbours for any matrix $\mathbf{M}$ as long as the condition number is sufficiently low. Let $\vec{a}$ be the vector from $\vec{x}^*_{cont}$ to the closest integer neighbour and $\vec{b}$ the vector from $\vec{x}^*_{cont}$ to the closest integer point which is not a surrounding neighbour of $\vec{x}^*_{cont}$. Without loss of generality, we can assume that $\vec{a}$ and $\vec{b}$ have the form $\vec{a} = [\vec{c}; d]$ and $\vec{a} = [\vec{c}; d+1]$. (The notation means that $\vec{a}$ has the same entries as $\vec{c}$ with an additional appended entry d.)

Let $g(\vec{y}) = \vec{y}^T \mathbf{M}\vec{y}/2$ with $\vec{y} = \vec{x} - \vec{x}^*_{cont}$ such that $||\vec{y}||^2\lambda_{min} \leq g(\vec{y}) \leq ||\vec{y}||^2\lambda_{max}$ holds $\forall \vec{y}$. We want to show that $g(\vec{a}) \leq g(\vec{b})$ which is necessarily true if $||\vec{a}||^2\lambda_{max} \leq ||\vec{b}||^2\lambda_{min}$. This yields

$$\frac{\lambda_{max}}{\lambda_{min}} \leq \frac{||\vec{c}||^2 + (d+1)^2}{||\vec{c}||^2 + d^2}. \tag{C.17}$$

It holds that $||\vec{c}||^2 \leq (n-1)/4$ since $||\vec{a}||^2$ is defined to be the vector to the closest neighbour and therefore $|a_i| \leq 1/2\,\forall i$. By incorporating this bound, we obtain the desired result of $\lambda_{max}/\lambda_{min} \leq 1 + 4/n$. □

We showed that under certain conditions, the discrete minimiser is among the surrounding neighbours. Intuitively, we can do better; namely, if the continuous minimiser is close to an integer in one variable, then the discrete minimiser can be obtained by rounding to the closest integer. The following theorem derives bounds when rounding single coordinates leads to optimal solutions.

Theorem C.3.3. *Let element i of $\vec{x}^*_{cont}$ be in the interval $[k, k+\delta]$ for an integer k. If the condition number κ fulfills*

$$\delta \leq \frac{\sqrt{\kappa^2 - (n-1)(\kappa^2-1)^2} - 1}{\kappa^2 - 1}, \tag{C.18}$$

the i-th element of the discrete minimiser is k.

Proof. We proceed in a similar way as beforehand by showing that no integer point with k at its i-th element has a higher value than the "opposite" point with entry $k+1$. With no loss of generality, let $\vec{a} = [\vec{c}; x]$ and $\vec{b} = [\vec{c}; 1-x]$ be the vectors from the continuous minimiser to two opposite integer points. Plugging this into $||\vec{a}||^2\lambda_{max} \leq ||\vec{b}||^2\lambda_{min}$ yields

$$\frac{\lambda_{max}}{\lambda_{min}} \leq \sqrt{\frac{||\vec{c}||^2 + (1-\delta)^2}{||\vec{c}||^2 + \delta^2}}. \tag{C.19}$$

Rearranging yields the quadratic $(\kappa^2-1)\delta^2 + 2\delta - 1 + (\kappa^2-1)||\vec{c}||^2 \leq 0$. Solving that for δ and observing that $||\vec{c}||^2 \leq n-1$ yields the desired result. □

Under this condition, we can guarantee that certain surrounding nodes can be omitted from checking and can, therefore, choose a threshold of $t = 1 - 2\delta$. Experimentally, however, choosing lower thresholds still yields optimal results, e.g. $t = ||\mathbf{\Delta}||_2$ - we motivate this choice in the following section.

C.3.2 Physical motivation for diagonal and spherical limit case

The constant interaction model is a good approximation when the quantum dots interact weakly and the associated tunnel coupling is small. As a result, we should expect the interdot capacitive coupling to be smaller than the coupling to the nearest gates; the diagonal elements will be considerably larger than the off-diagonals. As a result, we can write

$$\mathbf{c}_{dd} = \mathbf{D}(\mathbf{I} - \mathbf{\Delta}) \tag{C.20}$$

where $\mathbf{D}$ is a diagonal matrix, $\mathbf{I}$ is the identity matrix, and the elements $\mathbf{\Delta}$ are given by the ratio of the off-diagonal terms to the on. The inverse of this matrix can be approximated using the Neumann expansion

$$\mathbf{c}_{dd}^{-1} = \sum_{n=0}^{\infty} \mathbf{\Delta}^n \mathbf{D}^{-1} \approx (\mathbf{I} + \mathbf{\Delta})\mathbf{D}^{-1}. \tag{C.21}$$

Therefore, for small Δ where this approximation is valid, the $\mathbf{c}_{dd}^{-1}$ matrix is diagonally dominant and is a perturbation from the diagonal matrix. If $\mathbf{c}_{dd}$ were diagonal, the charges in the quantum dots would not interact. In this case, the lowest energy discrete charge configuration could be found by rounding the continuous solution to the nearest integer charge, as we prove in the next section.

It follows that for small but non-zero $\mathbf{\Delta}$ rounding will yield the lowest energy charge state except in the most ambiguous cases where a fractional component of the ith element of the continuous minimum, N_i^*, is close to 1/2. Therefore, the threshold should be proportional to some $\mathbf{\Delta}$ norm,

for example

$$t = ||\mathbf{\Delta}||_2 \tag{C.22}$$

C.4 Proofs of analytical results

C.4.1 Number of charge states

Theorem C.4.1. *The expected number of charge states needing to be considered by the thresholded algorithm with a threshold t is given by $(1+t)^{n_{dot}}$ where n_{dot} is the number of quantum dots.*

Proof. For random uniformly distributed gate voltages, the continuous minimum can also be considered uniformly distributed. Therefore, in 1d, the probability that the algorithm has to consider one charge state is $1-t$, and the probability that it has to consider two is t. Therefore, the expected number of points is $\mathbb{E}[N_{1d}] = t \cdot 2 + (1-t) \cdot 1 = 1 + t$. As the dimensions are independent $\mathbb{E}[N_{Nd}] = \mathbb{E}[N_{1d}]^N$. Therefore, if there are n_{dot} quantum dots the expected number of charge states is $(1+t)^{n_{\text{dot}}}$

□

C.4.2 Optimal gate voltages

Theorem C.4.2. *The gate voltages that minimise the charge state $\vec{Q}_d$ free energy, $F(\vec{V}_g, \vec{Q}_d)$ is*

$$\vec{V}^* = (\mathbf{R}\mathbf{c}_{gd})^+ \mathbf{R}\vec{Q}_d, \tag{C.23}$$

where $\mathbf{R}^T\mathbf{R} = \mathbf{c}_{dd}^{-1}$ is the Cholesky factorisation and $+$ denotes the Moore-Penrose pseudo inverse.

Proof. The free energy is given by

$$F(\vec{V}_g, \vec{N}_d) = \frac{1}{2}\left(\vec{Q}_d - \mathbf{c}_{gd}\vec{V}_g\right)^T \mathbf{C}^{-1}\left(\vec{Q}_d - \mathbf{c}_{gd}\vec{V}_g\right). \tag{C.24}$$

Using the Cholesky factorisation $\mathbf{c}_{dd}^{-1} = R^T R$, the minimisation problem can be reformulated as the following linear least-squares problem

$$\min_{\vec{V}_g} ||\mathbf{R}\vec{Q}_d - \mathbf{R}\mathbf{c}_{gd}\vec{V}_g||_2. \tag{C.25}$$

Solving this via the Moore-Penrose pseudo inverse yields the desired result. □

C.4.3 Virtual gates

This section derives how to construct virtual gates, namely determining how to change the gate voltages to obtain a specific change in the dot potentials.

Theorem C.4.3. *The optimal virtual gate matrix, α, used to construct virtual gates is*

$$\alpha = (\mathbf{c}_{gd}\mathbf{c}_{dd}^{-1})^+ \tag{C.26}$$

where + denotes the Moore-Penrose pseudoinverse. The ith row of this matrix encodes the electrostatic gate voltages required to change the ith dot's potential.

Proof. The quantum dot potential changes with gate voltages according to $\Delta\vec{V}_d = \mathbf{c}_{dd}^{-1}\mathbf{c}_{gd}\Delta\vec{V}_g$ according to (2.7). If this simple linear system is invertible, the Moore-Penrose pseudoinverse boils down to the regular inverse. If it is underdetermined (i.e., more gates than dots and full row rank), there are infinite solutions. In this case, the pseudoinverse picks the solution with minimal 2-norm. In case of an overdetermined linear system (i.e. more dots than gates or no full column rank) there is no solution. In this case, the pseudoinverse returns the least squares approximate solution. □