AI For Quantum Computing in Silicon

Brandon Severin

St Catherine's College

University of Oxford

Supervisors: Prof. G.A.D. Briggs & Prof. N. Ares

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas Term 2023

Abstract

Spin qubits in silicon-based quantum devices are a candidate quantum computing architecture because of their high fidelities, long coherence times and pathway to scalability. However, their potential for scaling is tainted by device variability. Each device must be tuned to operation conditions. Automated artificial intelligence-based tuning methods are necessary as individual devices scale and the dimensions of the tuning parameter space increase. This thesis presents algorithms for the automatic tuning of silicon-based quantum device architectures. I demonstrate a machine learning-based algorithm that is capable of tuning a 4-gate Si FinFET, a 5-gate GeSi nanowire and a 7-gate Ge/SiGe heterostructure double quantum dot device without human intervention. I achieve double quantum dot tuning times of 30, 10, and 92 minutes, respectively. I construct a new classifier of quantum transport features using machine learning and obtain novel insights into the double quantum dot parameter space across the different device architectures. I demonstrate the first algorithm for the automatic tuning of an ion-implanted donor in silicon device up to the point of readout calibration within 10 minutes. Modules relying on computer vision perform signal processing of quantum transport measurements synonymous with donor spin in silicon devices and enable tuning and characterisation faster than human experts. Finally, using machine learning I infer true qubit states from imperfect measurements and cross-examine our method on simulated data. I estimate initialisation fidelities of 99.34% for a Si-MOS qubit at 1 kelvin, further validating silicon-based architectures as a platform for quantum computing. These results show that AI-enabled automation is integral to the wave which carries silicon-based quantum devices towards the shores of universal fault-tolerant quantum computation.

Publications

The data presented in Chapter 5 is included in the following publication:

B. Severin, D. T. Lennon, L. C. Camenzind, F. Vigneau, F. Fedele, D. Jirovec, A. Ballabio, D. Chrastina, G. Isella, M. de Kruijf, M. J. Carballido, S. Svab, A. V. Kuhlmann, F. R. Braakman, S. Geyer, F. N. M. Froning, H. Moon, M. A. Osborne, D. Sejdinovic, G. Katsaros, D. M. Zumbühl, G. A. D. Briggs, and N. Ares. Cross-architecture tuning of silicon and SiGe-based quantum devices using machine learning. *arXiv:2107.12975*, 2021.

The data presented in Chapter 6 will be included in the following publication and uploaded to the arXiv prior to the *Viva Voce* examination:

B. Severin, T. Botzem, X. Yu, B. Wilhelm, H. G. Stemp, D. Schwienbacher, D. Holmes, A. S. Dzurak, K. M. Itoh, A. M. Jakob, D. N. Jamieson, A. Morello, and N. Ares. Automatic tuning of a donor in silicon quantum device using machine learning. *In preparation*.

The data presented in Chapter 7 is included in the following publication:

J. Y. Huang, R. Y. Su, W. H. Lim, M. Feng, B. van Straaten, **B. Severin**, W. Gilbert, N. D. Stuyck, T. Tanttu, S. Serrano, J. D. Cifuentes, I. Hansen, A. E. Seedhouse, E. Vahapoglu, N. V. Abrosimov, H. J. Pohl, M. L. W. Thewalt, F. E. Hudson, C. C. Escott, N. Ares, S. D. Bartlett, A. Morello, A. Saraiva, A. Laucht, A. S. Dzurak, and C. H. Yang. High-fidelity operation and algorithmic initialisation of spin qubits above one kelvin. *arXiv:2308.02111*, 2023.

Conference Presentations

Silicon Quantum Information Processing Workshop, Online, 2021. Contributed Talk: Cross-architecture tuning of silicon and SiGe-based quantum devices using machine learning.

Oxford Quantum Day, Clarendon Laboratory, University of Oxford, 2022. Poster: Cross-architecture tuning of silicon and SiGe-based quantum devices using machine learning.

Spin5, Pontresina, 2022. Poster: Cross-architecture tuning of silicon and SiGe-based quantum devices using machine learning.

Australian Research Council Centre of Excellence for Quantum Computation and Communication Technology (CQC²T), UNSW, Sydney, 2022. Invited Talk: Real-time tuning and characterisation of quantum devices using machine learning.

The American Physical Society's March Meeting, Las Vegas, 2023. Contributed Talk: Cross-architecture tuning of silicon and SiGe-based quantum devices using machine learning.

CQC²T Workshop, Hunter Valley, 2023. Invited Talk: Automatic tuning of a donor in silicon quantum device using machine learning.

Advances in Automation of Quantum Dot Devices Control, NIST, Maryland, 2023. Contributed Talk: Automatic tuning of a donor in silicon quantum device using machine learning.

Persistence is everything,

Acknowledgements

Natalia Ares - *You took a bet on me. You changed my life. I will forever be grateful. Thank you for everything.*

Andrea Morello, Giorgos Katsaros, Dominik Zumbühl, Silvano De Franceschi, Thaddeus Ladd, Steve Roberts, Andreas Fuhrer, Dominic O'Brien, John Morton, Andre Saraiva, Andrew Dzurak, Simon Benjamin, Harish Bhaskaran, Andrew Briggs, Federico Fedele, Danielle Holmes, Holly Harsant, Tom Neville, Lascelles Hussey, Keivan Samani, Andrej, M. A. Osborne, HomeDAO, Johana Saldaña, M. de Kruijf, A. Ballabio, Laurence Kirk, Ian Edward, Anna Pearson, Austen Robinson, Albie Tremlett, Chris Adambukulam, Dominic Lennon, Daniel Schwienbacher, The Bra Boys, William Wombell, Joel Pendleton, Maya Isarov, D. Sejdinovic, David Craig, Cameron Jones, G. Mahler, Arne Laucht, Riri, Barry Sanders, Henry Yang, Henry Howell, Nabs, Arjen Vaartjes, James Bedford, Markella, Marta, Michaela Eichinger, Eito Miyamura, Andrew Clealand, Debo Origano, Fabrizio, Michael Hall, Daisy Tremlett, Rujual Bains, Kiana Malmir, Steffanie, Yona Schell, Gadigal People, Sally Tremlett, Devis, Archie Young, Efi Barda, Mark Johnson, Sam Johnson, Simone Pincio, David Edwards (r.i.p.), Melanie A., Tim Botzem, Florian Vigneau, Sarah, Fay Hudson, Leon Camenzind, Nicola, Dario, Ed Lando, Qasi Gaba, Mario, Deep Lall, The Woshipful Company of Ironmongers, Daniel Jirovec, Gözde Ustun, Maggie, Santiago Serrano Ramirez, Lazar Lakic, Shannon Nicley, Sonali Goel, Fabio Ansaloni, Daniel Antoine-Donatein, Marek Rychetský, Alex Dickie, Toby Harsant, Seigo Tarucha, Tuomo Tanttu, Jason Smith, Esra Ertan, Jarryd Pla, Anasua Chatterjee, Aurélien Trichet, Rafael Eggli, Steve (Xi) Yu, yFe, Corey

Ostrove, Elena Vicario Santos, Han, Giulia Borini, Marthe Naudts, Ben Young, Sanmi Adekanye, Simon Sampson, Andreas Nickl, Philippa Moss, Wee Han Lim, Nicholas Sim, Paul Steinacker, Lucas Schorling, Vitalik Buterin, Andrew Kirima, Benjamin Wilhelm, Courtney, Adrian Taylor, Konrad Kopp, Sebastian Orbell, Anders Kringhøj, Ian Hammond, Femi Nylander, Sam Sutherland, Rupert Stonehill, Ferhat, Norbert Gottenberg (r.i.p.), Nard Dumoulin Stuyck, Matt Mai, Deems, Wyatt Vine, Will Minashi, Encode Club, Ian Sutton, The Rank Foundation, Robin Blume-Kohout, Valentina Gualtieri, Amanda Seedouse, James Little, Alona Shevchenko, Benjamin Barclay, Sammi Bammi, Kabir Gandhi, Josh King, Laura Dunn, Sharmaine, Miguel Carballido, Mark van Blankenstein, Giulia Galatolo, Shobhan Dhir, David Worlidge, Andrew Worrall, Dan Steward, Edward Laird, Ian Berkman, John the Porter, Zander, Larry, Md. Mamunur Rahman, Danilo, Andy Harsant, Tobias, Torbjørn Rasmussen, Steve Yanni, Max Fosh, Joseph Hickie, G. Isella, Jonas Schuff, Demis Hassabis, Layi Wang, Pippa Harsant, Martin Nurizzo, Paul Warren, Hyungil Moon, Joost van der Heijden, Naomi Mowat-Amie, Rocky Su, Alessandro Lodi, Farri Gaba, The Worshipful Company of Armourers and Brasiers, Arkady Fedorov, Jonathan Huang, The Ferreras-Willets, Michael Glossop, Andreas Kuhlmann, Lewis Williams (r.i.p.), James Lane, Max Schart, Lisa Gächter, Rohit, Holly Stemp, Mixie, Agnese Turra, Stani, Nina van Esbroeck, Arkadiy Kulik, Floris Braakman, Emin Gün Sirer, Michael Churchouse, Laura A O'Neill, Jaya, Pranav Vaidhyanathan, Anatoly Crachilov, Jamie Newton, Vincent Michal, Simon Svab, Will Gilbert, Serwan Asaad, Shazeaa Ishmael, Yikai Yang, Matthias M., Gerard J. Milburn, Hannah Chenok, Ingvlid Hansen, Charles Farrar-Bell, My Family, Will Yoon, and many more, James Baron, Annabella W., Simon Geyer, Oluwafemi Fakokunde, Irene Fernández de Fuentes, Vu Nguyen, Ajit Dash, Nathan Walemba, Zuska, Barnaby van Straaten, Hannah Jacobs, Chris Akinola, Toni Fola-Alade, Jonathan Dexter, Reo, Ashley, Alek Kloda, Will Okell, Vivek Wadhia, Lampros Bisdounis, Yolanda, Florian Froning, Mark Tremlett, D. Chrastina, Ferdinand Kuemmeth, Cassandra Warren, David Fauchier, Kristof Lommers, Jaime Saez Mollejo,

Kat, Karina Hudson, Gibbo, Ensar Vahapoglu, Michelle Simmons, Jésus Cifuentes, Colonel Charlie Knaggs, Jon Hopkins, MengKe Feng, Rima Tremlett, Rose A., Connor Vanderhook, Gemma Adams, Kushagra Aggarwal and Alissa Hummer.

Contents

Abbreviations

AI artificial intelligence

AWG arbitrary waveform generator

BNC Bayonet Neill–Concelman

CMOS complementary metal–oxide–semiconductor

CPMG Carr–Purcell–Meiboom–Gill

DAC digital-to-analog converter

DRAM dynamic random-access memory

FET field-effect transistor

FQT Fundamental Quantum Technologies

GPU graphics processing unit

HMM Hidden Markov model

ML machine learning

MOS metal-oxide-semiconductor

MOSFET metal-oxide-semiconductor field-effect transistor

NV nitrogen vacancy

PCB printed circuit board

PMMA polymethyl methacrylate

PSB Pauli spin blockade

RF radio frequency

RF-SET radio-frequency single-electron transistor

SET single electron transistor

Si-MOS silicon-metal-oxide-semiconductor

SPAM state preparation and measurement

List of Figures

List of Tables

Chapter 1

Introduction

> I think there is a world market for about five computers.
>
> Thomas J. Watson

In 1492 there were no horses in America. An animal now synonymous with American culture, politics, growth and identity only reappeared on the continent after the arrival of Christopher Columbus in 1493 [1]. 400 years later, American cowboys and cowgirls could not imagine life without a horse. Similarly, today we struggle to make it through an hour without checking on our tailored stream of information delivered by a mobile pocket computer. A pocket computer built on the horseback of the silicon transistor. A billion silicon transistors, all working harmoniously, carrying gigabytes of information. Globally, there are over a billion computers each with over a billion transistors, an unimaginable outcome for Thomas J. Watson of IBM who thought that the market for computers was limited to single digits.

We squeal; plucked from the ice-cold Steel-Age baby's bath water, only to find comfort, tightly wrapped and bundled in densely woven blankets of software running on silicon. We are now in the Silicon Age. Although it is a material of a grey metallic lustre in colour, its prominence continues to be reflected. The late Gordon Earle Moore observed in 1965 that the number of transistors on a microchip doubled

every two years. Computational power has continually increased; the modern-day smartphone can perform arithmetic faster than the supercomputers which landed man on the moon in 1969. Half of the world's top ten most valuable companies are either expert manufacturers of silicon transistors or exceptionally adept at developing software to perform calculations upon them. The proliferation and access to raw computing power throughout the new millennium have enabled our ability to process and harness the exponentially increasing amount of data created. Evident milestones of our capabilities are not limited to but include Deep Blue, AlphaZero, AlphaFold, Dall-E, and chatGPT. A short list of computers and artificial intelligences (AIs) that dethrone Chess grandmasters, conquer Go world champions, predict the structures of the molecular building blocks of life, generate realistic images from text and converse in a human-like manner based on context and prior conversation. However, in light of all of these achievements and our historical growth of computational prowess, computers are bound by their physical nature in the class of problems that they can solve.

My computer is limited. My current classical computer is limited. Despite all the computational power in the world were I to use all of it, there would still be problems I would be unable to solve in my lifetime and many generations yet to come. Some of these limitations have their advantages. Secure communication over the internet relies on the infinitesimally small statistical probability of a bad actor being able to solve hard problems [2]. In many cases, the only approach is to make a series of guesses at a solution and see if one of them is correct. There are however many disadvantages to limitations in computational power. Many of these limitations lie in solving problems which reside in the quantum realm of nature and could lead to significant advancements in medicine and science [3]. To solve problems like these, a sizable upgrade to current computers is needed. Increasing their memory will not help, and including graphical processing units (GPUs) will not make a noticeable impact on the time to reach a solution for multiple problems. But, we are possibly heading along the right path, which is altering how the computer processes information.

Each individual unit of information, a bit, can carry a single value in a classical computer - either a one or a zero. In a quantum computer each bit, or qubit, can carry more than a single classical bit of information. This is due to the quantum mechanical phenomenon of superposition. N qubits can carry as much information as 2^N classical bits. How can we make such a powerful computer? One of the key requirements is a two-level quantum system [4]. Many physical implementations have come about from experimentalists recognising the work they were carrying out in the lab could be used as a basis for quantum computing. These include trapped ions, superconducting resonant circuits, nitrogen vacancy (NV) centres in diamond, and semiconductor spin qubits. Deciding between these options can be done using a range of criteria, for example: relative advancement, ease of manufacture, or compatibility with current technology. But there are other important questions to answer such as how many qubits do we need for our quantum computer to solve problems accurately?

The exponential nature of the information-carrying capacity of qubits may lead one to believe that no more than a couple hundred qubits are required for quantum computation. A quick back-of-the-envelope calculation shows that 256 qubits could carry as much classical-bit information as there are atoms in the universe. However, qubits are not perfect and are prone to errors due to decoherence. There are additional errors from faulty gates, measurement or, quasiparticle poisoning for example [5], therefore additional qubits are needed to correct for these errors [6]. At least a million qubits will be required [6–8] for a universal fault-tolerant computer. A scalable architecture is key to achieving this.

Trapped ions [9] have shown high gate fidelities [10, 11] and long coherence times [11, 12]. Having a high gate fidelity is essential for reducing computational errors. The longer the coherence time, the more time is available for gate operations before the qubit decoheres (due to sources of electric or magnetic noise). In this context, the speed of gate operations is also important. IBM has created the concept of quantum volume to consider all these measures of qubit performance [13]. NV centres [14,

15] have the advantage of relying on photonics and therefore can play a role in both quantum computation and communication with the ability to also operate at room temperature. Other photonic-based qubits include those being developed in silicon devices [16, 17] however, these have relatively lower fidelities compared to alternative qubit implementations [18, 19]. Superconducting qubits [20, 21] are currently some of the highest-held platforms due to achievements of performing calculations using 53 superconducting qubits [22]. Arute *et al.* [22] claims that superconducting quantum processors will follow a quantum version of Moore's law, doubling their computational power every few years, without tackling the issue of scalability. Scalability is an unfortunate pinch point of superconducting qubits, with each qubit taking up to approximately 0.1 mm^2 in area [23]. This makes it easy to address individual qubits with microwave lines. Problems occur when solving well-known quantum algorithms [24, 25], millions of qubits are required to accommodate for error correction and one has to cool down a 1 m^2 chip to 20 mK temperatures. To traverse this challenge, superconducting qubit manufacturers are already building three-dimensional integrated systems, placing microwave wiring on a separate layer to the qubits [26].

Spin qubits in semiconductor devices are however a scalable architecture for quantum computing. Each qubit has a small footprint of approximately 100 x 100 nm^2 and all can be electrically addressed. By using silicon there is an opportunity to take advantage of the existing trillion-dollar semiconductor industry and its lithography fabrication expertise. These devices generally require sub 100 mK operating temperatures but due to their small size, their cryogenic scaling footprint is orders of magnitude less than that of superconducting qubits. Semiconductor devices can operate at even hotter temperatures such as above 1 kelvin [27], as I will introduce later. Fidelities [28] and coherence times [29] are more than promising for an implementation which is compatible with current complementary metal-oxide-semiconductor (CMOS) fabrication technology. Unfortunately, scalability is tainted by device variability, with each quantum device requiring the tuning of parameters to configure it to operating

conditions. Research labs currently work with devices that house on the order of two qubits, and rely on tuning their devices by hand. In some cases, the manual tune-up time can take days, if not weeks and in the worst cases months to realise a qubit. This approach is simply not scalable given the need for hundreds of millions of qubits to build a universal fault-tolerant quantum computer.

This thesis aims to supplicate the notion that before we can use a quantum computer we first need to be able to turn it on. Given the arduous nature of this task when carried out manually by experimentalists working with semiconductor-based qubits in the lab, we must create automated methods to turn on a quantum computer as we scale the number of qubits. I stress that research institutions and quantum hardware companies will be hard-pressed to find one billion PhD students to tune their billion qubit quantum computers manually. Instead, we can leverage our classical computational capabilities to conquer and reign control over the quantum realm. We can utilise AI and software to automatically navigate the high dimensional complex parameter space and turn on silicon-based quantum devices. So far, I have provided context to the landscape of computer innovation and the desire for quantum computation. Additionally, I have delved into why a spin in silicon-based architecture is suitable for a quantum computer. Next, I will cover the theory behind qubits, semiconductor quantum dots, qubit readout methods and machine learning tools relevant to the rest of the thesis. Following, I will review the literature on semiconductor-based qubits and current state-of-the-art machine learning methods for tuning semiconductor quantum devices. Before delving into the results of this thesis I will cover the methodology used to obtain the data included in the following chapters. In Chapter 5 I will demonstrate the first algorithm capable of tuning three different semiconductor device architectures and providing novel insight into the parameter space of the quantum devices. In Chapter 6 I will present the first algorithm to tune an ion-implanted donor spin in silicon device up to the point of readout calibration from scratch using machine learning. In Chapter 7, we will use machine learning

techniques to infer true qubit states in the presence of erroneous measurements of a silicon spin qubit device operating at 1 kelvin. Finally, I will provide a summary of what has been demonstrated in this thesis and my views on future work in the field of artificial intelligence for quantum computing in silicon.

Chapter 2

Theory

> Straight to the good stuff (said while rubbing hands together).
>
> Andre Saraiva

Qubits. What are they and why do we care? How do we make qubits and what are the requirements on their characteristics so that we can do something useful with them? I will delve into these questions within this chapter discussing spin qubits, quantum dots, and qubit readout methods.

2.1 Qubits

The qubit is the information-carrying unit of a quantum computer, analogous to a bit in a classical computer. The state of a qubit is represented as a linear combination of two orthogonal vectors, $\binom{1}{0}$ and $\binom{0}{1}$, $|0\rangle$ and $|1\rangle$ in Dirac notation respectively, which are the qubit's computational basis states. Mathematically we can write the state of the qubit $|\psi\rangle$ as,

$$|\psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle \tag{2.1}$$

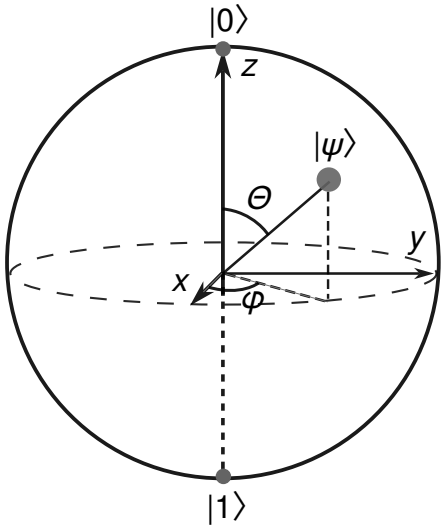


Figure 2.1: The Bloch sphere. The Bloch sphere is a unit sphere where the computational basis states of the qubit $|0\rangle$ and $|1\rangle$ reside at opposite poles of the z-axis. A general qubit state lies on the surface of the sphere and is defined by the angles ϕ and θ.

for any complex α and β, and under the normalisation condition that

$$|\alpha|^2 + |\beta|^2 = 1 \tag{2.2}$$

where α and β are the probability amplitudes of the basis states. Their moduli squared, $|\alpha|^2$ and $|\beta|^2$, represent the probabilities of finding the qubit in one of the two basis states $|0\rangle$ and $|1\rangle$ respectively after measuring the qubit.

We can take advantage of the normalisation condition for α and β and visualise all the possible states of the qubit as points on the surface of a unitary sphere, known as a Bloch sphere (Fig. 2.1). Using polar coordinates, a general qubit state on the Bloch sphere can be defined as

$$|\psi\rangle = \cos(\theta/2)\,|0\rangle + e^{i\phi}\sin(\theta/2)\,|1\rangle \tag{2.3}$$

where a general qubit state can be defined by two angles $\theta \in [0, \pi]$ and $\phi \in [0, 2\pi]$.

What good is encoding information if you can't do anything with it? A classical computer relies on logic operations on bits to perform computation, quantum computers

possess a similar quality - qubit logic gates. A single qubit operation, or gate, moves a qubit from one position on the Bloch sphere to another. A suite of single-qubit gates, defined by the Pauli matrices,

$$\sigma_x = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \sigma_y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \sigma_z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{2.4}$$

correspond to a rotation of π radians about the respective axis x, y, z of the Bloch sphere (Fig. 2.1). Similarly gates can operate on multiple qubits at the same time, for example a two-qubit gate such as the controlled-phase (CZ) gate which adds a phase to one of the qubits, the *target qubit*, conditional on the state of the other qubit, the *control qubit*.

To characterise qubits many metrics are utilised, here I will list the core ones used and mentioned within this thesis. The first is the (spin) *relaxation time*, known as T_1. This refers to the time in which a qubit, remains in the excited state, $|1\rangle$, before it relaxes to the ground state $|0\rangle$. The second metric is the *coherence time*, T_2. The coherence time is the length of time the qubit remains in a state of superposition, a state that is a linear combination of $|0\rangle$ and $|1\rangle$, before it decoheres and collapses into a single classical state. Gate *fidelity* is a measure of how often the desired outcome is achieved when an operation (or gate) is performed on a qubit and is quoted as a percentage. Similarly, readout fidelity is a measure of how accurately one can measure the qubit state. Fidelities must be greater than 99 % to achieve a universal fault-tolerant quantum computer otherwise, the accumulation of errors will overwhelm any meaningful result from a quantum algorithm, even if using state-of-the-art error correction techniques [30].

2.2 DiVincenzo's Criteria

To build a quantum computer, there are five key criteria that qubits and gates will need to meet, as laid out by DiVincenzo in 2000 [4],

1. A scalable physical system with well-characterized qubits
2. The ability to initialize the state of the qubits to a simple fiducial state, such as $|000...\rangle$
3. Long relevant decoherence times, much longer than the gate operation time
4. A "universal" set of quantum gates
5. A qubit-specific measurement capability

The criteria exemplify a form of a quantum computing paradox; we need to create qubits that are decoupled from their environment while we maintain excellent control over their behaviour. Moreover, as stated in item 1, the physical system in which the qubits are realised must be scalable and thoroughly understood. Architectural implementations that satisfy the remaining four items are hard to achieve, and very few have the current capability of satisfying all five.

DiVincenzo goes on to state that, "the embodiment of a qubit is simply a quantum two-level system" [4]. Therefore, a qubit can be encoded in for example the polarisation of a photon (horizontal or vertical), the energy states of an atom (ground or excited), or the spin of an electron (spin-up or spin-down) known as spin qubits. Here, we focus on spin qubits realised in semiconductor quantum dots.

2.3 Quantum dots

Through material manipulation and/or electrostatic potentials humans possess the ability to define quantum dots within semiconductor crystals. Quantum dots are

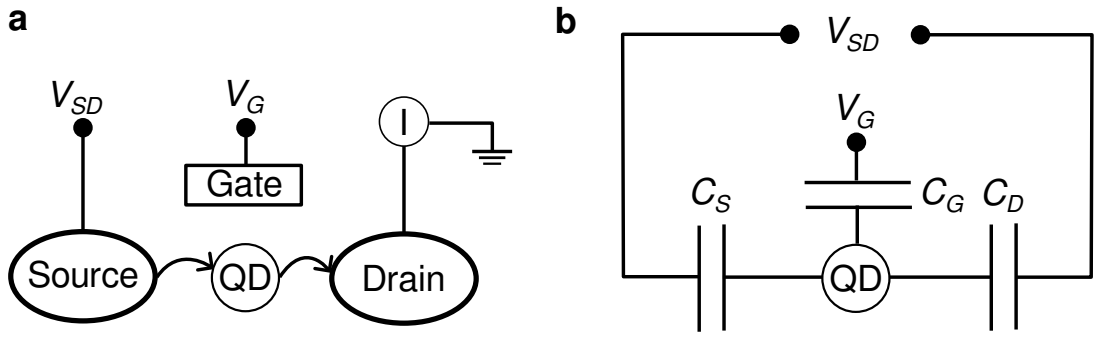


Figure 2.2: Schematic of the Constant Interaction model for a quantum dot. a) Schematic of a quantum dot device. A quantum dot (QD) can be defined by a material or electrostatic confinement potential. The quantum dot is connected to a (reservoir) source of charge carriers, and manipulating the voltage applied to a gate electrode V_G enables control of the energy of the quantum dot. Application of a bias voltage, V_{SD} creates the opportunity for charge carriers to flow from the source to the drain and a current to be measured. b) Constant Interaction model schematic. Capacitances, C_S, C_G, and C_D represent the coupling of the Coulomb interactions of the charge carriers within the QD to the source, gate and drain respectively.

artificial nanoscale electronic structures which can be filled with charge carriers (electrons or holes) [31], and posses a 0-dimensional density of states. The size of a quantum dot is comparable to the Fermi wavelength of the charge carrier, typically on the order of 100 nm. The number of charge carriers added or removed to the quantum dot, down to the single electron (or hole) regime, can be controlled through fine control of the confinement potential. The energy states of the charge carriers within the dot are quantised, and charge carriers obey the rules of atomic physics when filling these states [31]. Measurements of charge flowing through the quantum dot can be modelled by the Constant Interaction model.

2.3.1 Constant Interaction model

The Constant Interaction model (Fig. 2.2) is based on two main assumptions. 1) A single constant capacitance, C, represents the Coulomb interactions between electrons (or holes) within the dot and their interactions with the environment. C is the total of the capacitances connecting to the dot in the network including from the source, gate and drain of the device, C_S, C_G, and C_D respectively. 2) The Coulomb interaction is

independent of the number of electrons, N on the quantum dot. From Hanson *et al.* [31], the total energy, $U(N)$, of a single dot connected to source and drain reservoirs is,

$$U(N) = \frac{[-|e|(N - N_0) + C_S V_S + C_D V_D + C_G V_G]^2}{2C} + \sum_{n=1}^{N} E_n(B) \tag{2.5}$$

where $-|e|$ is the electron charge, N_0 is the number of background charges and V_S, V_D and V_G are the corresponding voltages applied to the source, drain and gate. $E_n(B)$ is the single-particle energy level which depends on confinement potential and the applied magnetic field, B. The electrochemical potential, $\mu(N)$ of the dot is:

$$\mu(N) \equiv U(N) - U(N-1) = (N - N_0 - \frac{1}{2})E_C - \frac{E_C}{|e|}(C_S V_S + C_D V_D + C_G V_G) + E_N \tag{2.6}$$

where $E_C = e^2/C$ is the charging energy, with $C = C_S + C_G + C_D$. The electrochemical potential contains an electrostatic part (first two terms) and a chemical part (last term). The electrochemical potential depends linearly on the gate voltage, the energy has a quadratic dependence. This dependence is the same for all N and the whole 'ladder' of electrochemical potentials can be moved up or down while the distance between levels remains constant. This makes the electrochemical potentials a convenient quantity for describing electron tunnelling. Applying a voltage bias between the source and the drain contacts, $V_{SD} = V_S - V_D$, creates a potential difference for charge carriers to flow. Moreover, it opens up a bias window for electrons to tunnel from the source reservoir to the dot, and then the drain. The bias window has an energy, $\mu_S - \mu_D = -|e|V_{SD}$. If an energy level of the dot sits within the bias window then electron transport is allowed. The electrochemical potentials of the successive dot energy levels are spaced by the addition energy, $E_{add}(N)$,

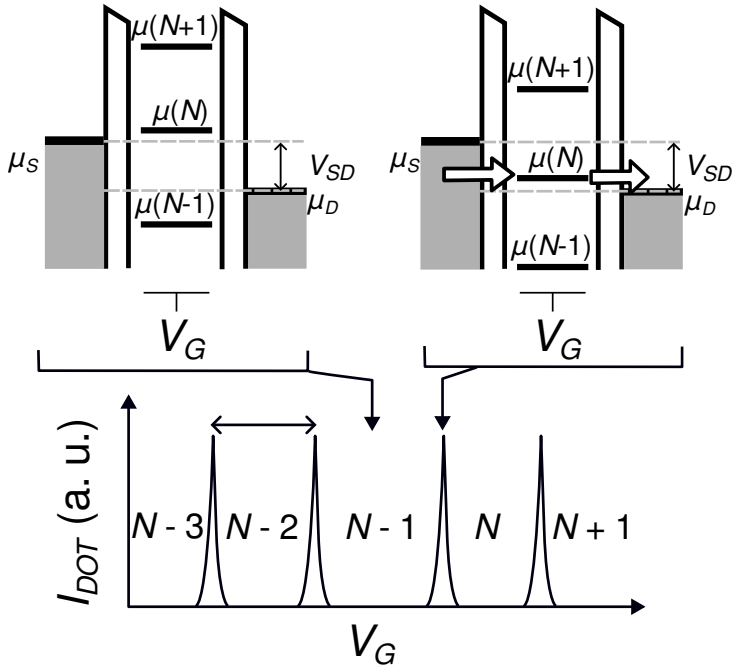


Figure 2.3: Quantum dot in the low bias regime. Application of bias voltage V_{SD} opens a window between the source and the drain electrochemical potentials, μ_S and μ_D, for charge carriers to flow. The window is small enough that a single electrochemical potential level of the dot, μ_N, can reside within the window when V_G is altered. When a dot level resides within the bias window as V_G is swept, current can flow and a peak in current (I_{DOT}) known as a Coulomb peak occurs. When a dot level exits the bias window, the current is halted and the dot is in a state called Coulomb blockade. Adapted from Ref. [31].

$$E_{add}(N) = \mu(N+1) - \mu(N) = E_C + \Delta E \tag{2.7}$$

The addition energy is made up of an electrostatic component of the charging energy, and the energy spacing between two discrete quantum energy levels, ΔE. ΔE can be zero in the event that two consecutive electrons are added to the same spin-degenerate level. We assume that the temperature is negligible compared to ΔE, corresponding to a temperature of approximately 1 K. The ladder of electrochemical potentials within the dot can be *plunged* or raised into and out of the bias window by control of V_G. There are two extremes in which the dot can operate, in the low bias regime where only a single dot level is within the bias window, and the high bias regime where multiple dot levels fall within the bias window.

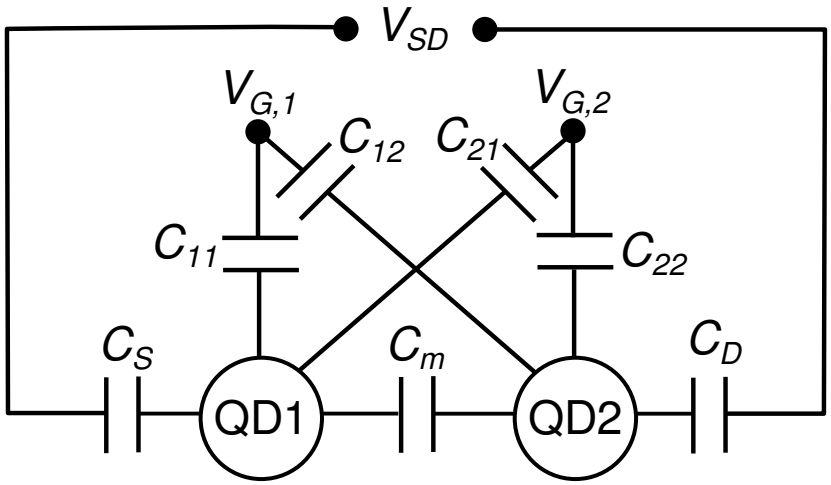


Figure 2.4: Schematic of the Constant Interaction model of a double quantum dot. Two quantum dots (QD1 and QD2) are capacitively coupled together by C_m. The respective gate electrodes ($V_{G,1}$ and $V_{G,2}$) for each dot (QD1 and QD2) are capacitively coupled (C_{12} and C_{21}) to the adjacent dot as well as their target dots (C_{11} and C_{22}).

In the low bias regime, if a dot level does not fall within the bias window, no current can flow through the dot, and the dot is in a state called Coulomb blockade (Fig. 2.3). Coulomb blockade can be lifted by altering V_G, such that $\mu_S \geq \mu(N) \geq \mu_D$ is satisfied enabling current to flow. An electron can tunnel onto the dot from the reservoir, and then tunnel off the dot into the drain. Once the dot is unoccupied, another electron can tunnel onto the dot from the source and so on. This behaviour is called single-electron tunnelling.

If we monitor the current flowing through the dot, I_{DOT} while sweeping the voltage V_G, we can observe characteristic peaks in current known as Coulomb peaks. These occur when a dot level falls within the bias window. The distance between successive Coulomb peaks corresponds to E_{add}, at which point the dot is in Coulomb blockade. The width of the Coulomb peaks is correlated to V_{SD} and can be increased to the point where Coulomb peaks merge, as V_{SD} becomes comparable to E_{add}.

2.3.2 Double quantum dots

The Constant Interaction model can be extended from a single dot to, two dots that are in series and capacitively coupled (Fig. 2.4). The electrochemical potential of dot

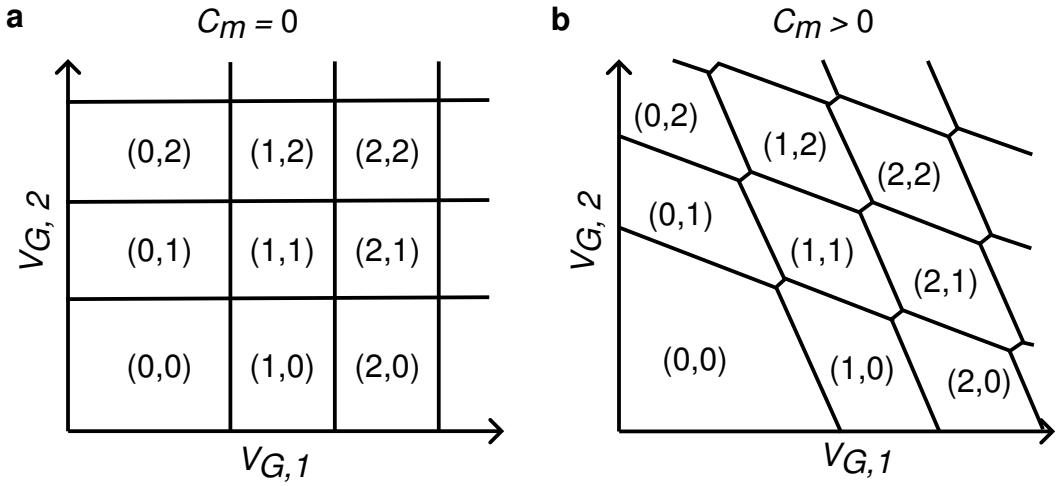


Figure 2.5: Charge stability diagrams, for the uncoupled (a) and coupled double quantum dots (b). The electron (or hole) number, (N_1, N_2) in each respective dot (QD1, QD2) changes at specific gate voltages denoted by the black lines. Cross capacitance between gate electrodes and adjacent dots is factored in (b) hence the diagonal characteristic to the black lines. Reproduced from Ref. [31].

1 is [31],

$$\begin{aligned}\mu_1(N_1, N_2) &\equiv U(N_1, N_2) - U(N_1 - 1, N_2) \\ &= (N_1 - \frac{1}{2})E_{C1} + N_2 E_{Cm} - \frac{E_{C1}}{|e|}(C_S V_S + C_{11}V_{G,1} + C_{12}V_{G,2}) \\ &\quad + \frac{E_{Cm}}{|e|}(C_D V_D + C_{22}V_{G,2} + C_{21}V_{G,1}) \quad (2.8)\end{aligned}$$

where C_{ij} is the capacitance between gate j and dot i, $C_S(C_D)$ is the capacitance from dot 1 (2) to the source (drain), E_{Ci} is the electrostatic coupling energy. The coupling energy E_{Cm} is the change in the energy of one dot when an electron is added to the other dot. $\mu_2(N_1, N_2)$ can be obtained by interchanging 1 and 2, as well as $C_D V_D$ and $C_S V_S$ in Eq. 2.8.

We can define a map, known as a charge stability diagram, which denotes at which voltages electrons get added (removed) to (from) each dot as we sweep $V_{G,1}$ and $V_{G,2}$. If there is no cross-capacitive coupling between gates and dots, $C_{12} = C_{21} = 0$, and the electrostatic coupling is zero, $E_{Ci} = 0$ then as we sweep $V_{G,1}$ and $V_{G,2}$ the

transitions between the number of electrons changing in each dot is defined by a series of horizontal and vertical lines as a function of $V_{G,1}$ and $V_{G,2}$. If the dots are capacitively coupled to each other and there is cross-capacitive coupling between gates and dots (*cross-talk*), the charge stability diagram appears as a hexagonal or "honeycomb" pattern. Each previous crossing point is split into two triple points at which three different charge states are energetically degenerate. The spacing between the triple points is defined by the interdot capacitance, C_m. In the low bias regime, quantum transport is only possible at the triple points.

Gate electrode voltages can be linearly combined to effectively remove the cross-talk between adjacent gate electrodes and quantum dots. The combination of gate electrode voltages in this manner is known as a virtual gate and results in orthogonal transition lines in the charge stability diagram within the virtual gate voltage space. Virtual gate construction effectively relies on calculating the gradient of the transition lines within a charge stability diagram. Methods to construct virtual gates include fitting measurements to the Constant Interaction model [32], the linear fit of transition lines [33], computer vision and machine learning, especially as the number of gate electrodes and device dimensionality increases [34].

2.4 Loss-DiVincenzo qubit

One of the simplest two-level quantum system and therefore qubit is the spin of an electron (spin-up or spin-down). Pioneered by Loss and DiVincenzo in 1998 [36], the computational basis states are $|\uparrow\rangle$ and $|\downarrow\rangle$, parallel or anti-parallel to an external magnetic field B_0 (Fig. 2.6). In the presence of an external magnetic field the energy levels of the spin states split, known as the Zeeman effect. The energy splitting, $E_Z = 2g\mu_B B_0 S$ with the electron g-factor, $g \approx 2$, in silicon, the Bohr magneton $\mu_B \approx 9.274 \times 10^{-24}$ JT^{-1}, and spin quantum number, $S = 1/2$. Manipulation of the qubit's state can be realised by electron spin resonance (ESR) or electric dipole spin

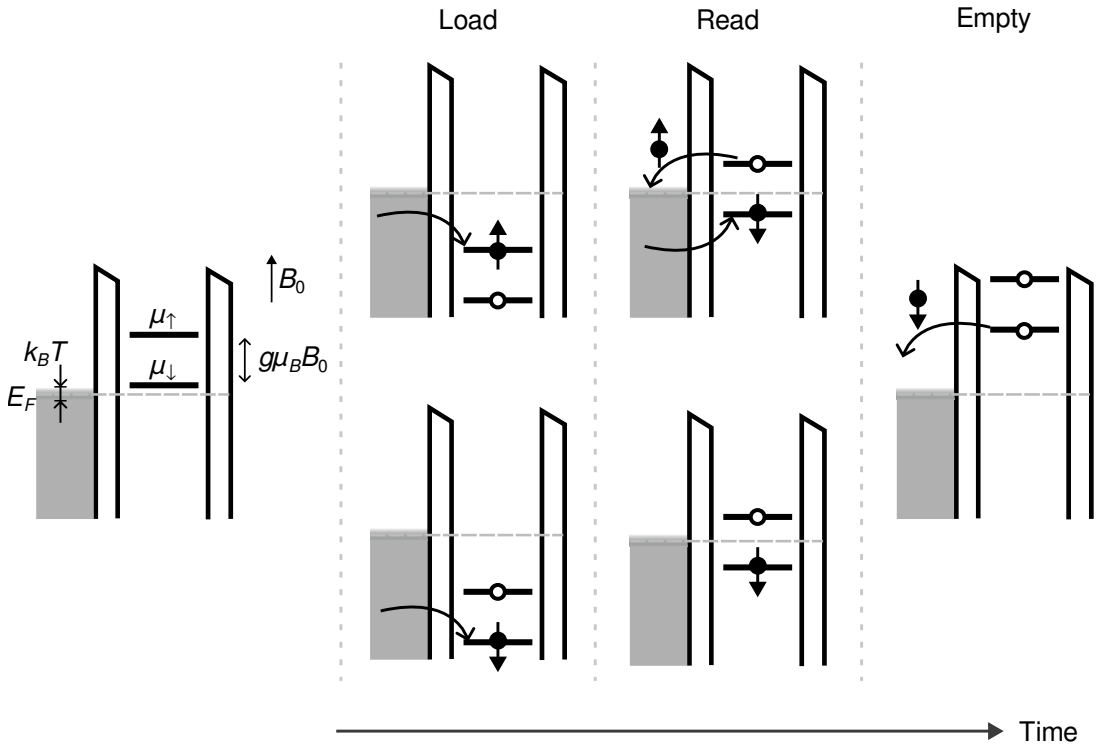


Figure 2.6: Loss-DiVincenzo qubit and Elzerman readout. An external magnetic field, B_0, Zeeman splits the dot electron spin-up, $\mu_\uparrow$, and spin-down, $\mu_\downarrow$, electrochemical potentials encoding a qubit into the spin of an electron. Electrons can be loaded onto the dot from a nearby reservoir with a Fermi energy E_F and thermal broadening of states k_BT at temperature T. A spin-up or spin-down electron is loaded onto the dot by plunging its electrochemical potential below the chemical potential of the reservoir and then manipulated by an externally oscillating magnetic field resonant with Zeeman splitting energy. The qubit state on the dot is read by raising the electrochemical potential of the dot such that a spin-up electron can tunnel off the dot, resulting in a current that can be measured or capacitively detected, and a spin-down electron can't tunnel. The dot is then emptied by raising the electrochemical potential of the dot level above the reservoir. The sequence of load, read, and empty occurs on the order of milliseconds. Adapted from Ref. [35].

resonance (EDSR). Via ESR the qubit is subjugated to an oscillating magnetic field with frequency, f_{AC}, resonant to E_Z which can be on the order of tens of gigahertz, where $\mu_B \approx 14.00\ \mathrm{GHzT^{-1}}$ and $B_0 = 2\ \mathrm{T}$. Electric dipole spin resonance relies on an oscillating electric field and a mechanism to couple the oscillating electric field to the electrons spin degree of freedom, such as spin-orbit or hyperfine interaction or an on-chip micromagnet.

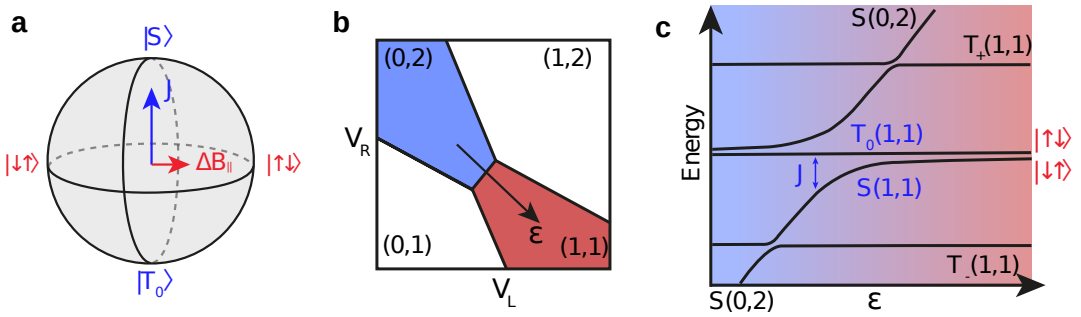


Figure 2.7: Singlet-triplet qubit. a) The Bloch sphere of the singlet-triplet, (S-T_0) qubit. The computational basis states are separated by the exchange energy splitting, J; a magnetic field gradient, $\Delta B_{||}$, between two quantum dots provides the second rotation axis. b) The charge stability diagram where the singlet-triplet qubit is realised. The arrow, ϵ, denotes the level of detuning between the two dot energy levels. The degree of exchange splitting can be controlled by pulsing along the detuning axis in gate voltage space. c) An energy level schematic of the various spin states as a function of detuning. A finite magnetic field leads to Zeeman splitting of the triplet states. The colour scheme represents where the system spin states are dominated by the exchange energy J or $\Delta B_{||}$. Reproduced from Ref. [37].

2.5 Singlet-triplet qubit

The singlet-triplet qubit (Fig. 2.7) encodes a qubit in the spin states of two electrons within a double quantum dot. The computational basis states (Fig. 2.7a) are the (ground) singlet state, $|S\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$, and the triplet state, $|T_0\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle)$. The total possible states of the two-spin system is four, one singlet and three triplet states corresponding to,

$$|S\rangle = \frac{|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle}{\sqrt{2}} \tag{2.9}$$

$$|T_+\rangle = |\uparrow\uparrow\rangle \tag{2.10}$$

$$|T_0\rangle = \frac{|\uparrow\downarrow\rangle + |\downarrow\uparrow\rangle}{\sqrt{2}} \tag{2.11}$$

$$|T_-\rangle = |\downarrow\downarrow\rangle \tag{2.12}$$

The double quantum dot is operated in the single electron regime, often at the (0,2) - (1,1) charge state transition where (n, m) denotes the number of electrons on the left and the right quantum dot respectively (Fig. 2.7b). At zero magnetic field, the three triplet states are degenerate. The T_0 (0,2) state is split from the singlet (0,2) ground state by E_{ST}, which is dominated by the exchange interaction due to the high wave-function overlap of electrons on the same dot. The $|S\rangle$ and $|T_0\rangle$ states are split by J the exchange energy. J is electrically tuneable and depends on the detuning, ϵ between the two dots. ϵ measures the relative energy level difference between the two charge states (1,1) and (0,2), while the average energy level between the two dots stays the same. Without interdot coupling and/or deep into the (1,1) charge state the exchange interaction vanishes and the singlet (1,1) and triplet (1,1) states are degenerate. The two electron spin states correspond to $|\downarrow\uparrow\rangle$ and $|\uparrow\downarrow\rangle$ and the degeneracy between these two spin states can be lifted by the Zeeman interaction (Fig. 2.7c). This is achieved by applying a magnetic field gradient across the quantum dots, $\Delta \mathrm{B}_{||}$, creating the second axis of rotation around the Bloch sphere. The magnetic field gradient can be created by on chip micromagnet in the vicinity of the quantum dots. The degeneracy of the triplet states can also be lifted by the Zeeman interaction in the presence of a finite external magnetic field and therefore confine the relevant state space to S and T_0. Singlet-triplet qubits have the benefit that the magnetic fields required to lift the degeneracy between the triplet states are on the order of a few hundred mT [38].

2.6 Readout methods

Qubit readout relies on converting the spin state of the electron (or hole) to something that we can easily measure in an electronic device which is charge. This process is called *spin-to-charge conversion* [31]. A charge sensor is commonly placed near the quantum dot spin qubits and used to perform spin-to-charge conversion. Broadly, the flow of

current through the charge sensor depends on the electrostatic environment and thus the quantum device's charge state. The charge sensor provides a near non-intrusive method of probing quantum devices and achieving spin-to-charge conversion.

Charge sensors may take the form of quantum point contacts (QPC) which are constrictions on the order of an electron's wavelength in a two-dimensional electron/hole gas. Changes in the QPC current is used as a way to monitor spin readout. Alternatively, one may use a single electron transistor (SET) as a charge sensor instead of a QPC. The SET is a nonlinear nanoelectronic device which behaves similarly to a quantum dot. One can detect changes in current flow through the SET by leveraging Coulomb (un)blockade as a method of spin-to-charge conversion.

One method of spin-to-charge conversion is energy selective readout also known as Elzerman readout [35]. Elzerman readout relies on energy-selective tunnelling of a spin-up or spin-down electron from the quantum dot to a nearby reservoir (Fig. 2.6). It is achieved by tuning the electrochemical potential of the Zeeman split energy levels such that the spin-up level is above the electrochemical potential level of the reservoir, and the spin-down level is below the reservoir level. This means that only a spin-up electron can tunnel off the dot whereas, a spin-down electron will not be able to tunnel. This measurement requires the tunnelling rate of the spin-up electron off the dot to be much greater than $1/T_1$. Such tunnelling events are detected using a nearby charge sensor that is electrostatically coupled to the quantum dot, as the current signal is typically too small to detect a single electron in transport. Elzerman readout later evolved to Morello readout [39, 40] where the electron spin qubit is both electrostatically coupled and tunnel coupled to the charge sensor. The charge sensor in Morello readout is an SET.

For Elzerman readout to be effective, high external magnetic fields (> 1 T) and low (electron) temperatures must be employed ($\sim$ 100 mK). This is to ensure that the $E_Z > k_B T$ and that E_Z is greater than the thermal and electromagnetic broadening of the electron states on the SET [40] or reservoir.

An alternative method of spin-to-charge conversion is Pauli spin blockade [41] which relies on the Pauli exclusion principle and the conservation of an electron's spin during tunnelling. The state of the qubit is read out by pulsing from the (1,1) charge state to the (0,2) charge state region of a double quantum dot. If the electrons are in a spin singlet state, tunnelling is allowed. However, if the electrons are in a spin triplet state, the device will be in a state of blockade because the Pauli exclusion principle prohibits the electron from making the transition from the triplet (1,1) state to the singlet (0,2) state and the triplet (0,2) state is too high in energy to be accessed. The change in the double quantum dot charge state is commonly measured using a nearby charge sensor.

2.7 Machine learning

Machine learning (ML) is a branch of artificial intelligence (AI) that empowers computers, through the use of algorithms and statistical models, to learn approximations of functions by inference rather than explicit programming. It is a core technology in various practical applications, including natural language processing, computer vision, recommendation systems, and autonomous vehicles. ML encompasses different types of learning, including supervised and unsupervised methods, and while it offers powerful capabilities, it also poses challenges related to data, biases, and model interpretability. Originating from the mid-twentieth century, ML techniques have gained prominence in the past decade due to the availability of large datasets (greater than 100,000 units) and computational power greater than 1 TFLOPS (1 trillion floating-point operations per second). Here, I present an overview of some of the machine learning techniques relevant to this thesis.

2.7.1 Supervised learning

Supervised machine learning models are used to predict an outcome or perform a specific task on unseen data based on past experience. Such tasks could be in the form of classification, for example, does this image contain a cat or a dog, or does this charge stability diagram show signatures of a single quantum dot or not. Other tasks could be that of regression, for example, predicting the price of a house based on its features such as the number of windows, bathrooms, bedrooms, and location. Supervised machine learning models are referred to as *supervised* because they are *trained* on labelled (training) data. It is in this process of training where a machine learning model builds or *learns* an approximation to an unknown function by minimising the error between the model's prediction and the true labelled data. By training a model and then testing its capabilities on unseen labelled data we can characterise its performance. For classification problems, the metrics used are accuracy and confusion matrices. The accuracy, A, of the model is defined as the total number of correct predictions, divided by the total number of samples,

$$A = \frac{T_P + T_N}{P + N} \tag{2.13}$$

where and T_P (T_N) is the sum of the of true positive (negative) outcomes, and P (N) is the sum of the positive (negative) samples. The confusion matrix, M_C is a 2 x 2 array whose elements summarise the prediction results of a classification problem. Despite being called a matrix, it is not used as an operator, it is merely used to give insight into the types of errors that the classifier is making. It is made up of the number of false positives, F_P, false negatives, F_N, true positives, and true negatives,

$$M_C = \begin{pmatrix} T_N & F_P \\ F_N & T_P \end{pmatrix} \tag{2.14}$$

To obtain a more reasonable view of a model's accuracy when dealing with

unbalanced datasets, where there are many more examples of one class than others, we can use a metric known as balanced accuracy, A_{bal}. The balanced accuracy in a binary classification problem is defined as the arithmetic mean of the true positive rate, *sensitivity*, and true negative rate, *specificity*, of each class.

$$A_{bal} = \frac{1}{2}\left(\frac{T_P}{T_P + F_N} + \frac{T_N}{T_N + F_P}\right) \tag{2.15}$$

The accuracy of ML models can be improved and computational costs of training can be reduced by engineering the features of the raw training data, known as *feature engineering*. For example, using our house example, not providing the number of windows during training, or converting the location to coordinates rather than the street address.

2.7.2 Receiver operating characteristic

The accuracy of a binary classifier is influenced by the threshold chosen for classification. For example, should we use classification outputs greater than 0.5 or 0.8 as a positive label? As we decrease the threshold, more items will be classified as positive. The chosen threshold influences the false positive and true positive rates, where the false positive rate is equal to 1 - specificity. We can plot the true positive rate (y-axis) against the false positive rate (x-axis) at different classifier thresholds, known as a receiver operating characteristic (ROC) curve.

The area under the ROC curve (AUC), is an aggregated measure of the classifier's performance across different classification thresholds. A classifier with an AUC of 1, performs predictions which are correct 100% of the time, whereas a classifier with an AUC of 0, is wrong 100% of the time. Using the ROC and AUC can provide a richer framework than confusion matrices when assessing machine learning classification models.

2.7.3 Type I and type II errors

Erroneous predictions from a classifier have their respective trade-offs depending on the problem at hand. For example, a spin qubit formation tuning algorithm that relies on a classifier for the presence of coupled double quantum dots in a charge stability diagram (Fig. 2.5) before it proceeds to tune gate voltages to optimise transport features at the triple points. A type I error (false positive), will lead the algorithm to waste time optimising gate voltages in a region of parameter space that will not lead to a qubit. Conversely, a type II error (false negative), will lead the algorithm to miss out on a potential qubit. In both cases, there is a mis-classification but the user of the model may care more about speed, rather than forming each possible qubit. For example, if there are many alternative voltages in which one can form a double quantum dot, then it is not worth spending experimental time in parts of the parameter space which will not lead to qubits. But, if the parameter space is sparse for potential double quantum dot formation, then the user may be willing to tolerate type I errors.

Decision Trees and Random Forests

A Decision Tree is a supervised machine learning technique that can be used for classification or regression. It relies on recursively splitting the training data into subsets based on the most significant features to make its predictions. Referred to as a tree, each node of the tree, starting at a single root node, represents a decision to be made on a feature and the branches of the node are the respective outcomes of that decision. The leaf nodes of the tree are the final class labels or values in classification or regression respectively. The selection of features to make decisions on at individual nodes can be done by numerous methods with a common one being the Gini impurity [42]. The Gini impurity (or Gini index) is a measure of the probability that a randomly chosen sample will be incorrectly classified by a specific node. Therefore, the lower the Gini impurity the better, as there is a lower likelihood of misclassification. The

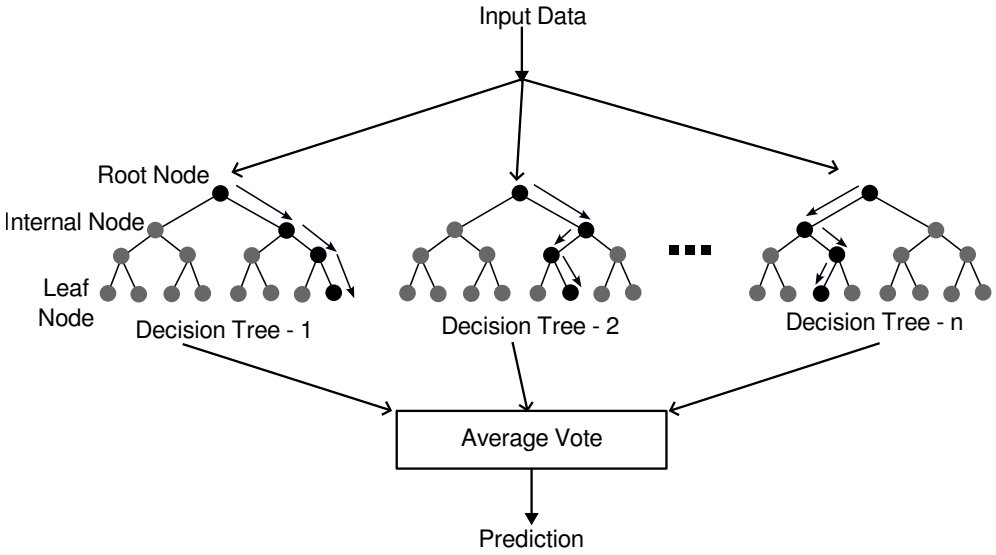


Figure 2.8: Random Forest. A Random Forest is a machine learning algorithm that is made up of an ensemble of Decision Trees each trained on a random subset of the training data. Outcomes from the decision trees are aggregated and averaged to produce the final prediction of the the machine learning model.

feature (and its relevant threshold) with the lowest Gini impurity is chosen as the root node of the tree. From the resulting decision outcome at that node, two child nodes are created from the root node. The lowest Gini impurity process is repeated to build a subtree from each child node, and so on... The entire Decision Tree is built up recursively until a stopping criteria is met such as the maximum tree depth being reached.

Decision Trees are a popular method because they can be visualised, and users can understand the decision pathways of the resultant model. However, Decision Trees tend to overfit, meaning that the trained model cannot generalise well to unseen data which effectively defeats the point of machine learning. A way of combating the overfitting problem is to group many different Decision Trees into an ensemble and form a forest.

Random Forests [43] are a type of supervised machine learning model which relies on the aggregation of predictions from many Decision Trees (Fig. 2.8). Each tree is trained on a randomly chosen subset of the training dataset, for the model's final prediction. Denoted as an ensemble approach, the risk of overfitting is reduced by

averaging predictions from multiple trees. Although they are less interpretable than a single Decision Tree, Random Forests carry out feature selection implicitly, reducing the need for manual feature engineering before training the model. Implicit feature selection occurs in that, features that lead to a greater reduction in Gini impurity at each node of the decision trees that make up the forest compared to other features are considered more important. Feature importance can be measured by calculating the mean decrease in Gini impurity for a feature used for splitting a node into child nodes across each tree of the random forest, and can be extracted after training.

2.7.4 Unsupervised learning

Unsupervised learning, as opposed to supervised learning, uses algorithms to learn patterns in an unlabelled training dataset without human intervention. A clustering algorithm will, in the ideal case, partition data into separate clusters based on their similarity. One of the most popular unsupervised ML methods is K-means clustering.

The number of clusters (K) is determined, typically using domain knowledge, in advance. The algorithm randomly initialises the centroids of the respective clusters in feature space. For each data point in the dataset, the Euclidean distance between the data point and each cluster centroid is calculated. The datapoint is assigned the cluster whose centroid it is closest to. The mean position of the data points in their respective clusters is calculated and the result is used as the new centroid of the cluster. The process of calculating the distance to the centroid and calculating the mean position is repeated until the position of the centroids stops changing. To reduce erroneous results the K-means clustering algorithm is run repeatedly with different random centroid initialisation starting points, and the final centroids with the mean lowest Euclidean distance to their respective data points are picked from the repeated runs. K-means clustering relies on significant assumptions about the underlying data, which are the clusters are normally distributed with a spherical covariance matrix that is the same for all clusters [44]. If these assumptions are not satisfied, K-means is

likely to fail and is not suitable for the application.

2.7.5 Multi-armed bandits

The multi-armed bandit problem appears in machine learning as it exemplifies the exploration-exploitation tradeoff dilemma. The term originates from the idea of a gambler, at a series of slot machines (known as single-armed bandits), who has to decide which machines to play, how many times to play each machine, in what order and whether to continue playing an individual machine (exploit) or try other machines (explore). The gambler's aim of course is to maximise their winnings, and in effect minimise their losses, considering fixed resources (money). Similarly, a machine learning agent or algorithm would aim to maximise reward, considering a fixed number of trials. As with the gambler, each action the agent takes is associated with a given reward, but that reward is a random variable. The agent has the difficulty of trialling different actions while also selecting the actions that maximise the reward over the number of trials available.

The agent has a few options in how to solve this problem of exploration vs. exploitation. At one end of the spectrum, the agent may continually explore, equivalent to trialling each slot machine repeatedly for the entire number of iterations. Conversely, the agent may take an exploit-only strategy, trialling each slot machine once, then only playing the one which initially gave the highest reward for the rest of their financial time in the casino. There are more intelligent solutions available such as epsilon-greedy, upper confidence bound (UCB) and Thompson sampling. The epsilon-greedy method consists of the agent taking the most rewarding action by default, but there is some probability (epsilon) that in a given iteration the agent will choose to explore, and thus selecting a different action at random. When following the upper confidence bound method the agent selects actions based on their perceived reward and a measure of the uncertainty of the reward. Following on, Thompson sampling, based on Bayesian statistics, relies on the construction of a statistical model of the rewards but an action

is selected based on a randomly drawn belief. The advantage of Bayesian methods lies in the seamless automation of exploration-exploitation tradeoff with no additional machinery.

Gaussian process

Gaussian processes are a non-parametric supervised learning technique that can be used for either regression or classification. Non-parametric means that the number of parameters of the model (and hence expressivity) grows with data. They are computationally manageable as one only requires two parameters to define a Gaussian curve, its mean and variance, however as we will discuss later, their computational requirements explode as the number of dimensions and data points increase. Moreover, they are leveraged within the realm of Bayesian statistics in the form of Gaussian process Bayesian optimisation. The ability of a Gaussian Process model to quantify its uncertainty in specific parts of the domain is useful for optimisers to decide which point in the domain to evaluate next to find a solution to the problem at hand efficiently. Additionally, the information of uncertainty can be leveraged by solutions for the exploration-exploitation trade-off as we have discussed previously. In this thesis I use Gaussian processes in Chapters 5 and 6. Here I will give a brief introduction to Gaussian processes.

A Gaussian process is a collection of random variables, separated in either the domain of time or space, any finite number of which have a joint Gaussian distribution [45]. The Gaussian Process, $\mathcal{GP}$, consists of a mean function, $\mu(x)$ and a kernel (also known as a covariance function), $k(x, x')$ which defines the correlations between different variables, giving the Gaussian process function, $f(x)$, as,

$$f(x) \sim \mathcal{GP}(\mu(x), k(x, x')) \tag{2.16}$$

The mean function $\mu(x)$ represents the expected value of the GP at any given input point x, and greatly influences the extrapolation of the predicted functions, i.e.

predictions far away from training data. The choice of a prior mean function $\mu_0(x)$, depends on the problem at hand, typically mean functions are set to a constant if one is only interested in interpolation. However, if extrapolation is required one may construct a specific mean function based on domain knowledge.

The kernel allows us to incorporate structure and correlation into our models and typically a kernel expresses that covariance decreases with increasing distance between variables. The choice of a prior kernel $k_0(x)$ also depends on the problem at hand and the goals of the user, as the kernel encodes our assumptions about the function we wish to learn. For example, does one expect the function we are trying to model to be rapidly varying and periodic, or relatively smooth.

Given observed data points $x_1, ..., x_n$ and their respective function evaluations $f(x_1), ..., f(x_n)$, we can predict the function value at a new input x by calculating the conditional distribution of $f(x)$ given the data observed so far $f(x_{1:n})$ using Bayes' rule [45],

$$f(x)|f(x_{1:n}) \sim \mathcal{N}(\mu_n(x), \sigma_n^2(x)) \tag{2.17}$$

$$\mu_n(x) = k(x, x_{1:n})k(x_{1:n}, x_{1:n})^{-1}(f(x_{1:n}) - \mu_0(x_{1:n})) + \mu_0(x) \tag{2.18}$$

$$\sigma_n^2(x) = k(x, x) - k(x, x_{1:n})k(x_{1:n}, x_{1:n})^{-1}k(x_{1:n}, x). \tag{2.19}$$

$f(x)|f(x_{1:n})$ is called the posterior probability distribution. $\mu_n(x)$ and $\sigma_n^2(x)$ are the posterior mean and posterior variance. The computational complexity of the Gaussian process scales with $O(n^3)$. This is because to calculate the posterior mean and variance of the Gaussian process to predict a new value, one must perform the inverse of the kernel matrix. Computation of the matrix inverse scales with $O(n^3)$, is numerically unstable and susceptible to condition errors. To reduce the computational

costs of the matrix inversion and improve stability, Cholesky decomposition is used which still scales on the order of $O(1/3n^3)$.

Kernels

There are a few commonly chosen kernels, one of which is the radial basis function (RBF), also known as the squared exponential [45],

$$k_{SE}(r) = \exp(-\frac{r^2}{2\ell^2}) \tag{2.20}$$

where $r = |x - x'|$, the distance between inputs x and x', hence the term *radial* in the name. ℓ determines the characteristic length scale which is roughly the distance in input space one must move before the function value changes significantly. ℓ also gives an impression of how far you can extrapolate beyond your data. We can multiple k by a positive constant σ_f^2, which behaves as a scale factor to get any desired process variance. The squared exponential function is popular because there are only two hyperparameters to optimise, σ_f^2 and ℓ. It is a universal kernel, meaning that under some conditions it is capable of learning any continuous function given enough data [46, 47]. Moreover, the squared exponential function is infinitely differentiable and therefore the resultant GP is very smooth but, perhaps too smooth to reflect real physical data. Another downfall is that the characteristic length scale is likely determined by the smallest distance over which the function changes drastically, which means that extrapolation of smooth regions in the data may be difficult if there is a non-smooth region within the data.

Another group of kernel functions is the Matérn class of kernel functions [45],

$$k_{\text{Matern}}(r) = \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\frac{\sqrt{2\nu} r}{\ell} \right)^{\nu} K_\nu \left(\frac{\sqrt{2\nu} r}{\ell} \right), \tag{2.21}$$

With positive parameters ν and ℓ, and where Γ is the gamma function, and K_v is a modified Bessel function [48]. The Matén kernel functions are simplified when ν is

set to a half-integer, $\nu = p + 1/2$, where p is a non-negative integer. The general expression can be derived from [48], giving [45],

$$k_{\nu=p+1/2}(r) = \exp\left(-\frac{\sqrt{2\nu}r}{\ell}\right)\frac{\Gamma(p+1)}{\Gamma(2p+1)}\sum_{i=0}^{p}\frac{(p+i)!}{i!(p-i)!}\left(\frac{\sqrt{8\nu}r}{\ell}\right)^{p-i}. \quad (2.22)$$

Throughout this thesis, the Matérn 5/2 kernel is used, where $\nu = 5/2$ [45],

$$k_{\nu=5/2}(r) = \left(1 + \frac{\sqrt{5}r}{\ell} + \frac{5r^2}{3\ell^2}\right)\exp\left(-\frac{\sqrt{5}r}{\ell}\right) \quad (2.23)$$

ν allows us to control the differentiability and therefore tune the smoothness of the process. As $\nu \to \infty$ the Matérn kernel converges to the squared exponential function. Setting $\nu = 1/2$ results in a very rough process, and sets the assumption that the function is not differentiable. Setting $\nu = 5/2$, results in a twice differentiable smooth process similar to that of the squared exponential function, but unlike the squared exponential function, the process can be smooth without being limited by the characteristic length scale being too small. This feature has made the Matérn 5/2 kernel commonly used within the machine learning community and is why it is used throughout this thesis. Conversely, periodic or linear kernels are not used because we do not expect the locations of features which we search for when tuning a quantum device (e.g. Coulomb peaks or two-level fluctuations) to vary periodically or linearly within the voltage space.

The hyperparameters of the Gaussian process, such as σ_f^2 and ℓ can be inferred from data by finding the maximum a posteriori estimate (MAP) using a nonlinear optimizer and a prior over hyperparameter values. Alternatively one may find the maximum likelihood estimate (MLE), where given observations $f(x_{1:\,n})$ we calculate the likelihood of these observations under the prior $P(f(x_{1:n}|\eta))$, where η represents the hyperparameters of the process e.g. σ_f^2 and ℓ. Then we can calculate η that maximises the likelihood. Choosing one over the other depends on the situation, a

rule of thumb is to use to MLE when one has informative data, and MAP when one has informative data and/or priors.

2.7.6 Bayesian optimisation

Bayesian optimisation is a technique in machine learning used to efficiently find the global maximum (or minimum) of an unknown and expensive-to-evaluate objective function whose outputs can be noisy. Bayesian optimisation uses a probabilistic surrogate model, typically a Gaussian process, to approximate the objective function. Gaussian processes are chosen due to their ability to capture prediction uncertainty in their probabilistic estimate of the objective function.

Another function, the acquisition function, is used to determine where to sample the objective function next. The choice of acquisition function and its parameters balances exploration and exploitation based on the surrogate model's predictions and uncertainties. A commonly used acquisition function, relies on the *expected improvement* which is defined as [49],

$$\mathrm{EI}_n(x) \colon = E_n\left[[f(x) - f_n^*]^+\right], \tag{2.24}$$

where f_n^* is the largest observed value among the n times we have evaluated f so far. The *improvement* in the value of the best observed point after a new observation $f(x)$ is $f(x) - f_n^*$, if positive, and zero otherwise which we write as $[f(x) - f_n^*]^+$. $E_n[\cdot] = E_n[\cdot|x_{1:n}, y_{1:n}]$ is the *expectation* taken under the posterior distribution given function evaluations $f(x_1, ..., x_n)$. The posterior distribution is given by our Gaussian process surrogate model (Equation 2.17). The expected improvement can be evaluated as described by [50] resulting in [49],

$$\mathrm{EI}_n(x) = [\Delta_n(x)]^+ + \sigma_n(x)\varphi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right) - |\Delta_n(x)|\Phi\left(\frac{\Delta_n(x)}{\sigma_n(x)}\right), \tag{2.25}$$

where $\Delta_n(x) := \mu_n(x) - f_n^*$ is the expected difference between the proposed point x and the previous best, Φ is the cumulative distribution function of the standard normal distribution, φ is the probability density function of the standard normal distribution. Therefore, the Expected Improvement acquisition function [50] then evaluates f at the point with the largest expected improvement,

$$x_{n+1} = \mathrm{argmax}\mathrm{EI}_n(x). \tag{2.26}$$

Expected improvement is based on the assumption that we are only willing to return a previously evaluated point as our final solution, and the primary benefit of sampling occurs through an improvement at the point sampled [49]. Expected Improvement is commonly chosen as an acquisition function because it is simple to implement, inexpensive to evaluate and naturally balances exploration and exploitation evaluating points of high expected quality (large $\Delta_n(x)$) versus high uncertainty (large $\sigma_n(x)$).

In summary, the typical Bayesian optimisation workflow consists of initially evaluating the objective function at a series of random points. Then using the evaluated points, fit a Gaussian process (the surrogate model) to the objective function. Use an acquisition function for example, Expected Improvement, to determine where to evaluate the objective function next, and use the result to update the surrogate model. Repeat the steps of deciding where to evaluate, getting the result and updating the surrogate model until a stopping criterion is met.

Chapter 3

Literature Review

What is the story you want to tell?

Andrew Briggs

The story I want to tell begins with the desire for quantum computers (Chapter 1). A decision is then made to base the architectural building block of a quantum computer on semiconductor quantum devices in the argument of scalability. The experience the semiconductor industry has in the routine manufacture of semiconductor chips housing billions of transistors, allows us to envisage a world where we can scale from two qubit devices to hundreds of millions of qubits using similar manufacturing techniques and with a minimal footprint. Moreover, leveraging classical telecommunications techniques such as multiplexing, will enable us to address our 100 million qubit chip despite the limitations of the number of electrical lines that may fit within a dilution refrigerator. This story follows the scientific advances which support the reasoning behind this decision and lay a foundation for quantum computation. Automatic tuning procedures opens the door to tuning up quantum dots simultaneously rather than using manual sequential tuning procedures, a must-have as quantum devices grow and outpace human and manual capabilities. Unfortunately, device variability limits the ability for physics informed models to be applied with simple conditional logic as a general tuning procedure across different quantum devices, as such, we look to more

unsupervised methods. This is where machine learning (ML) is placed as a cornerstone. This will be the main focus of my literature review.

3.1 Semiconductor Spin Qubits

Initially, qubit development in semiconductors was performed in GaAs devices due to their low disorder and relaxed confinement requirements. In general, electron spin qubits in silicon are more challenging to realise due to the larger electron effective mass, thus smaller wavefunction. This results in more stringent lithographic fabrication requirements when compared to GaAs spin qubits. The characteristic size of a quantum dot in silicon is formed over areas of the order 10-20 nm [51]. Secondly, difficulties continue to arise due to the Si lattice symmetry and resulting conduction band valley degeneracy, leakage states may be thermally populated. Moreover, the valley splitting is affected by unavoidable fabrication defects, whether it be inhomogeneities at the oxide interfaces or even step edges in Si-nanowires or Si/SiGe heterostructures [51, 52]. Whereas GaAs quantum dots are typically formed over 100-300 nm sized areas [31, 41, 51], reducing fabrication precision requirements. Due to their large size, fabrication of GaAs quantum dots within university clean rooms is relatively easy which led to their early demonstration as qubits [41]. The GaAs devices are based on a heterostructure of GaAs and AlGaAs doped with Si to introduce free electrons. By stacking the GaAs and AlGaAs layers, free electrons accumulate at the AlGaAs/GaAs interface forming a two-dimensional electron gas. By tuning the voltages applied to gate electrodes patterned on the sample quantum dots can be formed and single electrons can be isolated. Many silicon devices such as metal-oxide-semiconductor (MOS) devices and heterostructures use gates in the same manner to define quantum dots. The parasitic nuclear spin bath within GaAs devices and the potential large-scale industrial fabrication of silicon devices have led to GaAs being left behind in favour of silicon. Here, I will briefly discuss reasons for moving to silicon devices and the types of silicon

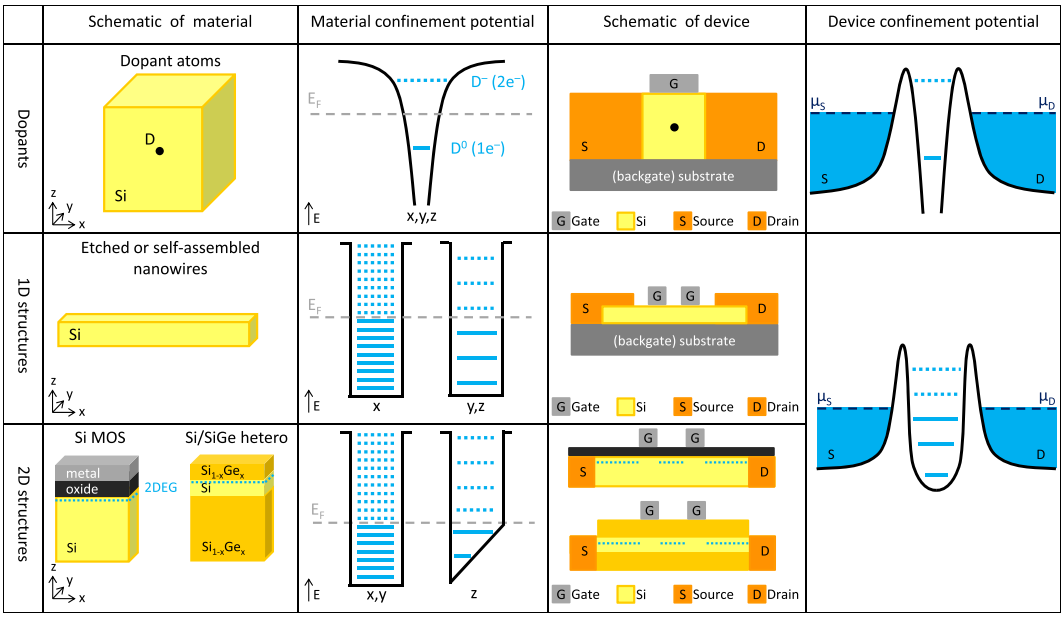


Figure 3.1: Schematic of silicon devices used for quantum dots. First column is a schematic of the materials. Second column: Confinement potentials experienced by electrons within the material. Occupied electron states are indicated by dashed blue lines and occupied electron states are denoted by solid lines up to the Fermi energy E_F. Third column: schematic of the device with gates and source and drain contacts. Fourth column: schematic of device band structure. Gate electrodes can raise/lower the electron energy occupation levels within the potential well relative to the source (μ_s)/drain (μ_d) to control the tunnelling of electrons/holes from the source to drain reservoirs. Reproduced from [52].

devices which are the main candidates for quantum computing.

3.1.1 Silicon devices

There are three main reasons for using silicon as the material of choice for semiconductor qubits. Firstly, it is the basis of the microelectronics industry. An industry which has led to the best silicon fabrication processes in the world only needs to devote a small fraction of its capabilities to accelerate quantum computer development. Secondly, it is possible to obtain a near-perfect clean magnetic environment. The lack of hyperfine interactions in purified ^{28}Si, the most common isotope, promise long spin coherence times [29, 53–56]. Finally, it is one of the most scalable architectures. This point is partly linked to the first where the manufacturing technology present lays a path

for quantum computer-on-a-chip-like architecture. Proposals of architectures which house millions of qubits [57, 58] have already been made along with error correcting codes [59, 60]. There are also three main device types, Figure 3.1, based on silicon: donors (e.g. ^{31}P dopants) in silicon, gate-defined dots in metal-oxide-semiconductors (MOS), and SiGe heterostructures. There are other implementations such as (near) one-dimensional structures including Si-FinFETs and Si nanowires. All types rely on electron (or hole) spins for qubits except donors in silicon which often rely on nuclear spins [29, 61] as well. Electron spin qubits are typically realised in Si/SiGe and Si-MOS devices. Holes spin qubits can be realised in Si-MOS as well as Si FinFET devices. Hole spin qubits are also realised in Ge/SiGe heterostructures and Ge/Si core/shell nanowires.

The difference between electrons and holes stretches beyond their charge; they are different in their spin(-orbit) properties. Due to the p-type Bloch wave function of valence band holes in Si and Ge, holes have a corresponding orbital angular moment quantum number $l = 1$. The implications of this are fairly large. The wavefunctions of holes have reduced overlap with nuclear sites in the host lattice resulting in a reduced contact hyperfine interaction and with it, reduced dephasing. Secondly, this results in the need to consider the total angular momentum operator $\boldsymbol{J} = \boldsymbol{L} + \boldsymbol{S}$, where $\boldsymbol{L}$ is the orbital momentum operator and $\boldsymbol{S}$ is the spin operator. This leads to the formation of heavy holes and light holes [52], the system of which is described by the Luttinger-Kohn Hamiltonian [62]. Moreover, the presence of strain and confinement (including applied electric fields), such as in a quantum dot in Si or Ge, can give rise to Rashba spin-orbit interaction [63]. The spin-orbit interaction is the coupling of the orbital and spin degree of freedom of a particle. The spin-orbit interaction facilitates a coupling of oscillating electric fields to the spin of the hole qubit enabling fast electrical control of a hole spin qubit using electric dipole spin resonance (EDSR). This, however, opens up the qubit to be sensitive to electrical noise. The electric tunability of the spin-orbit interaction is accentuated when confinement is limited down to one dimension, such

as in a nanowire, which is called direct-Rashba spin-orbit interaction [64, 65]. The direct-Rashba spin-orbit interaction offers non-monotonic electric tunability enabling one to turn on and off spin-orbit interaction for certain device geometries at finite electric fields. Therefore, this allows us to turn off the coupling between the qubit and the environment, and find sweet spots [66] for fast driving and low noise. Moreover, the effective mass of holes is less than that of electrons meaning that lithographic fabrication requirements are less stringent compared to electron spin quantum device counterparts. Despite these differences in terms of angular momentum, in the presence of strain and strong confinement such as in a quantum dot, we can typically describe hole spin states in the same manner as electrons [67].

Device types will be discussed in turn and their characteristics analysed as the building block of a quantum computer. However, before delving into different devices a brief background on electron transport in devices will be given.

Electron (or hole) transport in quantum dot devices occurs under a bias from source to drain. The current flow is controlled by a series of gate voltages. Gates which alter the electrochemical potential of the quantum dots are known as plunger gates. This is because they can 'plunge' (i.e. raise and lower) the electron (or hole) energy occupation levels within the potential well of the quantum dot relative to that of the source and drain. Tunnel barrier gates control the size of the potential barriers between quantum dots or a quantum dot and the source/drain. Therefore if the barrier potential is 'wide', tunnelling is forbidden. By sweeping the tunnel barrier gate voltages and measuring the current flowing through the device, the device parameter space can be split into two regions. There is a region of high current and a region of low/near-zero current. In the low/near-zero current region, the current flow is described as 'pinched-off'.

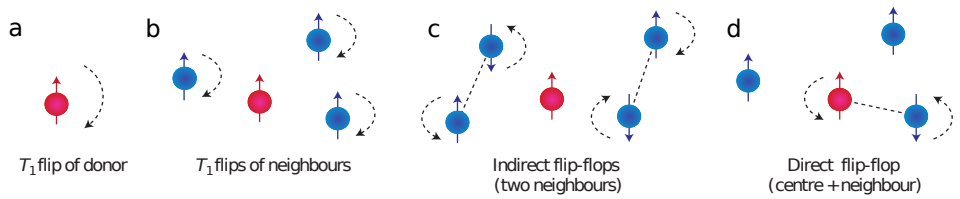


Figure 3.2: Diagram showing different types of electron spin relaxation and flip-flop mechanisms for donors in silicon. a) and b) correspond to (T_1) spin relaxations whether of a single electron donor spin or that induced by nearby electron donor spins relaxing respectively. c) Indirect flip-flop, decoherence of a central spin due to spin flip-flops in neighbouring donor pairs can also occur. Fluctuating fields are produced by neighbouring spin (blue) flips and result in dephasing of the central spin (pink). d) Direct flip-flop, the central spin is decohered by being involved directly in a flip-flop event with a neighbouring spin. T_1 relaxation process dominates donor electron decoherence at temperatures above 8 K (a). Below 4 K T_2 times are dominated by indirect flip-flop processes (c). At transition temperatures from 4-8 K induced spin flips from neighbours dominate (b). Reproduced from [54].

Donors in silicon

Donors in silicon [68] hold some of the longest coherence times, 0.5s and 35.6s demonstrated on electron spins and ^{31}P nuclear spins respectively, in silicon or that of any physical implementation [29]. This is often argued to be due to the excellent potential confinement of the electron in the bulk material resultant from the symmetrical field created by the ^{31}P impurity. Whereas non-uniform interfaces found in MOS and heterostructures can create random fluctuations in confinement potentials; Muhonen *et al.* [29] thoroughly demonstrates that Si/SiO_2 interfaces are not the main source of decoherence for electron spin qubits in donors in silicon devices. A reasonable assumption is made that paramagnetic spin noise is negligible due to the low temperature of 100 mK and the high 1.5 T applied field greatly reduces any paramagnetic spin fluctuations. All possible different sources of noise were considered and it was verified successfully that the main noise source was external to the sample across two different devices [29]. This is a solvable engineering challenge to increase electron spin coherence times but the quantitative effect on coherence times if solved is not discussed by Muhonen *et al.* [29]. Despite demonstrations of high fidelities [61], in some cases exceeding 99.99% [29] there is still the challenge of manufactura-

bility of devices which is not addressed by Muhonen *et al.* [29]. Even with precise placement of ^{31}P donors in silicon, there will be interactions between electron spins and neighbouring donor electron spins leading to decoherence of the electron spin. There are proposals for how these interactions can be disregarded. Tyryshkin *et al.* [54] does well to explain the different types of interactions between donor electron spins at different temperatures in silicon, Figure 3.2. Furthermore, it is recognised that, at transition temperatures between spin interaction regimes, there is a third spin interaction in effect which is not a linear combination of the interactions occurring at the temperature extremes [54]. Reducing donor spin interactions by applying a magnetic field gradient (10 μT mm^{-1}) to increase the resonance offset between nearby spin donor pairs is briefly discussed [54]. Recently, the manufacture of donor spin qubits has included implanting donor spins as molecules such as PF_2, due to the finer control over the implantation depth with a heavier molecule [69]. Once the PF_2 molecule is ion-implanted into the silicon lattice, the sample is annealed, after which the ^{31}P nucleus remains in place and the fluorine diffuses away to the surface. Holmes *et al.* [69] correctly show that no fluorine molecules are present and coupled to the ^{31}P donor spin nucleus through ESR spectra alongside nuclear magnetic resonant pulses (NMR) pulses resonant with the fluorine nucleus. Holmes *et al.* [69] unfortunately observe a ^{29}Si impurity coupling to the implanted ^{31}P despite isotopic purification of the silicon lattice. This emphasises that even with precise placement of ion-implanted donor ions, including via molecules, nano-apertures and ion-detection [70], they must reside in a magnetically clean environment with very few impurities, such as ^{29}Si atoms that will lead to qubit decoherence. Donor in silicon devices continue to demonstrate their exquisite capabilities from the single-shot readout of an electron spin in 2010 [40] to the demonstration of achieving error correction threshold fidelities in 2022 [71]. Replacing the ^{31}P nucleus with a ^{123}Sb ion opens the door to realising *qudits*, qubits with a state space with dimensions greater than 2, from the $7/2$ ^{123}Sb nucleus spin [72, 73].

Quantum dots in MOS

The main argument for quantum dots in metal-oxide-semiconductor (MOS) devices is that of scalability in manufacturing. They are the most similar implementation to that of current MOS technology. They do not require atomic precision placement of atoms but, do require reproducible feature size and quality. Those features include gate electrodes which form quantum dots via applying a voltage to these gates, inducing an electric field in the silicon. Multiple gate electrodes per quantum dot provide the additional benefit of being able to tune different quantum dot devices to similar operation regimes despite discrepancies in manufacturing quality control which there will be at this early stage of development. Maurand *et al.* [74] demonstrates the ability that MOS devices have in creating spin qubits and displaying spin effects such as Pauli spin blockade. However, the coherence times are disappointingly short at 245 ns via Hahn echo with very little explanation of why this is the case. Proposed reasons point to impurities in the material playing the role as a reminder of how important quality control of the fabrication process will be in these devices. Veldhorst *et al.* [75] removes any doubts in MOS devices with coherence times of 28 ms using a Carr-Purcell-Meiboom-Gill (CPMG) pulse sequence, a pulse sequence capable of reducing the effects of spin dephasing. This is accordingly compared to 200 μs CPMG coherence times achieved in GaAs [76]. Reasonable proposals such as low frequency noise from the superconducting magnet, are put forward for the sources of decoherence but it is acknowledged that further experiments are necessary to confirm this. Once again the argument of scalability of MOS devices is presented by but more importantly is the demonstration of a two qubit logic gate by Veldhorst *et al.* [28] two years later. This delivers on the requirement set out by DiVincenzo [4] for quantum computation and later thought-out in spin qubits by Loss and DiVincenzo [36]. A serious consideration of architecture for scaling silicon MOS spin qubits is presented by Veldhorst *et al.* [57]. One of the key acknowledgements by the author, despite their inherent biases, is that there is not a clear-cut path to the scaling of silicon MOS

qubits despite what many may claim in the literature. This is particularly down to the minimum feature size, the separation between gates, set at $\sim$ 7 nm [57]. This would lead to an area of $\sim$ 63 x 63 nm^2 per qubit [57], and if we take a module to carry 480 qubits, as per the architecture laid out by Veldhorst *et al.* [57], for a one million qubit computer it can get quite large fairly quickly. In addition to this there would need to be appropriate space to house classical computer transistors to aid with the controlling of the gate voltages and readout of the qubits. Housing this circuitry will most likely require further advancements in current manufacturing technology to minimise thermal and volumetric impact on the qubit module. If we consider heat dissipation from the control electronics catering to each module it will be difficult to maintain a consistent temperature of 20 mK. This issue is roughly addressed by Veldhorst *et al.* [57] and estimates that the module could operate at 100 mK in a typical dilution fridge. But it is acknowledged that this could be a possible bottleneck for scalability and may rely on silicon spin qubits to perform at higher temperatures as previously demonstrated [77] to provide scalability. Yang *et al.* [78] goes on to operate two qubits at 1.5 K in a silicon complementary metal–oxide–semiconductor (CMOS) device with coherence times of 2 ms and fidelities of 98.6%. It is appropriately shown how these qubits would fit into the architecture envisaged by Veldhorst *et al.* [57] but still operate at temperatures that could be provided by a pumped ^{4}He system. This greatly reduces the costs and engineering difficulties from thermal management in the scaling of MOS-based qubits.

Quantum dots in heterostructures

One of the attractions to SiGe heterostructures is the lack of the amorphous Si/SiO_2 interface which causes charge defects in MOS devices. 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 [79]. The high tunability of the SiGe dots is demonstrated by Lawrie *et al.* [80] 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 dot cross capacitances between the different devices. Although the devices are different in terms of layout they have much lower cross capacitances than that seen in GaAs devices and the Si-MOS device shows the lowest, facilitating their operation. Lawrie *et al.* are fair in their discussion, presenting the challenges which face heterostructures for scaling devices such as automated tuning and wiring logistics for each gate with supporting work that looks to help solve these problems [81, 82]. Xu *et al.* [83] 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 dynamic random-access memory (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 [83]. Xu *et al.* [83] verifies that the capacitor storing the dot gate potential doesn't affect the gate pulses. This is shown by performing electronic circuit simulations which allow for pulses up to 20 GHz frequencies. Alternative gate geometries can be considered instead. Borsoi *et al.* [84] demonstrates a 16 (4 x 4) quantum dot array in a Ge/SiGe heterostructure, with a single gate electrode addressing more than one quantum dot at a time. Heterostructure devices have firmly planted their foot down as a contending architecture with not only the demonstration of qubit arrays [85] but also the achievement of error-correcting threshold fidelities [86, 87] alongside their

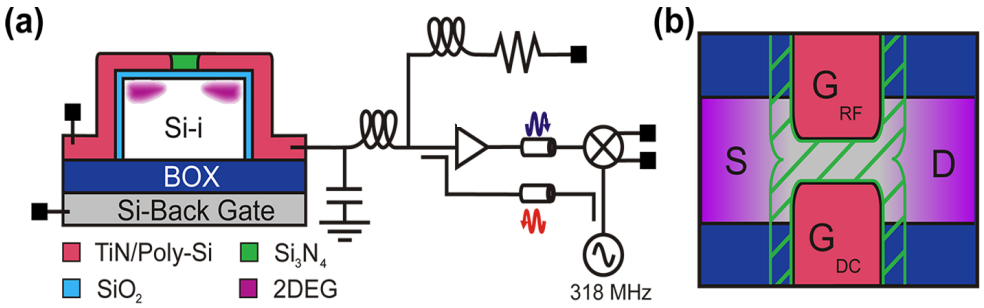


Figure 3.3: a) Nanowire device cross section and connected reflectometry setup for readout. The electrons or holes are localised at the top corners of the device. b) Overhead digram of nanowire device, the Si_3N_4 (hatched green) spaces apart the two top gates G_{DC} and G_{RF}. Reproduced from [91].

ion-implanted donor in silicon counterparts [71].

Others: nanowires & FinFETs

Nanowires can be composed of multiple materials such as a 10 nm diameter germanium core and a 2.5 nm silicon shell [88]. A hole gas is formed within the Ge core and can be depleted through the application of positive voltages to the gate electrodes, above which the nanowire is suspended. Froning *et al.* [88] successfully demonstrates the ability to form single, double and triple quantum dots within a 5-gate Ge/Si Core/Shell nanowire opening the door to spin qubit experiments. Leveraging the electrically tunable g-factor of holes in germanium, Froning *et al.* [90] later successfully demonstrates complete electrical control of hole spin qubits in Ge/Si core/shell nanowires. Froning *et al.* [90] also shows the ability to tune the hole qubit Rabi frequency, the spin rotation rate, by an order of magnitude with only a change of millivolts in gate electrode voltages.

1D structures such as silicon nanowires offer a compact way of achieving gate-defined quantum dots while still being compatible with large-scale manufacture. Similar to MOS devices in their fabrication the gates reside above the nanowire (Fig. 3.3) but are split at the top so a pair of gates resides along the length of the nanowire. Betz *et al.* [91] explains that the quantum dots reside in the top corners of the

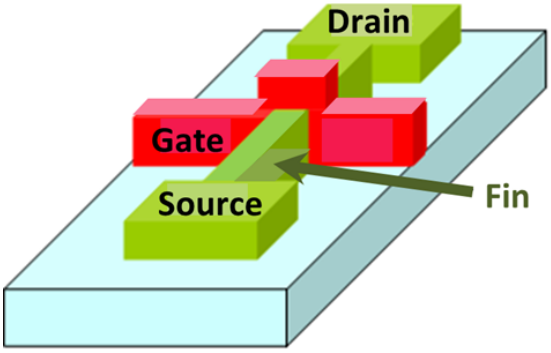


Figure 3.4: Schematic of FinFET gate structure. The gates wrap around three sides of the 'fin' which consists of a silicon channel. The wrap around gate results in a more intense electric field than in a planar MOS device enabling high control of current flow at shorter channel lengths allowing faster switching speeds.

nanowire as that is where the electric field from the gates is the strongest. Betz *et al.* goes on to demonstrate Pauli spin blockade, placing the device on the ladder as a contender for semiconductor qubits. Betz *et al.* is able to demonstrate this using radio-frequency (RF) reflectometry measurements [92]. This future proofs the device, as RF measurements allow for scalable readout via multiplexing techniques. Multiple quantum dots along the length of the silicon channel were seen recently in work by Ansaloni *et al.* [93]. Measuring each individual quantum dot in current transport with a local charge sensor is cumbersome, and source to drain currents only flow in very small regions of the parameter space.

Si FinFETs [94] appear as siblings to nanowires as their geometry is very similar, Figure 3.4. The main difference is that the top gates in a FinFET wrap around the silicon channel, the 'fin'. FinFETs are one of the largest breakthroughs in the semiconducting industry since the invention of the metal-oxide-semiconductor field-effect transistor (MOSFET) in the 1960s. The wrap-around gate results in a more intense electric field than in a planar MOS device enabling high control of current flow at shorter channel lengths allowing faster switching speeds. The intense electric field from the wrap-around gates is taken advantage of, defining a quantum dot with a single gate. Kuhlmann *et al.* [95] intelligently demonstrates both electron and hole quantum dots in the same FinFET transistor. The argument for an ambipolar device

is flexibility in circuitry with the ability to benchmark the performance of holes against electron quantum dots. Work by Kuhlmann *et al.* supports earlier claims of tunability and control of quantum dots in FinFETs [96] in a compact package. FinFET's place as a qubit contender is solidified by Camenzind *et al.* [38] demonstrating hole spin qubits in a FinFET at 4 kelvin. Camenzind *et al.* [38] understandably compares hole spin qubits to their hot electron counterparts in other device architectures [78, 97], and claims superiority on the basis of Rabi quality factor, a measure of the spin oscillation rate multiplied by the decay time of the spin oscillations. Delivering on the promise of scalability we are starting to see linear qubit arrays realised in FinFETs produced by advanced semiconductor manufacturing techniques [98]. Recent work on the manufacture of FinFETs shows that there is still, "plenty of room at the bottom" [99]. These industry-standard devices can be made even smaller (single atom thick) [100] ingraining them in the field as contenders for semiconductor qubits.

3.1.2 Readout

Briefly, qubit readout techniques namely, transport measurements and RF reflectometry will be discussed.

The qubit state is read out by spin-to-charge conversion [31]. One could measure the current flowing through the device to detect this, but the current would be so small that one has to measure a series of electrons over time (multiple-shot). For a functional quantum computer, reading out the state of qubits will need to be single-shot, accurate and fast [4]. For the sake of scalability, the physical readout implementation must also be compact. RF reflectometry [92] possess all of these key traits. It is compact as one can use the same gates which are used to tune quantum dots to measure them [91, 101] thus no extra gates are needed. Its speed is demonstrated by Botzem *et al.* [101] who uses RF reflectometry to tune quantum dots in GaAs and benchmarks it against typical current transport measurements performing orders of magnitude faster. The low integration times necessary for RF reflectometry enables single-shot readout [102].

3.1.3 Discussion

There is a large range of silicon devices and many of which are suitable for scalable quantum computing. Gate-defined quantum dot devices are attractive for numerous reasons but particularly their compatibility with industry standard manufacturing techniques and flexible tuning of quantum dots. Despite fabrication excellencies, there will always be defects which will alter the specific voltages required to define a dot from one device to the next. Automation will be required to tune quantum dots such that one can finally turn on a quantum computer with a million qubits.

3.2 Tuning Quantum Devices

Automated tuning of quantum dot devices is at a stage of very early development. There are few groups who publish work on it, and those that do, offer patch solutions. A complete algorithm which can take you from a 'fresh' device to an array of prepared qubits is currently unavailable. However, light is beginning to shine through the small pile of existing automation algorithms, inspiring those in the field to combine together what is left in the shadows and build greater self-operating tools.

There are multiple algorithms [81, 103, 104] which do take us from a 'fresh' device to coarsely defined double quantum dots which is one of the initial automation challenges of the field and possibly one of the most helpful in the lab. These algorithms not only save experimentalists time but are necessary steps towards the scaling of semiconductor qubits and 'turning on' a quantum computer. Standard distributed control approaches have not been used for tuning because there are no forward dynamics models available for tuning quantum devices. However, we are starting to find ways to bridge the gap between reality and models of quantum devices using machine learning [105]. In the following sections, I will introduce a range of tuning algorithms, some follow a series of simple conditional steps, and others involve machine learning in their workflow.

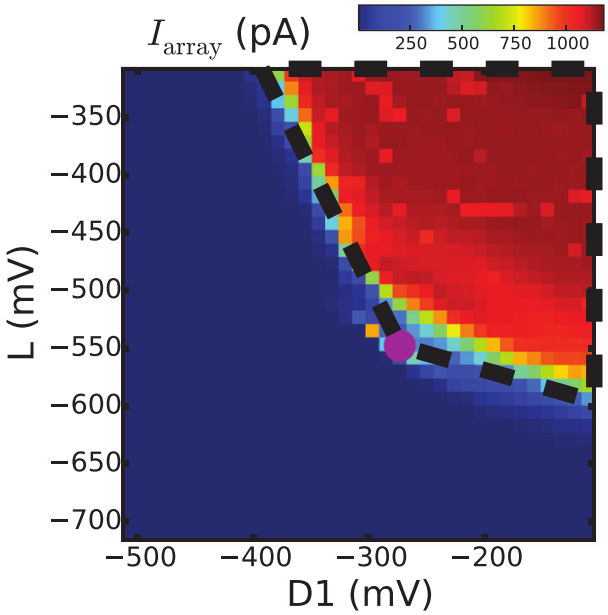


Figure 3.5: A current map generated by sweeping the adjacent tunnel barrier gate voltages L and D1 while measuring the current I_{array} flowing from the source to the drain of a linear quantum dot array device. The black dashed line identifies the border between regions of high current and low current. The purple dot marks a promising location to look for single dot features such as Coulomb peaks. These two steps become key components in later auto-tuning quantum dot algorithms. Reproduced from [81].

3.2.1 Tuning without machine learning

When creating an automation algorithm one can take many pathways in designing its workflow. The first and most common choice is a human-like approach [81, 104], in other words copy the human workflow and put it into code that communicates with the required instruments. An initial approach Baart *et al.* [81] involved no machine learning and relied heavily on previous knowledge of the device to coarsely tune two quantum dots from a linear four dot array. By measuring current traces the pinch-off for each gate is found. Then single dots are tuned by using plunger gates whilst relying on previous knowledge of appropriate voltages for this device. Baart *et al.*'s [81] algorithm is not general because it relies on familiarity of the device at hand but, it is foundational work. Baart *et al.* [81] have successfully identified key components for future tuning algorithms which are, locating pinch-off, searching for Coulomb peaks

and mapping out the surface that separates regions of high current from low current, Figure 3.5. Volk *et al.* [106] presents a "N+1" method of tuning a linear array of eight qubits. Similar to Baart *et al.* each dot is tuned sequentially with a charge sensor whilst noting the cross capacitances between all of the gates. The cross capacitances are then used to generate 'virtual gates' which give the ability to group gates such that we can change the charge state of each quantum dot independently without affecting neighbouring quantum dots. This is repeated for each dot. The algorithm in [106] lends itself well however to one type of device, that is linear arrays. As seen in Section 3.1.1, devices may take a range of shapes and layouts and it will be important to have a general algorithm that can be applied to all of them. The reasoning behind this narrative is that a quantum computer based on spins in semiconductors may be composed of different devices architectures serving different purposes for example, one device type for the processor and another for memory.

3.2.2 Tuning with machine learning

Machine learning can be used to play the role of the human by verifying the steps taken by the algorithm such as work by Darulová *et al.* [104]. For example, a Random Forest classifier is used to determine whether pinch-off occurs for individual gates, rather than a human looking at each current trace for each gate. If a single gate does not demonstrate pinch-off then the device is considered untunable and useless. Arguably the classifier is generalised by choosing only particular features from fitting measured current traces, such as pinch-off voltage and the change in voltage at pinch-off, rather than the whole trace as a feature set. This means one can take traces with different current values and still use the classifier without encountering feature dimension problems. After all the gates have been characterised, the device is then tuned by first sweeping the gates one by one, the central-barrier, inner-barriers, outer-barriers and plungers in that order. 1D-current traces are taken along the way to determine if there are single electron transport features. When discovered a 2D current

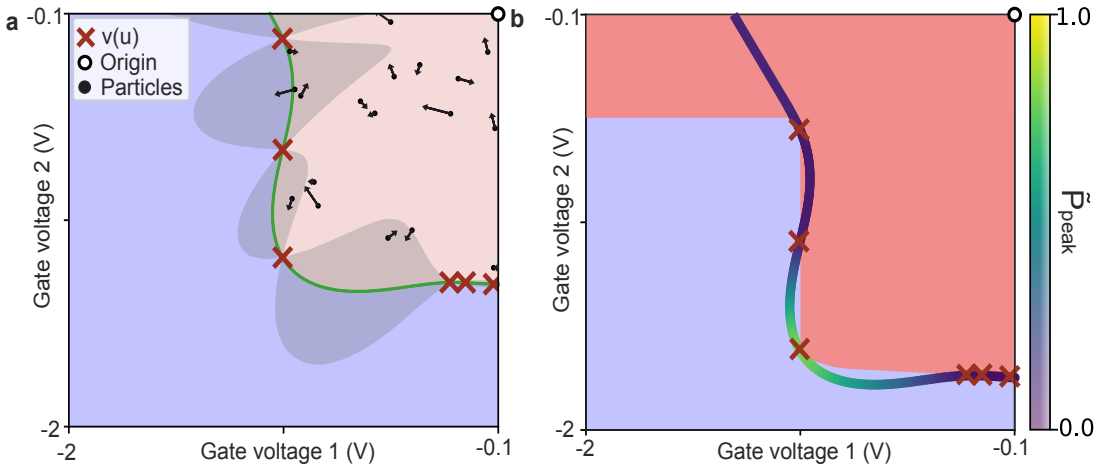


Figure 3.6: Sampling phase of double dot tuning algorithm. Schematic current maps of the device, pink signifies regions of high current, blue signifies regions of relatively low current. Gate voltage space is limited to two dimensions for illustration. Red 'x's are points in the voltage space on the hypersurface which have been sampled, $v(u)$. a) The green line signifies the model hypersurface, with uncertainties from the Gaussian process model (grey), which separates the region of high/non-zero current from low/near-zero current. Simulated particles undergoing Brownian motion (black) are used to sample the hypersurface randomly. b) The colour gradient of the line signifies the probability of finding Coulomb peaks at the model hypersurface. Reproduced and edited from [103].

map is taken as a function of the plunger gate voltages. The output current map is classified by a multi-layered perceptron neural network for the existence of good/bad double dot features. This is very efficient in time because it minimises the number of 2D-current maps taken by the algorithm. Unfortunately, the time taken to tune a double dot successfully is not clearly stated in the article. The algorithm is restricted despite claims of generality because it relies on knowledge of the gate architecture for both characterisation and tuning stages. Therefore, for this algorithm to be used on other devices the tuning procedure would have to be modified and most likely rely on a typical human workflow for guidance. In addition, the field is moving away from the traditional gate architectures seen in older GaAs devices and more towards foundry fabricated silicon devices [93] to favour manufacturability. Respectfully, Darulová *et al.* [104] tried multiple machine learning techniques for both characterising gates and classifying double dot features and used those which produced the best results.

A second approach is to use available mathematical and A.I. tools, Moon *et al.* [103] to become faster than humans in the hope of finding possible shortcuts to obtaining an end result. By removing the typical human workflow one can create a more general algorithm which can be applied to a range of devices as it does not rely on human habits and knowledge from previous attempts at tuning similar devices. When tuning quantum dots by relying on transport features there are particular gate voltage values which split the device parameter into two regions, relatively high current flowing and relatively low current flowing (pinched-off). These voltage values define a hypersurface in an N-dimensional space where N is the number of gates that control the flow of current through the device, typically N is the the range of 3 to 7 for double quantum dot devices. This hypersurface is usually located by performing 1D current traces with each gate until pinch-off is reached [81, 104]. However, one can speed up the process by randomly sampling points in voltage space and measuring the current [103] in that direction from an arbitrarily assigned origin. After multiple points (at least 30) on the hypersurface have been located then using a Gaussian process model, a prediction of the hypersurface can be built, Figure 3.6. As previously identified [81], points in voltage space on this hypersurface and near to it in the pinch-off region often display single dot features identifiable as Coulomb peaks in a 1D current trace. If Coulomb peaks are present then a low resolution 2D-current map is taken using the plunger gates of the device, and this map is scored against the expected 'honeycomb lattice' characteristic of a double dot regime. If the score is above a predetermined threshold then this regime of voltages is believed to contain a double dot regime and a high resolution scan is taken by the algorithm for viewing in post by a human. This process of sampling the hypersurface and investigating the regions around it is repeated. In each iteration, the information from the investigation stage (are Coulomb peaks present?) is fed back into the Gaussian process model of the hypersurface. This results in a better prediction of the hypersurface and the best location to sample the hypersurface next based on the probability of finding Coulomb

peaks (see Chapter 2.7.6 for Bayesian optimisation). To reduce tuning times the number of 2D-current maps are minimised by relying on targeted sampling of the hypersurface and accurate classification of Coulomb Peaks in 1D traces. However, the accuracy of the Coulomb peak identification method used is not clear [103]. This may not be as easy as identifying pinch-off [104] gate voltages for different devices. Thus, this could be a point of clarification and an improvement to generalise an algorithm which is already one of the most general as it requires no knowledge of device gate architecture for the hypersurface sampling stage [103]. Moreover, the computational complexity of training Gaussian process models and data sparsity [107] are issues in scaling Gaussian processes to higher dimensions [108, 109], therefore techniques such as dimensionality reduction [110, 111] may be required as devices increase in size and N grows. The algorithm only requires knowledge of the plunger gate identities to produce 2D charge stability diagrams (current maps) in the investigation stage. The performance of the algorithm is demonstrated by tuning two double dot GaAs devices. A point set registration analysis which maps different hypersurfaces to each other, is performed on the effect of thermally cycling two different devices. Each device had 8 gate electrodes but on one of the devices leakage currents were associated with a gate. This gate was set at 0V and excluded from the tuning algorithm. This further proves generalisation as the algorithm appears to be able to tune devices regardless of the number of gates or whether all gates are functioning unlike other contemporaries which would classify such a device as untunable [104]. Using the algorithm these double dots can be tuned in 70 minutes, arguably faster than humans, when bench marked against tuning times of human experts (3 hours) and even more so a pure random search algorithm (680 hours) [103]. This algorithm [103] was demonstrated in a GaAs device where electrons were charge carriers. In Chapter 5 I demonstrate an algorithm that tunes double quantum dots in three different architectures where holes are charge carriers. Moreover, the algorithm successfully tunes the devices regardless if their mode of operation is depletion or accumulation of charges, by changing the starting

point of the algorithm in the gate voltage space bounds and searching for pinch-off rather than turn on. In the same way that there are different manual approaches to tuning a device for example searching for turn-on instead of pinch-off, automatic tuning approaches can be built to cater to different device types by searching for different features or being adaptable in their search given some prior knowledge such as mode of operation or voltage bounds.

Quantum device fine tuning

Once quantum dots have been roughly tuned to form a double quantum dot, these dots must be finely tuned. This has the aim of optimising a set of charge transitions for qubit operation. Van Esbroeck *et al.* [112] has produced some of the earliest work in the field of automated fine tuning relying on a variational auto encoder to optimise tunnel rates and inter-dot coupling in GaAs. This is done by optimising the shape of bias triangles, transport features corresponding to double dot regimes, Figure 3.7. The algorithm requires no knowledge of the device architecture but it has only been demonstrated tuning three pre-selected gates out of the eight gates. Although the coupling of the additional gates to the bias triangles is described as 'weak' it would have been ideal to see how the algorithm would have performed with more gates under its control.

Suppose my double quantum dots have been finely tuned, yet I am in an unknown charge state and I would like to acquire a particular one. A charge state signifies the number of electrons present in each quantum dot. This pre-qubit tuning is necessary for qubit operations. An algorithm presented by Durrer *et al.* [113] relies on convolutional neural networks to recognise in a series of low resolution 2D current maps when the zero charge state is reached and where the charge transition lines are crossed. Despite the neural nets being trained on 10^5 (after data augmentation) labelled current maps to identify (0,0) charge states and charge transition lines, the algorithm is successful in locating the desired charge state 57% of the time. One could

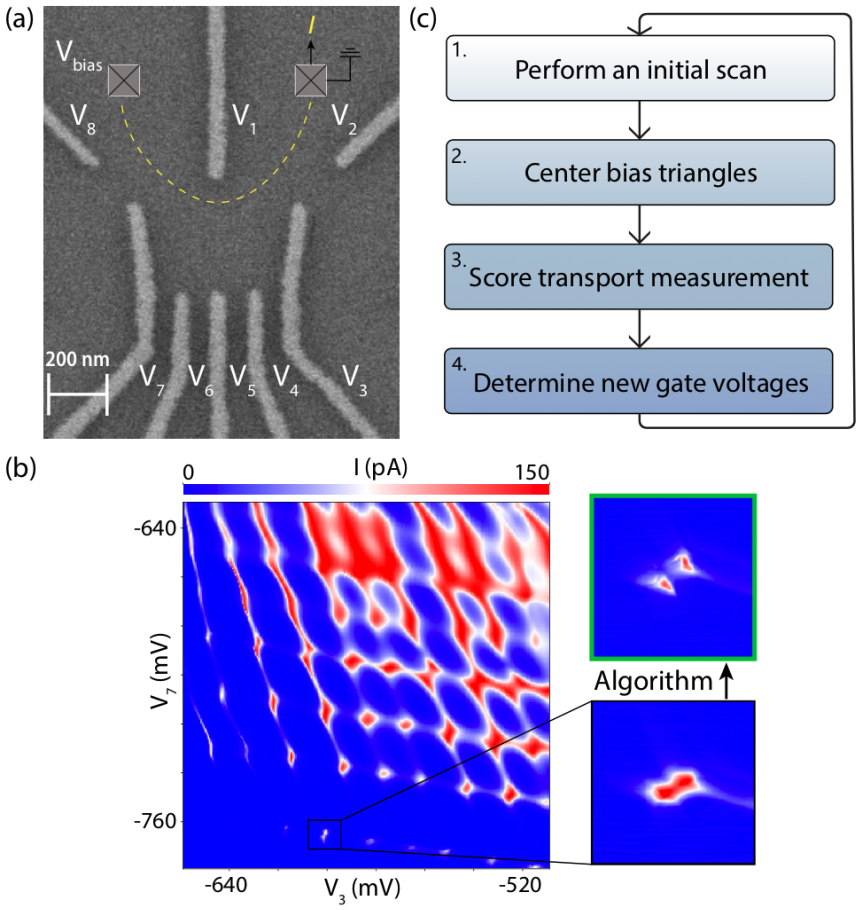


Figure 3.7: Quantum device fine tuning. a) SEM image of device gate architecture. b) Current map showing double dot features, inset: the before and after of the algorithm fine tuning a bias triangle. c) Workflow diagram of algorithm. Reproduced from [112].

argue that this is due to poor signal to noise ratio in the experiment. If this is the case, the neural nets will have to be trained and tested on higher resolution current maps to find the appropriate charge state. The time to take higher resolution current maps will drastically increase the time to locate a given charge state. Unfortunately the time taken by the algorithm to find a desired charge state is not clear, and it is not bench-marked against other approaches. These approaches could include locating the (0,0) charge state by taking a series of 1D current traces and identify when Coulomb peaks are no longer observed. This would certainly be faster and simpler than neural net image recognition techniques and may lead to the same if not greater accuracy. If one chose to remain on the neural net route then, more targeted sampling of the

voltage space [114] could be employed. This would reduce the number of 2D current maps required and provide the time for higher resolution current maps to be acquired, increasing the signal to noise ratio and the accuracy of the neural network. However, Durrer *et al.* [113] has been able to overcome a challenge that Kalantre *et al.* [82] deemed difficult. That is, locating desired charge states using neural networks and Durrer *et al.*'s method simply relies on the calibration step of finding the (0,0) charge state [113]. Kalantre *et al.* [82] was one of the earliest to apply machine learning to identify double quantum dots and single quantum dots albeit using a convolutional neural network trained on simulated data based on the Thomas-Fermi model. They proved this method was successful with tests on real data from a silicon nanowire with over 90% accuracy. Furthermore, Kalantre *et al.* goes on to outline a method of auto-tuning a quantum dot device using neural networks which is similar to the approach taken by Durrer *et al.* [113].

The majority of these automated tuning algorithms have been tested in well-established GaAs devices. There is space therefore to develop and demonstrate automated quantum dot tuning algorithms for a range of silicon devices. The broad range of silicon devices and the less established device fabrication techniques place a high bar for the successful demonstration of a general tuning algorithm in these devices.

3.3 Conclusion

In conclusion, the literature supports the use of silicon over other materials such as GaAs due to the possibility of leveraging industry fabrication expertise and a low nuclear spin environment due to isotropic purification resulting in reduced dephasing. The ability to leverage industry fabrication expertise is most applicable to Si-MOS devices rather than heterostructures such as Ge/SiGe or Si/SiGe. The scalable and reliable manufacture of silicon-based devices is caught on the stringent lithographic

fabrication requirements for electron spin qubit devices. However, these requirements are relaxed if hole spin qubits are used in place of electrons and if the host material is Germanium due to the relative effective masses of charge carriers. Automatic tuning algorithms and reliable and reproducible quantum dot fabrication will need to meet each other in the middle of the scaling journey of spin qubit quantum architectures. All devices will require automated tuning of gate voltages to reliably form arrays of quantum dots and to achieve optimal qubit control. However, the burden on the automatic tuning algorithm is reduced for well fabricated and reliable quantum dots, in which case, a capacitance model may be suitable for guiding automated tune-up. A range of algorithms involving machine learning have been developed but do not provide a complete tuning solution, let alone demonstrate fully automated tuning in a range of silicon devices. Moreover, few algorithms in existent are arguably capable of tuning extended quantum dot arrays in their current form without significant modification. This is especially true, as readout methods of larger quantum arrays are beginning to come to fruition as device size grows. For prototype devices such as double quantum dots, the literature shows that algorithms which use machine learning may be the most successful in tuning them. In this thesis, I explore the automatic tune-up of prototype devices using more unsupervised methods which involve machine learning.

Chapter 4

Methodology

> People pay their mortgages and feed their children by doing this stuff.
>
> Andrea Morello

The experiments which led to the results presented in this thesis were carried out across a range of laboratories. Algorithms were run remotely in the Zumbül Group, University of Basel and Katsaros Group, IST Austria during the COVID-19 pandemic to acquire the results shown in Chapter 5. The Algorithms used in Chapters 6 and 7 were developed at both the University of Oxford and the University of New South Wales (UNSW). The experiments in Chapters 6 and 7 were completed in the Fundamental Quantum Technologies (FQT) laboratory of Prof. Andrea Morello and the Diraq laboratory at the ARC Centre of Excellence for Quantum Computation and Communication Technology in UNSW, Sydney, Australia. Here I discuss the methods commonly used across the laboratories I worked with/in to carry out experiments to acquire the data presented within this thesis. This will include a discussion on sample preparation, control electronics, software interfaces, algorithmic control and building software.

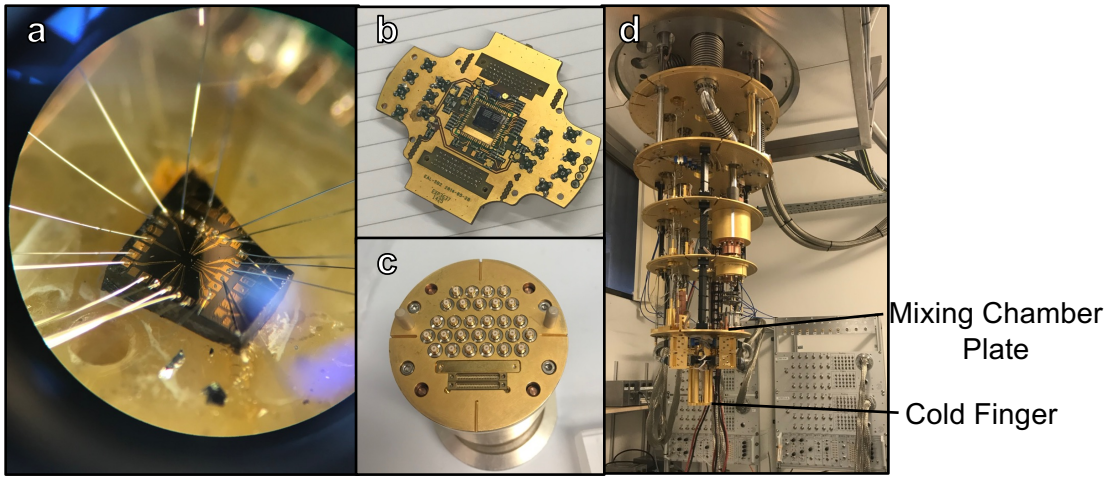


Figure 4.1: Sample preparation. a) A semiconductor chip sample viewed from an optical microscope. Wire bonds (aluminium wires), create an electrical contact between the sample's gate-electrodes, source and drain contacts to the bonding pads of the printed circuit board (PCB). b) The PCB (pictured) behaves as an electrical interface between the semiconductor chip and the electrical lines used to send signals to control the sample. c) A cylindrical sample enclosure pictured from the top where electrical lines enter the enclosure to reach the sample via nano-D and SMP connectors. d) An open dilution refrigerator where the shields and outer vacuum can have been removed. Locations where the sample can be bolted for quantum transport experiments are labelled. The cylindrical sample enclosure (c) is bolted to the Cold Finger which can reach a base temperature of 20 mK. Alternative sample enclosures can be bolted to the Mixing Chamber Plate and can reach a temperature of 20 mK.

4.1 Sample preparation

The semiconductor chip sample (Fig. 4.1a) is mounted onto a printed circuit board (PCB), also known as a 'sample board' using PMMA as an adhesive. The PCB is the interface between the signals sent from the external control electronics and the sample (Fig. 4.1b). The gate-electrodes of the sample are connected to the control electronics via wire bonds which provide an electrical connection between the gate-electrodes and the relevant bond pads on the PCB. There are different types of bond pads on the PCB, those that are for DC signals, and those that correspond to signals that can be either AC or DC from the respective control electronic. Bond pad features are taken into consideration when planning the orientation of the sample on the PCB and the respective wire bonds between gate electrodes and PCB bond pads.

As discussed in Chapter 2 it is important that electron temperature and therefore

the device temperature is less than the charging energy of the quantum dot. To achieve this and therefore the millikelvin temperatures required, the sample is placed within a dilution refrigerator. This is done by placing the PCB, holding the mounted sample, within a sample enclosure (Fig. 4.1c) which is then firmly bolted to the coldest stage of the dilution refrigerator, the Cold Finger (Fig. 4.1d). The Cold Finger typically has a base temperature of 20 mK. Samples can also be bolted to the Mixing Chamber Plate which has a temperature of 20 mK when the fridge is at base temperature. The sample enclosure must have excellent thermal contact with the dilution refrigerator to maximise thermalisation. The shape and mounting process of the sample enclosure depends on the style of the dilution refrigerator. For example, within my lab at the University of Oxford, we used Oxford Instruments dilution refrigerators which are compatible with a cylindrical sample enclosure called a *Puck*. Whereas in the Fundamental Quantum Technologies lab in Sydney, I used cuboidal sample enclosures which were compatible with their Blue Fors branded dilution refrigerators. Control electronics are then connected to the sample enclosure during the loading process into the dilution refrigerator which, in turn, connects to the PCB and then finally to the device.

4.2 Classical electronics

The brands of breakout boxes, voltage sources and DACs can vary greatly between laboratories and individual experimental setups. Particular groups will have preferred features, such as affordability, ease of use, or brand familiarity and loyalty. In some cases, laboratories will build their equipment, for example, the homemade breakout boxes used by the FQT lab. Despite subtle differences, the equipment serves the same purpose and I will briefly outline the core features using the QTRay Racks manufactured by TU Delft as an example which are common across laboratories in which work for my thesis was carried out. Moreover, the layout of the instrument

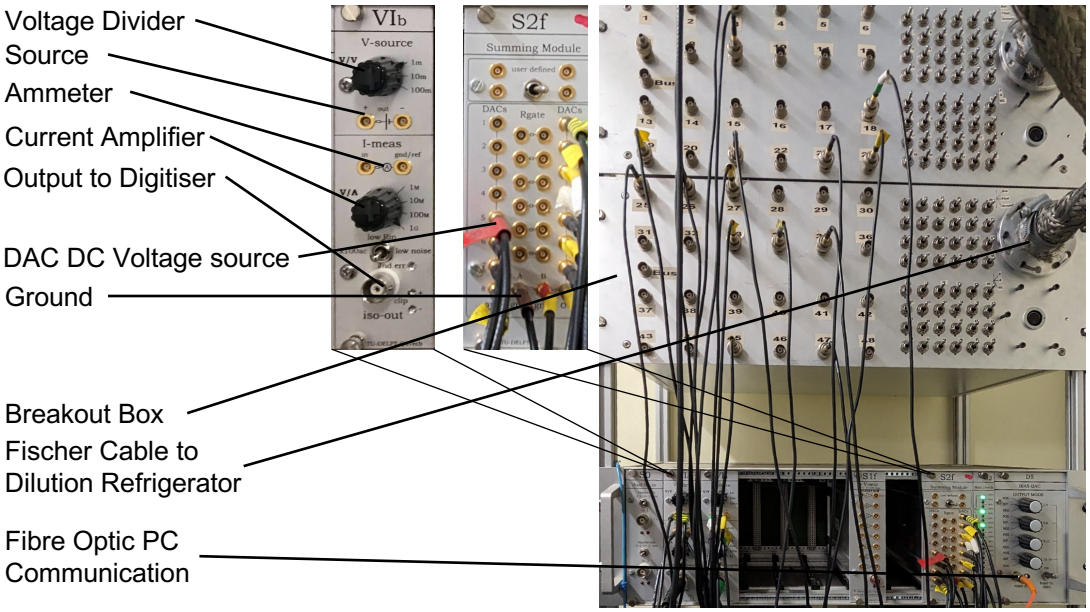


Figure 4.2: Classical electronics. A QTRay Rack houses the breakout box, the digital-to-analogue (DAC) DC voltage sources, voltage dividers, ground pins and current amplifiers for quantum transport measurements. Communication with the DAC occurs via a fibre optic cable.

enables easy explanation and walkthrough of breakout electronics, voltage sources, voltage dividers and current amplifiers in a single rack.

The DC lines connected to the sample, run from the Cold Finger, up the stages of the fridge, and exit at the top of the dilution refrigerator at room temperature where they are connected to a breakout box via a Fischer Cable (Fig. 4.2). The AC lines follow a similar path from the sample and are connected to the breakout box via an SMA cable, and then an AC source such as a Keysight M3302A arbitrary waveform generator (AWG) or a Quantum Machines OPX. From the breakout box, each DC line/channel is connected to a DAC DC voltage source via a Bayonet Neill–Concelman (BNC) connector. Individual DC lines can be grounded or floated using a switch on the breakout box. The DACs have a voltage range of 4V and are operated in bipolar mode, resulting in maximum and minimum voltages of 2 V and -2 V respectively. The VIb module (Fig. 4.2) is a combined V-source, I-measure and IsoOut, hence the letters "V" and "I" in the name. It provides a source voltage, current measurement with a noise floor down to 5 fA/sqrtHz(at 1G V/A) and an isolation amplifier (used to

provide electrical isolation and electrical safety of the output signals), in one module. The S2f module (Fig. 4.2) is a summing module (hence the letter "S" in the name) and serves as a patch area for DAC voltage sources. The DAC S2f DC voltage source module has a voltage resolution is 60 μV due to the 16-bit resolution of the DAC. The inclusion of voltage dividers is common to achieve higher voltage resolutions and can be custom made or purchased. Serial commands and the status of the DAC are sent between the DAC and the PC via a fibre optic to USB connection. The source-drain bias is set by the VIb module which has a built-in voltage divider. When performing quantum transport measurements the current is measured using the ammeter in the VIb module which is then amplified by 1 $\mathrm{GV/A}$. The amplification value is factored out at the software level after data acquisition. The measured current is acquired and sent to a digitiser connected to a PC. The drain is connected to the ground on the S2f module.

4.3 Instrument control

Various software packages were used to programmatically control the classical electronics (Fig. 4.3), as implementations are built customarily between laboratories, but all used in this work were written in the Python programming language. In Chapter 5, `Pygor` developed by Dominic T. Lennon [115] was used across the University of Basel and IST Austria. In Chapter 6, `SilQ` developed by Serwan Asaad and Mark Johnson [116] was used in the FQT lab for data acquisition and instrument control specifically for ion-implanted donor in silicon devices. Our collaborators in Chapter 7 used custom `Qua` scripts in Python for data acquisition and control of their AC voltage source.

The purpose of all of these various packages is to act as a human-understandable wrapper around the lower-level instrument drivers. Therefore, rather than running hard-to-read scripts of regular measurement sequences which are instrument-dependent, users can run commands such as `get_current()` or `set_val()` to measure the

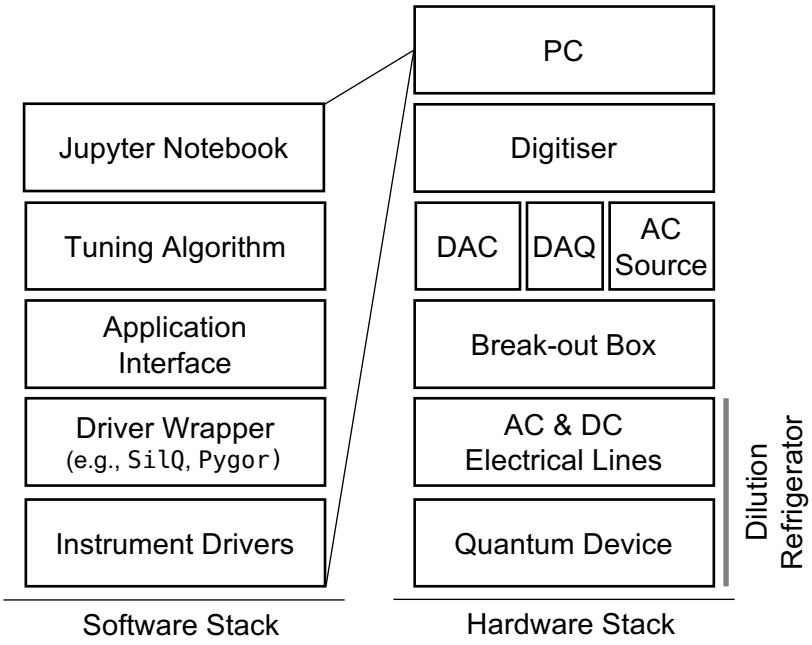


Figure 4.3: Software and hardware stack. The software is written in the Python programming language. Each laboratory possesses its own Driver Wrapper around the lower level instrument drivers for easy instrument control and data acquisition (DAQ). The application interface enables the tuning algorithm to be hardware agnostic and portable. The Jupyter Notebook provides data visualisation capabilities and an interactive console for the control of the hardware stack using the Driver Wrapper of choice.

current flowing through the device or set a specific voltage to a specific gate electrode, while the underlying necessary instrument drivers are correctly chosen via a configuration file.

`Pygor` was originally designed to perform as a two-computer, server-client architecture. This enabled the tuning algorithm logic and heavy computation to run on a more powerful PC (the client), and then send commands via the internet to a less powerful laboratory PC which would control the instruments and acquire data (the server). The acquired data would then be sent back to the client's PC, and the sequence would be repeated. However, I realised that the computational power required for the algorithm developed in Chapter 5 could be satisfied by a standard laboratory PC, but portions of the algorithm which relied on multiprocessing needed to be run on a Unix-like computer operating system. Therefore, I ran the client and server on the same PC in the laboratory, but if the operating system of the laboratory PC was Windows-based, I ran the client on the Windows Linux Subsystem giving me access to a Unix-based operating system on the same PC. This greatly reduced the installation time of `Pygor` when setting up experiments remotely as we removed the need to specify open ports across university department networks and lift certain firewalls which could put the university network at risk. Moreover, the entire experiment could be accessed by authorised remote desktop to a single PC within the host institution's laboratory, which favoured data sharing security controls and promoted ease of use.

In between the driver wrapper and the tuning algorithm software sits an application interface. The interface serves a crucial role in the software stack, allowing the tuning algorithms developed to be agnostic towards hardware and instrument control software across laboratory setups. By defining an interface and data types, tuning algorithms can be easily ported between laboratories, with minimal coding required other than constructing an application interface. The interface is made up of commands such as, set DAC voltages, read the current state of the DAC and acquire a current measurement.

The tuning algorithms developed in this body of work do not require any form of human interaction; they control the experiment autonomously. They have procedural characteristics and are iterative, repeatedly progressing through a sequence of steps. An overview of a tuning algorithm workflow is, acquire a measurement, signal process the data acquired, make a decision on the next measurement to take and repeat. Each portion of the workflow routinely relies on machine learning for example, a Random Forest classifier at the signal processing stage and Gaussian Process Bayesian optimisation to decide what measurement to take next. Each portion of the tuning algorithm is halted when a maximum number of iterations is reached or a stopping criterion is met. Development of the relevant stopping criteria relies on our understanding as experimentalists as what we'd expect and desire the respective signals to look like at different stages of the tuning process.

Humans interact with the entire software stack via a Jupyter Notebook on a PC where they can use their instrument driver wrapper of choice and tuning algorithm to control their instrument electronics. Jupyter Notebooks, behave as an interactive python console and possess the ability to display graphical images inline enabling experimentalists to acquire measurements such as charge stability diagrams and display them in real time. This can be helpful for manual tuning for the experimentalist as well as observing the online performance of a tuning algorithm at its various stages.

4.4 Software development

Being a thesis which consists of algorithms for automated quantum device control, I will briefly give an overview of the software architectural styles I obeyed and software development practises I attempted to follow.

The architecture of the software developed followed a 'wide instead of deep' [117] approach with each portion and task of the overall programme separated into various modules consisting of distinct components (Fig. 4.4). For example, the

```
brandonseverin@oums-brandon:~/Documents/development/unsw/donorsearch$ tree src/donorsearch -I __pycache__
src/donorsearch
├── acquisitionroutines
│   ├── daqprocedures.py
│   ├── __init__.py
│   ├── searchprocedures.py
│   └── set.py
├── assets
│   ├── pulses.py
│   └── tallulaheast
│       └── OPX
│           ├── OPX_config.py
│           └── random_pulse_train.csv
├── helperfunctions
│   ├── arraymanipulations.py
│   ├── filesystemtools.py
│   ├── __init__.py
│   ├── plotting.py
│   ├── stringmethods.py
│   └── timer.py
├── __init__.py
├── interfaces
│   ├── __init__.py
│   ├── isilqhybrid.py
│   ├── isilq.py
│   ├── itallulaheast.py
│   └── typealiases.py
```

Figure 4.4: Codebase file tree. An excerpt of the codebase file tree of one of the algorithms developed in this thesis. The codebase consists of a total of 16 directories and 46 files and is organised as a Python Package. The `interfaces` directory contains application interfaces for the tuning algorithm to run on different laboratory setups and serves as an example for future interface development.

different modules were made up of acquisition routines, models, helper functions, signal processing, interfaces and pipelines. Throughout my research, I moved away from object-oriented programming and towards a procedural programming paradigm with the inclusion of functional paradigm elements. This shift led to codebases (e.g. `donorsearch` in Chapter 6) that allowed for fast iterative development, modularity, and composability, as each routine or procedure was made up of smaller procedures which could be swapped in or out at will. Moreover, the testing of each component, unit testing, was easier to perform because the risk of not including global state changes was reduced, such as in an object-oriented approach (e.g. CATSAI in Chapter 5).

How the software is developed can greatly influence the software's quality, readability, portability, scalability and adaptability should changes be required during later stages of development. The practices I followed to maximise these qualities are as follows:

- **Long Stretches of Focused Time**: Developing software requires long stretches of uninterrupted time (at least 3 hours). Remove distractions and meetings to allow for focused time. A software developer needs to keep as

much of the program in their near-term memory as possible. This holistic understanding of the codebase takes at least 30 minutes to build in each coding session and is easily destroyed by distractions.

- **Development Environment**: Employ a lightweight development environment that supports autocompletion and type hints, such as Visual Studio Code (VS Code) or Vim with the relevant plugins.
- **Package and Environment Management**: Set up a virtual environment using `conda` to manage project dependencies and ensure reproducibility. Back up the virtual environment to `.yaml` file to allow portability across different PCs and operating systems.
- **Version Control**: Employ Git for code versioning, enabling the ease of tracking and managing the integration of features into the codebase.
- **Feature Development Management**: Utilise platforms like GitHub's Issues and Projects to track features and tasks. Take advantage of Kanban boards to manage tasks and prioritise different features. Re-evaluate the development of features weekly.
- **Codebase Structure**: From the inception of the project, organise the software as a Python package. This approach enhances portability and minimises potential installation and testing complications in the future.
- **Testing**: Rigorously test the software as it is being developed via a Python package style installation, utilising tools like `pytest`, `coverage`, and actual *in-the-field testing* to ensure reliability and functionality. Getting early and frequent feedback from *in-the-field testing* and users will keep development goals on track to achieve a good working product.
- **Code Formatting**: Employ a code formatter (e.g. `black`) to maintain a uniform code layout and improve code readability.

- **Type Hints**: Add type hints to every function or callable to enhance code clarity and facilitate error detection. Rely on tools such as `Pylance` or `MyPy` for static analysis and type checking. This greatly reduces type errors occurring in production.

- **Documentation**: Accompany each function/callable with docstrings unless their inclusion is deemed unnecessary. Utilise third-party tools to streamline this process, ensuring that the code evolves alongside its documentation without additional manual effort.

- **Mainline Development**: An individual should focus on developing one feature at a time, assigning each feature to its dedicated Git branch. Merge these branches with the main branch upon completion. The life cycle of a feature before it is merged should be as short as possible, no more than a few days. This practice, even when working alone, enables efficient code management and facilitates the isolation of potential issues without affecting the entire codebase.

Chapter 5

Cross Architecture Tuning in Silicon using ML

> The implications go far beyond my beloved chessboard... Not only do these self-taught expert machines perform incredibly well, but we can actually learn from the new knowledge they produce.
>
> Gary Kasparov

The work in this chapter is from the published paper by Severin *et al.* [118] which was completed with collaborators from the University of Basel and IST Austria.

In the late 20^{th} century Humanity entered a new period: The Silicon Age. Coming out of the tail-end cusp of the industrial revolution, silicon alongside the ability to wield element-14 with its neighbours into miniature switches gave birth to much of the world we interact with today. As we continue to wade through this period and look forward to future computational architectures, it is hard to ignore silicon's opportunity for scalable qubit realisations. To reliably leverage the scalability characteristic of silicon one must possess a series of algorithms, or better, a single algorithm, to tune across

different architectures of gate-defined double quantum dots in silicon and SiGe-based foundry fabrication-ready devices, preparing the road to millions of qubits on a chip.

The motivation of this chapter is to demonstrate a general algorithm that can coarsely tune double quantum dots in a range of device architectures that, for example, may have very different cross talk, voltage ranges, noise characteristics and material systems, with minimal modifications.

In this chapter, I demonstrate that it is possible to automate the tuning of a 4-gate Si FinFET, a 5-gate GeSi nanowire and a 7-gate Ge/SiGe heterostructure double quantum dot device from scratch with the same algorithm. I achieve tuning times of 30, 10, and 92 minutes, respectively. The algorithm also provides insight into the parameter space landscape for each of these devices, allowing for the characterisation of the regions where double quantum dot regimes are found. These results show that overarching solutions for the tuning of quantum devices are enabled by machine learning.

5.1 Introduction

Before we can use a quantum computer, we first need to be able to turn it on [118]. There are many stages to this initial step, particularly for quantum computing architectures based on semiconductors. Silicon and SiGe devices can encode promising spin qubits [36], demonstrating excellent fidelities, long coherence times and a pathway to scalability [57, 67, 119–122]. Many of these key characteristics revolve around the material itself providing the opportunity to be purified to a near-perfect magnetically clean environment resulting in very weak to no hyperfine interactions. As the material of choice of the microelectronics industry, gate-defined quantum dots in silicon and SiGe have great potential for the fabrication of circuits consisting of a large number of qubits, an essential requirement to achieving a universal fault-tolerant quantum computer [6, 8].

Multiple gate electrodes provide the ability to tune differing devices into similar operating regimes. These gate voltages define a large parameter space to be explored. Each device architecture and material realisation defines a specific parameter space. The time-consuming challenge of tuning semiconductor devices becomes intractable as we combine different device architectures in the realisation of complex quantum circuits with millions of components. The development of machine learning algorithms for quantum device tuning [32, 81, 82, 101, 103, 104, 106, 113, 123–126] is exceptionally challenging when looking for such overarching solutions, successful on very different types of devices which may need to cater to specific purposes.

Here I demonstrate that it is possible to tune quantum dots in three different device architectures and material systems completely automatically. This machine learning-based algorithm, which I call 'Cross-Architecture Tuning Solution using AI' (CATSAI), requires only the following hyperparameters to be set once, for each type of device, in a configuration file: source-drain bias, safety voltage bounds, resolution and size of acquisition current maps and traces, the offset current noise floor, and Coulomb peak segmentation threshold (see Appendix A). The origin and gate voltage sweep directions can be arbitrarily selected for devices operating with accumulation or depletion mode gate electrodes, and either holes or electrons as majority charge carriers. An advanced signal processing classification method handles charge switches and other noise patterns.

I demonstrate the CATSAI algorithm for a Si accumulation-mode ambipolar FinFET [38, 95, 127], a depletion-mode Ge/Si core/shell nanowire [88–90] and a laterally-defined device in a Ge/SiGe heterostructure [80, 128–130], operating with holes as charge carriers. I show that CATSAI outperforms random search and human experts on all devices. The machine learning-based approach also reveals the size and characteristics of the double quantum dot regime within the multidimensional parameter space defined by each gate voltage architecture. The learnings from the automatic tuning of double quantum dot devices will help us understand how machine

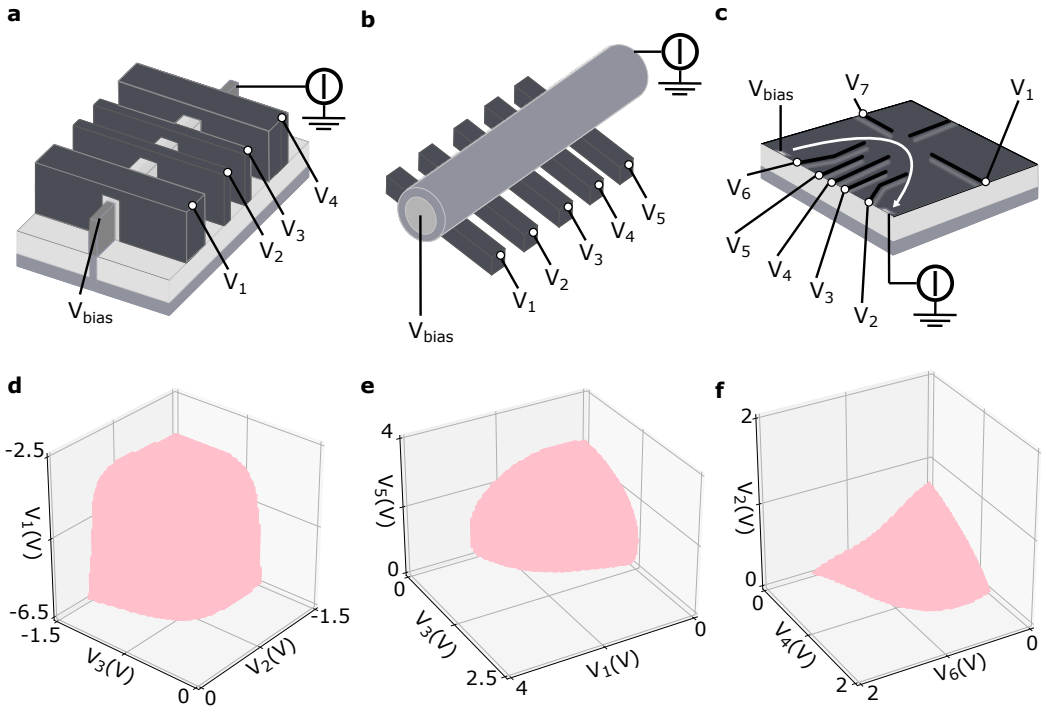


Figure 5.1: Device schematics. Si FinFET (a), GeSi nanowire (b) and Ge/SiGe heterostructure (c) device architectures and their corresponding current pinch-off hypersurfaces for hole transport calculated using a Gaussian process model for one of the tuning algorithm runs (d, e, f). Three gates are plotted for illustrative purposes with the remaining gates on each device set to a constant value. The bias was kept constant throughout the experiment. CATSAI was given control over the gate electrodes V_1 - V_4, V_1 - V_5, and V_1 - V_7 on the FinFET, nanowire and heterostructure, respectively.

learning and other automated tunining methods may be used for tuning up larger devices in the future.

5.2 Methods

5.2.1 The devices

Double quantum dots are defined by applying DC voltages to the gate electrodes $V_1 - V_4$ for the FinFET, $V_1 - V_5$ for the nanowire, $V_1 - V_7$ for the heterostructure (Fig. 5.1). For the FinFET, the lead gate electrodes V_1 and V_4, open and close the quasi 1D silicon channel to charge carriers by controlling the size of the tunnel barrier between

the quantum dots and the source and drain. The left and right plunger gate electrodes V_2 and V_3, control the occupation of the left and right quantum dot respectively. A current is driven through the FinFET by applying a bias voltage V_{bias} of 7.6 mV (+ 3.8 mV at the source, - 3.8 mV at the drain) to NiSi contacts [95]. The gate voltages of the FinFET are operated such that the charge carriers are holes confined by accumulation. For the nanowire, gates V_2 and V_4 act as left and right plunger gates for the quantum dots formed within the 1D channel with the remaining gates mainly controlling the tunnel barriers. Hole quantum dots are formed in depletion mode. V_{bias} was set to 4 mV. For the Ge/SiGe heterostructure, V_5 and V_3 operate as the left and right plunger gate electrodes respectively, with the remaining gate electrodes utilised as barrier gates. The white arrow denotes the flow of current. V_{bias} was set to 0.5 mV and the charge carriers are holes confined in depletion mode. The values of V_{bias} are set to be above typical charging energies for single quantum dots in each device. The choice of V_{bias} can be left to an optimiser. For the heterostructure, experiments were performed at 300 mK, for the nanowire at 1.5 K and for the FinFET at 800 mK.

Voltages applied to the gate electrodes of the devices can cause the current flow to pinch-off, transitioning from a relatively high current to a near-zero value. These voltages where pinch-off occurs define a hypersurface within the entire voltage space for each device. CATSAI has no knowledge of the device architecture and generates a model of the hypersurface after a given number of iterations. The resulting hypersurface for different devices is shown in Fig. 5.1d–f. Three gates are plotted for ease of visualisation and the remaining gates are kept constant at their average value at pinch-off across the hypersurface (see Appendix A). The hypersurfaces corresponding to different devices present different curvatures, leading to different tuning landscapes. The FinFET hypersurface (Fig. 5.1d) is near symmetrical in the plunger gates plane, $V_2 - V_3$. This is expected as these gate electrodes are nominally identical. Although V_1 is wider than the plunger gates, its effect is not stronger.

The curvature of the nanowire's hypersurface is similar in the planes $V_1(V_5) - V_3$, since these planes are defined by the outer-middle barrier gates (Fig. 5.1e). The heterostructure's hypersurface has almost planar dependence on gate voltages $V_{2,4,6}$ (Fig. 5.1f). The hypersurface's curvature in the V_2 – V_4 plane is evidently similar to that in the V_6 – V_4 plane, in agreement with the gate architecture. This hypersuface is qualitatively different to that reported for a relatively similar gate architecture patterned on a different heterostructure (AlGaAs/GaAs) [103]. The more pronounced curvature of the hypersurfaces corresponding to the FinFET and the nanowire are expected given the larger gate couplings that are typically observed in FinFET and nanowire devices. Hypersurface characterisation could be used to inform device design and quantify device variability. Despite the stark differences in gate voltage landscapes, which evidence the difficulties of cross-architecture tuning, CATSAI is able to tune across all three device architectures.

5.2.2 The CATSAI algorithm

CATSAI is designed to coarsely tune double quantum dot devices using transport measurements. As quantum devices evolve, and for example double quantum dot tuning is carried out in the presence of/with a charge sensor and in larger dot arrays, parts of CATSAI may need to be adapted and forged with other algorithms [131] to cater to alternative tuning and measurement techniques (Chapter 8). This would include providing automatic gate compensation across wide voltage ranges during the tuning process with a charge sensor [131]. The purpose of CATSAI is to tune double quantum dot devices.

CATSAI's workflow consists of three stages, the initialisation stage, the sampling stage and the investigation stage (Fig. 5.2). In the initialisation stage V_{bias} is fixed, and the current range, i.e. the maximum and minimum current flowing through the device, is determined by measuring the current both with all the gate electrodes set to 0 V and to their maximum permissible magnitude. To avoid damage to the device the

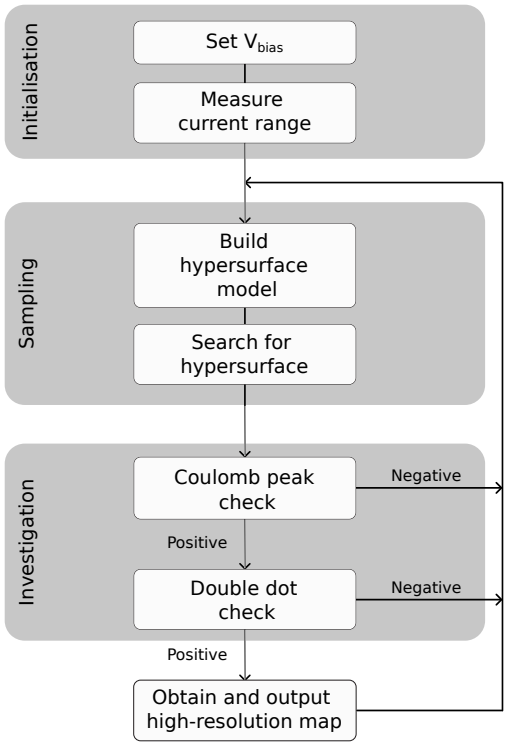


Figure 5.2: Outline of CATSAI's workflow. The initialisation stage consists of setting V_{bias} then measuring the maximum and minimum (offset) current flowing through the device. The sampling stage detects pinch-off locations in gate voltage space. The algorithm selects a unit vector in gate voltage space $\mathbf{u}$ based on the model it generates of the hypersurface and of the probability of finding Coulomb peaks in a given location in gate voltage space. In the investigation stage the algorithm uses the plunger gates to sequentially acquire current traces and maps which are sent to the relevant classifiers. The Coulomb peak detector is a random forest classifier which determines whether Coulomb peaks are present (positive) or not (negative) within a current trace. In each iteration, the algorithm outputs a high-resolution current map if the double dot check score function is passed. After the investigation stage, the algorithm returns to the sampling stage.

algorithm is given voltage bounds in which it can operate each gate electrode. The bounds are measured manually in advance of running the algorithm.

Sampling stage

After the initialisation stage, the algorithm turns to the sampling stage. The algorithm selects a unit vector $\mathbf{u}$ in the gate voltage space of the device based on a Gaussian process model of the hypersurface as shown in Fig. 5.1d–f, and a weighting from the probability of finding Coulomb peaks at a given location in voltage space $\tilde{P}_{\mathrm{peak}}(\mathbf{v})$.

This vector $\mathbf{v}$ consists of all the gate voltages considered for tuning. The algorithm then sweeps the gate voltages along that direction until pinch-off occurs. The algorithm identifies the onset of pinch-off as a current drop below a certain threshold (0.5% of the measured current range). The N-dimensional hypersurface is delimited by the pinch-off voltages of the N gate electrodes for each device. The algorithm's performance is highly sensitive to the pinch-off threshold. If incorrectly specified the found hypersurface will not lead to desired quantum transport features, for example evidenced by an inability for the gate voltages to pinch-off the current flow. The measured current range of semiconductor quantum devices is typically on the order of nanoamps, with the minimum (offset) current on the order of a picoamp. Therefore, a suitable threshold value can be deduced as a fraction of the measured current range, which I chose as 0.5%, to target the transition in voltage space between the device conducting and being completely pinched-off.

The model of the hypersurfaces $\tilde{p}(r|\mathbf{u})$, (Fig. 5.1d-f) predicts how far in gate voltage space from the origin r, is the hypersurface for a given $\mathbf{u}$. The probability of finding a Coulomb peak $\tilde{P}_{\text{peak}}(\mathbf{v})$ at a given point in voltage space $\mathbf{v}$ can be modelled by breaking it down into two components $\tilde{P}_{\text{peak}}(\mathbf{v}) = \tilde{P}_{\text{valid}}(\mathbf{v})\tilde{P}_{\text{peak|valid}}(\mathbf{v})$. The probability of finding pinch-off $\tilde{P}_{\text{valid}}(\mathbf{v})$ and if Coulomb peaks will be found there $\tilde{P}_{\text{peak|valid}}(\mathbf{v})$, are modelled using a Gaussian process classifier.

In the sampling stage, I used a numerical Markov Chain Monte Carlo approach from Ref. [132] to sample from the predicted hypersurface, due to challenges of uniformly sampling arbitrarily shaped hypersurfaces [103]. The approach simulates the Brownian motion of particles initialised at the origin within the volume in voltage space defined by the hypersurface, and each collision location of a particle with the hypersurface provides a candidate sample point. It uses $\tilde{p}(r|\mathbf{u})$ to approximate the location of the hypersurface during sampling. N_p particles are initialised at the origin, where $N_p = 200$, and each particle may take steps in voltage space sampled from a normal distribution with mean of 0 and variance of 25 mV. If the particle exits

the volume defined by the predicted hypersurface after a given step, the particle's last location within the volume is stored as a candidate point. If the particle leaves the voltage safety bounds, the particle is re-initialized at the origin. This process is repeated until all N_p particles have collided with the hypersurface.

Each candidate point is assigned a weight proportional to the corresponding value of $\tilde{P}_{\text{peak}}(\mathbf{v})$. The algorithm uses Thompson sampling [133] to select candidate points such that a point lies on the hypersurface and leads to the detection of a Coulomb peak. I used Thompson sampling, and gave uniform priors to $\tilde{P}_{\text{valid}}(\mathbf{v})$ and $\tilde{P}_{\text{peak|valid}}(\mathbf{v})$ which construct $\tilde{P}_{\text{peak}}(\mathbf{v})$, such that the algorithm will initially sample the hypersurface uniformly. As observations are made, the algorithm will exclude regions of the hypersurface where pinch-off was not detected or Coulomb peaks were not detected.

There is no prior knowledge other than the gate voltage bounds for the initial model of $\tilde{p}(r|\mathbf{u})$ The range of r spans from 0 to r_{bound}, where $r_{\text{bound}} = \sqrt{(\Delta {V_1}^2 + \ldots, \Delta {V_N}^2)}$ where ΔV_i is the difference between the maximum and minimum gate voltage bounds on gate electrode V_i. The prior distribution is set as, $\mu_r(\mathbf{u}) = r_{\text{bound}}/2, k_r(\mathbf{u}, \mathbf{u} = (r_{\text{bound}})^2/4$, such that r_{bound} and the origin will be two standard deviations away from $m_r(\mathbf{u})$

All Gaussian process models used the Matérn 5/2 kernel for the prior covariance with characteristic length scales of ℓ_q where $q = 1, \ldots, N$. I used the Matérn 5/2 kernel because I did not expect the the hypersurface or the location of quantum features to vary linearly or periodically with gate voltage space but, I did expect the hypersurface to be reasonably smooth (Chap. 2.7.5). The length scales are periodically optimised to the Maximum a posterior estimate based on a prior gamma distribution and acquired data. For $\tilde{p}(r|\mathbf{u})$, the gamma distribution is set to have a mean of 0.4 and variance of 0.1^2, for $\tilde{P}_{\text{valid}}(\mathbf{v})$ the mean is set to 500 and variance of 100^2, for $\tilde{P}_{\text{peak|valid}}(\mathbf{v})$ the mean is 50 and 20^2 for the variance.

Investigation stage

At the start of the investigation stage, once pinch-off is found in a given gate voltage direction, a high-resolution current trace is performed. This current trace, which starts at the pinch-off location and runs diagonal within the plane defined by the plunger gates, was set to have a fixed length of 128 pixels and resolution 1.56 mV/pixel for the nanowire and 0.78 mV/pixel for the FinFET and the heterostructure. Given the expected charging energies (Chap. 2) of the devices and relative impact a gate voltage has on the electrochemical potential of a quantum dot (gate lever arm), I chose the length of the current trace such that the resolutions were adequate to capture multiple quantum transport features such as Coulomb peaks within a single trace. The algorithm's sensitivity to this choice is relatively low. The plunger gates, selected before running the algorithm, are those expected to predominantly shift the electrochemical potential in left and right dots. Using a random forest classifier [43, 134], the algorithm determines whether Coulomb peaks are present in the current trace. Due to prior training, this approach is robust against noise and switches unlike simple peak-finding packages which are much more likely to be tricked that a trace of noise corresponds to hole/electron transport as they typically rely on the sole identification of local maxima. I chose a random forest classifier due to its simplicity and speed to train, resistance to overfitting and limited readily accessible current trace data from a range of different device types. Scarcity of suitable data makes it inappropriate to use more computational intensive approaches, such as deep learning, without significant data augmentation or a reliable simulator capable of representing Coulomb peaks and various noise characteristics. This random forest classifier is key to the success of CATSAI across device types with different noise characteristics (see Appendix A).

If Coulomb peaks are found by the classifier then a low-resolution current map (16×16 pixels, 5 mV/pixel for the nanowire and 9 mV/pixel for the FinFET and the heterostructure) is taken by sweeping the plunger gates. The current map is believed to contain double quantum dot features if it scores above a threshold, which

is fixed and can be optimised. I use the same score function as in Ref. [103]. If double quantum dot features are believed to be present, a high-resolution current map (48×48 pixels, 4.2 mV/pixel for the nanowire and 2.5 mV/pixel for the FinFET and the heterostructure) is taken. At the end of the iteration, CATSAI returns to the start of the sampling stage. CATSAI proceeds to update the hypersurface model and $\tilde{P}_{\mathrm{peak}}(\mathbf{v})$ with the knowledge garnered of pinch-off and Coulomb peak locations respectively. CATSAI runs for a certain number of iterations. A posteriori, to gauge the algorithm's performance, humans can verify if the double quantum dot features were successfully identified by the algorithm.

CATSAI is benchmarked against a version of this algorithm which does not use a weighted hypersurface model to influence the sampling of the hypersurface. It instead samples a point in the voltage parameter space of the device at random and carries out the investigation stage for each iteration. I call this version of CATSAI 'Random Search', although it is important to highlight that it still relies on peak detection. In this manner one may gain insight into the online performance of the Coulomb peak detector. Moreover, it enables cross-validation of CATSAI's sampling method. This would not be possible if CATSAI were benchmarked against alternative published tuning procedures which would also require significant alteration to perform on both one-dimensional and planar devices. Moreover, previously published algorithms lack the signal processing and classification routines required for the silicon and SiGe-based devices investigated in this work.

5.2.3 Tuning across architectures and material systems

To make the algorithm general across different charge carriers and modes in which gate electrodes are designed to act (depletion or accumulation), the origin, bound, and direction of the gate-voltage space exploration used in the sampling stage are set in a configuration file (Fig. 5.3). The algorithm starts in the gate voltage configuration which delivers the highest current and sweeps gate voltages in the

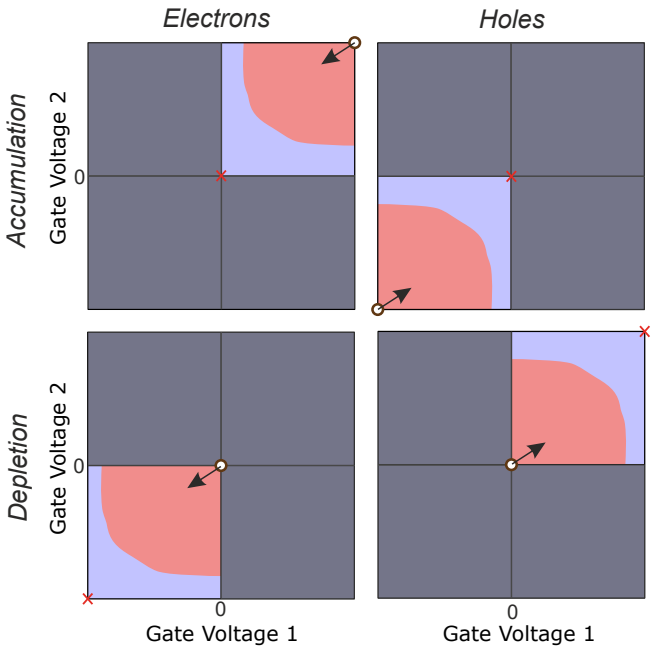


Figure 5.3: Gate-voltage space exploration. Different charge carriers (gate operation modes) are represented in different columns (rows). Each panel illustrates the initial placement of the origin (white circle), search boundary (red cross), and search direction (black arrow). The gate voltage space is divided into regions of near-zero (blue) and non-zero (pink) current. Regions of voltage space which cannot be explored due to the gate voltage bounds set to avoid device damage are greyed out.

direction of decreasing current with the aim of locating the boundary between the two regions. This flexibility in the search of gate voltage space, combined with a noise-tolerant classification of Coulomb peaks in the investigation stage, makes CATSAI robust across device architectures and material systems. The Coulomb peak detector is trained on current traces acquired in different Si FinFET and GeSi nanowire devices (see Appendix A). This random forest classifier can successfully handle both noise and charge switches, resulting in a robust Coulomb peak detection. The number of false positives in the classification that are accepted for the next step of the investigation stage is thus reduced, significantly shortening device tuning times.

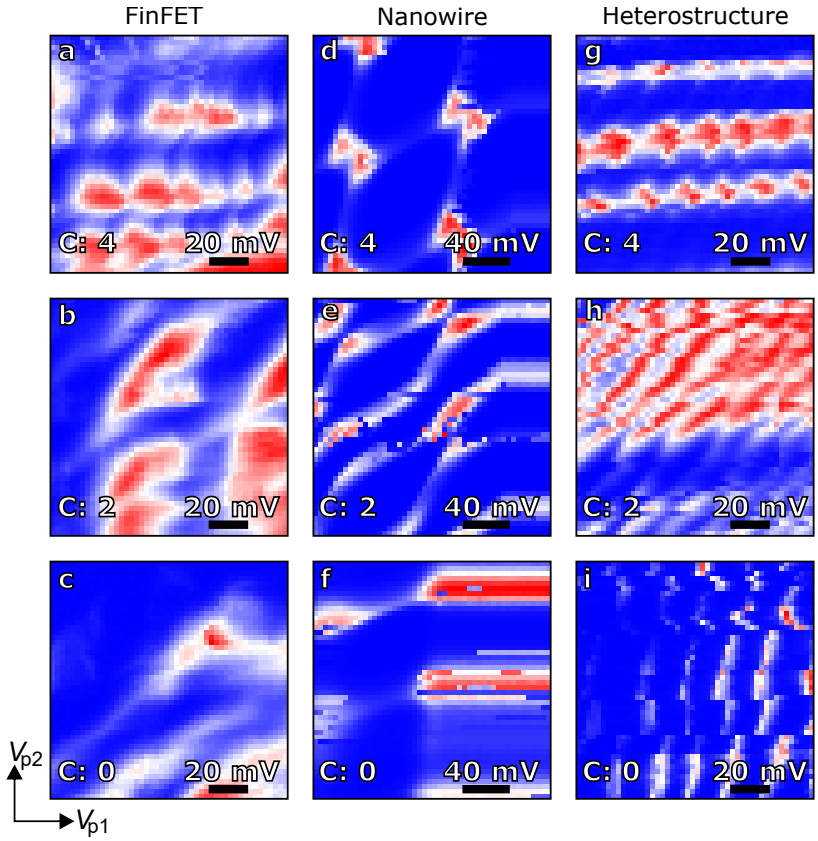


Figure 5.4: Device tuning. Examples of current map outputs on the different devices in which CATSAI was run. High resolution maps are generated during the investigation stage by sweeping the plunger gates of each device $V_{p1,p2}$; for the FinFET $V_{3,2}$ (a,b,c), the nanowire $V_{4,2}$ (d,e,f) and the heterostructure $V_{3,5}$ (g,h,i). These current maps are labelled *a posteriori* by humans to verify whether they correspond to the double quantum dot regime. C indicates the number of humans out of four who labelled the current map as corresponding to a double quantum dot regime. Red (blue) indicates regions of high (low) current in each map.

5.3 Results

The algorithm was run for 250 iterations for all experiments performed. The number of iterations that the algorithm runs without a hypersurface model, i, which can be separately optimised, was fixed to twelve in this case. A few examples of output current maps produced by CATSAI for the different devices considered are displayed in Fig. 5.4. The double quantum dot regimes pictured in Fig. 5.4 show that the Algorithm is capable of identifying double dot regimes despite device characteristics that might be far from ideal.

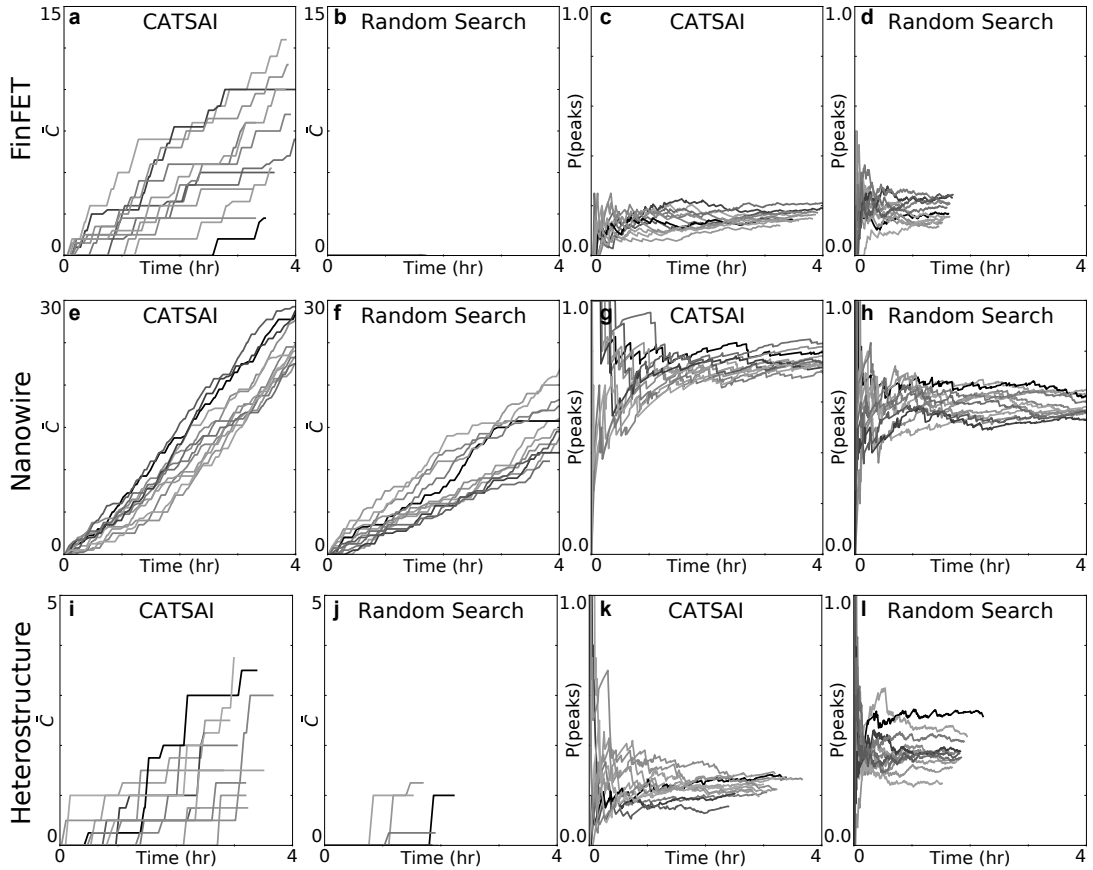


Figure 5.5: Benchmarking the Algorithm's performance. The cumulative sum of the average number of double quantum dot regimes verified by humans $\bar{C}$ (first and second columns) and probability of finding Coulomb peaks $\mathrm{P(peaks)}$ (third and fourth columns), as a function of laboratory time for each run of CATSAI and Random Search algorithms. Rows correspond to the different devices. Only the first 4 hours of each tuning run are shown for ease of visualisation. CATSAI outperforms Random Search in the number of double quantum dot regimes located for all devices. The value of $\bar{C}$ remains at 0 in many of the Random Search runs, and thus are not visible in the plots of $\bar{C}$ as a function of time. The increase in $\mathrm{P(peaks)}$ as a function of laboratory time observed for the CATSAI runs after the first 12 iterations can be explained by the algorithm 'learning' a better model of the hypersurface as the Gaussian process regression acquires more observations.

To benchmark the performance of the algorithm, the output current maps were labelled by human experts at the end of the tuning experiment to verify whether they corresponded to the double quantum dot regime (see Appendix A). The human experts were unaware whether the current maps to be labelled were the output of CATSAI or Random Search. I define C as the number of humans who labelled a current map as containing double quantum dot features. In each iteration of the

	Tuning Times (minutes)	
Device	CATSAI	Random Search
GeSi Nanowire	9.5 (6.7, 12)	17 (9.9, 26)
Si FinFET	30 (26, 37)	-
Ge/SiGe Het.	92 (71, 120)	360 (190, 830)

Table 5.1: Median device tuning times with 80% credibility intervals (equal tailed) corresponding to CATSAI and Random Search algorithm runs for all devices considered. The Random Search tuning time for the FinFET is unknown as no double quantum dot regimes were located.

algorithm, I cumulatively sum the value of C normalised by the total number of human labellers (four). The resulting quantity, $\bar{C}$, provides a measure of the number of double dot regimes found by the tuning algorithm while considering disagreements between human labellers.

Figure 5.5a–j shows $\bar{C}$ as a function of laboratory time for 12 runs of CATSAI and Random Search for each of the devices considered. CATSAI outperforms Random Search in the total number of double quantum dot regimes located in all cases. The Random Search algorithm did relatively well in locating double quantum dot regimes in the nanowire but did not locate any double quantum dot regime in the FinFET (Fig. 5.5b) and struggled to locate more than one double quantum dot regime in the Ge/SiGe heterostructure device (Fig. 5.5j).

The probability of Coulomb peaks estimated for a given number of iterations, $\mathrm{P(peaks)}$, is plotted as a function of laboratory time for each algorithm run and each device in Fig. 5.5c–l. The trend of $\mathrm{P(peaks)}$ as a function of laboratory time observed in most CATSAI runs is similar for the FinFET, nanowire and the heterostructure devices. $\mathrm{P(peaks)}$ has a gradual upward trend in many of the experimental runs after the first 30 minutes to an hour and then saturates over laboratory time at different values between the devices. For the FinFET device and the heterostructure, the values of $\mathrm{P(peaks)}$ are on average larger for Random Search than for CATSAI runs. However, the number of double dots found by Random Search is still less than CATSAI in both devices.

CATSAI tuned all devices faster than Random Search. The median tuning times are 10 minutes for the nanowire, 30 minutes for the FinFET, and 90 minutes for the heterostructure (Table 5.1). The Random Search algorithm was surprisingly quick at tuning the nanowire, while unable to tune the FinFET successfully within 12 runs of the algorithm, which totals a laboratory time of 19 hours. The difference between the upper and lower credibility interval of the tuning times achieved in the heterostructure device is an order of magnitude smaller than that achieved by Random Search. I estimate tuning times and their respective 80% credibility intervals as described in Ref. [103]: To infer the expected tuning times μ_t and probabilities $\mathrm{P}(\mathrm{peaks})$ and $\mathrm{P}(\mathrm{success}|\mathrm{peaks})$, where success corresponds to double quantum dot features found, I used Bayesian inference with a Jeffreys prior. To infer the probabilities, let p denote the probability of either $\mathrm{P}(\mathrm{peaks})$ and $\mathrm{P}(\mathrm{success}|\mathrm{peaks})$. The Jeffreys prior for a binomial distribution is $p \sim \mathrm{Beta}(0.5, 0.5)$. If I observe k successful events over n trials, then the posterior distribution of p is $p|k, n \sim \mathrm{Beta}(0.5 + k, 0.5 + n - k)$.

To infer μ_t, I assume that the rate of success over time follows a Poisson distribution $k \sim \mathrm{Poisson}(\lambda t_{tot})$ where t_{tot} is the total time for an algorithm run, and λ is the rate parameter. The expected time between two consecutive successes (i.e. an acquired charge stability diagram displays double quantum dot features) is $\mu_t = 1/\lambda$. The Jeffreys prior for a Poisson distribution is $\lambda \sim \mathrm{Gamma}(0.5, 0)$ therefore, the posterior is $\lambda|k, t_{tot} \sim \mathrm{Gamma}(0.5 + k, t_{tot})$. Therefore $\mu_t|k, t_{tot} \sim$ Inv-Gamma$(0.5 + k, t_{tot})$.

Finally, multiple labellers are used to infer μ_t, $\mathrm{P}(\mathrm{peaks})$ and $\mathrm{P}(\mathrm{success}|\mathrm{peaks})$. Labellers are given data $\mathcal{D} = \{C_i\}_{i=1,\ldots,n}$ where C_i is a high resolution charge stability diagram produced by the algorithm at the final step of the investigate stage. If no high-resolution scan was acquired for iteration i then C_i is an empty array. $\mathcal{D}$ requires the labelling function $\psi_j : \psi_j(C_i) = 0$ if there are no double quantum dot transport features, or 1 otherwise. The set of all labellers is $\Psi = \{\psi_j\}_{j=1,\ldots,4}$. Labellers may not agree, and therefore I need to marginalise this effect. Let θ denote a parameter to be inferred μ_t, $\mathrm{P}(\mathrm{peaks})$ or $\mathrm{P}(\mathrm{success}|\mathrm{peaks})$, then $p(\theta|\mathcal{D}, \psi) = p(\mathcal{D}|\theta, \psi)p(\theta)/p(\mathcal{D})$. ψ

only affects the likelihood; to minimise the effect of ψ the posterior can be marginalised over ψ and approximated by samples of ψ [103],

$$p(\theta|\mathcal{D}) = \mathbb{E}_{\psi}\left[p(\theta|\mathcal{D},\psi)\right] \approx \frac{1}{|\Psi|}\sum_{\psi\in\Psi} p(\theta|\mathcal{D},\psi), \tag{5.1}$$

where the number of labellers is $|\Psi|$. The cummulative distribution function of $\theta|\mathcal{D}$ is [103],

$$P(\theta < z|\mathcal{D}) \approx \frac{1}{|\Psi|}\sum_{\psi\in\Psi} P(\theta < z|\mathcal{D},\psi). \tag{5.2}$$

The parameter space in which double quantum dots were found, by labeller majority vote, across all experimental runs varies greatly between devices (Fig. 5.6). In the 3D-parameter space shown in Fig. 5.6 the double quantum dot regime volumes are 1.21 V^3, 17.7 V^3, 0.215 V^3 for the FinFET, nanowire and heterostructure respectively. The FinFET (Fig. 5.6a) displays an almost planar voltage space across V_3, V_1 and V_4 which encapsulates the double quantum dots found. The nanowire (Fig. 5.6b) displays a much larger region in voltage space where double quantum dots were found across V_1, V_3 and V_5. In the parameter space pictured, there are areas of the double quantum dot regime volume which are more dense with double quantum dots than others. At the upper bounds of V_3 and V_5, double dots are present but sparse in the parameter space. The heterostructure displays the smallest volume across the parameter space of V_2, V_4 and V_6 and the double quantum dots are distributed more sparsely within the volume when compared to the two other devices.

Beyond the 3D volumes displayed in Fig. 5.6, the Algorithm explores the full dimensionality of the available parameter space (4, 5, and 7 dimensional for the FinFET, nanowire and heterostructure, respectively), allowing us to gain new insights into the variability of operating regimes. The double quantum dot regime convex hull volumes, defined by the shape of the smallest convex set which encapsulates all double quantum dots found in the entire parameter space, are 16.6×10^{-2} V^4, 23.8 V^5,

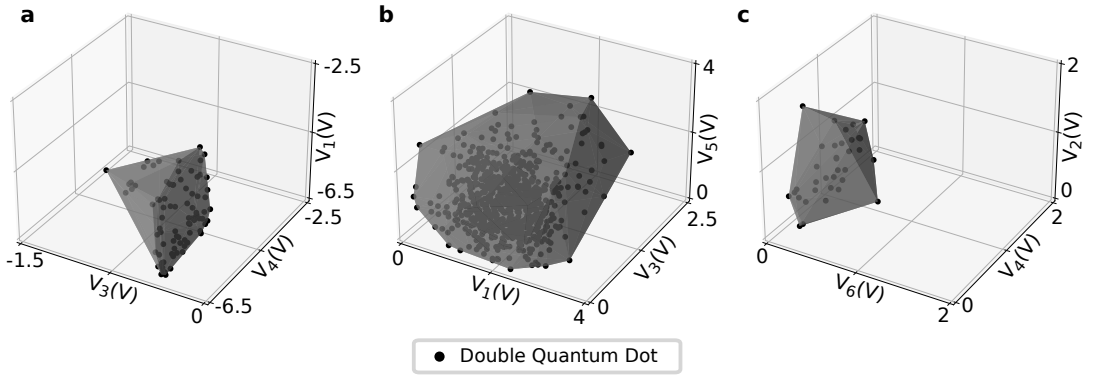


Figure 5.6: Double quantum dot regime volumes. Regions of voltage space (grey) encapsulate and define a volume where double quantum dots were found (black points) across all experimental runs of both random search and CATSAI in the FinFET (a), nanowire (b), and heterostructure (c). Three barrier gates are plotted for illustrative purposes, except for the FinFET (a) where the two lead gates and a plunger gate are plotted.

26.3×10^{-4} V^7 for the FinFET, nanowire and heterostructure, respectively. I name this metric, the Double Dot Voltage Space Volume (DDV). This metric is concerned with all N-dimensions of the gate voltage space of each device, as opposed to the double dot regime volume discussed earlier, which refers to the three dimensional plots in Fig. 5.6. Naturally, the convex hull volume calculation is sensitive to outliers as it encapsulates all double quantum dots found and confirmed by human labels. However, I use the convex hull because it offers a simple estimate of the upper bound on the size of the voltage space in which one may find quantum double dot features. Techniques such as alpha-shapes [135], could be employed to provide a more nuanced representation of the shape and structure of the DDV.

5.4 Discussion

Although accurate most of the time, the score function that the algorithm uses to detect double quantum dot regimes can sometimes be tricked by charge switches, as observed in Fig. 5.4i. The ideal double dot regimes found by the algorithm are tuned to a point where a specialised fine tuning algorithm such as that developed by van Esbroeck, N. M. *et al.* [112] would be suitable to follow on and optimise transport

features. However, this is out of the scope of this work, as here I focus on the coarse tuning of different device architectures.

For the Random Search and the first i iterations of CATSAI, the algorithm chooses pinch-off locations randomly, and thus $\mathrm{P}(\mathrm{peaks})$ does not show a definite trend (Fig. 5.5). For the subsequent iterations, I expect CATSAI to learn which are the promising locations in gate voltage space, and $\mathrm{P}(\mathrm{peaks})$ should thus increase as a function of time. Empirically, I observe a non-monotone increase, which is expected as the algorithm does not have a formal monotone guarantee. The saturation after 1–2 hours is expected given that transport features can only be found in a limited portion of the gate voltage space. The value of $\mathrm{P}(\mathrm{peaks})$ from the Random Search runs in the FinFET device and the heterostructure is inflated due to false positive classifications by the Coulomb peak detector, confirmed by human labels of all the current traces (see Appendix A). Random search performs differently across devices because it does not sample the hypersurface, nor the voltage space intelligently. Random search merely samples the voltage space at random and proceeds with the investigation stage. Therefore, the performance of random search is related to the probability of finding and correctly classifying single and double quantum dot features, the latter is linked to the DDV scaled by the search parameter space, which varies drastically between devices.

The difference between median tuning times (Table 5.1) for different devices begs the question whether the dimensionality of the gate voltage space is the key factor affecting tuning times or if there is a more subtle characteristic at play. The faster median tuning times were achieved in those devices for which the gate voltage space has fewer dimensions, i.e. the FinFET and the nanowire. Although the nanowire does have greater gate electrode dimensionality than the FinFET, faster tuning times are still observed for the nanowire. There would seem to be more double quantum dot regimes in the nanowire gate voltage space than there are in that of the FinFET. This hypothesis is reinforced by the lack of double quantum dot regimes found in

the FinFET by Random Search and it is in agreement with the experience of human experts when tuning these devices.

A reason for the lack of double quantum dot regimes found by Random Search in the FinFET is the sharp pinch-off that occurs as a function of the lead gate electrodes. The probability of finding lead gate voltages that enable current flow and plunger gate voltages that lead to double quantum dot regimes is inherently low. As mentioned previously, faster tuning times for FinFETs would thus be expected for CATSAI and Random Search if the lead gate voltages, V_1 and V_4, are fixed.

For FinFET in the 3D-parameter space shown in Fig. 5.6, V_1 and V_4 display their symmetrical weighting by forming the base of the plane which is almost square, and centred at similar points in voltage space with an approximate area of 2 V x 2 V. This supports the thesis that tuning times could be reduced by grouping the lead gate electrode voltages. In the nanowire, relatively high number of double dots across a wide parameter space provides evidence as to why the nanowire has the shortest tuning time and confirms that it has the most double dot regimes. The combination of the two factors of: small double quantum dot regime volume and sparse double dot distribution, supports why the tuning times were the longest for heterostructure in addition to its higher gate electrode dimensionality.

Access to DDVs allows us to understand the sensitivity of the quantum dots' confinement potential to each gate voltage value, and to examine the role of each gate electrode as plunger, barrier gates, etc. To put the DDVs into perspective, they can be scaled by the size of the parameter space of each device defined by the gate voltage bounds. The DDVs of the FinFET, nanowire and heterostructure occupy 0.174%, 5.95%, and 0.00206% of the device parameter spaces (95.1 V^4, 400 V^5, and 128 V^7) respectively. Comparisons between these values must take into consideration the effect of gate voltages on the confinement potential for each device architecture, the presence of disorder, strain, and material characteristics. Still, this percentage gives us an insight into the ease of tuning each device architecture. The Algorithm's

Device	FinFET	Nanowire	Heterostructure
Material	Si	Ge/Si	Ge/SiGe
Charge carriers	Holes	Holes	Holes
Operation mode	Accumulation	Depletion	Depletion
Gates controlled	4/4	5/5	7/7
Search space	95.1 V^4	400 V^5	128 V^7
1D-trace (mV/pixel)	0.78	1.56	0.78
Low-res 2D-map (mV/pixel)	9	5	9
High-res 2D-map (mV/pixel)	2.5	4.2	2.5
CATSAI times (mins)	30 (26, 37)	9.5 (6.7, 12)	92 (71, 120)
Random times (mins)	N/A	17 (9.9, 26)	360 (190, 830)
DDV	16.6×10^{-2} V^4	23.8 V^5	26.3×10^{-4} V^7
DDV/Search space (%)	0.174	5.95	0.00206

Table 5.2: CATSAI data summary table. The table includes the resolution of the traces and 2D maps taken in the investigation stage. The median devices tuning times from Table 5.1 are also included. For a breakdown of individual gate voltage limits for each gate electrode, the reader is referred to Appendix A.

proficiency in effectively exploring diverse DDVs across various device architectures showcases its versatility and robust capabilities. The DDV metric can be used to explore differing quantum device architectures and materials, and thus has a wide range of applicability.

5.5 Conclusion

I demonstrated fully automated tuning of gate defined double quantum dots across devices differing in material compositions and gate architectures (Table 5.2) with minimal modifications to the algorithm between devices. The algorithm also has provided insight into the parameter space of the quantum devices in the form of different shaped hypersurfaces that define pinch-off and different size double dot voltage space volumes.

I achieved fast tuning times in a Si FinFET, a GeSi nanowire and a Ge/SiGe heterostructure device, three different types of devices with very different characteristics. The tuning times reported are as low as 30, 10 and 92 minutes respectively. The capability to tune these devices from scratch completely automatically, prepares the

pathway laid out for the scaling of semiconductor qubits that lend themselves to industrial scale manufacture.

An analysis of the hypersurfaces corresponding to different device types and material systems could minimise variability and boost device performance by an informed device design. The size of the gate voltage space is also an important consideration in this context. While the FinFET and the nanowire gate-voltage spaces at mV resolution have approximately 10^{14} and 10^{17} pixels respectively, the median tuning times are only different by a factor of 3, and surprisingly the median tuning time is shorter for the nanowire device.

The heterostructure, with a gate voltage space at mV resolution of 10^{23} pixels, shows a median tuning time only 3 times longer than the nanowire. This would suggest that other factors, such as the design of the gate architecture and the disorder potential, might have a very significant role in how quickly a device can be tuned. Faster tuning times could be achieved by using device information, for example by grouping gate electrodes with similar functions. While the size of the gate voltage space is determined both by device properties and fabrication methods, the volume of the hypersurface and the volume of gate voltage space in which transport features are found could be useful to quantify device variability and to characterise and design different device architectures. This includes calculating how the hypersurface of a particular architecture is different between devices or thermal cycles via point set registration as well as Coulomb peak occurrence and Coulomb peak sensitivity within parts of the voltage space [115]. Additionally, my introduction of the new metric, Double Dot Voltage Space Volume (DDV), opens the door to understanding the sensitivity of the quantum dots' confinement potential to each gate voltage value, and to examine the role of each gate electrode as plunger, barrier gates, etc. between devices.

I expect the Algorithm to be successful in tuning geometries where gate electrode cross-talk is more considerable. Moreover, the machine learning-based approach

is geared towards navigating intricate parameter spaces rather than relying on a procedural algorithm workflow.

Radio-frequency reflectometry measurements would also lead to faster tuning times and the possibility of efficiently tuning large device arrays. This work evidences the potential of machine learning-based algorithms to find overarching solutions for the control of complex quantum dot systems.

Chapter 6

Donor Search

This is where the magic happens

Andrea Morello

The work contained in this chapter is based on a manuscript in preparation entitled, "Automatic tuning of a donor in silicon quantum device using machine learning" and was born out of a collaboration with Andrea Morello's Fundamental Quantum Technologies Laboratory in UNSW, Sydney, NSW, Australia.

Single shot readout [40], 30-second coherence times [29], and error correction threshold fidelities [71] have been realised in ion-implanted donor spin qubits in silicon. However, the realisation of a scalable ion-implanted donor spin qubit architecture is dependent on the reliable tuning of such devices to operational conditions, a challenge which has been previously untackled. I demonstrate the ability to locate and tune charge transitions in an ion-implanted donor in silicon device up to the point of readout calibration on the order of minutes automatically using machine learning. My algorithm tunes gate voltages to achieve two-level fluctuations in the readout current trace. Two-level fluctuations signify that the tunnel rate of an electron is slow enough to be detected by the experiment's readout bandwidth. Moreover, two-level fluctuations mean that the tunnel rates are slow enough that we can successfully detect electron spin readout and therefore form qubits (Chapter 2.4). The motivation for this chapter

is to demonstrate an algorithm capable of automatically tuning an ion-implanted ^{31}P donor device from zero gate voltage such that the tunnel rates on and off the electron site are near equal, and thus poised for spin-selective readout. My Algorithm enables both automatic characterisation and tuning faster than human experts. These results show once more, the capabilities of AI and computer vision tools for tuning quantum devices.

6.1 Introduction

The promised quantum computing architecture treasures held within ion-implanted donor spin qubits in silicon have led to their repeated pursuit and demonstration as a contending scalable quantum computing architecture ever since their proposition in 1998 [68]. Taking advantage of the core material of the electronics industry and its ever-finessed fabrication methods, placing a ^{31}P ion in originally natural silicon [136] and later an isotopically enriched ^{28}Si lattice, has been the source of single shot readout [40], 30-second coherence times [29] and error correction threshold fidelities [71]. However, the realisation of quantum computers built upon ion-implanted donors in silicon is not only 'dependent on future refinements of conventional silicon electronics' [68], but also the development of the approaches required to tune such devices to operational conditions automatically.

Universal fault-tolerant quantum computing requires more than 10^8 qubits [8] depending on the quantum algorithm, error correcting code, and quantum volume of the underlying architecture used. Therefore, an automated process for the turn-on and control of the quantum architectural units is necessary. To achieve such scale regardless of device variability, new signal processing routines and machine learning-enabled tools are required for automatic methods to analyse, classify and make intelligent decisions based on data acquired from quantum devices concurrently during the tuning process [126]. The vast waters of automatic tuning of ion-implanted

donors were once uncharted, until now.

Here I present, an algorithm called `donorsearch` that automatically tunes an ion-implanted ^{31}P donor device up to the point of electron spin readout calibration from scratch. My Algorithm utilises techniques inspired by tuning algorithms for gate-defined quantum dot devices [81, 101, 103, 104, 112, 118, 124, 137], while leveraging novel computer vision and embedded unsupervised machine learning-enabled methods to process and classify quantum transport signals synonymous with donor in silicon devices.

I demonstrate `donorsearch` on a phosphorus ion-implanted donor in silicon device. Unlike in gate-defined quantum dots, where gate electrodes are utilised to shape the quantum transport landscape into desired features, tuning an ion-implanted donor device is a true needle in a haystack problem. Voltages must be applied to gate electrodes to probe the extensive parameter space in search of a single implanted ^{31}P donor ion and its electron. Then, the gate electrode voltages need to be finely tuned such that the donor electron can tunnel off of the ^{31}P ion and be read out accordingly. To tackle this tenacious tuning task, `donorsearch` comprises three stages: coarse tuning, a handshake and fine tuning. The signal processing methods and modular architecture of `donorsearch` enables the Algorithm to be used as both a tuning and a characterisation mechanism for ion-implanted donor in silicon devices.

`donorsearch` is the first algorithm that caters to the automatic tune-up of donor in silicon devices and therefore devices with some of the longest coherence times in the solid state. The presence and evolution of tuning algorithms are important for the realisation of quantum computers built upon silicon-based nuclear spin architectures in order to automatically tune, initialise and optimise donor spin qubits. This must also occur alongside continual improvements in the fabrication of ion-implanted silicon electronics [69, 138, 139].

6.2 Methods

6.2.1 The device

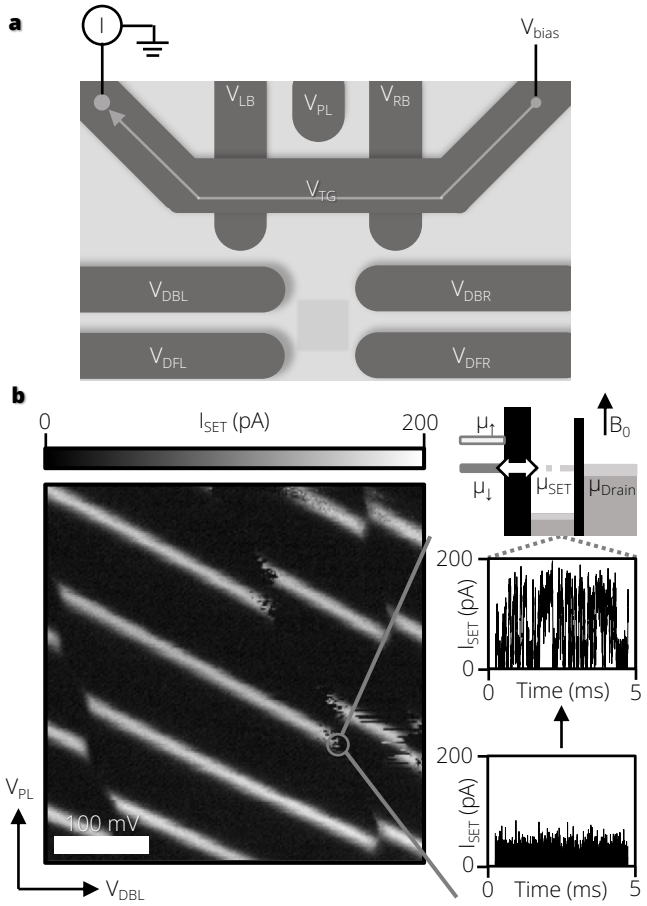


Figure 6.1: a) Ion-implanted donor in silicon device. Readout of the donor electron site is carried out by the single electron transistor (SET) defined by the gate electrodes V_{TG}, V_{LB}, V_{RB}, and V_{PL}. V_{bias} is applied at the source. The flow of current from the source to the drain underneath V_{TG} is shown by the arrow and measured at the drain. Four gates, V_{DFL}, V_{DFR}, V_{DBL} and V_{DBR}, are used to alter the electrochemical potential of the electron site. The approximate ion-implantation site window is denoted by the dark grey square. The Algorithm was given control of various gate electrodes during different stages of the tuning procedure. b) Algorithm overview. A charge stability diagram is acquired by sweeping V_{PL} and V_{DFL} and measuring the current flowing through the SET, I_{SET}. The `donorsearch` algorithm locates charge transitions (example encircled) within the charge stability diagram. `donorsearch` proceeds to tune the gate electrode voltages at the charge transitions to the point of random telegraph current signal observed on the SET. This corresponds to the electrochemical potential of the electron site ($\mu_{\downarrow}$) spin-down level becoming level with the SET's electrochemical potential (μ_{SET}), allowing for random loading and unloading of the spin-down level of the electron site in the presence of a magnetic field B_0, of 1.1 T.

`donorsearch` was demonstrated on a ^{31}P ion-implanted device in silicon which incorporates a single electron transistor (SET) for readout (Fig 6.1a) [40]. The ^{31}P donor is implanted via a focused ion beam within the implantation window outlined in Fig 6.1a. The electrochemical potential of the electron donated to the silicon lattice by the ^{31}P ion is controlled by the donor gate electrodes V_{DFL}, V_{DFR}, V_{DBL} and V_{DBR}. The electrochemical potential of an electron site in the vicinity of the donor gates can be plunged or raised by applying positive or negative voltages respectively to the donor gates. The SET is defined by the gate electrodes V_{TG}, V_{LB}, V_{RB}, and V_{PL}. By setting a bias voltage, V_{bias}, to the source contact and a positive voltage to V_{TG}, V_{LB}, V_{RB}, electrons are accumulated below V_{TG} and a conductive channel is created, allowing current to flow (I_{SET}) from the source to drain contacts. The conductive channel can be pinched off by the barrier gate electrodes V_{LB} and V_{RB} therefore creating the SET which is tunnel-coupled to the source and drain. The electrochemical potential of the SET can be raised or lowered by either the top (V_{TG}) or plunger (V_{PL}) gate electrodes.

The Algorithm, similar to how a human would tune a device, acquires a charge stability diagram of the donor electron site and the SET (Fig. 6.1b). Charge stability diagrams are acquired by sweeping a donor gate electrode (V_{DFL}) and the SET plunger gate electrode (V_{PL}) and measuring I_{SET}. Peaks in current, Coulomb peaks, signify voltages at which the electrochemical potential of the SET allows for electrons to tunnel from the source to the drain. Troughs in current signify the states where the SET is in Coulomb blockade. The relative capacitive coupling of V_{DFL} and V_{PL} to the SET can be extracted from the gradient of the Coulomb peak slopes within the charge stability diagram. Breaks in the Coulomb peaks signify a charge transfer event occurring at an electron site, such as an electron un-loading (loading) off (onto) an implanted donor (ion), i.e. charge transitions, detected by the SET. `donorsearch` identifies charge transitions within the charge stability diagram and proceeds to alter the gate voltages at the charge transition such that a random telegraph signal in I_{SET} is detected. In

the presence of a magnetic field (B_0) the electron site energy levels are Zeeman split. A random telegraph current signal corresponds to the electrochemical potential of the spin-down level of the electron site ($\mu_\downarrow$) being level with the electrochemical potential of the SET (μ_{SET}) [39, 40]. Achieving $\mu_\downarrow = \mu_{\mathrm{SET}}$ is the aim of the Algorithm, as it results in a productive orientation in gate-voltage space to carry out the calibration of readout levels and pulse sequences to confirm and observe spin selective readout. All the experimental runs were carried out in a dilution refrigerator at a base temperature of 20 mK and a B_0 of 1.1 T.

6.2.2 The `donorsearch` algorithm

The `donorsearch` algorithm consists of three main stages: coarse tuning, the hand-shake and fine tuning (Fig. 6.2). Each stage progressively fixates on a reduced number of regions in voltage space that may realise a donor electron spin qubit. Before starting `donorsearch`, V_{bias} is manually set to 1 mV and is fixed throughout the entire tuning procedure. This value for V_{bias} was chosen due to prior experience from tuning other devices of the same architecture. All gate electrodes are initialised to a value of 0 V.

The coarse tuning stage commences by sweeping all gate electrodes to their maximum and minimum permissible bounds to acquire the current range of the device. The current range acquisition is for characterisation purposes and is not used as information for the algorithm. The SET is then checked for turn on in the following manner; a current trace is acquired while simultaneously sweeping V_{TG}, V_{LB} and V_{RB} from 0 V to 2 V, their maximum permissible bound. All other gate electrodes are kept at 0 V. `donorsearch` checks for a factor of 100 change in current and a range greater than at least 1 nA in the output trace. If these conditions are not satisfied, turn-on is not achieved; the user is notified and the Algorithm stops. If successful, the approximate turn-on voltage is extracted by finding the voltage corresponding to the current value 100 times greater than the minimum current (offset) value in the output trace.

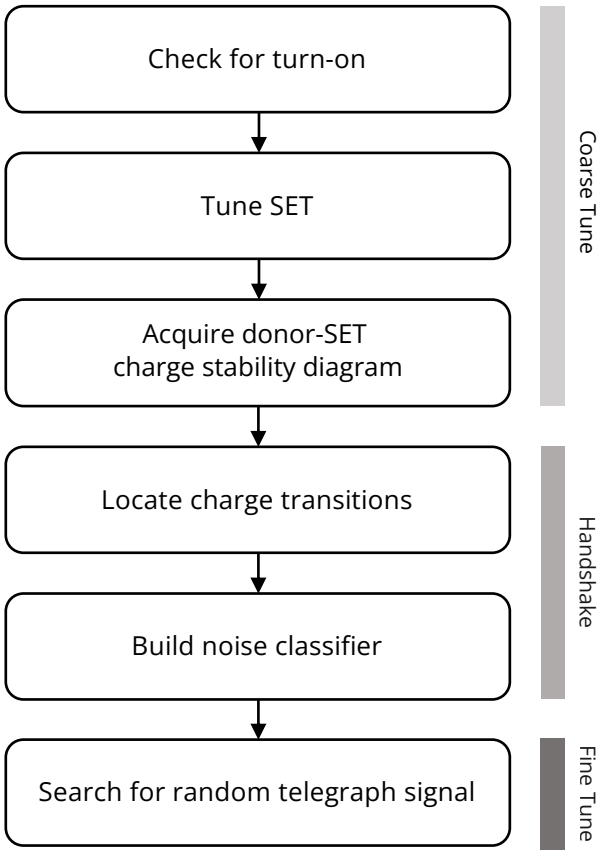


Figure 6.2: Outline of `donorsearch`'s workflow. V_{bias} is fixed before commencing the Algorithm. The coarse tuning stage consists of a procedural method to characterise, check the functioning of and tune the SET as well as acquire a charge stability diagram of the donor-SET gate voltage space. The handshake stage consists of signal processing routines to locate charge transitions within the acquired donor-SET charge stability diagram. Multiple current traces are acquired in the vicinity of the charge transitions to build a noise classifier utilised in the fine tuning stage. In the fine tuning stage, the Algorithm takes control of 7 out of the 8 gate electrodes and attempts to tune each charge transition site to the point of observing random telegraph signal in I_{SET}.

The SET is tuned by acquiring pinch-off current traces for each of the barrier gates and applying the respective pinch-off voltages to the barrier gates. V_{TG} is kept at 2 V while pinch-off current traces are acquired by sequentially sweeping V_{LB} and V_{RB} from their upper bound to their lower bound, and then returning to their upper bound before sweeping the next gate electrode. The output current trace of each sweep is first normalised, then smoothed with a Gaussian filter before applying a curve fit to facilitate the extraction of the corresponding pinch-off voltages (see Appendix B), as

carried out in other algorithms for tuning gate-defined quantum dots [101, 104]. If the curve fitting procedure is unsuccessful, it is assumed by `donorsearch` that pinch-off is unattainable, the user is notified and the Algorithm stops. The SET is therefore tuned by applying the respective extracted pinch-off voltages to V_{LB} and V_{RB} and keeping V_{TG} at its maximum bound of 2 V.

The coarse tuning stage is completed with `donorsearch` acquiring a 400 mV $\times$ 400 mV charge stability diagram of the donor electron site and SET (see Appendix B) by sweeping V_{DFL} and V_{PL}. This is an exploitative approach as it relies on belief and knowledge that typically one would find charge transitions in these devices in this region of voltage space. A more explorative version of `donorsearch` could be employed where the entire Algorithm workflow as described is unchanged, and it is repeated at various initial voltages collectively applied to the donor gate electrodes.

The charge stability diagram of the donor-SET voltage space acquired in the coarse tuning stage is passed to the charge transition locator module at the start of the handshake stage. Through a series of image analysis and computer vision techniques, the location of the charge transitions and the gradient of the Coulomb peaks in terms of $V_{\mathrm{PL}}/V_{\mathrm{DFL}}$ are extracted from the charge stability diagram. The location of each transition is visited as a starting point to acquire ten 30 ms current traces in the vicinity of each transition. This consists of recording I_{SET} for 30 ms at a 500 kHz sampling rate without altering the voltages applied to the gate electrodes at each point sampled in voltage space. The current traces acquired are used to build the noise classifier which is utilised in the fine tuning stage.

In the fine tuning stage, the Algorithm is given control of all the gate electrodes except V_{TG} which remains fixed at 2 V. Each charge transition identified during the handshake stage is visited in the fine tuning stage. The Algorithm searches in gate-voltage space within 5 mV of the transition on each gate electrode, acquiring 30 ms current traces at each point, with the aim of observing random telegraph signal in I_{SET}. To search the voltage space the Algorithm may use either Gaussian process

Bayesian optimisation or random search, similar to the approach taken in *Harmless Bayesian Optimisation* [140]. Both are used in this work with random search acting as a benchmark, as random search is simple to implement, requires no heuristics, and makes no assumptions about the underlying search space. Moreover, due to the noisiness of evaluating current traces for random telegraph signal, random search is less likely to miss out on promising locations of voltage space compared to more directed search approaches. An acquired current trace is first passed to the noise classifier, previously built in the handshake phase, to check whether the trace has characteristics of a two-level signal. The two-level signal is then scored by dynamically generating a current threshold and calculating how much time the signal spends above the threshold, T_{a}, and below the threshold, T_{b}. The current threshold is generated by applying a double Gaussian fit to the current trace and extracting the split between the two Gaussians as the threshold (see Appendix B). The score is calculated as the absolute difference between T_{a} and T_{b}, divided by the length of the current trace (30 ms). The Gaussian process Bayesian optimisation search method aims to locate the coordinates in voltage space with the lowest score value, corresponding to a score of 0, where $T_{\mathrm{a}} = T_{\mathrm{b}}$. The worst score is 1, meaning that the values within the current trace reside entirely below (or above) the current threshold for the duration of the current trace. A random telegraph signal is deemed to be located when the score is less than 0.1, at which point the Algorithm stops the search and fine tunes the next charge transition site. A score of 0.1 means that T_{a} and T_{b} differ by no more than 10% of the total current trace duration, corresponding to a difference of 3 ms. Current traces classified as noise (and not a two-level signal) by the noise classifier are given the worst score value of 1. The Algorithm samples up to 50 points in voltage space at each charge transition site before attempting to fine tune the next transition.

The Gaussian process kernel was Matérn 5/2, and the characteristic length scales ℓ_q, where $q = 1, \ldots, N$ where $N = 7$ corresponding to the 7 gates controlled by the algorithm, and σ_f the covariance amplitude, are optimised using maximum likelihood

estimation. I used the Matérn 5/2 because I expected the features (two-level current trace score) to vary with gate voltage in a smooth manner, not linearly or periodically due to the presence of cross talk and charge traps. I chose negative Expected Improvement as the desired acquisition function because the charge transition locator reduces the search space for the fine tuning therefore, initially less explorative sampling techniques are required such as the hypersurface sampling technique used in Chapter 5.

The charge transition coordinates that were successfully fine tuned to achieve random telegraph signal are output by the algorithm. These can then be visited by the user for spin-readout calibration. This would consist of performing a three-level pulse to calibrate the read-level [40]. The three-level pulse involves a sequence of three voltage pulses used to load and selectively read out electron spin qubits. The determination of the voltage coordinate for the read pulse is done iteratively to achieve spin-selective readout. The three-level pulse requires generating a virtual gate to pulse along the direction of the Coulomb peak of the transition to compensate for the cross-talk between donor (plunger) gate electrodes and the SET (electron site) [39]. The virtual gate is constructed by `donorsearch` automatically via the extraction of the Coulomb peak gradient during the image analysis process in the handshake phase.

Further details of the modules of the handshake phase are discussed in the following sections. This will include a discussion of K-means clustering aiding with the separation of the feature space for the noise classifier. Following that, a double Gaussian fit is used in the fine tuning stage to score two-level signals.

Charge transition locator

The novel charge transition locator module included in `donorsearch` extracts vital information from the donor-SET charge stability diagram (Fig. 6.3). The input charge stability diagram acquired in the coarse tuning stage is thresholded and binarised using Otsu's method [141]. High current regions of the charge stability diagram are

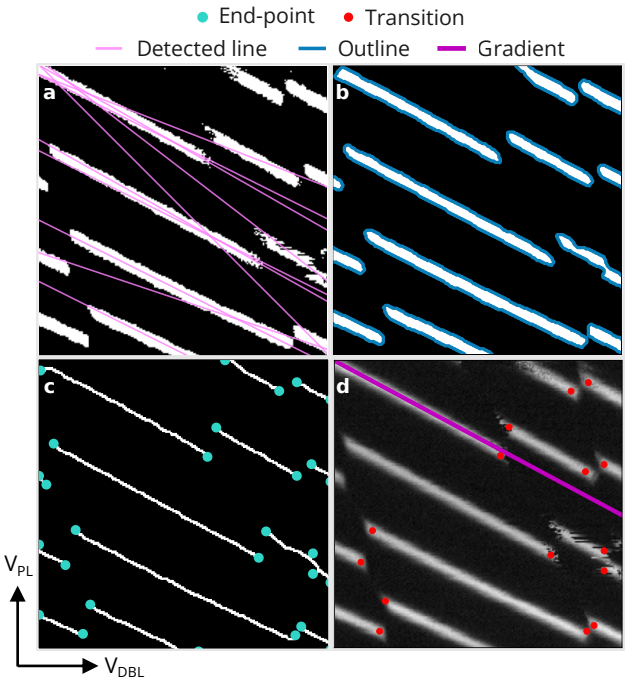


Figure 6.3: Overview of the charge transition locator module. The charge transition module is composed of multiple computer vision and image analysis techniques to extract the location of the charge transitions within a charge stability diagram. a) The charge stability diagram is first thresholded resulting in a binarised image, and lines within the image are detected using the Hough transform. b) Artificial Coulomb peaks are created by first Gaussian filtering the binarised image. Then, contours are located and therefore the outlines (blue) of the effectively smoothed (artificial) Coulomb peaks. c) The medial axis and the end-points (turquoise) of the artificial Coulomb peaks are identified. d) The original input charge stability diagram is used as a mask to filter out false Coulomb peak end-points at the edges of the image and false end-points generated in the previous step (c). The module returns the locations of the charge transitions in gate voltage space and the gradient of the Coulomb peaks, inferred to be the average slope of the lines detected in the first step of the charge transition locator module (a).

automatically separated from the background by exhaustively searching for a threshold which minimises intra-class intensity variance; high current regions are assigned a pixel value of 1, and low current regions a pixel value of 0. The lines in the resulting binary image are extracted by applying the Hough transform [142, 143], and the average gradient of the lines is inferred to be the Coulomb peak gradient (Fig. 6.3a, d). The gradient value can then be used to construct a virtual gate for read-level calibration via a three-level pulse sequence [39, 40].

To locate each charge transition site the charge stability diagram must first be

denoised and cleaned to remove artefacts as a result of the acquisition process (Fig. 6.3b). A Gaussian filter is applied to the binary image to smoothen the edges of the Coulomb peaks. Contours within the image and therefore the outline of the smoothed Coulomb peaks are extracted by applying the Marching Squares algorithm, the two-dimensional version of the Marching Cubes algorithm [144, 145]. The generated outline of the smoothed Coulomb peaks is used as a shape boundary which is internally filled to denoise the Coulomb peaks and create what I call artificial Coulomb peaks. The denoising and artefact removal step is crucial to reduce the number of false positive charge transition sites, such that the Algorithm does not waste time tuning false charge transition sites. Noise and artefacts can appear to be charge transitions if relying on fast rudimentary methods based on image gradient changes which are included in the `donorsearch` package but not used in this work. Image gradient-change-based methods may be effective for high-resolution stability diagrams without artefacts and good signal-to-noise ratio, typically those performed via much slower DC-source measurements.

The medial axis of each artificial Coulomb peak is calculated using the medial axis transform [146] resulting in a skeletonised image (Fig. 6.3c). The end-points of the medial axes, and therefore potential charge transition sites, are found by identifying pixels in the skeletonised binary image where the sum of nearest neighbour pixel values is equal to 1.

After further filtering of the end-points, the Algorithm returns the charge transition locations and the average Coulomb peak slope is found in the input charge stability diagram (Fig. 6.3d). The end-points are filtered as some may not correspond to charge transitions, for example, edges of the image or residual noise around the edges of the artificial Coulomb peaks may be interpreted as the end of a Coulomb peak. To filter the end-points the binary image (Fig. 6.3a) is used as a mask. Each end-point $\pm\Delta$ pixels along the average Coulomb peak slope must lie within the boundary of the binary image and the binary values at those pixels must be opposite. Moreover, no

end-point can be within r pixels of another. I set Δ and r to 5 and 3 pixels respectively; end-points which satisfy the conditions are believed to be charge transitions. This harsh filtering approach results in some false negative charge transitions but that is outweighed by the time saved in not fine tuning potential false positives. For tuning donor in silicon devices, I prioritise tuning speed due to the well-defined and abundant charge transitions in voltage space (Fig. 6.1). The fine tuning task is to tune the tunnel rates at a given charge transition such that one can measure spin-readout effectively. To realise a qubit, only a single charge transition with a measurable tunnel rate is required therefore I am willing to trade type I errors (false positives) for type II errors (false negatives) to maximise tuning speed.

Noise classifier

The current trace noise classifier was built using unsupervised embedded learning (Fig. 6.4). The classifier enables the Algorithm to separate current traces containing a two-level signal, signifying tunnelling events detected by the SET, from current traces which contain no tunnelling events, or noise. A 10-step random walk in gate-voltage space is initiated at each identified charge transition site (see Appendix B). The Algorithm controls all gates except V_{TG} which remains fixed at 2 V during the random walk. At each step, a 30 ms current trace is acquired. All the current traces acquired are scaled by the minimum current measured during the set of random walks. Features of each current trace, X_1 and X_2, are extracted before applying a K-means clustering algorithm with a target of two clusters,

$$X_1 = \frac{m-n}{i} \tag{6.1}$$

$$X_2 = \frac{m}{n} \tag{6.2}$$

where m, n, and i are respectively the maximum, minimum and mean of the 50 μs moving average of the current trace. In the case where the tunnelling rate of

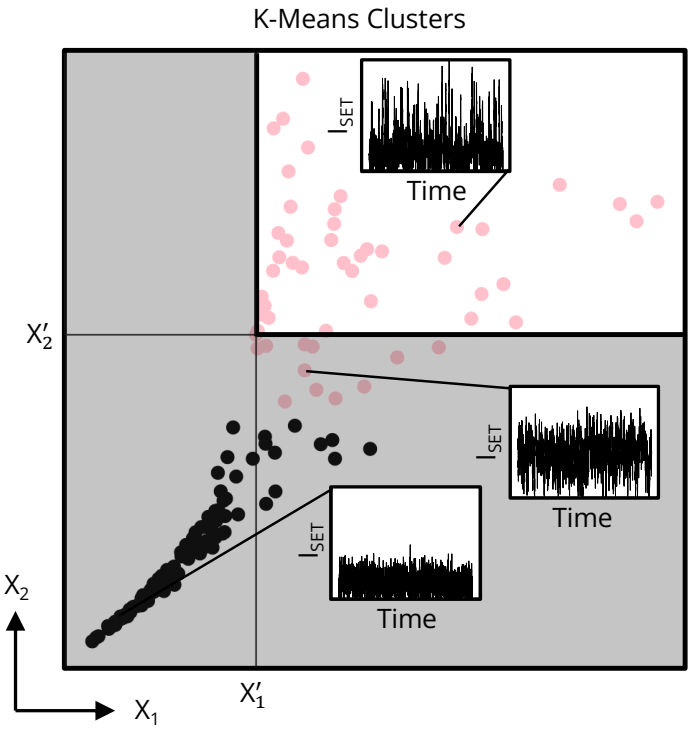


Figure 6.4: K-means clusters map of current trace features, X_1 and X_2, for the construction of the noise classifier. The K-means clustering algorithm targets two clusters. The clusters do not satisfy K-means assumption of circular groupings (Chap. 2.7.4) but, their rough identification can be used to separate regions of the feature space. Current traces containing mainly noise are clustered in the bottom left of the map, identified by the cluster of black points. Current traces containing two-level characteristics are identified by the pink coloured point cluster. There can be some bleeding of types between the clusters (middle current trace cutout). The feature values used as classification thresholds by the noise classifier in the fine tuning stage, (X_1', X_2'), are found by locating the first pink point along the X_1 axis. Current traces acquired after feature extraction which fall in the non-shaded region of the cluster map are considered two-level signals and checked for random telegraph signal whereas those which fall in the shaded region are considered as noise and not checked for random telegraph signal by `donorsearch`.

the electron is too slow to be detected within a 30 ms current trace or faster than the acquisition bandwidth of 500kHz, the current trace appears as just noise leading to a small difference between m and n resulting in near-zero values for X_1 and X_2 tending towards 1. As tunnelling events are detected, the difference between m and n increases resulting in an increase in X_2. X_1 also increases, and it varies greatly with the number of tunnelling events in a given trace due to the presence of the average current in its denominator. I use the 50 μs moving average to reduce the short-term jitter and improve the effective signal-to-noise ratio in I_{SET} when calculating X_1 and X_2. Current traces without two-level characteristics cluster in the bottom left of the K-means map, identified by the black points as the first cluster, whereas those that have two-level characteristics (pink points) form the second cluster which fans out to the upper right quadrant of the map. The clusters do not satisfy the standard K-means assumptions (Chap. 2.7.4), i.e. circular clusters, and are poorly identified. Therefore, we use what is provided by K-means to separate the feature space into different regions, that is one corresponding to two-level fluctuations and the other to noise. To extract the features that separate the two clusters efficiently, the algorithm scans along the X_1 axis and takes the coordinates of the first pink cluster point found (X_1', X_2') as the threshold features to be considered a two-level signal. The rejection area, shaded in grey, although not defined by the true separation boundary between the clusters, has the added benefit of reducing the number of false positive two-level current traces as a result of the feature threshold extraction method.

During the search for random telegraph signal in the fine tuning stage, X_1 and X_2 are first calculated and compared to X_1' and X_2' for each current trace acquired. If both X_1 and X_2 are greater than X_1' and X_2' respectively, the current trace is considered to contain two-level characteristics and is scored for random telegraph signal, otherwise, the acquired current trace is considered as noise and rejected.

Alternative methods of classification were attempted such as comparing the quality of single and double Gaussian fits, or setting a fixed current threshold in advance.

These proved not to be robust enough, completely disregarding two-level signals where the peak-to-peak values were relatively small or letting through too many false positive two-level signals. The unsupervised embedded learning method used in this work enables the Algorithm to 'learn' characteristics of noise and two-level current traces based on the device it is tuning. This method avoids the requirement of acquiring large and diverse data sets alongside time-consuming human labels to pre-train a classifier which is required for supervised methods [118].

6.3 Results

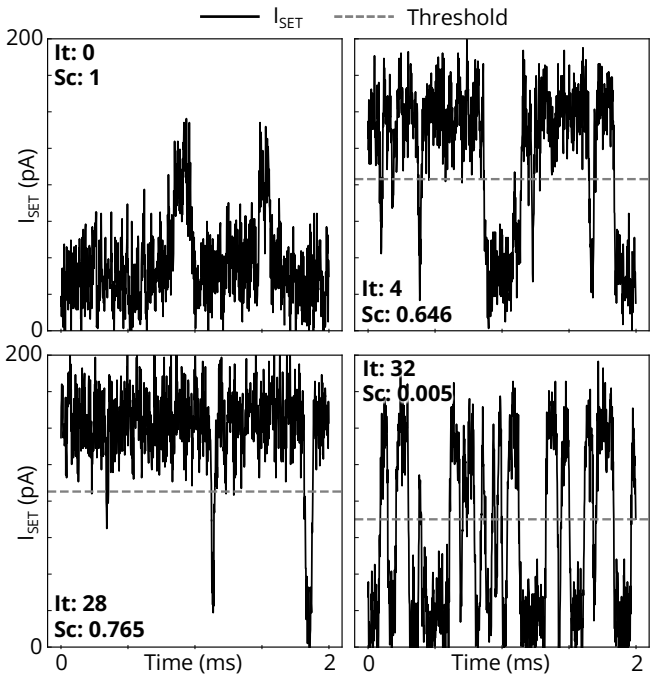


Figure 6.5: Current traces acquired during the search for random telegraph signal. Plotted are the first 2 ms of four current traces acquired at various iterations (It: 0, 4, 28, 32) during the search for random telegraph signal at a particular charge transition. Each current trace is scored (Sc) during the tuning process. Current traces which are classified as noise, iteration 0, receive a score of 1. Those that are classified as a two-level signal are scored using a dynamic current threshold (dashed line). Random telegraph signal is achieved once a score less than 0.1 is reached.

The Algorithm was run from an initial configuration of 0 V, for 28 times, using both random search and Gaussian process Bayesian optimisation in the fine tuning

stage for 14 repeats of each. Example current traces acquired during the fine tuning stage of a particular charge transition are shown in Fig. 6.5. Current traces deemed by `donorsearch` to contain random telegraph signal were labelled by two humans. I use the human labels to gauge the Algorithm's performance in tuning from nothing to random telegraph signal, the precursor to read-level calibration.

The charge transition locator module (Fig. 6.3) detected 13 charge transitions on average with a standard deviation of 2 charge transitions across all the 28 experimental runs. It calculated the Coulomb peak gradient to be -0.525 $V_{\mathrm{PL}}/V_{\mathrm{DFL}}$ on average with a deviation of 0.045 $V_{\mathrm{PL}}/V_{\mathrm{DFL}}$ across all experimental runs.

To gauge the performance of the noise classifier construction method (Fig. 6.4), current traces collected to train the noise classifier from an experimental run of `donorsearch` were labelled by two humans to check which current traces contained two-level signals. A balanced accuracy score of 88% was achieved by the noise classifier when using the unanimous vote of human labels as a positive classification of a current trace containing a two-level signal. This corresponds to the point (0.03, 0.80) on the ROC. A balanced accuracy of 86% was achieved when using the unanimous vote of human labels as a positive classification of a current trace not containing a two-level signal and being classified as noise. This corresponds to the point (0.28, 0.99) on the ROC.

The dynamic threshold for scoring current traces is applied to traces which pass the noise classifier (Fig. 6.5). The combination of the noise classifier and the dynamic current threshold enables reliable scoring of the current traces during the search for random telegraph signal in the fine tuning stage. Current traces which are noise or contain relatively low peaks in current (~ 100 pA) are classified as noise by the classifier leading to a score of 1. The average current threshold applied was 73 pA with a deviation of 9.5 pA, when analysing current traces across all experimental runs where all human labellers have agreed that the trace contained random telegraph signal. The following accuracy measures correspond to the point (1, 1) on the ROC as the

	Tuning Times (s)	
Stage	Gaussian Process	Random Search
Coarse	318 (315, 323)	
Handshake	117 (104, 145)	
Fine	132 (108, 163)	228 (167, 329)

Table 6.1: Median device tuning times broken down by each stage of the algorithm with 80% credible intervals (equal tailed). `donorsearch` fine tuning stage tuning times are split based on search method used, Gaussian process Bayesian optimisation or random search.

algorithm only returns the current traces which it predicts as a positive (i.e. containing random telegraph signal) for human inspection and labelling. When comparing the unanimous human labels to the current traces output by `donorsearch` deemed to contain random telegraph signal, `donorsearch` has an accuracy score of 66%. If I split the implementations of the fine tuning search methods, Gaussian process Bayesian optimisation results in an accuracy of 77% and random search has an accuracy of 50%. The overall accuracy increases to 82%, if I remove the unanimous vote requirement of the human labels and rely on a single human label confirming the presence of random telegraph signal. This accuracy increase is reflected in the different fine tuning implementations with the accuracies for Gaussian process Bayesian optimisation and random search being, 90% and 70%, respectively.

The human labels can be utilised to calculate the tuning times of fine tuning stage of `donorsearch` (Tab. 6.1). The tuning times for the fine tuning stage are calculated using the same Bayesian multilabeller-statistics used in Chapter 5. The times for the coarse and handshake stages are grouped regardless of the fine tuning search method chosen, because they follow the same procedure. The median tuning times for the coarse and handshake stages are 318 and 117 seconds respectively. The majority of the time spent in the coarse tuning stage is taken up by the initial checks and SET tuning as the acquisition of the donor-SET charge stability diagram takes 25 seconds out of the 318 seconds. The handshake stage is the fastest part of the tuning algorithm taking a median time of 117 seconds. 0.61 seconds is spent executing the charge

transition locator module, and the remaining time is for building the noise classifier. The fine tuning stage is the second fastest part of the `donorsearch` algorithm when using Gaussian process Bayesian optimisation to search for random telegraph signals taking a median time of 132 seconds. The machine learning-based search method is 40% faster than random search when comparing median tuning times.

6.4 Discussion

The charge transition locator illustrated in Fig. 6.3 behaved reliably across tuning runs for identifying charge transitions and extracting Coulomb peak gradients. Deviations in the number of transitions found by the module between experimental runs were mainly caused by the harsh filtering employed at the end of the module which was necessary to reduce the number of false positive charge transitions. Despite the rough long edges of the Coulomb peaks, the de-noising methods used in the charge transition locator module were successful, as charge transitions are routinely found at the ends of Coulomb peaks and not along their length.

The noise classifier has the benefit of being both accurate and built in situ with a minimal number of training samples. Current traces misclassified as two-level signals tended to have either relatively high noise levels or contain a periodic signal which points towards an alternative source of noise in the experimental set-up. Given the shape of the cluster of traces which contain only noise in the K-means cluster map (black points Fig. 6.4), future versions of `donorsearch` could use the K-means cluster map as a method of classifying the presence of two-level signals in a parameter space and the relative tunability of devices. A region of parameter space may contain charge transitions but, they may not be able to be tuned to a point where one can observe a two-level signal in I_{SET}. This can be because the tunneling rate of the electron at the transition is faster than the sampling rate of the measurement of I_{SET}. The lack of two-level signals in I_{SET} would be signified by the absence of the pink cluster in

Fig. 6.4 and the relationship between X_1 and X_2 to be well approximated by a linear fit. By leveraging the unsupervised learning characteristic of the noise classifier its versatility can be utilised to create more explorative tuning algorithms.

In the identification of random telegraph signals, experimental runs using Gaussian process Bayesian optimisation significantly outperform those relying on random search in accuracy by at least 20% regardless of the human label voting metric chosen. The poor performance in the random search runs points towards a failure from the noise classifier. The noise classifier is natively trained each run on traces near, $\pm$ 2 mV, to the charge transition. Whereas during the fine tuning stage, the bounds of the search space are increased to $\pm$ 5 mV which will likely lead to current traces acquired by `donorsearch` that trick the noise classifier due to its limited training set. This is further reflected by 27% of random telegraph signals found by random search were found at the point in voltage space on the first iteration of the fine tuning stage. This point in voltage space corresponds to the original location of the charge transition in gate-voltage-space returned by the charge transition locator module. Future tuning procedures could rely on more data collected across a wider parameter space to build the noise classifier while sacrificing tuning speed. Or a noise classifier pre-trained via supervised learning could be utilised. One could incorporate alternative search methods, such as greedy search where the algorithm moves in the direction of gate voltage space where the score appears to be best. But, there is the risk of getting stuck in local minima due to the complex nature of the parameter space.

A significant portion of the tuning procedure (52 %) is spent in the coarse tuning stage preparing the SET (Table 6.1). Moreover, the coarse tuning time variation across 14 runs is minimal with the 80% interval possessing a range of 8 seconds. The speed of the SET tuning method is limited by the slow 2 V DC gate-voltage sweeps and current measurements at a resolution of 1 pixel per 2 mV swept. Time is saved by the fast 2D scan acquisition of donor-SET charge stability diagram (see Appendix B). If the lever arms of V_{PL} and V_{DFL} were weaker than observed here one may need to

increase the size of the charge stability diagram acquired to observe charge transitions. This could be achieved with a large and slow DC gate-voltage scan or maintain speed by acquiring multiple smaller fast 2D scans at different DC gate-voltage intervals. If solely concerned with tuning speed and not device characterisation one could reduce the resolution by a factor of 2 of the SET gate electrode voltage sweeps without disturbing signal processing routines for detecting turn-on and pinch-off. To improve efficiency further, one may employ alternative sweeps to detect turn-on and tune to pinch-off. For example, defining a pinch-off current threshold target based on a percentage of the current range of the device and utilising binary search on the voltages applied to the SET gate electrodes to achieve pinch-off. However, it is useful to obtain high-resolution current traces for SET characterisation purposes and potential charge reset analysis by cryogenic illumination, where incident light is used to remove charge traps affecting gate pinch-off and turn-on voltages [147].

The process of building the noise classifier is the dominant portion of the handshake stage in terms of tuning time, taking a median time of 117 (104, 145) seconds with 80% credible intervals (equal-tailed). The range in build times depends on the number of charge transitions located in the previous step of the handshake stage as each transition will be visited and explored to build the noise classifier. The time spent in the noise classifier build stage could be reduced by reducing the number of current traces acquired in the vicinity of each transition. It is unlikely that a diverse enough dataset would be acquired to build the noise classifier via K-means clusters (Fig. 6.4), were the number of current traces acquired reduced from the 10 samples per transition acquired by `donorsearch` by default. Nonetheless, my Algorithm is able to create a classifier which has an in-the-field accuracy of 86% in under two minutes, and without the inclusion of augmented or simulated data.

The Gaussian process Bayesian optimisation search routine outpaces random search tuning speeds in the search for random telegraph signal by 40%. Similar behaviour has been observed in machine learning-enabled search methods benchmarked against

random search in the coarse tuning of gate-defined double dots in GaAs [103] and silicon-based architectures (Chap. 5) [118]. Although the limits of the fine tuning search space are tightly confined at a range of 10 mV at the located transition site on each gate electrode, the high dimensionality of the search space (7 gate electrodes), makes it difficult for random search to outperform a more intelligent search method. Random search does still perform surprisingly well given the dimensions, size of the search space and parameter space in which the desired random telegraph signal can be found. The fastest random search tuning times venture into the range of that achieved by Gaussian process Bayesian optimisation. These occurrences are unreliable given the 80% confidence interval of random search possessing a lower bound of 167 seconds and a range of 162 seconds, three times that of the Gaussian process confidence interval range. These results from my Algorithm show that machine learning-enabled search methods result in both reliable and relatively fast tuning of quantum devices.

6.5 Conclusion

`donorsearch` is the first algorithm of its kind which can tune an ion-implanted donor in silicon device up to the point of spin-readout calibration from scratch. I have shown tuning times are on the order of 10 minutes putting `donorsearch` on par, if not ahead of human experts tuning such devices to achieve two-level fluctuations in current. Median fine tuning times are almost halved if Gaussian process Bayesian optimisation is used in the fine tuning stage instead of random search. However, random search performs arguably well especially when compared to random search implementations of other algorithms (e.g. CATSAI in Chapter 5). I discussed that this could be due to the charge transition locator tool and with it, the reduced search space during tuning. The charge transition locator tool does not require any pre-training. `donorsearch` incorporates both coarse and fine tuning stages into a single package. The fine tuning stage is aided by an efficient and lightweight noise classifier which is

built via unsupervised embedded learning enabling fast tuning times.

Further fine tuning elements may be automated such as readout calibration, relaxation time measurements and electron spin resonance. This would require new and potentially artificial intelligence-assisted signal processing methods to detect particular quantum features such as 'spin-tail' effects [40] as evidence for spin-dependent readout. Moreover, an automated and intelligent method to benchmark and optimise properties such as Rabi frequencies and readout contrasts may provide insight into the current undecipherable incongruity of tuning these properties [148]. The previously unexplored waters of automatic tuning in ion-implanted donors in silicon, soon to be densely charted, will become archives of knowledge, opening doors to scalability, a wealth of information and a deeper understanding of these devices.

Chapter 7

ML for SPAM Analysis

> If variety is the spice of life, marriage is the big can of leftover Spam.
>
> Jonny Carson

The narrative of this thesis has been: 1. We need hundreds of millions of qubits to build a fault-tolerant quantum computer. 2. Silicon-based architectures are suitable because they can host many qubits per unit volume and the experience of a trillion-dollar semiconductor industry can be leveraged for their manufacture. 3. We need to use AI-enabled automation to get these qubits to operating conditions and control them. Few in the semiconducting spin qubit community are willing to address the elephant in the room, temperature. Yes, silicon-based architectures are more compact than their superconducting counterparts, but there is still not enough cooling power in current dilution refrigerators to cool a billion qubit silicon chip alongside control electronics to 20 mK. Our heralded scalable architecture will need to operate at (relatively) hotter temperatures of above 1 K. What does this mean for the control and readout fidelity of our qubits? In this chapter, I tackle this issue head-on, born out of a collaboration carried out between myself, a colleague of mine, Barnaby van Straaten, and the Australian full-stack quantum computing company, Diraq. Diraq were experimenting with qubits operating above 1 K temperatures, referred to as 'hot

qubits'. Their qubits showed remarkable tolerance to these temperatures, with single and two-qubit gate fidelities above 99%. However, evaluating state preparation and measurement fidelities was proving unreliable and inconsistent. The cause of these issues was that some aspects of the measurement sequence caused a finite probability of the spin state flipping. The motivation of this chapter and my contribution was to estimate state preparation and measurement errors of their hot spin qubits given a series of measurement traces. Additionally, provide a gauge of uncertainty of the state preparation and measurement error parameters. This chapter is based on the results from our collaboration which contributed to Ref. [27] and any use of the words "we" or "our" in this chapter refers to the co-authors of Ref. [27].

7.1 Introduction

Qubits - little beastly creatures. It is as though solving one problem leads you to the next, or perhaps that is simply the nature of research and innovation. Although the argument is held that spin qubits in silicon are a scalable architecture due to our capability to manufacture high qubit density at an industrial scale, it is forgotten that control and fidelity remain a problem at high temperatures.

Temperature must especially be considered when one factors in the mass and real estate of on-chip control electronics required to conduct a symphony of a billion qubits. Unable to provide the cooling power to operate a quantum processing unit (QPU) at millikelvin temperatures, so-called 'hot qubits' are obliged to operate at 1 K, orders of magnitude hotter [78, 97, 119, 149, 150]. Where noise is increased at higher temperatures, successful qubit operation has been demonstrated while suffering from relatively poor state preparation and measurement (SPAM) and low gate fidelities [38, 78, 97, 149], compared to previous experiments showcasing spin qubits in silicon in low noise millikelvin temperature environments. The ability to obtain a firm grasp of SPAM errors and fidelities is vital to prove the feasibility of spin qubits at elevated

temperatures.

Here I present a machine learning-based method, named `error-causation`, which analyses SPAM errors for spin qubit parity readout [151]. This method, based on a Hidden Markov model (HMM) [152], extracts the initialisation and readout fidelity of qubits based on a series of repeated parity readout measurements. The model also has the ability to calculate the probability of a spin flip occurring during a sequence of measurements and infer the true underlying spin state of the system based on observed data.

I demonstrate `error-causation` on simulated and experimental data from a silicon-metal-oxide-semiconductor (Si-MOS) device, obtaining initialisation fidelities up to $99.34 \pm 0.27\,\%$ at temperatures of 1 K [27]. I also investigate the prediction errors of `error-causation` on simulated data and show 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. The lightweight nature of the `error-causation` package means that it can be adapted and run online to provide real-time feedback for initialisation and reset of qubit states. This unleashes the potential of fast active reset of multiple qubits in the presence of live SPAM error analysis.

7.2 Methods

7.2.1 The device

Experimental data was obtained from two Loss-DiVincenzo qubits realised in a Si-MOS gate-defined double quantum dot device (Fig. 7.1).The plunger gates, P1 and 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 of each electron

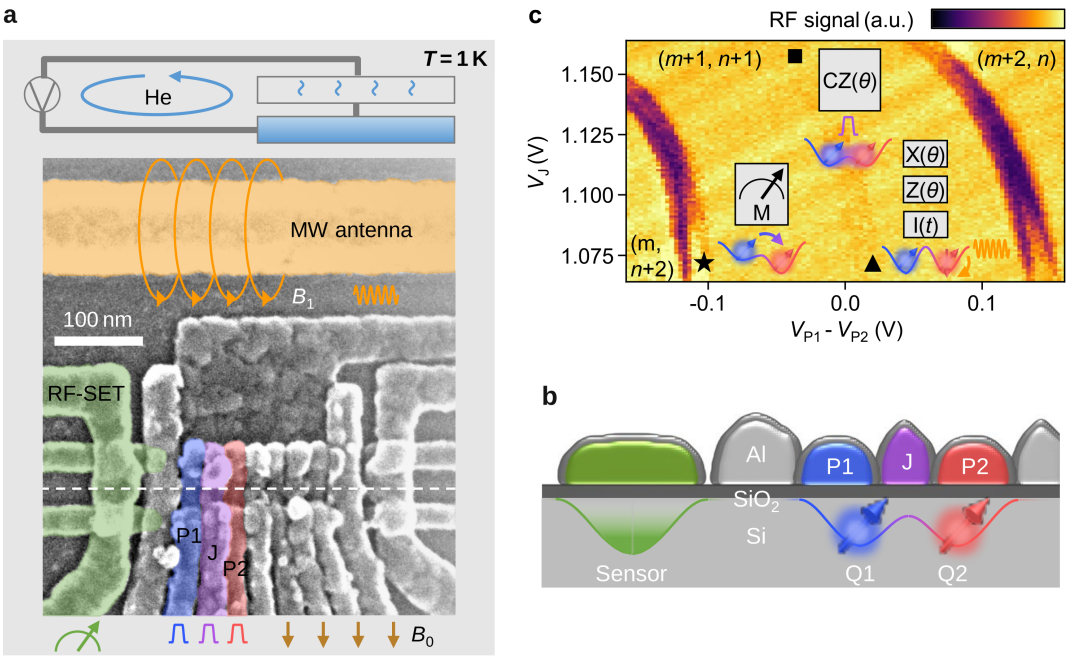


Figure 7.1: Si-MOS Device architecture. a) An SEM image of a device similar to that used in this work. Active gate electrodes and microwave antennae are highlighted in colour. B_0 and B_1 with their corresponding arrows are the external DC and the antenna-induced AC magnetic fields, respectively. The 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 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 (⋆), triangle (▲) and square (■), respectively. The insets schematically show the operations that are performed at each position. Reproduced from Ref. [27].

spin respectively. States where the spins of the electrons are parallel ($|\downarrow\downarrow\rangle$ and $|\uparrow\uparrow\rangle$), are referred to as 'even'. States where the spins of the electrons are antiparallel ($|\downarrow\uparrow\rangle$ and $|\uparrow\downarrow\rangle$), are referred to as 'odd'.

The states were measured via parity readout, which is a method based on Pauli spin blockade (PSB) [151]. A radio-frequency single-electron transistor (RF-SET) operating at 210 MHz was used for readout of the spin states. The RF-SET behaves as a charge sensor and is capacitively coupled to the adjacent quantum dots. Parity readout, instead of singlet-triplet readout, is the dominant readout method of the

quantum dots due to the large Zeeman energy difference between the two qubits [151]. 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 0 and 1 respectively. The classification data is then analysed a posteriori by an HMM, from the Python package hmmlearn [153], to extract the state preparation and measurement errors.

7.2.2 Hidden Markov models

Hidden Markov models (HMMs) are a classic method used in machine learning and statistics for modelling sequences such as speech [154] and proteins [155]. An HMM defines a probability distribution over sequences of observations (or emissions) $\vec{m} = \{m_0, m_1, m_2, ..., m_{N-1}\}$ by invoking another sequence of unobserved hidden discrete states $\vec{s} = \{s_0, s_1, s_2, ..., s_{N-1}\}$. These hidden states evolve according to a Markov chain, defined by a transition matrix, A, such that $A_{ij} := P(s_{n+1} = i \mid s_n = j)$. The probability of an observation is conditioned on the hidden state according to an emission matrix, θ, defined as $\theta_{ij} := P(m_n = i \mid s_n = j)$. In my case the emissions are discrete and binary and the emission model is a Bernoulli distribution. Finally, the probability vector, $\vec{\pi}$, encodes the probability of starting in a hidden state $\vec{\pi}_i = P(s_0 = i)$ [156].

To learn the parameters of an HMM, there are three important algorithms:

1. The *Baum-Welch* algorithm [157], which given a set of observations, $\vec{m}$, performs expectation maximisation to obtain the most likely set of HMM parameters specified by $(\vec{\pi}, A, \theta)$.

2. The *Forward-Backward* algorithm, which computes the likelihood of a set of observations, $\vec{m}$, given some HMM parameters, $(\vec{\pi}, A, \theta)$.

3. The *Viterbi* algorithm [158], which given a set of observations, $\vec{m}$, and a set of HMM parameters $(\vec{\pi}, A, \theta)$ it finds the most likely set of hidden states, $\vec{s}$.

7.2.3 The Categorical HMM

Fitting a model to the experimental data relies on defining a categorical HMM with two hidden states and two observables, also known as emission states. It is a categorical HMM because the observables are discrete and take on distinct values from a finite set for example, $[0, 1]$. This is opposed to, for example, Gaussian HMMs, where the observables are continuous and are assumed to be generated from a Gaussian distribution. Here, we take one of the simplest and traditional approaches, a standard expectation-maximisation-like HMM model. Future work could leverage advances in inference such as MAP, or variational Bayes methods [159], especially as future experimental work reveals more information about quantum device behaviour which may serve as reasonable priors.

I utilise a categorical HMM where the hidden state space is the qubit eigenstates, such that $\vec{s}$ consists of a sequence of the two hidden states $|0\rangle$ and $|1\rangle$, corresponding to the even and odd configuration respectively. The observation space is the measurement outcomes (after a threshold has been applied to the RF-SET data) such that $\vec{m}$ represents a sequence of even and odd state measurements, consisting of emission states of 0 and 1 respectively. With this model defined, the Baum-Welch algorithm is given the experimental observations and yields the best fitting initialisation probability vector $\vec{\pi}$, transition matrix A, and emission matrix θ. These vectors and matrices encode parameters of interest such that:

1. If $P_{\text{init, even}}$ denotes the initialisation fidelity into the even state, then $\vec{\pi} = [P_{\text{init, even}}, 1 - P_{\text{init, even}}]^T$.

2. If the fidelities of reading out the even and odd states are $P_{\text{read, even}}$ and $P_{\text{read, odd}}$,

then the emission matrix is

$$\theta = \begin{bmatrix} P_{\text{read, even}} & 1 - P_{\text{read, even}} \\ 1 - P_{\text{read, odd}} & P_{\text{read, odd}} \end{bmatrix}. \tag{7.1}$$

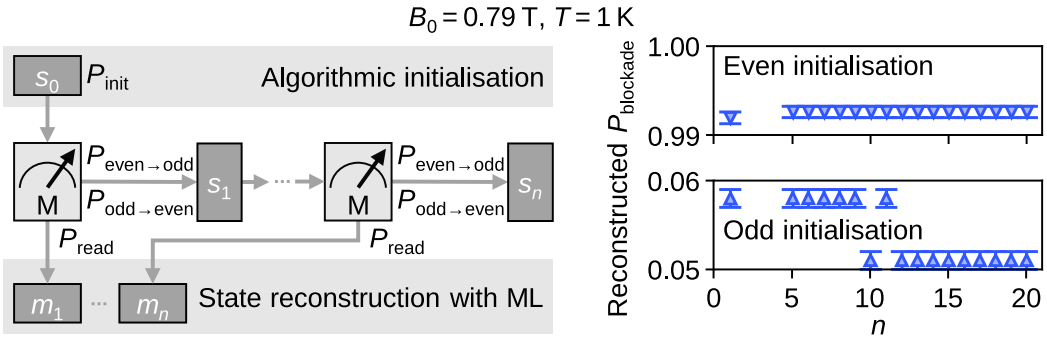
3. If $P_{\text{even}\to\text{odd}}$ and $P_{\text{odd}\to\text{even}}$ denote the probability of a measurement causing a transition from the even to the odd state and vice versa, then the transition matrix is

$$A = \begin{bmatrix} 1 - P_{\text{even}\to\text{odd}} & P_{\text{even}\to\text{odd}} \\ P_{\text{odd}\to\text{even}} & 1 - P_{\text{odd}\to\text{even}} \end{bmatrix}. \tag{7.2}$$

These vectors and matrices provide a method for disentangling the initialisation, readout and state transition errors from one another. To quantify the uncertainty in these parameters I used the Cramér-Rao bound [160], 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

$$\text{cov}_{\vec{\phi}}\left(\text{est}_{\vec{\phi}}(\vec{y})\right) \geq I\left(\vec{\phi}; \vec{y}\right)^{-1} \tag{7.3}$$

where $I(\vec{\phi}; \vec{y})_{ij} = -\partial^2 \log L(\vec{\phi}; \vec{y})/\partial\phi_i \partial\phi_j$ is the Fisher information matrix. Therefore, I 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 my case, the parameters are $\vec{\phi} = [P_{\text{init, even}}, P_{\text{read, even}}, P_{\text{read, odd}}, P_{\text{even}\to\text{odd}}, P_{\text{odd}\to\text{even}}]^T$ and the data is the measurement observations, such that $\vec{y} = \vec{m}$. The Forward-Backward algorithm is used to compute the likelihood $L(\vec{\phi}; \vec{m})$, and the gradients are calculated numerically using finite differences.



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

Figure 7.2: 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. State-preparation-and-measurement (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), $P_{\mathrm{even\to odd}}$ and $P_{\mathrm{odd\to even}}$, fidelities. Using the model, the underlying state and corresponding PSB measurement probability P_{blockade}, can be reconstructed. Reproduced from Ref. [27].

7.3 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, $n = 20$. The Hidden Markov model was used to reconstruct the states and perform state preparation and measurement error analysis (Fig. 7.2). Using the algorithmic initialisation developed in Ref. [27] 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, I can infer the state preparation, P_{init}, and measurement, P_{read}, fidelities as well as reconstruct the underlying qubit states based on the measurement outcomes $m_1...m_n$. Additionally, I can infer the probability of respective state changes, $P_{\mathrm{even\to odd}}$ and $P_{\mathrm{odd\to even}}$. The initialisation fidelities were inferred as $99.34 \pm 0.27\,\%$ and $94.67 \pm 0.73\,\%$, for $P_{\mathrm{init,even}}$ and $P_{\mathrm{init,odd}}$ respectively. The readout fidelities were inferred as $99.34 \pm 0.08\,\%$ and $96.15 \pm 0.44\,\%$, for $P_{\mathrm{read,even}}$ and $P_{\mathrm{read,odd}}$ respectively. The probability for a

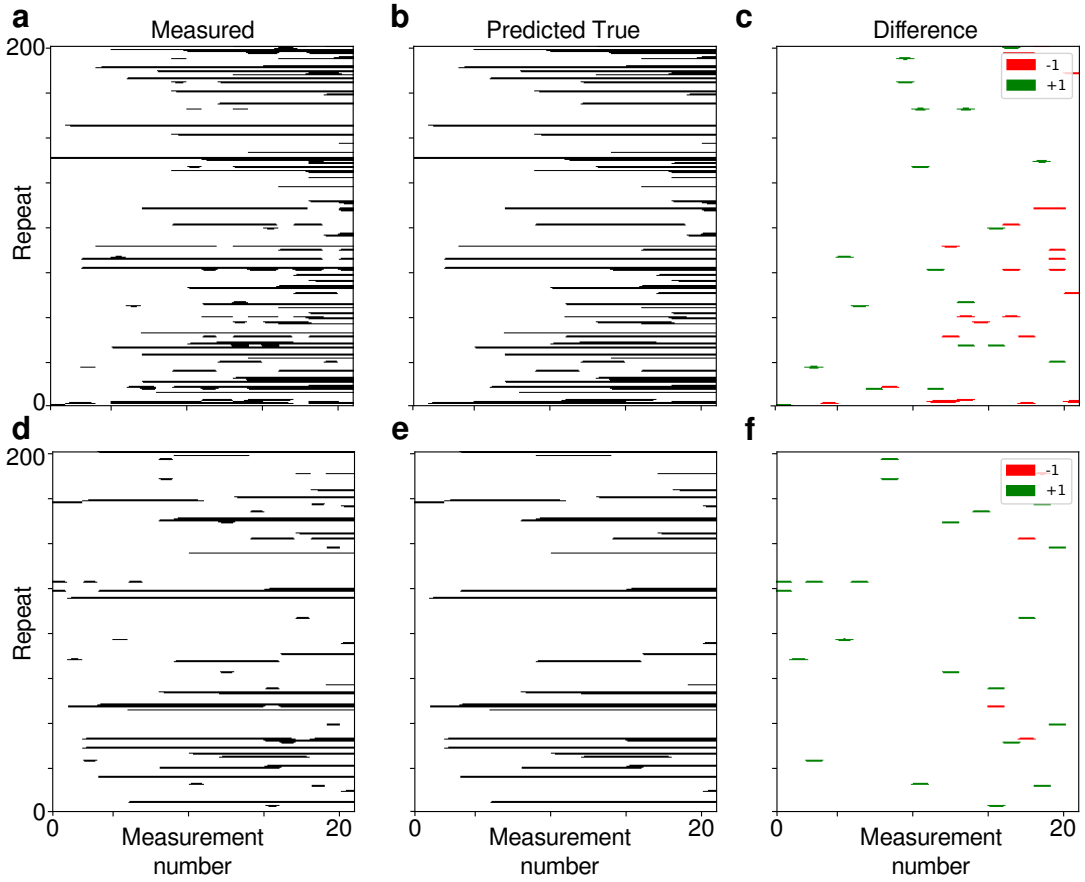


Figure 7.3: Experimental and simulated parity readout measurement outcomes and predicted states. The first 200 out of 1000 repeats of a 20-measurement sequence are shown. a) Real parity readout measurements from the Si-MOS device. b) HMM Predictions of the true underlying hidden state. c) The difference 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.

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_{blockade}, based on state reconstruction increases from $99.2\,\%$ to $99.3\,\%$ when $n = 5$ and the system is initialised into the even state. Conversely, P_{blockade} decreases from $5.8\,\%$ to $5.1\,\%$ at $n = 12$ when the system is initialised into the odd state.

To test and verify the performance of our HMM I generated simulated parity readout measurement outcomes, with the aim of the simulated dataset possessing similar qualitative characteristics to that of the real data from the Si-MOS device (Fig.

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

7.3). 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 performance of the Baum-Welch algorithm of the HMM and the parameters used to generate 1000 repeats of a sequence of 20 measurements, where each measurement is the classified PSB parity readout value, are shown in Table 7.1. 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 fall within reasonable bounds of the ground truth probabilities with the largest uncertainty being $\pm$ 0.004 and corresponding fitted value differing by 0.0017 to the ground truth of 0.0200. A reasonable uncertainty bound is 0.007 as this corresponds to the higher end of uncertainties seen during randomised benchmarking and gate tomography [161] of qubit gates on this and other devices under similar conditions [27, 78].

I checked the reliability of our measure of uncertainty by generating 1000 datasets of 1000 repeats of a 20-measurement sequence. I plotted histograms of the initialisation, spin-flip and readout probabilities output by the Baum-Welch algorithm in Fig. 7.4. The same ground truth values (Tab. 7.1) were used for each dataset and are shown by the dashed vertical lines in Fig. 7.4. The horizontal error bars correspond to one

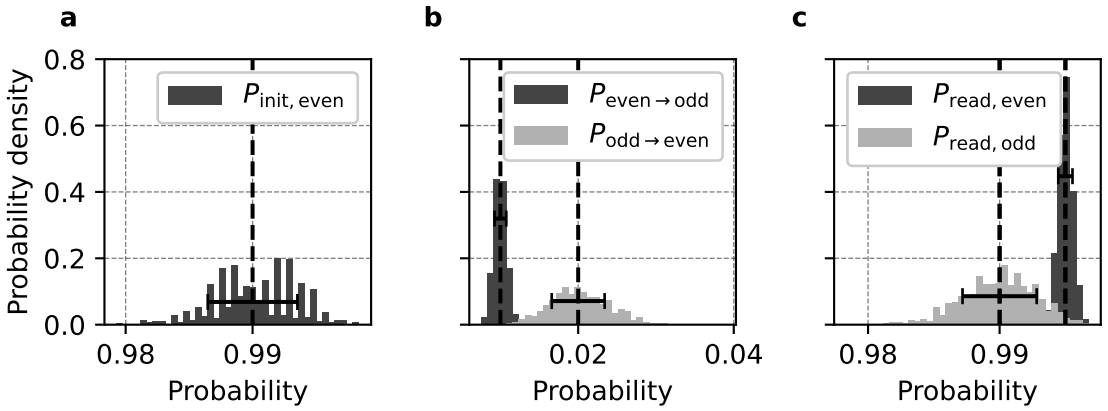


Figure 7.4: Histograms of the values fitted by the Baum-Welch algorithm when supplied 1000 randomly generated datasets, each consisting of 1000 repeats of a 20-measurement sequence. In addition, the ground truth value used to generate the dataset and the expected uncertainty, as per the Cramér-Rao bound, are overlaid as a vertical line and horizontal 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.

standard deviation computed by the Cramér-Rao bound.

I investigated the dependence of the Cramér-Rao uncertainty bounds on dataset size (Fig. 7.5). I computed the log-likelihood of each state probability parameter from increasing fractions of a simulated dataset consisting of 1000 repeats of a 20-measurement sequence. 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 dataset. The Cramér-Rao uncertainties decrease with increasing dataset sizes across all SPAM parameters.

7.4 Discussion

The low uncertainty on the SPAM parameters on the simulated data, where the ground truth is known (Fig. 7.3), strengthens the confidence in the SPAM parameter values in the Si-MOS device (Fig. 7.2). 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 which becomes a factor when $n > 20$ (Fig. 7.3).

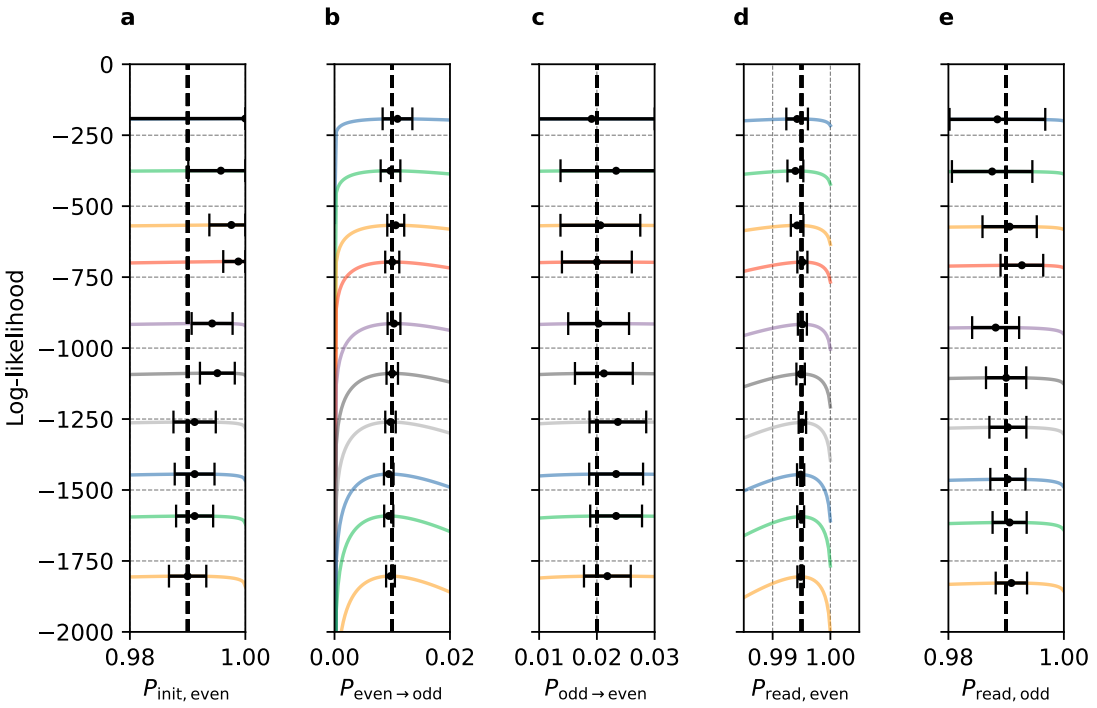


Figure 7.5: (a-e) Slices through the log-likelihood landscape for the Hidden Markov model detailed in Table 7.1 generating a dataset of increasing fractions of a dataset of 1000 repeats of a 20-measurement sequence. From top to bottom, the curves correspond to $10\%, 20\%, ..., 100\%$. The ground truth value used to generate the dataset for each parameter slice is overlaid as a dashed line. For each dataset size, the most likely parameter value and its corresponding uncertainty are plotted as a point with error bars.

In both the simulated and real dataset SPAM analysis, I estimate high values of uncertainty for P_{init} (Fig. 7.2 & Tab. 7.1). This is because, relative to the other events such as spin-flips or readout, initialisation only happens a limited number of times (Fig. 7.2), i.e. the number repeats of measurement sequences is lower therefore fewer statistics result in a relatively high uncertainty. $P_{\text{read,even}}$ has the lowest uncertainty out of all the parameters as there as so many readout examples to train on.

Approximately two-thirds of the SPAM parameters predicted 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. 7.4). 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.

Additionally, the Cramér-Rao bound decreases with increasing dataset size (Fig. 7.5) providing further support for its use as a suitable measure of uncertainty. Empirically, for large datasets, a parameter's uncertainty ϵ, scales with the dataset size, N, such that $\epsilon \propto 1/\sqrt{N}$. Moreover, the decrease in uncertainty is reflected in the log-likelihood maxima becoming more prominent with increasing dataset size. As demonstrated in both the real and simulated datasets (Fig. 7.2 & Tab. 7.1), 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 datasets (Fig. 7.5).

7.5 Conclusion

I successfully demonstrated the use of 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 $99.34 \pm 0.27\,\%$ and $99.34 \pm 0.08\,\%$ respectively. I verified the use and performance of our HMM for SPAM error analysis on simulated data and extracted the ground truth simulated states, while also achieving SPAM probabilities and respective uncertainties that provide confidence in our method.

The lightweight nature of HMMs lends itself to being implemented on field-programmable arrays, which will enable fast active reset of qubits at elevated temperatures based on online measurements and live signal processing. Such developments are vital contributions to the growing drive towards rapid cryogenic stage on-chip multiplexing, signal processing and control of quantum devices [162–168], an inevitable evolution to achieve a universal fault-tolerant quantum computer based on spins in semiconductors.

Chapter 8

Conclusion & Outlook

What did you discover?

Andrew Briggs

In 1492 there were no horses in America [1]. In 2019, there were no automatic methods for the tuning of quantum dots in silicon across a range of architectures [169]. The silicon spin qubit community was uncertain that automatic methods were needed for both the scaling and efficient iterative development of their architecture. Additionally, they did not have a solid grasp of how to build the required technology. This body of work has cast rays of light into the unknown shadows of automation for silicon-based spin qubit architectures. Here, I will lay out the nature and texture of that light which now shines, and outline the ways in which we can continue to reveal the shapes of scalability among the dimly lit shadows.

8.1 Conclusion

I started this thesis by building and demonstrating the first cross-architecture tuning solution using artificial intelligence (CATSAI) for gate-defined double quantum dots in silicon and germanium. By training CATSAI to recognise single quantum dot features in a range of silicon devices using a Random Forest classifier, CATSAI tuned a silicon

FinFET, a Ge/Si core/shell nanowire and a Ge/SiGe heterostructure from scratch to double quantum dots within minutes. The efficient sampling of the parameter space by CATSAI provided insight into the nature of the devices being tuned which has never been seen before. This included the shapes of the surfaces within the parameter space which divide regions of relatively high current from low current and the volumes of the parameter space in which a double quantum dot can be formed.

Building on the knowledge gained from developing CATSAI, I created a more exploitative algorithm, `donorsearch`, for the automatic tuning of an ion-implanted donor spin qubit silicon device up to the point of spin-readout calibration within 10 minutes. By incorporating computer vision techniques I was able to automatically identify charge transitions. By using embedded unsupervised learning and Gaussian process Bayesian optimisation I automatically tuned the voltages applied to gate-electrodes to obtain random telegraph signal in the current flowing through the readout transistor - the precursor to calibrating spin readout. The signal processing techniques developed for these devices allow for rapid device characterisation and analysis. This is the first automatic tuning workflow for ion-implanted donor spin devices and is the closest we have gotten to achieving a complete zero-to-qubit tuning solution in silicon.

Finally, I demonstrated the use of machine learning to extract qubit state preparation and measurement parameters for a two-qubit Si-MOS device operating at temperatures above one kelvin. Using Hidden Markov Models, I extracted the true underlying qubit state in the presence of faulty state preparation and readout on both simulated and real data. This contributed to work which showcases silicon qubits operating at elevated temperatures, providing further support for their use as a scalable quantum computing architecture.

Looking forward, I see roads diverging in a yellow wood. I will now describe the pathways I see laid before me.

8.2 Machine learning for quantum dots

Machine learning can be used to achieve a complete end-to-end tuning algorithm. End-to-end means that we can go from nothing to spin qubits completely automatically. An operator would load their silicon device into a dilution fridge, hook up the tuning software to their instruments via an interface and click run. At some point along the direction of the arrow of time, qubits are found by the algorithm. There are multiple ways in which I envision such an algorithm working. One could build a sequential pipeline consisting of the algorithms developed in this work and others referenced. For example, the coarse tuning of double quantum dots could be performed by CATSAI (Chapter 5). Subsequently, a fine tuning algorithm such as that developed by Van Esbroeck *et al.*[112] could be used to optimise electron transport features to form bias triangles. Nguyen *et al.*'s algorithm [170] could be utilised for the efficient search of the parameter space to locate the bias triangles. Finally, one could include a Pauli spin blockade classifier to confirm the presence of qubit spin physics, such as the one developed by Schuff *et al.* [171]. Alternatively, the fine tuning stage could be performed by CATSAI, progressively moving its origin start point towards the region of parameter space where double dots are found, reducing the size of the search space while continually scoring and ranking double quantum dots found in the investigation stage.

A simpler non-machine learning search method could be used during the coarse tuning stage to tune to double quantum dots faster, similar to that carried out by `donorsearch` and envisioned by D. T. Lennon [115]. By taking advantage of our knowledge of gate-electrode purposes, i.e. plunger gates and tunnel barrier gates, we can carry out procedural measurements such as finding pinch-off sequentially on the relevant gates to coarsely tune a double quantum dot. I believe algorithms of this nature are promising for device architectures with linear gate arrangements such as FinFETs, nanowires and linear arrays. However, I am unsure that they will generalise well to architectures where gate-electrodes have high degrees of cross-talk

or are overlapping. Moreover, their procedural nature gives the impression of inherent brittleness, unable to adapt for example, if a single gate electrode is broken.

8.2.1 Challenges

There are multiple steps to be traversed to build an end-to-end algorithm. An accurate double quantum dot classifier is needed; this could then be incorporated into a handshake stage to confirm that coarse tuning is finished and fine tuning may commence. Such a simple feature classifier will be an interesting feat as we have seen from Chapter 5 that not even human experts can agree on the presence of a double quantum dot. For the fine tuning stage, algorithms that have previously been demonstrated on a given materials system may find it difficult to perform as well in a more noisy and charge-switchy environment, and may require retraining on additional data.

At the confirmation of spin physics stage, we already have a classifier developed by Schuff *et al.* [171] which confirms the presence of Pauli Spin Blockade in FinFETs. It is yet to be demonstrated to be effective in other material systems. Additionally, it may be easier to confirm the presence of a qubit via ESR or EDSR. Utilising ESR or EDSR will require new signal processing and classification modules. Finally, there is still work to be carried out for the automatic optimisation of qubits properties, setting a whole new host of challenges.

Here, I am focusing on the tuning problem of a double quantum dot. We need billions of quantum dots to build a quantum computer. Tuning and signal approaches will have to adapt accordingly as silicon-based architectures scale and we observe features in transport or reflectometry yet to be seen in the semiconductor spin qubit community. Measuring quantum dots and qubits in transport becomes challenging as the array of quantum dots scales. It is therefore likely that future quantum dot tuning approaches will build upon foundations laid by van Straaten *et al.* [137] and Hickie *et al.* [131], and take advantage of RF-reflectometry and frequency multiplexing for

measurement, compensation and control.

To achieve what I have laid out, the community will require access to more data from quantum devices. Ideally, this will be in the form of an open-access structured database, that we see in other communities, for example, the Universal Protein Knowledgebase [172] in the Life Sciences, with the inclusion of accurate metadata. The open-access nature will enable the rapid development of machine learning models and signal processing routines to accelerate the field. In regard to the hardware, the move towards reliable (foundry) fabricated devices is crucial for the growth of the field. Proof of concepts have been demonstrated and now is the time to truly leverage our knowledge from the semiconducting industry. I can envision a near future where silicon foundries sell qubit-ready chips due to their ability to rapidly iterate fabrication and automatic tuning on an industrial scale.

Appendix A

CATSAI

A.1 Supplementary Methods

A.1.1 3D hypersurface plots

The 3D plot of the hypersurface for each device was generated by relying on the same method that CATSAI uses to generate the hypersurface of each device as it proceeds to coarsely tune it. The main difference being that no sampling is involved; the surface is generated by a model that makes use of the pinch-off locations detected during an algorithm run selected at random (CATSAI run 10). The model of the hypersurface used was a Gaussian Process (Matérn52 Kernel). This model is then used as an interpolation method to generate the 3D plots; regularly spaced points in gate voltage space are considered and the model is used to determine whether these points lie within the hypersurface. The gate electrodes not considered for the plots are kept constant at their respective mean gate voltage values for which pinch-off was observed during the experiment (Table A.3).

A.1.2 CATSAI's workflow

For the first i (12) iterations of the sampling stage (Fig. A.1), the algorithm selects a vector $\boldsymbol{u}$ at random in the gate voltage space of the device since the algortithm

is unaware of the characteristics of the device. This vector consists of all the gate voltages considered for tuning. The algorithm then sweeps the gate voltages along that direction until pinch-off occurs.

After the i^{th} iteration, a model of the hypersurface is built using a Gaussian process and $\boldsymbol{u}$ is chosen by incorporating the knowledge gained during the peak detection module in the investigation stage. The algorithm achieves this by generating a set of candidate pinch-off locations on the hypersurface and using the probability of finding Coulomb peaks in a given location of gate voltage space, $\tilde{\mathrm{P}}_{\mathrm{peaks}}$, as a weighting for the choice of $\boldsymbol{u}$ [103]. Using Thompson sampling [133], the algorithm then selects one of the candidate pinch-off locations, defining a new $\mathbf{u}$. In each of the following iterations, the pinch-off locations and the information gathered by the peak detection are used to update the hypersurface model and $\tilde{\mathrm{P}}_{\mathrm{peaks}}$, respectively.

For the low resolution current map score function in the investigation stage, the algorithm is given the noise floor and the current at which Coulomb peaks can be segmented, a current value between the noise floor and the peak of the Coulomb peaks, the segmentation threshold. For simplicity, measurements of the safe bounds, noise floor and segmentation threshold are taken manually before running CATSAI; these measurements can easily be automated.

A.1.3 Coulomb peak detector

Due to the different types of current noise observed for each of the devices considered, a robust Coulomb peak detector was required. We thus developed a Random Forest Coulomb peak detector.

A set of 128-pixel current traces was obtained running the tuning algorithm developed by Moon *et al.* [103] on different devices to those for which CATSAI was tested (Table A.1); two different 5-gate GeSi nanowires (400 mV-long current traces), and a single 3-gate Si FinFET (200 mV-long current traces). We gathered 1095 current traces from GeSi nanowire device 1, 1321 from GeSi nanowire device 2, and 4306 from

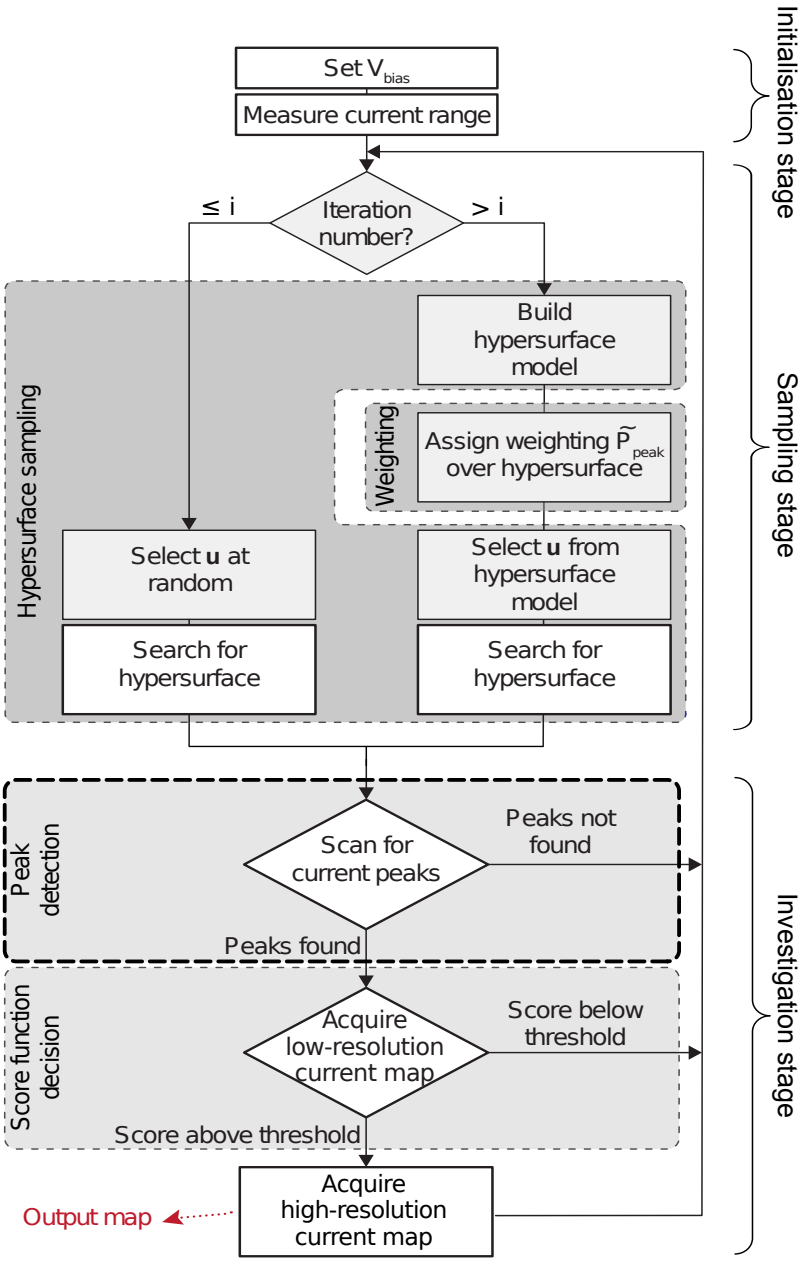


Figure A.1: CATSAI's workflow. For the first i iterations (left-hand branch of the sampling stage), the algorithm selects $\boldsymbol{u}$ at random and travels along it until the hypersurface is found. After the i^{th} iteration (right-hand branch of the sampling stage), the algorithm selects $\boldsymbol{u}$ based on the model it generates of the hypersurface and of the probability of finding Coulomb peaks in a given location in gate voltage space, $\tilde{\mathrm{P}}_{\mathrm{peaks}}$. In the investigation stage, the algorithm sweeps the plunger gates to generate current traces and low-resolution and high-resolution current maps if the conditions are satisfied for each classifier. Figure adapted from [103].

the Si FinFET device 1. I labelled the 6722 current traces, from which there were 553 labelled as positive (current traces containing Coulomb peaks) and the remainder

(6169 current traces) were labelled as negative. 553 negative examples were randomly picked from the shuffled 6169 negative examples, to make up an even dataset of 1106 current traces. The breakdown of the data subsets include, for the positives: 115 traces from GeSi nanowire device 1, 100 from GeSi nanowire device 2 and 338 from the Si FinFET device 1. For the negative subset: 83 from GeSi nanowire device 1, 113 from GeSi nanowire device 2, and 357 from the Si FinFET device 1. Randomly chosen current traces from the even dataset of 1106 current traces were used to train and test the Random Forest Coulomb peak detector; 70% of the traces chosen were used to train the classifier, and 30% were used to test it. No characteristic feature engineering or data pre-processing was done other than normalisation. The characteristic features the Random Forest classifier was trained on were the normalised current values of each trace at each pixel point, thus each sample had 128 characteristic features. The classifier relies on the Scikit-learn's ensemble RandomForestClassifier package [134]. An accuracy of 84% was achieved. The Random Forest classifier was then retested on 1562 current traces from a 5-gate Ge/SiGe heterostructure device 1 and an accuracy of 92% was achieved (Table A.1, Test 2). This relatively high accuracy contrasts the Coulomb peak detector used in Ref. [103], which achieved an accuracy of 20% classifying the current traces obtained for the Ge/SiGe heterostructure device 1.

A.1.4 Coulomb peak detector: online performance

I labelled all the current traces after the CATSAI and Random Search experimental runs were complete. My labels were compared against the labels predicted by the Random Forest Coulomb peak detector used in the experiment (Table A.2). The accuracy of the Coulomb peak detector is as follows, FinFET: 82.1%, 71.3%, nanowire: 86.0%, 82.3%, and heterostructure: 63.7%, 79.7% for all the Random Search and Full Decision runs respectively.

A.1.5 Algorithm configuration for the different type of devices studied

Across all devices the initialisation of the algorithm is set to 12 iterations (the first 5% of the total number of iterations for each run). In this work we did not apply any pruning rules [103]. When searching for the hypersurface, the algorithm looks for current drops below 0.5% of the maximum current range. The parameters chosen to run the algorithm can be separately optimised. The model of the hypersurface is built via a Gaussian Process as in Ref. [103].

Other configuration parameters depend on the type of device to be explored (Table A.4 & A.5). These parameters include: voltage bounds (origin and limit) set for each gate electrode to prevent device damage, the value at which the bias voltage is fixed, the noise and segmentation thresholds, and the size in gate voltage space of current traces (diag_trace), as well as low and high resolution current maps (2d_lowres and 2d_highres, respectively).

During the investigation stage the current traces have a length of 128 pixels, the low resolution current maps have a size of 16 $\times$ 16 pixels, and the high resolution have a size of 48 $\times$ 48 pixels. The dimensions of the traces and the scans in voltage space are device dependent (Table A.5).

The bias voltages were chosen to be slightly larger than typical charging energies expected for single quantum dots in each device. The noise and segmentation thresholds were chosen according to expected values; these can easily be replaced by a fixed percentage of the maximum-minimum current range across devices. The size of current traces and current maps in the investigation stage was larger for the GeSi nanowires, since the gate lever arms in these devices is often smaller compared to the other devices. These hyperparameters could also be optimised in future implementations. All hyperparameters were tested in advance to ensure that transport features were observable with the set configuration.

A.1.6 Labelling procedure

The current maps that were classified by the Algorithm as corresponding to a double dot regime were checked and labelled by human beings at the end of the experiment to benchmark the Algorithm's performance (Table A.6 & A.7). There is often disagreement between humans about what current maps correspond to a double quantum dot regime. The current maps for each type of device were thus labelled by 4 different and independent human labellers. Three datasets were collected, one for each device (nanowire, heterostructure and FinFET). For each device, the current maps collected by Random Search and CATSAI were grouped together and shuffled to avoid labellers' confirmation bias. Median tuning times were calculated using a Bayesian model based on the resultant labels as in Ref. [103].

A.2 Supplementary Tables

Table A.1: Devices used throughout this work. All devices used for training and or testing are different to the devices used in the experiment. Devices used for the experiment algorithm runs only are numbered as zero.

Device	Train	Test 1	Test 2	Algorithm run
GeSi Nanowire 0	-	-	-	x
GeSi Nanowire 1	x	x	-	-
GeSi Nanowire 2	x	x	-	-
Si FinFET 0	-	-	-	x
Si FinFET 1	x	x	-	-
Ge/SiGe Heterostructure 0	-	-	-	x
Ge/SiGe Heterostructure 1	-	-	x	-

Table A.2: Confusion matrices of the Coulomb peak detector for all of the experimental runs comparing the true human labels against the predicted labels of the Random Forest Coulomb Peak detector.

Si FinFET (CATSAI)		
	Pred. Neg.	Pred. Pos.
True Neg.	1862	211
True Pos.	651	276

Si FinFET (Random Search)		
	Pred. Neg.	Pred. Pos.
True Neg.	2455	512
True Pos.	24	9

GeSi Nanowire (CATSAI)		
	Pred. Neg.	Pred. Pos.
True Neg.	156	31
True Pos.	501	2312

GeSi Nanowire (Random Search)		
	Pred. Neg.	Pred. Pos.
True Neg.	1114	277
True Pos.	144	1465

Ge/SiGe Het. (CATSAI)		
	Pred. Neg.	Pred. Pos.
True Neg.	1625	313
True Pos.	199	380

Ge/SiGe Het. (Random Search)		
	Pred. Neg.	Pred. Pos.
True Neg.	1878	1079
True Pos.	10	33

Table A.3: Bounds used for the 3D hypersurface plots.

Device	V_1	V_2	V_3	V_4	V_5	V_6	V_7
Si FinFET, origin (V)	-6.5	-1.5	-1.5	-5.0	-	-	-
Si FinFET, limit (V)	-2.5	0.0	0.0	-5.0	-	-	-
GeSi Nanowire, origin (V)	0.0	0.56	0.0	1.1	0.0	-	-
GeSi Nanowire, limit (V)	4.0	0.56	2.5	1.1	4.0	-	-
Ge/SiGe Heterostructure, origin (V)	0.48	0.0	0.74	0.0	0.79	0.0	0.41
Ge/SiGe Heterostructure, limit (V)	0.48	2.0	0.74	2.0	0.79	2.0	0.41

Table A.4: Gate voltage space explored by CATSAI and Random Search algorithms for each of the devices considered.

Device	V_1	V_2	V_3	V_4	V_5	V_6	V_7
Si FinFET, origin (V)	-6.5	-1.5	-1.5	-6.5	-	-	-
Si FinFET, limit (V)	0.0	0.0	0.0	0.0	-	-	-
GeSi Nanowire, origin (V)	0.0	0.0	0.0	0.0	0.0	-	-
GeSi Nanowire, limit (V)	4.0	2.5	2.5	4.0	4.0	-	-
Ge/SiGe Heterostructure, origin (V)	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Ge/SiGe Heterostructure, limit (V)	2.0	2.0	2.0	2.0	2.0	2.0	2.0

Table A.5: Differences in the configuration of the algorithm for each of the devices considered.

Device	V_{bias} (mV)	Noise Thd. (pA)	Seg. Thd. (pA)
Si FinFET	7.6	2	20
GeSi Nanowire	4	2	1000
Ge/SiGe Heterostructure	0.5	10	30

Device	diag_trace (mV)	2d_lowres (mV)	2d_highres (mV)
Si FinFET	100	80×80	120×120
GeSi Nanowire	200	150×150	200×200
Ge/SiGe Heterostructure	100	80×80	120×120

Table A.6: Total number of current maps labelled as positive (i.e. corresponding to the double quantum dot regime) found by each labeller (L1, L2, L3, L4) for each device and for each run of CATSAI. Runs marked with a an asterisk were excluded because the cryostat temperature was slightly higher than base temperature.

Experiment	Iterations	Time (hours)	L1	L2	L3	L4
Si FinFET, run 1	250	3.47	2	2	2	3
Si FinFET, run 2	250	4.17	12	12	10	10
Si FinFET, run 3	250	3.62	5	5	5	5
Si FinFET, run 4	250	4.15	9	6	6	7
Si FinFET, run 5	250	3.30	9	9	6	8
Si FinFET, run 6	250	3.90	9	9	7	9
Si FinFET, run 7	250	3.30	3	2	1	3
Si FinFET, run 8	250	3.86	13	13	7	13
Si FinFET, run 9	250	3.25	4	4	4	4
Si FinFET, run 10	250	3.81	10	11	8	11
Si FinFET, run 11	250	3.57	5	5	5	6
Si FinFET, run 12	250	3.83	13	13	13	13
GeSi Nanowire, run 1	250	8.42	45	58	74	48
GeSi Nanowire, run 2	250	8.26	46	61	80	54
GeSi Nanowire, run 3	250	8.57	38	60	77	49
GeSi Nanowire, run 4	250	9.18	40	64	79	46
GeSi Nanowire, run 5	250	8.21	38	52	73	47
GeSi Nanowire, run 6	250	8.90	38	64	78	54
GeSi Nanowire, run 7	250	8.12	39	46	70	46
GeSi Nanowire, run 8	250	8.68	46	59	79	48
GeSi Nanowire, run 9	250	9.05	50	67	84	48
GeSi Nanowire, run 10	250	9.31	51	64	78	52
GeSi Nanowire, run 11	250	9.38	50	64	82	54
GeSi Nanowire, run 12	250	9.02	43	63	78	55
Ge/SiGe Heterostructure, run 1	250	3.38	2	4	5	3
Ge/SiGe Heterostructure, run 2	250	2.50	2	3	2	2
Ge/SiGe Heterostructure, run 3	250	2.39	1	1	0	1
Ge/SiGe Heterostructure, run 4*	250	3.17	1	2	0	1
Ge/SiGe Heterostructure, run 5	250	3.04	3	2	2	1
Ge/SiGe Heterostructure, run 6	250	3.66	2	3	4	3
Ge/SiGe Heterostructure, run 7	250	3.19	1	1	1	2
Ge/SiGe Heterostructure, run 8	250	2.81	2	1	2	1
Ge/SiGe Heterostructure, run 9	250	3.19	1	1	1	1
Ge/SiGe Heterostructure, run 10	250	3.22	1	0	1	1
Ge/SiGe Heterostructure, run 11	250	2.91	3	4	1	2
Ge/SiGe Heterostructure, run 12	250	3.50	1	2	2	1
Ge/SiGe Heterostructure, run 13*	250	3.42	2	2	2	3
Ge/SiGe Heterostructure, run 14*	250	3.31	4	3	5	3
Ge/SiGe Heterostructure, run 15	250	2.99	3	4	4	4

Table A.7: Total number of current maps labelled as positive (i.e. corresponding to the double quantum dot regime) found by each labeller (L1, L2, L3, L4) for each device and for each run of Random Search. Runs marked with a an asterisk were excluded because the cryostat temperature was slightly higher than base temperature.

Experiment	Iterations	Time (hours)	L1	L2	L3	L4
Si FinFET, run 1	250	1.62	0	0	0	0
Si FinFET, run 2	250	1.68	0	0	0	0
Si FinFET, run 3	250	1.69	0	0	0	0
Si FinFET, run 4	250	1.58	0	0	0	0
Si FinFET, run 5	250	1.64	0	0	0	0
Si FinFET, run 6	250	1.62	0	0	0	0
Si FinFET, run 7	250	1.51	0	0	0	0
Si FinFET, run 8	250	1.45	0	0	0	0
Si FinFET, run 9	250	1.49	0	0	0	0
Si FinFET, run 10	250	1.52	0	0	0	0
Si FinFET, run 11	250	1.63	0	0	0	0
Si FinFET, run 12	250	1.56	0	0	0	0
GeSi Nanowire, run 1	250	4.40	11	18	23	15
GeSi Nanowire, run 2	250	4.06	5	13	20	10
GeSi Nanowire, run 3	250	4.44	9	17	28	11
GeSi Nanowire, run 4	250	3.82	3	12	21	8
GeSi Nanowire, run 5	250	4.66	12	20	30	14
GeSi Nanowire, run 6	250	4.58	10	22	32	17
GeSi Nanowire, run 7	250	4.17	11	11	22	13
GeSi Nanowire, run 8	250	3.92	7	14	21	10
GeSi Nanowire, run 9	250	4.53	14	23	30	17
GeSi Nanowire, run 10	250	4.37	12	19	23	16
GeSi Nanowire, run 11	250	4.59	11	20	30	14
GeSi Nanowire, run 12	250	4.21	19	23	28	18
Ge/SiGe Heterostructure, run 1	250	2.22	1	1	1	1
Ge/SiGe Heterostructure, run 2	250	1.83	0	0	0	0
Ge/SiGe Heterostructure, run 3	250	1.82	0	0	0	0
Ge/SiGe Heterostructure, run 4	250	1.85	0	0	0	0
Ge/SiGe Heterostructure, run 5	250	1.89	0	1	0	0
Ge/SiGe Heterostructure, run 6	250	1.82	0	0	0	0
Ge/SiGe Heterostructure, run 7	250	1.72	0	0	0	0
Ge/SiGe Heterostructure, run 8	250	1.68	0	0	0	0
Ge/SiGe Heterostructure, run 9	250	1.69	1	2	1	1
Ge/SiGe Heterostructure, run 10	250	1.81	0	0	0	0
Ge/SiGe Heterostructure, run 11	250	1.95	0	0	0	0
Ge/SiGe Heterostructure, run 12	250	1.52	1	1	1	1
Ge/SiGe Heterostructure, run 13*	250	1.64	0	0	1	0

Appendix B

donorsearch

B.1 Supplementary Methods

B.1.1 Experimental setup and control

The device consisted of a natural silicon wafer which was topped with a 900 nm epitaxial layer of isotopically enriched ^{28}Si. A two-step thermal-oxide, SiO_2, separated the ^{28}Si enriched substrate and Al gate electrodes on the device's top surface which were patterned using electron beam lithography. The implantation of the ^{31}P donor was performed prior to gate-electrode patterning. Implantation was followed by annealing the device at 1000°C for five seconds. Although the device featured an on-chip microwave antenna, it was not utilized in this work. The device was wire bonded to a proprietary printed circuit board and was stored within a copper sample enclosure. The sample enclosure was placed within a Halbach array of neodymium magnets [173] which resulted in a magnetic field of approximately 1.1 T applied to the sample. The sample enclosure was securely attached to the mixing chamber plate of a Bluefors BF-LD400 dilution refrigerator, capable of reaching a base temperature of 20 mK.

A Stanford Research Systems SIM900 Mainframe containing SIM928 isolated DC-voltage sources supplied voltages to the SET gate electrodes, V_{TG}, V_{LB}, and V_{RB} via proprietary factor 8 resistive voltage dividers and to the source, V_{bias}, via a factor

1000 voltage divider.

The DC-voltages applied to the donor gates and plunger gate electrodes, V_{DBR}, V_{DBL}, V_{DFR}, V_{DFL} and V_{PL}, were controlled by National Instruments PXIe 4322 within a PXIe-1088. Additionally V_{DFL} and V_{PL} were connected to a Keysight M3300A arbitrary waveform generator which was used to provide dynamic voltage signals enabling fast acquisition of donor-SET charge stability diagrams. The AC and DC signals were combined using impedance-matched combiners with a voltage division factor of 2.5 before being supplied to V_{DFL} and V_{PL} individually. DC-signals supplied to V_{DBR}, V_{DBL}, V_{DFR}, and V_{DFL} were divided by a factor of 8.

The SET current was converted to a voltage using a FEMTO DLPCA-200 trans-impedance amplifier with a gain of 10^7 V/A and 50-kHz bandwidth. The signal was then passed through Stanford Research Systems SIM910 JFET preamplifier, the gain was set to 1V/V as it acts as a ground connection breaker between the SET and what followed the preamplifier. The preamplifier was followed by a Stanford Research Systems SIM965 analogue 50kHz low-pass band filter and the converted signal was digitised and recorded by the Keysight M3300A at a sampling rate of 500 kHz. `donorsearch` features an open and flexible interface back-end which interfaced with SilQ [116] software for instrument control, which wraps around the QcoDeS [174] acquisition framework and instrument drivers.

B.1.2 `donorsearch`'s workflow and parameters

During the coarse tuning stage, 0 V and 2 V were the corresponding lower and upper bounds of V_{TG}, V_{LB}, and V_{RB}. During the fine-tuning stage, each gate electrode had a search range of 10 mV except V_{TG} which was fixed at 2 V. The V_{TG}, V_{LB}, and V_{RB} gates were swept over a range of 2 V with a resolution of 1000 points during the SET tuning process.

Pinch-off current traces are first passed through a Gaussian filter [175] (sigma = 0.1), normalised and then fit to

$$f(x, A, B, C) = A(1 + \tanh(Bx + C)) \tag{B.1}$$

Where A, B and C define the amplitude, slope and shift offset respectively [104]. The pinch-off current trace fit enables the calculation of the pinch-off voltage which we define as the voltage at which the maximum of the second derivative of $f(x, A, B, C)$ occurs.

`donorsearch` has the option between random search or Gaussian process Bayesian optimisation as search methods during the fine-tuning stage. Random search is carried out by performing uniform random samples within the bounds of the gate electrodes. The Gaussian process Bayesian optimiser [176] aims to minimise the score, uses the Matérn 5/2 kernel (Chap. 2.7.5) and relies on negative expected improvement (Chap. 2.7.6) as its acquisition function which it aims to minimise over the posterior distribution. The optimiser acquires the first ten samples randomly before approximating the voltage space with a Gaussian process.

During the fine-tuning stage, each two-level current trace acquired is scored for random telegraph signal after passing through the noise classifier. The scoring method relies on applying a double Gaussian fit to the current trace and extracting the split between the two Gaussians as the current threshold. The software used to apply the double Gaussian fit is the Quantum Technology Toolbox (QTT) developed by QuTech [177] which relies on the Non-Linear Least Square Minimization and Curve-Fitting for Python (LMFIT) package [178]. The separation, d, of the two Gaussians is defined as [177],

$$d = \frac{\mu_2 - \mu_1}{(\sigma_1 + \sigma_2)} \tag{B.2}$$

and the split, s, is

$$s = \mu_1 + d\sigma_1 \tag{B.3}$$

where $\mu_{1/2}$ and $\sigma_{1/2}$ are the mean and standard deviation of the respective Gaussians and $\mu_2 > \mu_1$.

B.1.3 Noise classifier random walk parameters

The current traces for training the noise classifier are acquired within the vicinity of each charge transition detected by the charge transition locator module. The algorithm takes a 10-step random walk in gate-voltage space acquiring current traces at each step. Each step in gate-voltage space is determined by what we call a Gaussian dice roll. The change in voltage applied to each gate is sampled from a standard normal distribution with a standard deviation corresponding to 0.5 mV. These samples are added to the latest gate voltages with the result being the next point in gate-voltage space to move to in the random walk.

B.1.4 Labelling procedure

To check the Algorithm's performance in tuning to random telegraph signal, the current traces claimed by the Algorithm to contain random telegraph signal were labelled by two humans. Current traces from all the experimental runs were gathered together and shuffled randomly, then labelled by each human separately. This was done to reduce any bias towards the different branches of `donorsearch`'s fine tuning stage (machine-learning enabled or random search) and peer bias from other human labellers. Fine tuning times were calculated using Bayesian model of multi-labeller statistics as in Ref. [103].

B.2 Supplementary Figures and Tables

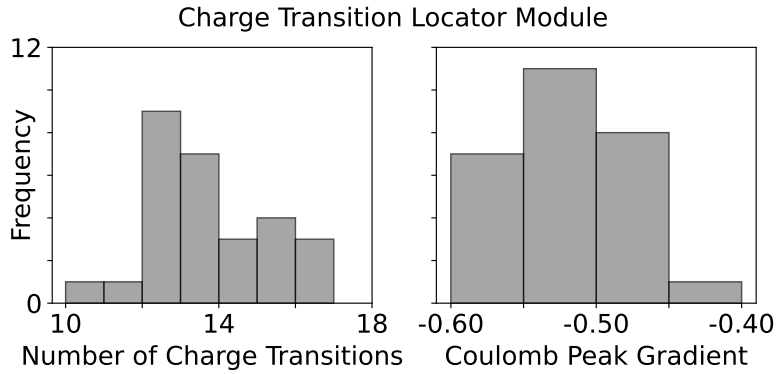


Figure B.1: Charge transition locator module performance. Bar charts of the number of charge transitions and range of Coulomb peak gradients extracted by `donorsearch` across all experimental runs.

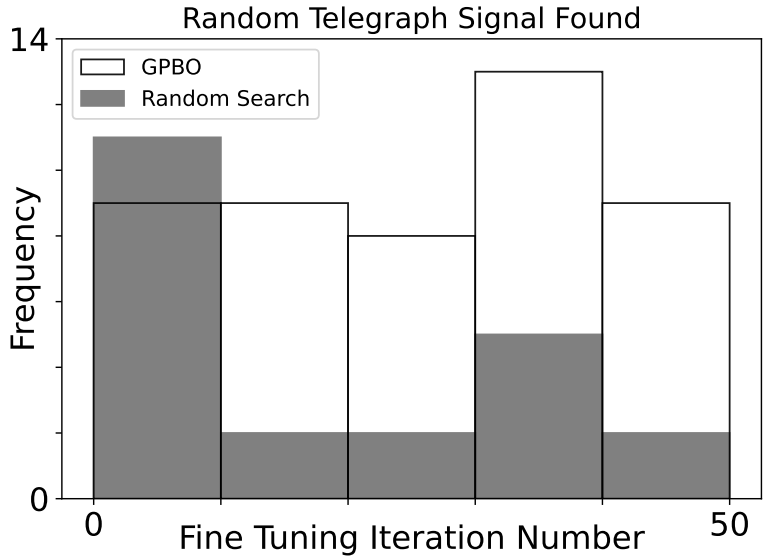


Figure B.2: Search for random telegraph signal success rate. Bar chart of at which iteration did the respective search methods in the fine tuning stage, Gaussian process Bayesian optimisation (GPBO) and random search, find random telegraph signal based on unanimous human labels.

Table B.1: Total number of current traces labelled as positive (i.e. containing random telegraph signal) found by each labeller (Labeller 1 and 2) for all experimental runs including Gaussian process Bayesian optimisation (GPBO) and random search implementations for the fine tuning stage. The run marked with an asterix (*) was used as a sample run to gauge the performance of the noise classifier.

Experiment	Experiment Time (s)	Labeller 1	Labeller 2
GPBO run 1	623.07	1	1
GPBO run 2	582.61	6	6
GPBO run 3*	619.69	4	4
GPBO run 4	415.53	5	5
GPBO run 5	490.22	6	5
GPBO run 6	371.92	5	3
GPBO run 7	443.48	4	3
GPBO run 8	556.69	3	3
GPBO run 9	417.17	4	3
GPBO run 10	474.93	4	3
GPBO run 11	502.52	1	1
GPBO run 12	376.04	6	6
GPBO run 13	547.13	3	3
GPBO run 14	432.62	4	2
Random Search run 1	497.57	3	1
Random Search run 2	496.32	1	1
Random Search run 3	399.42	1	1
Random Search run 4	370.24	2	2
Random Search run 5	421.48	2	2
Random Search run 6	304.15	3	2
Random Search run 7	360.96	3	2
Random Search run 8	422.88	2	1
Random Search run 9	506.86	3	1
Random Search run 10	349.36	4	3
Random Search run 11	583.70	1	0
Random Search run 12	422.78	2	2
Random Search run 13	483.90	1	1
Random Search run 14	430.16	3	3

Table B.2: Confusion matrices of the noise classifier tested from one of the experimental runs when using unanimous human labels to gauge its performance as a noise classifier and as a two-level signal classifier.

Noise Classifier		
	Pred. Neg.	Pred. Pos.
True Neg.	39	15
True Pos.	1	105

Two-level Signal Classifier		
	Pred. Neg.	Pred. Pos.
True Neg.	111	4
True Pos.	9	36

Table B.3: Confusion matrices of the current traces deemed by `donorsearch` to contain random telegraph signal at the end of search for random telegraph signal relying on Gaussian process Bayesian optimisation (GPBO) or random search and overall experimental runs. Confusion matrices are shown in the unanimous form where all human labellers agree on the presence of random telegraph signal and in the single labeller form where a signal labeller confirmed the presence of random telegraph signal in a given current trace.

GPBO - Unanimous

	Pred. Neg.	Pred. Pos.
True Neg.	0	14
True Pos.	0	48

GPBO - Single

	Pred. Neg.	Pred. Pos.
True Neg.	0	6
True Pos.	0	56

Random Search - Unanimous

	Pred. Neg.	Pred. Pos.
True Neg.	0	22
True Pos.	0	22

Random Search - Single

	Pred. Neg.	Pred. Pos.
True Neg.	0	13
True Pos.	0	31

Overall - Unanimous

	Pred. Neg.	Pred. Pos.
True Neg.	0	36
True Pos.	0	70

Overall - Single

	Pred. Neg.	Pred. Pos.
True Neg.	0	19
True Pos.	0	87

Bibliography

[1] C. Luís, C. Bastos-Silveira, E. G. Cothran, and M. D. M. Oom. Iberian origins of new world horse breeds. *Journal of Heredity*, 97(2):107–113, 2006.

[2] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[3] R. P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6-7):467–488, 1982.

[4] D. P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48(9-11):771–783, 2000.

[5] G. Catelani, J. Koch, L. Frunzio, R. J. Schoelkopf, M. H. Devoret, and L. I. Glazman. Quasiparticle relaxation of superconducting qubits in the presence of flux. *Physical Review Letters*, 106:077002, 7, 2011.

[6] J. Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2(July):79, 2018.

[7] R. Van Meter, C. Horsman, B. R. V. Meter, and C. Horsman. A blueprint for building a quantum computer. *Communications of the ACM*, 56(10):84, 2013.

[8] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Physical Review A - Atomic, Molecular, and Optical Physics*, 86(3):032324, 2012.

[9] C. Monroe, D. M. Meekhof, B. E. King, W. M. Itano, and D. J. Wineland. Demonstration of a fundamental quantum logic gate. *Physical Review Letters*, 75(25):4714–4717, 1995.

[10] C. J. Ballance, T. P. Harty, N. M. Linke, M. A. Sepiol, and D. M. Lucas. High-fidelity quantum logic gates using trapped-ion hyperfine qubits. *Physical Review Letters*, 117(6), 2016.

[11] T. P. Harty, D. T. Allcock, C. J. Ballance, L. Guidoni, H. A. Janacek, N. M. Linke, D. N. Stacey, and D. M. Lucas. High-fidelity preparation, gates, memory, and readout of a trapped-ion quantum bit. *Physical Review Letters*, 113(22):220501, 2014.

[12] Y. Wang, M. Um, J. Zhang, S. An, M. Lyu, J.-N. Zhang, L.-M. Duan, D. Yum, and K. Kim. Single-qubit quantum memory exceeding 10-minute coherence time. *Nature Photonics*, 11(10):646–650, 2017.

[13] A. W. Cross, L. S. Bishop, S. Sheldon, P. D. Nation, and J. M. Gambetta. Validating quantum computers using randomized model circuits. *Physical Review A*, 100(3):032328, 2019.

[14] L. Childress and R. Hanson. Diamond NV centers for quantum computing and quantum networks. *MRS Bulletin*, 38(2):134–138, 2013.

[15] C. E. E. Bradley, J. Randall, M. H. H. Abobeih, R. C. C. Berrevoets, M. J. J. Degen, M. A. A. Bakker, M. Markham, D. J. J. Twitchen, and T. H. H. Taminiau. A ten-qubit solid-state spin register with quantum memory up to one minute. *Physical Review X*, 9(3):031045, 2019.

[16] J. L. O'Brien. Optical quantum computing. *Science*, 318(5856):1567–1570, 2007.

[17] A. Politi, M. J. Cryan, J. G. Rarity, S. Yu, and J. L. O'Brien. Silica-on-silicon waveguide quantum circuits. *Science*, 320(5876):646–649, 2008.

[18] D. Llewellyn, Y. Ding, I. I. Faruque, S. Paesani, D. Bacco, R. Santagati, Y.-J. Qian, Y. Li, Y.-F. Xiao, M. Huber, M. Malik, G. F. Sinclair, X. Zhou, K. Rottwitt, J. L. O'Brien, J. G. Rarity, Q. Gong, L. K. Oxenlowe, J. Wang, and M. G. Thompson. Chip-to-chip quantum teleportation and multi-photon entanglement in silicon. *Nature Physics*, 16:148–153, 2, 2020.

[19] R. Santagati, J. W. Silverstone, M. J. Strain, M. Sorel, S. Miki, T. Yamashita, M. Fujiwara, M. Sasaki, H. Terai, M. G. Tanner, C. M. Natarajan, R. H. Hadfield, J. L. O'Brien, and M. G. Thompson. Silicon photonic processor of two-qubit entangling quantum logic. *Journal of Optics*, 19:114006, 11, 2017.

[20] M. H. Devoret, J. M. Martinis, and J. Clarke. Measurements of macroscopic quantum tunneling out of the zero-voltage state of a current-biased josephson junction. *Physical Review Letters*, 55(18):1908–1911, 1985.

[21] Y. Nakamura, C. D. Chen, and J. S. Tsai. Spectroscopy of energy-level splitting between two macroscopic quantum states of charge coherently superposed by josephson coupling. *Physical Review Letters*, 79(12):2328–2331, 1997.

[22] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, B. Burkett, Y. Chen, Z. Chen, B. Chiaro, R. Collins, W. Courtney, A. Dunsworth, E. Farhi, B. Foxen, A. Fowler, C. Gidney, M. Giustina, R. Graff, K. Guerin, S. Habegger, M. P. Harrigan, M. J. Hartmann, A. Ho, M. Hoffmann, T. Huang, T. S. Humble, S. V. Isakov, E. Jeffrey, Z. Jiang, D. Kafri, K. Kechedzhi, J. Kelly, P. V. Klimov, S. Knysh, A. Korotkov, F. Kostritsa, D. Landhuis, M. Lindmark, E. Lucero, D. Lyakh, S. Mandrà, J. R. McClean, M. McEwen, A. Megrant, X. Mi, K. Michielsen, M. Mohseni, J. Mutus, O. Naaman, M. Neeley, C. Neill, M. Y. Niu, E. Ostby, A. Petukhov, J. C. Platt, C. Quintana, E. G. Rieffel, P. Roushan, N. C. Rubin, D. Sank, K. J. Satzinger, V. Smelyanskiy, K. J. Sung, M. D. Trevithick, A. Vainsencher, B. Villalonga, T. White, Z. J. Yao, P. Yeh, A. Zalcman,

H. Neven, and J. M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[23] C. Wang, X. Li, H. Xu, Z. Li, J. Wang, Z. Yang, Z. Mi, X. Liang, T. Su, C. Yang, G. Wang, W. Wang, Y. Li, M. Chen, C. Li, K. Linghu, J. Han, Y. Zhang, Y. Feng, Y. Song, T. Ma, J. Zhang, R. Wang, P. Zhao, W. Liu, G. Xue, Y. Jin, and H. Yu. Towards practical quantum computers: transmon qubit with a lifetime approaching 0.5 milliseconds. *NPJ Quantum Information*, 8:3, 1, 2022.

[24] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Annual ACM Symposium on Theory of Computing*, volume Part F1294, pages 212–219, New York, New York, USA. Association for Computing Machinery, 1996.

[25] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[26] J. Chow, O. Dial, and J. Gambetta. Ibm quantum breaks the 100-qubit processor barrier. *IBM Research Blog*, 2, 2021.

[27] J. Y. Huang, R. Y. Su, W. H. Lim, M. Feng, B. van Straaten, B. Severin, W. Gilbert, N. D. Stuyck, T. Tanttu, S. Serrano, J. D. Cifuentes, I. Hansen, A. E. Seedhouse, E. Vahapoglu, N. V. Abrosimov, H.-J. Pohl, M. L. W. Thewalt, F. E. Hudson, C. C. Escott, N. Ares, S. D. Bartlett, A. Morello, A. Saraiva, A. Laucht, A. S. Dzurak, and C. H. Yang. High-fidelity operation and algorithmic initialisation of spin qubits above one kelvin, 2023. arXiv: 2308.02111.

[28] M. Veldhorst, C. H. Yang, J. 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, 2015.

[29] J. T. Muhonen, J. P. Dehollain, A. Laucht, F. E. Hudson, R. Kalra, T. Sekiguchi, K. M. Itoh, D. N. Jamieson, J. C. McCallum, A. S. Dzurak, and A. Morello. Storing quantum information for 30 seconds in a nanoelectronic device. *Nature Nanotechnology*, 9(12):986–991, 2014.

[30] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg. Surface code quantum computing with error rates over 1%. *Physical Review A*, 83:020302, 2, 2011.

[31] R. Hanson, L. P. Kouwenhoven, J. R. Petta, S. Tarucha, and L. M. Vandersypen. Spins in few-electron quantum dots. *Reviews of Modern Physics*, 79(4):1217–1265, 2007.

[32] C. J. van Diepen, P. T. Eendebak, B. T. Buijtendorp, U. Mukhopadhyay, T. Fujita, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen. Automated tuning of inter-dot tunnel coupling in double quantum dots. *Applied Physics Letters*, 113(3), 2018.

[33] C. Volk, A. M. Zwerver, U. Mukhopadhyay, P. T. Eendebak, C. J. van Diepen, J. P. Dehollain, T. Hensgens, T. Fujita, C. Reichl, W. Wegscheider, and L. M. Vandersypen. Loading a quantum-dot based "Qubyte" register. *NPJ Quantum Information*, 5(1):1–8, 2019.

[34] G. A. Oakes, J. Duan, J. J. L. Morton, A. Lee, C. G. Smith, and M. F. G. Zalba. Automatic virtual voltage extraction of a 2x2 array of quantum dots with machine learning, 2020. arXiv: 2012.03685.

[35] J. M. Elzerman, R. Hanson, L. H. V. Beveren, B. Witkamp, L. M. Vandersypen, and L. P. Kouwenhoven. Single-shot read-out of an individual electron spin in a quantum dot. *Nature 2004 430:6998*, 430:431–435, 6998, 2004.

[36] D. Loss and D. P. DiVincenzo. Quantum computation with quantum dots. *Physical Review A*, 57(1):120–126, 1998.

[37] F. Fedele. *Spin interactions within a two-dimensional array of GaAs double dots*. PhD thesis, 2020.

[38] L. C. Camenzind, S. Geyer, A. Fuhrer, R. J. Warburton, D. M. Zumbühl, and A. V. Kuhlmann. A hole spin qubit in a fin field-effect transistor above 4 kelvin. *Nature Electronics 2022 5:3*, 5(3):178–183, 2022.

[39] A. Morello, C. C. Escott, H. Huebl, L. H. W. van Beveren, L. C. L. Hollenberg, D. N. Jamieson, A. S. Dzurak, and R. G. Clark. Architecture for high-sensitivity single-shot readout and control of the electron spin of individual donors in silicon. *Physical Review B - Condensed Matter and Materials Physics*, 80(8), 2009.

[40] A. Morello, J. J. Pla, F. A. Zwanenburg, K. W. Chan, K. Y. Tan, H. Huebl, M. Möttönen, C. D. Nugroho, C. Yang, J. A. Van Donkelaar, A. D. Alves, D. N. Jamieson, C. C. Escott, L. C. Hollenberg, R. G. Clark, and A. S. Dzurak. Single-shot readout of an electron spin in silicon. *Nature 2010 467:7316*, 467(7316):687–691, 2010.

[41] 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.

[42] L. Jost. Entropy and diversity, 2006.

[43] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[44] H. H. Bock. Probabilistic models in cluster analysis. *Computational Statistics and Data Analysis*, 23(1):5–28, 1996. Classification.

[45] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. 2018.

[46] D. K. Duvenaud. Automatic model construction with gaussian processes. *PhD Thesis, University of Cambridge*, June, 2014.

[47] C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7, 2006.

[48] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, number v. 55, no. 1972 in Applied mathematics series. U.S. Government Printing Office, 1968.

[49] P. I. Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

[50] D. R. Jones, M. Schonlau, and W. J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13, 4, 1998.

[51] A. Chatterjee, P. Stevenson, S. D. Franceschi, A. Morello, N. P. de Leon, and F. Kuemmeth. Semiconductor qubits in practice. *Nature Reviews Physics*, 3, 3, 2021.

[52] F. A. Zwanenburg, A. S. Dzurak, A. Morello, M. Y. Simmons, L. C. Hollenberg, G. Klimeck, S. Rogge, S. N. Coppersmith, and M. A. Eriksson. Silicon quantum electronics. *Reviews of Modern Physics*, 85(3):961–1019, 2013.

[53] W. M. Witzel, M. S. Carroll, A. Morello, Ł. Cywiński, and S. Das Sarma. Electron spin decoherence in isotope-enriched silicon. *Physical Review Letters*, 105(18):187602, 2010.

[54] A. M. Tyryshkin, S. Tojo, J. J. Morton, H. Riemann, N. V. Abrosimov, P. Becker, H. J. Pohl, T. Schenkel, M. L. Thewalt, K. M. Itoh, and S. A. Lyon. Electron spin coherence exceeding seconds in high-purity silicon. *Nature Materials*, 11(2):143–147, 2012.

[55] K. Saeedi, S. Simmons, J. Z. Salvail, P. Dluhy, H. Riemann, N. V. Abrosimov, P. Becker, H. J. Pohl, J. J. Morton, and M. L. Thewalt. Room-temperature quantum bit storage exceeding 39 minutes using ionized donors in silicon-28. *Science*, 342(6160):830–833, 2013.

[56] M. Steger, K. Saeedi, M. L. Thewalt, J. J. Morton, H. Riemann, N. V. Abrosimov, P. Becker, and H. J. Pohl. Quantum information storage for over 180 s

using donor spins in a 28Si "semiconductor vacuum". *Science*, 336(6086):1280–1283, 2012.

[57] M. Veldhorst, H. G. J. Eenink, C. H. Yang, and A. S. Dzurak. Silicon CMOS architecture for a spin-based quantum computer. *Nature Communications*, 8(1), 2017.

[58] B. Buonacorsi, Z. Cai, E. B. Ramirez, K. S. Willick, S. M. Walker, J. Li, B. D. Shaw, X. Xu, S. C. Benjamin, and J. Baugh. Network architecture for a topological quantum computer in silicon. *Quantum Science and Technology*, 4(2), 2019.

[59] Z. Cai, M. A. Fogarty, S. Schaal, S. Patomäki, S. C. Benjamin, and J. J. L. Morton. A silicon surface code architecture resilient against leakage errors. *Quantum*, 3:212, 2019.

[60] J. O'gorman, N. H. Nickerson, P. Ross, J. J. Morton, and S. C. Benjamin. A silicon-based surface code quantum computer. *NPJ Quantum Information*, 2(1):1–14, 2016.

[61] J. J. Pla, K. Y. Tan, J. P. Dehollain, W. H. Lim, J. J. Morton, F. A. Zwanenburg, D. N. Jamieson, A. S. Dzurak, and A. Morello. High-fidelity readout and control of a nuclear spin qubit in silicon. *Nature*, 496(7445):334–338, 2013.

[62] R. Winkler, S. Papadakis, E. De Poortere, and M. Shayegan. *Spin-orbit coupling in two-dimensional electron and hole systems*, volume 41. Springer, 2003.

[63] E. Rashba. Properties of semiconductors with an extremum loop. i. cyclotron and combinational resonance in a magnetic field perpendicular to the plane of the loop. *Sov. Phys.-Solid State*, 2:1109, 1960.

[64] C. Kloeffel, M. Trif, and D. Loss. Strong spin-orbit interaction and helical hole states in ge/si nanowires. *Physical Review B - Condensed Matter and Materials Physics*, 84, 19, 2011.

[65] C. Kloeffel, M. J. Rančić, and D. Loss. Direct rashba spin-orbit interaction in si and ge nanowires with different growth directions. *Physical Review B*, 97, 23, 2018.

[66] S. Bosco, B. Hetényi, and D. Loss. Hole spin qubits in si finfets with fully tunable spin-orbit coupling and sweet spots for charge noise. *PRX Quantum*, 2, 1, 2021.

[67] G. Scappucci, C. Kloeffel, F. A. Zwanenburg, D. Loss, M. Myronov, J.-J. J. Zhang, S. De Franceschi, G. Katsaros, M. Veldhorst, S. D. Franceschi, G. Katsaros, M. Veldhorst, S. De Franceschi, G. Katsaros, and M. Veldhorst. The germanium quantum information route. *Nature Reviews Materials 2020*:1–18, 2020.

[68] B. E. Kane. A silicon-based nuclear spin quantum computer. *Nature*, 393(6681):133–137, 1998.

[69] D. Holmes, B. Wilhelm, A. M. Jakob, X. Yu, F. E. Hudson, K. M. Itoh, A. S. Dzurak, D. N. Jamieson, and A. Morello. Improved placement precision of implanted donor spin qubits in silicon using molecule ions, 2023. arXiv: 2308.04117.

[70] A. M. Jakob, S. G. Robson, H. R. Firgau, V. Mourik, V. Schmitt, D. Holmes, M. Posselt, E. L. H. Mayes, D. Spemann, A. Morello, and D. N. Jamieson. Scalable Atomic Arrays for Spin-Based Quantum Computers in Silicon, 2023. arXiv: 2309.09626.

[71] M. T. Madzik, S. Asaad, A. Youssry, B. Joecker, K. M. Rudinger, E. Nielsen, K. C. Young, T. J. Proctor, A. D. Baczewski, A. Laucht, V. Schmitt, F. E. Hudson, K. M. Itoh, A. M. Jakob, B. C. Johnson, D. N. Jamieson, A. S. Dzurak, C. Ferrie, R. Blume-Kohout, and A. Morello. Precision tomography of a three-qubit donor quantum processor in silicon. *Nature 2022 601:7893*, 601(7893):348–353, 2022.

[72] S. Asaad, V. Mourik, B. Joecker, M. A. Johnson, A. D. Baczewski, H. R. Firgau, M. T. Madzik, V. Schmitt, J. J. Pla, F. E. Hudson, K. M. Itoh, J. C. McCallum, A. S. Dzurak, A. Laucht, and A. Morello. Coherent electrical control of a single high-spin nucleus in silicon. *Nature 2020 579:7798*, 579(7798):205–209, 2020.

[73] I. F. de Fuentes, T. Botzem, A. Vaartjes, S. Asaad, V. Mourik, F. E. Hudson, K. M. Itoh, B. C. Johnson, A. M. Jakob, J. C. McCallum, D. N. Jamieson, A. S. Dzurak, and A. Morello. Navigating the 16-dimensional Hilbert space of a high-spin donor qudit with electric and magnetic fields, 2023. arXiv: 2306.07453.

[74] 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, 2016.

[75] 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.

[76] H. Bluhm, S. Foletti, I. Neder, M. Rudner, D. Mahalu, V. Umansky, and A. Yacoby. Dephasing time of GaAs electron-spin qubits coupled to a nuclear bath exceeding 200μs. *Nature Physics*, 7(2):109–113, 2011.

[77] 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, 2013.

[78] C. H. Yang, R. C. Leon, J. 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, 2020.

[79] E. Kawakami, T. Jullien, P. Scarlino, D. R. Ward, D. E. Savage, M. G. Lagally, V. V. Dobrovitski, M. Friesen, S. N. Coppersmith, M. A. Eriksson, and L. M. 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, 2016.

[80] W. I. Lawrie, H. G. Eenink, N. W. Hendrickx, J. M. Boter, L. Petit, S. V. Amitonov, M. Lodari, B. Paquelet Wuetz, C. Volk, S. G. Philips, G. Droulers, N. Kalhor, F. Van Riggelen, D. Brousse, A. Sammak, L. M. Vandersypen, G. Scappucci, and M. Veldhorst. Quantum dot arrays in silicon and germanium. *Applied Physics Letters*, 116(8):080501, 2020.

[81] T. A. Baart, P. T. Eendebak, C. Reichl, W. Wegscheider, and L. M. Vandersypen. Computer-automated tuning of semiconductor double quantum dots into the single-electron regime. *Applied Physics Letters*, 108(21):213104, 2016.

[82] S. S. Kalantre, J. P. Zwolak, S. Ragole, X. Wu, N. M. Zimmerman, M. D. Stewart, and J. M. Taylor. Machine Learning techniques for state recognition and auto-tuning in quantum dots. *NPJ Quantum Information*, 5(1):1–10, 2019.

[83] 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. *Applied Physics Letters*, 117(14):144002, 2020.

[84] F. Borsoi, N. W. Hendrickx, V. John, M. Meyer, S. Motz, F. van Riggelen, A. Sammak, S. L. de Snoo, G. Scappucci, and M. Veldhorst. Shared control of a 16 semiconductor quantum dot crossbar array. *Nature Nanotechnology 2023*:1–7, 2023.

[85] N. W. Hendrickx, W. I. Lawrie, M. Russ, F. van Riggelen, S. L. de Snoo, R. N. Schouten, A. Sammak, G. Scappucci, and M. Veldhorst. A four-qubit germanium quantum processor. *Nature*, 591(7851):580–585, 2021.

[86] A. Noiri, K. Takeda, T. Nakajima, T. Kobayashi, A. Sammak, G. Scappucci, and S. Tarucha. Fast universal quantum gate above the fault-tolerance threshold in silicon. *Nature 2022 601:7893*, 601:338–342, 7893, 2022.

[87] X. Xue, M. Russ, N. Samkharadze, B. Undseth, A. Sammak, G. Scappucci, and L. M. Vandersypen. Quantum logic with spin qubits crossing the surface code threshold. *Nature 2022 601:7893*, 601:343–347, 7893, 2022.

[88] F. N. Froning, M. K. Rehmann, J. Ridderbos, M. Brauns, F. A. Zwanenburg, A. Li, E. P. Bakkers, D. M. Zumbühl, and F. R. Braakman. Single, double, and triple quantum dots in Ge/Si nanowires. *Applied Physics Letters*, 113(7):73102, 2018.

[89] F. N. Froning, L. C. Camenzind, O. A. van der Molen, A. Li, E. P. Bakkers, D. M. Zumbühl, and F. R. Braakman. Ultrafast hole spin qubit with gate-tunable spin–orbit switch functionality. *Nature Nanotechnology*, 16(3):308–312, 2021.

[90] F. N. M. Froning, M. J. Rančić, B. Hetényi, S. Bosco, M. K. Rehmann, A. Li, E. P. A. M. Bakkers, F. A. Zwanenburg, D. Loss, D. M. Zumbühl, and F. R. Braakman. Strong spin-orbit interaction and g -factor renormalization of hole spins in Ge/Si nanowire quantum dots. *Physical Review Research*, 3(1):13081, 2021.

[91] A. C. Betz, R. Wacquez, M. Vinet, X. Jehl, A. L. Saraiva, M. Sanquer, A. J. Ferguson, and M. F. Gonzalez-Zalba. Dispersively detected pauli spin-blockade in a silicon nanowire field-effect transistor. *Nano Letters*, 15(7):4622–4627, 2015.

[92] 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):2789–2793, 2010.

[93] F. Ansaloni, A. Chatterjee, H. Bohuslavskyi, B. Bertrand, L. Hutin, M. Vinet, and F. Kuemmeth. Single-electron control in a foundry-fabricated two-dimensional qubit array, 2020. arXiv: 2004.00894.

[94] D. Hisamoto, W.-c. C. Lee, J. Kedzierski, H. Takeuchi, K. Asano, C. Kuo, E. Anderson, T.-j. J. King, J. Bokor, C. Hu, J. Jeffrey, and C. Hu. FinFET — A Self-Aligned Double-Gate MOSFET. *IEEE Transactions on Electron Devices*, 47(12):2320–2325, 2000.

[95] A. V. Kuhlmann, V. Deshpande, L. C. Camenzind, D. M. Zumbühl, and A. Fuhrer. Ambipolar quantum dots in undoped silicon fin field-effect transistors. *Applied Physics Letters*, 113(12):122107, 2018.

[96] F. M. Alkhalil, J. I. Perez-Barraza, M. K. Husain, Y. P. Lin, N. Lambert, H. M. Chong, Y. Tsuchiya, D. A. Williams, A. J. Ferguson, and H. Mizuta. Realization of fully tunable FinFET double quantum dots with close proximity plunger gates. In *Proceedings of the IEEE Conference on Nanotechnology*, 2012.

[97] L. Petit, H. G. Eenink, M. Russ, W. I. Lawrie, N. W. Hendrickx, S. G. Philips, J. S. Clarke, L. M. Vandersypen, and M. Veldhorst. Universal quantum logic in hot silicon qubits. *Nature 2020 580:7803*, 580(7803):355–359, 2020.

[98] A. M. Zwerver, T. Krähenmann, T. F. Watson, L. Lampert, H. C. George, R. Pillarisetty, S. A. Bojarski, P. Amin, S. V. Amitonov, J. M. Boter, R. Caudillo, D. Corras-Serrano, J. P. Dehollain, G. Droulers, E. M. Henry, R. Kotlyar, M. Lodari, F. Lüthi, D. J. Michalak, B. K. Mueller, S. Neyens, J. Roberts, N. Samkharadze, G. Zheng, O. K. Zietz, G. Scappucci, M. Veldhorst,

L. M. Vandersypen, and J. S. Clarke. Qubits made by advanced semiconductor manufacturing. *Nature Electronics 2022 5:3*, 5(3):184–190, 2022.

[99] R. P. Feynman. There's plenty of room at the bottom. *California Institute of Technology, Engineering and Science magazine*, 1960.

[100] M. L. Chen, X. Sun, H. Liu, H. Wang, Q. Zhu, S. Wang, H. Du, B. Dong, J. Zhang, Y. Sun, S. Qiu, T. Alava, S. Liu, D. M. Sun, and Z. Han. A FinFET with one atomic layer channel. *Nature Communications*, 11(1):1–7, 2020.

[101] T. Botzem, M. D. Shulman, S. Foletti, S. P. Harvey, O. E. Dial, P. Bethke, P. Cerfontaine, R. P. McNeil, D. Mahalu, V. Umansky, A. Ludwig, A. Wieck, D. Schuh, D. Bougeard, A. Yacoby, and H. Bluhm. Tuning methods for semiconductor spin qubits. *Physical Review Applied*, 10(5):054026, 2018.

[102] P. Pakkiam, A. V. Timofeev, M. G. House, M. R. Hogg, T. Kobayashi, M. Koch, S. Rogge, and M. Y. Simmons. Single-Shot Single-Gate rf Spin Readout in Silicon. *Physical Review X*, 8(4):1–5, 2018.

[103] H. Moon, D. T. Lennon, J. Kirkpatrick, N. M. van Esbroeck, L. C. Camenzind, L. 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):1–18, 2020.

[104] 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):054005, 2019.

[105] D. L. Craig, H. Moon, F. Fedele, D. T. Lennon, B. van Straaten, F. Vigneau, L. C. Camenzind, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, D. Sejdinovic, and N. Ares. Bridging the reality gap in quantum devices with physics-aware machine learning. *Phys. Rev. X*, 14:011001, 1, 2024.

[106] C. Volk, A. Chatterjee, F. Ansaloni, C. M. Marcus, and F. Kuemmeth. Fast charge sensing of Si/SiGe quantum dots via a high-frequency accumulation gate. *Nano Letters*, 19(8):5628–5633, 2019.

[107] R. Bellman. Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences*, 42(10):767–769, 1956.

[108] M. Binois and N. Wycoff. A survey on high-dimensional gaussian process modeling with application to bayesian optimization. *ACM Transactions on Evolutionary Learning and Optimization*, 2:1–26, 2021.

[109] H. Liu, Y.-S. Ong, X. Shen, and J. Cai. When gaussian process meets big data: a review of scalable gps. *IEEE Transactions on Neural Networks and Learning Systems*, 31(11):4405–4423, 2020.

[110] E. Snelson and Z. Ghahramani. Variable noise and dimensionality reduction for sparse gaussian processes. *ArXiv*, abs/1206.6873, 2006.

[111] R. Tripathy, I. Bilionis, and M. Gonzalez. Gaussian processes with built-in dimensionality reduction: applications to high-dimensional uncertainty propagation. *Journal of Computational Physics*, 321:191–223, 2016.

[112] N. M. Van Esbroeck, D. T. Lennon, H. Moon, V. Nguyen, F. Vigneau, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. D. Briggs, D. Sejdinovic, and N. Ares. Quantum device fine-tuning using unsupervised embedding learning. *New Journal of Physics*, 22(9):095003, 2020.

[113] R. Durrer, B. Kratochwil, J. V. Koski, A. J. Landig, C. Reichl, W. Wegscheider, T. Ihn, and E. Greplova. Automated tuning of double quantum dots into specific charge states using neural networks. *Physical Review Applied*, 13(5):054019, 2020.

[114] D. T. Lennon, H. Moon, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. Briggs, M. A. Osborne, E. A. Laird, and N. Ares. Efficiently measuring a quantum device using machine learning. *NPJ Quantum Information*, 5(1):1–8, 2019.

[115] D. T. Lennon. *Efficient Quantum Device Tuning using Machine Learning*. PhD thesis, 2022.

[116] S. Asaad and M. A. Johnson. Silq measurement software.

[117] T. Peters. PEP 20 - The Zen of Python. *Python Software Foundation*, 2004.

[118] B. Severin, D. T. Lennon, L. C. Camenzind, F. Vigneau, F. Fedele, D. Jirovec, A. Ballabio, D. Chrastina, G. Isella, M. de Kruijf, M. J. Carballido, S. Svab, A. V. Kuhlmann, F. R. Braakman, S. Geyer, F. N. M. Froning, H. Moon, M. A. Osborne, D. Sejdinovic, G. Katsaros, D. M. Zumbühl, G. A. D. Briggs, and N. Ares. Cross-architecture tuning of silicon and sige-based quantum devices using machine learning, 2021. arXiv: 2107.12975.

[119] 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 donors—hot, dense, and coherent. *NPJ Quantum Information*, 3(1):1–10, 2017.

[120] T. F. Watson, S. G. Philips, E. Kawakami, D. R. Ward, P. Scarlino, M. Veldhorst, D. E. Savage, M. G. Lagally, M. Friesen, S. N. Coppersmith, M. A. Eriksson, and L. M. Vandersypen. A programmable two-qubit quantum processor in silicon. *Nature*, 555(7698):633–637, 2018.

[121] N. W. Hendrickx, W. I. L. Lawrie, M. Russ, F. van Riggelen, S. L. de Snoo, R. N. Schouten, A. Sammak, G. Scappucci, and M. Veldhorst. A four-qubit germanium quantum processor, 2020. arXiv: 2009.04268.

[122] K. Takeda, A. Noiri, T. Nakajima, J. Yoneda, T. Kobayashi, and S. Tarucha. Quantum tomography of an entangled three-qubit state in silicon. *Nature Nanotechnology 2021*:1–5, 2021.

[123] J. D. Teske, S. S. Humpohl, R. Otten, P. Bethke, P. Cerfontaine, J. Dedden, A. Ludwig, A. D. Wieck, and H. Bluhm. A machine learning approach for

automated fine-tuning of semiconductor spin qubits. *Applied Physics Letters*, 114(13):133102, 2019.

[124] 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), 2019.

[125] J. P. Zwolak, T. McJunkin, S. S. Kalantre, J. Dodson, E. MacQuarrie, D. Savage, M. Lagally, S. Coppersmith, M. A. Eriksson, and J. M. Taylor. Autotuning of double-dot devices in situ with machine learning. *Physical Review Applied*, 13(3):034075, 2020.

[126] N. Ares. Machine learning as an enabler of qubit scalability. *Nature Reviews Materials 2021 6:10*, 6(10):870–871, 2021.

[127] S. Geyer, L. C. Camenzind, L. Czornomaz, V. Deshpande, A. Fuhrer, R. J. Warburton, D. M. Zumbühl, and A. V. Kuhlmann. Self-aligned gates for scalable silicon quantum computing. *Applied Physics Letters*, 118(10):104004, 2021.

[128] 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):263113, 2013.

[129] A. Hofmann, D. Jirovec, M. Borovkov, I. Prieto, A. Ballabio, J. Frigerio, D. Chrastina, G. Isella, and G. Katsaros. Assessing the potential of Ge/SiGe quantum dots as hosts for singlet-triplet qubits, 2019. arXiv: 1910.05841.

[130] D. Jirovec, A. Hofmann, A. Ballabio, P. M. Mutter, G. Tavani, M. Botifoll, A. Crippa, J. Kukucka, O. Sagi, F. Martins, J. Saez-Mollejo, I. Prieto, M. Borovkov, J. Arbiol, D. Chrastina, G. Isella, and G. Katsaros. A singlet-triplet hole spin qubit in planar Ge. *Nature Materials*, 20(8):1106–1112, 2021.

[131] J. Hickie, B. van Straaten, F. Fedele, D. Jirovec, A. Ballabio, D. Chrastina, G. Isella, G. Katsaros, and N. Ares. Automated long-range compensation of an rf quantum dot sensor, 2023. arXiv: 2310.02135.

[132] H. Narayanan and P. Niyogi. Sampling hypersurfaces through diffusion. In A. Goel, K. Jansen, J. D. P. Rolim, and R. Rubinfeld, editors, *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 535–548, Berlin, Heidelberg. Springer Berlin Heidelberg, 2008.

[133] W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

[134] F. Pedregosa, G. G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, É. Duchesnay, G. G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011.

[135] H. Edelsbrunner, D. Kirkpatrick, and R. Seidel. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–559, 1983.

[136] D. N. Jamieson, C. Yang, T. Hopf, S. M. Hearne, C. I. Pakes, S. Prawer, M. Mitic, E. Gauja, S. E. Andresen, F. E. Hudson, A. S. Dzurak, and R. G. Clark. Controlled shallow single-ion implantation in silicon using an active substrate for sub- 20-keV ions. *Applied Physics Letters*, 86(20):1–3, 2005.

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

[138] S. G. Robson, P. Räcke, A. M. Jakob, N. Collins, H. R. Firgau, V. Schmitt, V. Mourik, A. Morello, E. Mayes, D. Spemann, and D. N. Jamieson. Near-surface electrical characterization of silicon electronic devices using focused kev-range ions. *Physical Review Applied*, 18(3):034037, 2022.

[139] A. M. Jakob, S. G. Robson, V. Schmitt, V. Mourik, M. Posselt, D. Spemann, B. C. Johnson, H. R. Firgau, E. Mayes, J. C. Mccallum, A. Morello, D. N. Jamieson, A. M. Jakob, S. G. Robson, D. Spemann, B. C. Johnson, J. C. Mccallum, D. N. Jamieson, V. Schmitt, V. Mourik, H. R. Firgau, A. Morello, M. Posselt, H.-Z. Dresden-Rossendorf, and E. Mayes. Deterministic shallow dopant implantation in silicon with detection confidence upper-bound to 99.85% by ion–solid interactions. *Advanced Materials*, 34(3):2103235, 2022.

[140] M. O. Ahmed, B. Shahriari, and M. Schmidt. Do we need "harmless" bayesian optimization and "first-order" bayesian optimization. *NIPS BayesOpt*, 2016.

[141] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern*, SMC-9(1):62–66, 1979.

[142] R. O. Duda and P. E. Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.

[143] P. V. Hough. Method and means for recognizing complex patterns:US 3069654 A, 1962.

[144] S. Van Der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. Scikit-image: Image processing in python. *PeerJ*, 2014(1):e453, 2014.

[145] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *ACM SIGGRAPH Computer Graphics*, 21(4):163–169, 1987.

[146] H. Blum. A transformation for extracting new descriptors of shape. In W. Wathen-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, 1967.

[147] M. Wolfe, B. Coe, T. Kovach, T. McJunkin, B. Harpt, D. Savage, M. Lagally, G. Bernhardt, R. McDermott, S. Kolkowitz, and M. Eriksson. How cryogenic

illumination resets gate defined semiconductor quantum devices. In *APS March Meeting Abstracts*, Z74.004, 2023.

[148] M. A. I. Johnson, M. T. Madzik, F. E. Hudson, K. M. Itoh, A. M. Jakob, D. N. Jamieson, A. Dzurak, and A. Morello. Beating the thermal limit of qubit initialization with a Bayesian Maxwell's demon. *Physical Review X*, 12(4), 2021.

[149] L. Petit, J. M. Boter, H. G. Eenink, G. Droulers, M. L. Tagliaferri, R. Li, D. P. Franke, K. J. Singh, J. S. Clarke, R. N. Schouten, V. V. Dobrovitski, L. M. Vandersypen, and M. Veldhorst. Spin lifetime and charge noise in hot silicon quantum dot qubits. *Physical Review Letters*, 121(7):076801, 2018.

[150] L. Petit, M. Russ, G. H. Eenink, W. I. Lawrie, J. S. Clarke, L. M. Vandersypen, and M. 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, 2022.

[151] A. E. Seedhouse, T. Tanttu, R. C. Leon, R. Zhao, K. Y. Tan, B. Hensen, F. E. Hudson, K. M. Itoh, J. Yoneda, C. H. Yang, A. Morello, A. Laucht, S. N. Coppersmith, A. Saraiva, and A. S. Dzurak. Pauli blockade in silicon quantum dots with spin-orbit control. *PRX Quantum*, 2(1):010303, 2021.

[152] L. E. Baum and T. Petrie. Statistical inference for probabilistic functions of finite state markov chains. *Annals of Mathematical Statistics*, 37(6):1554–1563, 1966.

[153] G. Varoquaux, S. Lebedev, A. Lee, C. Farrow, M. Danielson, and Anmol. Hmmlearn, 2023.

[154] B. H. Juang and L. R. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, 33:251–272, 3, 1991.

[155] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology: applications to protein modeling. *Journal of Molecular Biology*, 235:1501–1531, 5, 1994.

[156] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen. The Infinite Hidden Markov Model. *Advances in Neural Information Processing Systems*, 14, 2001.

[157] L. R. Rabiner and B. H. Juang. An Introduction to Hidden Markov Models. *IEEE ASSP Magazine*, 3(1):4–16, 1986.

[158] A. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269, 1967.

[159] M. J. Beal. Variational algorithms for approximate bayesian inference. *PhD Thesis*, May, 2003.

[160] H. Cramér. *Mathematical methods of statistics*. 1946.

[161] T. Evans, W. Huang, J. Yoneda, R. Harper, T. Tanttu, K. Chan, F. Hudson, K. Itoh, A. Saraiva, C. Yang, A. Dzurak, and S. Bartlett. Fast bayesian tomography of a two-qubit gate set in silicon. *Phys. Rev. Appl.*, 17:024068, 2, 2022.

[162] 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), 2014.

[163] 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, 2015.

[164] 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, 2016.

[165] J. P. G. Van Dijk, B. Patra, S. Subramanian, X. Xue, N. Samkharadze, A. Corna, C. Jeon, F. Sheikh, E. Juarez-Hernandez, B. P. Esparza, H. Rampurawala, B. R. Carlton, S. Ravikumar, C. Nieva, S. Kim, H. J. Lee, A. Sammak, G. Scappucci, M. Veldhorst, L. M. Vandersypen, E. Charbon, S. Pellerano, M. Babaie, and F. 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, 2020.

[166] A. Ruffino, T. Y. Yang, J. Michniewicz, Y. Peng, E. Charbon, and M. F. 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, 2021.

[167] Y. Peng, A. Ruffino, T. Y. Yang, J. Michniewicz, M. F. Gonzalez-Zalba, and E. 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, 2022.

[168] 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, 2023.

[169] J. P. Zwolak and J. M. Taylor. Colloquium : Advances in automation of quantum dot devices control. *Reviews of Modern Physics*, 95(1):011006, 2023.

[170] V. Nguyen, S. B. Orbell, D. T. Lennon, H. Moon, F. Vigneau, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. Briggs, M. A. Osborne, D. Sejdinovic, and N. Ares. Deep reinforcement learning for efficient measurement of quantum devices. *NPJ Quantum Information 2021 7:1*, 7(1):1–9, 2021.

[171] J. Schuff, D. T. Lennon, S. Geyer, D. L. Craig, F. Fedele, F. Vigneau, L. C. Camenzind, A. V. Kuhlmann, G. Andrew, D. Briggs, D. M. Zumbühl, D.

Sejdinovic, and N. Ares. Identifying Pauli spin blockade using deep learning. *Quantum*, 7:1077, 2023.

[172] T. U. Consortium. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 2022.

[173] C. Adambukulam, V. K. Sewani, H. G. Stemp, S. Asaad, M. T. Madzik, A. Morello, and A. Laucht. An ultra-stable 1.5 T permanent magnet assembly for qubit experiments at cryogenic temperatures. *Review of Scientific Instruments*, 92(8), 2021.

[174] J. H. Nielsen, W. H. Nielsen, M. Astafev, Alexcjohnson, D. Vogel, S. Chatoor, G. Ungaretti, MerlinSmiles, Adriaan, S. Pauka, P. Eendebak, QSaevar, P. Eendebak, R. van Gulik, N. Pearson, Damazter, A. Corna, S. Droege, Damazter2, ThorvaldLarsen, A. Geller, Euchas, V. Hartong, S. Asaad, C. Granade, L. Drmić, S. Borghardt, and Mltls. QCoDeS/Qcodes: QCoDeS 0.2.1, 2019.

[175] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, A. Vijaykumar, A. P. Bardelli, A. Rothberg, A. Hilboll, A. Kloeckner, A. Scopatz, A. Lee, A. Rokem, C. N. Woods, C. Fulton, C. Masson, C. Häggström, C. Fitzgerald, D. A. Nicholson, D. R. Hagen, D. V. Pasechnik, E. Olivetti, E. Martin, E. Wieser, F. Silva, F. Lenders, F. Wilhelm, G. Young, G. A. Price, G. L. Ingold, G. E. Allen, G. R. Lee, H. Audren, I. Probst, J. P. Dietrich, J. Silterra, J. T. Webber, J. Slavič, J. Nothman, J. Buchner, J. Kulick, J. L. Schönberger, J. V. de Miranda Cardoso, J. Reimer, J. Harrington, J. L. C. Rodríguez, J. Nunez-Iglesias, J. Kuczynski, K. Tritz, M. Thoma, M. Newville, M. Kümmerer,

M. Bolingbroke, M. Tartre, M. Pak, N. J. Smith, N. Nowaczyk, N. Shebanov, O. Pavlyk, P. A. Brodtkorb, P. Lee, R. T. McGibbon, R. Feldbauer, S. Lewis, S. Tygier, S. Sievert, S. Vigna, S. Peterson, S. More, T. Pudlik, T. Oshima, T. J. Pingel, T. P. Robitaille, T. Spura, T. R. Jones, T. Cera, T. Leslie, T. Zito, T. Krauss, U. Upadhyay, Y. O. Halchenko, and Y. Vázquez-Baeza. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods 2020 17:3*, 17(3):261–272, 2020.

[176] T. Head, MechCoder, G. Louppe, I. Shcherbatyi, Fcharras, Z. Vinícius, Cmmalone, C. Schröder, Nel215, N. Campos, T. Young, S. Cereda, T. Fan, Rene-rex, K. (Shi, J. Schwabedal, Carlosdanielcsantos, Hvass-Labs, M. Pak, SoManyUsernamesTaken, F. Callaway, L. Estève, L. Besson, M. Cherti, K. Pfannschmidt, F. Linzberger, C. Cauet, A. Gut, A. Mueller, and A. Fabisch. Scikit-optimize/scikit-optimize: v0.5.2, 2018.

[177] P. Eendebak. Quantum Technology Toolbox (QTT), 2018.

[178] M. Newville, R. Otten, A. Nelson, T. Stensitzki, A. Ingargiola, D. Allan, A. Fox, F. Carter, Michał, R. Osborn, D. Pustakhod, Lneuhaus, S. Weigand, A. Aristov, Glenn, C. Deil, Mgunyho, Mark, A. L. R. Hansen, G. Pasquevich, L. Foks, N. Zobrist, O. Frost, Stuermer, Azelcer, A. Polloreno, A. Persaud, J. H. Nielsen, M. Pompili, and S. Caldwell. Lmfit/lmfit-py: 1.2.1, 2023.