

On Scalable Integrated Charge State Tuning for Semiconductor Quantum Dot Devices

Fabian Ant3nio Hader

Information

Band / Volume 120

ISBN 978-3-95806-884-1

Forschungszentrum Jülich GmbH
Peter Grünberg Institut (PGI)
Integrated Computing Architectures – ICA (PGI-4)

On Scalable Integrated Charge State Tuning for Semiconductor Quantum Dot Devices

Fabian António Hader

Schriften des Forschungszentrums Jülich
Reihe Information / Information

Band / Volume 120

ISSN 1866-1777

ISBN 978-3-95806-884-1

Bibliografische Information der Deutschen Nationalbibliothek.
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte Bibliografische Daten
sind im Internet über <http://dnb.d-nb.de> abrufbar.

Herausgeber und Vertrieb: Forschungszentrum Jülich GmbH
Zentralbibliothek, Verlag
52425 Jülich
Tel.: +49 2461 61-5368
Fax: +49 2461 61-6103
zb-publikation@fz-juelich.de
www.fz-juelich.de/zb

Umschlaggestaltung: Grafische Medien, Forschungszentrum Jülich GmbH

Druck: Grafische Medien, Forschungszentrum Jülich GmbH

Copyright: Forschungszentrum Jülich 2026

Schriften des Forschungszentrums Jülich
Reihe Information / Information, Band / Volume 120

D 464 (Diss. Duisburg-Essen, Univ., 2025)

ISSN 1866-1777
ISBN 978-3-95806-884-1

Vollständig frei verfügbar über das Publikationsportal des Forschungszentrums Jülich (JuSER)
unter www.fz-juelich.de/zb/openaccess.



This is an Open Access publication distributed under the terms of the [Creative Commons Attribution License 4.0](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

*The most beautiful thing we can
experience is the mysterious. It is
the source of all true art and science.*

Albert Einstein

Hiermit versichere ich, die vorliegende Dissertation selbstständig, ohne fremde Hilfe und ohne Benutzung anderer als den angegebenen Quellen angefertigt zu haben. Alle aus fremden Werken direkt oder indirekt übernommenen Stellen sind als solche gekennzeichnet. Die vorliegende Dissertation wurde in keinem anderen Promotionsverfahren eingereicht. Mit dieser Arbeit strebe ich die Erlangung des akademischen Grades Doktor der Ingenieurwissenschaften (Dr.-Ing.) an.

Ort, Datum

Name

Acknowledgments

First and foremost, I would like to express my sincere gratitude to Prof. Dr.-Ing. Stefan van Waasen for his trust in me and for the opportunity to pursue my doctoral research under his guidance. His consistent support has served as a reliable foundation for the entirety of this work.

Additionally, I would like to express my profound gratitude to Prof. Dr.-Ing. Lotte Geck, who has played a pivotal role as my official supervisor and has provided thoughtful support throughout this journey.

Particularly, I would like to express my profound gratitude to Dr.-Ing. Jan Vogelbruch, my daily sparring partner, mentor, and teacher in developing my scientific mindset and methodology. Your sharp mind, patience, and clarity of thought shaped the way I think and work today. I am deeply thankful for the countless discussions, the challenges you assisted me in navigating, and the exacting standards of scientific rigor you instilled in me.

In addition, I would like to express my heartfelt gratitude to colleagues in the office, whose support – both emotional and scientific – was of inestimable value during this endeavor. In particular, I want to thank Lea Schreckenber, Florian Rössing, and Jonas Bühler. Not only have they been reliable colleagues but also exceptional friends. You helped keep me grounded, motivated, and sane through the highs and lows of this PhD.

I would also like to extend my gratitude to members of the RWTH Quantum Technologies Group, namely Hendrik Bluhm, Simon Humpohl, Tobias Hangleiter, Till Huckemann, Paul Surrey, and Han Na We, for helpful discussions and for providing measurement data that contributed to this thesis.

Finally, I would like to express my profound gratitude to my family for their unwavering support, encouragement, and belief in me, even during the most challenging periods. To Marieluise Oden, your patience, understanding, and love provided me with the necessary strength and balance to successfully navigate this journey. I am deeply grateful to all of you for helping me keep perspective, staying by my side, and reminding me of what truly matters.

Abstract

Semiconductor quantum computing leverages quantum dots as qubits, where achieving high qubit fidelity is one essential requirement for scalability. The formation of qubits and associated sensor dots necessitates meticulous iterative voltage adjustments, a tuning process that is ordinarily executed manually by researchers. While various automated approaches have been proposed, only a few address scalability in the context of a system with a million qubits or consider shifting key data analysis tasks into the cryostat. The latter is necessary to alleviate wiring constraints and to minimize data transmission to room temperature electronics.

This thesis explores scalable solutions for automated in-cryostat data analysis, identifying charge transition detection as a pivotal task. A simulation model is developed to generate realistic charge stability diagrams with known ground truth. The model's output is benchmarked against experimental data to verify fidelity and diversity.

Utilizing this model, a comprehensive evaluation of charge transition detection methods is conducted, comparing classical algorithms and a range of machine learning approaches. In consideration of energy-efficient integration, this study includes an initial investigation into reducing the complexity of convolutional neural networks. Subsequent to their optimization using simulated data, the approaches are benchmarked on both simulated and experimental data from diverse qubit samples. When applied to contemporary data qualities, evidence is presented that classical approaches are inadequate in terms of robustness and accuracy while numerous machine learning models demonstrate compelling performance. Among the latter, convolutional neural networks like U-Net are identified to be particularly eligible due to their advantageous structure for efficient hardware integration.

Although the U-Net architecture already demonstrates a strong potential for network size reduction under current data quality constraints, a cryogenic hardware integration remains a significant challenge. Consequently, a subsequent study explores if enhanced sensor dot tuning and sensor compensation reduce the complexity of charge transition analysis and if they enable the utilization of ray-based detection methodologies. The findings indicate that particularly sensor compensation has a substantial impact on the data quality and allows a streamlining of the analytical process. However, achieving robust ray-based analysis remains feasible only if consistently reaching the highest observed measurement quality and utilizing machine learning. Even in this case, classical methods and a proposed averaging and thresholding circuit fall short of the requisite detection accuracy. Only with further improvements toward optimal data quality and additional enhancements to the proposed circuit, an integrated solution without machine learning appears attainable.

Finally, the energy efficiency of scaled-down neural networks is assessed using state-of-the-art hardware accelerators. The findings suggest that the cryogenic integration of machine learning-based charge transition detection is feasible, thus propelling the development of scalable and energy-efficient quantum control architectures.

Zusammenfassung

Halbleiterbasiertes Quantencomputing bedient sich Quantenpunkten als Qubits, wobei eine hohe Verlässlichkeit eine zentrale Prämisse für die Skalierbarkeit darstellt. Die Bildung von Quantenpunkten bedingt eine sorgfältige, iterative Kalibrierung die üblicherweise manuell durchgeführt wird. Obwohl zahlreiche Vorschläge für automatisierte Verfahren vorliegen, bleiben Aspekte wie die Skalierbarkeit im Kontext eines Systems mit Millionen Qubits oder die Verlagerung zentraler Datenanalyseaufgaben auf Kryostatenebene zur Überwindung von Leitungsempfängen zur Raumtemperaturelektronik meist unberücksichtigt.

In der vorliegenden Arbeit werden skalierbare Lösungen zur automatisierten Kalibrierung untersucht und die Detektion von Ladungsübergängen als zentrale Herausforderung identifiziert. Hierzu wird ein Simulationsmodell entwickelt, das realistische Ladungsstabilitätsdiagramme mit zugehörigen Referenzdaten erzeugt. Die Ergebnisse der Simulation werden mit experimentellen Daten validiert.

Basierend auf diesem Modell erfolgt eine umfassende Evaluation von Methoden zur Erkennung von Ladungsübergängen, wobei klassische Algorithmen und diverse Machine Learning (ML) Ansätze gegenübergestellt werden. Unter Berücksichtigung einer energieeffizienten Integration wird zudem die Komplexitätsreduktion von Convolutional Neural Networks (CNNs) untersucht. Nach der Optimierung auf Basis simulierter Daten erfolgt die Evaluierung der Verfahren auf simulierten und experimentellen Datensätzen verschiedener Qubit-Proben. Es wird ersichtlich, dass klassische Ansätze für die gegenwärtigen Datenqualitäten hinsichtlich Robustheit und Genauigkeit nicht ausreichend sind, während viele ML-Verfahren überzeugende Leistungen erzielen. Insbesondere CNN-Architekturen wie U-Net erweisen sich aufgrund ihrer Struktur als vielversprechend für eine effiziente Hardware-Integration.

Obwohl das U-Net bereits unter der derzeitigen Datenqualität ein hohes Potenzial für eine Netzwerkgrößenreduktion aufweist, bleibt eine kryogene Hardware-Integration eine erhebliche Herausforderung. Daher wird weiterführend untersucht, ob eine verbesserte Kalibrierung und Kompensation der Sensorquantenpunkte die Komplexität der Ladungsübergangsanalyse reduziert und den Einsatz strahlbasierter Methoden ermöglicht. Die Ergebnisse zeigen, dass insbesondere die Sensorkompensation einen signifikanten Einfluss auf die Datenqualität ausübt und eine Vereinfachung des Analyseprozesses ermöglicht. Eine robuste Analyse auf Basis strahlbasierter Verfahren ist jedoch nur bei hoher Messqualität unter Einbindung von ML realisierbar. Klassische Methoden sowie ein vorgeschlagener Schaltkreis auf Basis von Mittelwertbildung und Schwellenwerten erreichen nicht die erforderliche Detektionsgenauigkeit. Nur bei weiterer Verbesserung in Richtung optimaler Messqualität und zusätzlicher Erweiterung des Schaltkreises wird eine vollständig integrierte Lösung ohne ML möglich sein. Eine abschließende Bewertung der Energieeffizienz skalierten neuronaler Netzwerke auf modernen Hardwarebeschleunigern weist darauf hin, dass die kryogene Integration einer ML-basierten Ladungsübergangsdetektion realisierbar ist. Damit wird die Entwicklung skalierbarer und energieeffizienter Quantenkontrollarchitekturen entscheidend vorangetrieben.

Contents

Abstract	vii
Zusammenfassung	ix
List of Figures	xv
List of Tables	xvii
Abbreviations	xix
1 Introduction to Quantum Computing	1
1.1 Quantum Mechanics	1
1.2 Quantum Computing	2
1.2.1 Circuit-Based Quantum Computers	3
1.2.2 Quantum Annealers	3
1.2.3 Topological Quantum Computers	4
2 Tuning Semiconductor Quantum Dot Spin Qubits	5
2.1 Quantum Dots	5
2.1.1 Quantum Dots in Quantum Computing	6
2.1.2 Challenges	6
2.2 Semiconductor Spin Qubits	7
2.2.1 Qubit Realizations	7
2.2.2 Measurement Realizations	10
2.3 Qubit Tuning and Automation Approaches	13
2.3.1 Turn-on and Initialization	13
2.3.2 Sensor Tuning	13
2.3.3 Dot Regime Tuning	13
2.3.4 Charge State Detection and Single-Electron Regime	14
2.3.5 Inter-Dot Coupling and Tunnel Barrier Fine Tuning	14
2.3.6 Further Fine-Tuning and Stabilization of Qubit Parameters	14
2.4 Data Analysis Techniques	15
2.4.1 Noise Analysis	15
2.4.2 Pattern Recognition Using Classical Processing Pipeline	17
2.4.3 Pattern Recognition Using Deep Learning Pipeline	21
2.5 Motivation	23
3 Experimental Setup and Identification of Crucial Tuning Steps	25
3.1 Qubit Experiment	25

3.2	General Tuning Workflow	28
3.3	Crucial Steps for the Automation of Sensor Dot Tuning	36
3.4	Characteristical Scans for the Selection of Suitable Coulomb Peaks	37
3.5	Crucial Steps for the Automation of Spin Qubit Tuning	38
3.6	Charge Stability Diagrams	38
3.7	Conclusion	40
4	Simulation of Charge Stability Diagrams for Automated Tuning Solutions (SimCATS)	41
4.1	Simulation Model	43
4.1.1	Dot Occupation Model	43
4.1.2	Sensor Model	46
4.1.3	Distortions Model	47
4.2	Parameter Extraction	50
4.2.1	Extraction of Occupation Data Parameters	50
4.2.2	Extraction of Sensor Parameters	50
4.2.3	Extraction of Distortion Parameters	51
4.3	Evaluation	53
4.3.1	Visual Assessment	53
4.3.2	Metrics for Generative Models	53
4.3.3	Performance Analysis	56
4.4	Outcome and Perspective	57
5	Automated Charge Transition Detection in Quantum Dot Charge Stability Diagrams	59
5.1	Datasets	59
5.2	Metrics and Evaluation Methods	62
5.3	Charge Transition Detection Approaches	63
5.3.1	Classical Approaches	63
5.3.2	Machine Learning Approaches	65
5.4	Training and Optimization	69
5.5	Evaluation	69
5.6	Outcome and Perspective	74
6	Sensitivity and Efficiency Refinement with Improved Sensor Tuning	77
6.1	Automated Selection of the Coulomb Peak Oscillation Area	77
6.2	Noise Sensitive Coulomb Peak Selection	78
6.2.1	Noise Analysis Results	78
6.2.2	Flank Candidate Selection and MSG Criterion	80
6.2.3	Suggested Noise Estimator and Application to Measured Data	82
6.3	Charge Transition Detection Complexity Reduction Using Compensated Sensor	84
6.3.1	Compensated Dataset Generation and Optimization of Approaches	85
6.3.2	Results with Sensor Compensation	87
6.3.3	Results with Sensor Compensation and Superior Data Quality	89

6.3.4	Suggested Detection Methods	91
6.4	Energy Efficiency of Scaled Machine Learning Models	93
6.5	Unified Sensor and CSD Simulation Framework	96
6.6	Outcome and Perspective	98
7	Conclusion and Outlook	101
A	Basic Condensed Training Routine for Machine Learning Networks	107
B	Visual Analysis of Feature Maps and Kernels	109
	Bibliography	113
	Publications Related to This Work	143
	Curriculum Vitae	145

List of Figures

2.1	Material stacks and corresponding band diagrams for semiconductor spin qubits	8
2.2	Examples of qubit samples of different materials and topologies	9
2.3	Visualization of the potential landscape for a DQD	10
2.4	Spatial configuration of a SD positioned in close proximity to a DQD	12
2.5	Comparison of the conventional classical pattern recognition pipeline and the deep learning approach	18
3.1	Arrangement of the DQD array and SD gates	26
3.2	Cross-sectional view of the interior of a dilution refrigerator	27
3.3	Diagram of the signal path	27
3.4	Overview of qubit tuning steps	29
3.5	Sequence diagram for the highest level automated qubit tuning functionalities	31
3.6	Sequence diagram for the sample characterization	31
3.7	Sequence diagram for the coarse tuning	32
3.8	Sequence diagram for the sensor tuning	34
3.9	Wide scan data of a SD	36
3.10	Example narrow scan from SD measurements	37
3.11	Visualization of the characteristic structure of CSDs	39
3.12	Example of a CSD over a large voltage range	40
4.1	Interfaces of the simulation class of the simulation framework SimCATS	43
4.2	Representation of the separation between regions with fixed numbers of electrons using TCTs	44
4.3	One-dimensional TCT description	45
4.4	Simulation of the sensor response	47
4.5	Examples of the simulated sensor response and distortions	48
4.6	Example of simulation results	54
4.7	Average execution time of different simulation approaches	57
5.1	Visualization of parameters used to define the TCTs for the geometric simulation in SimCATS	61
5.2	Examples for CSDs of the final evaluation datasets	62
5.3	Bar plot visualizing the metrics for all detectors across the three test datasets	73
5.4	Exemplary predictions on the test datasets	74
6.1	Example of the fully automatic approach for the SD noise analysis	79
6.2	Example of candidate regions for a sensitive SD operating point	80
6.3	Visualization of the MSG criterion	81
6.4	Boxplots of the noise estimation errors on the evaluation dataset	82

6.5	Results of the investigation regarding the sensitivity of the quality of results to broader dispersion of the estimation	83
6.6	Visualization of the effects of sensor compensation on the CSD data using SimCATS	86
6.7	Schematic of the HIAT	87
6.8	Timing diagram of the HIAT	87
6.9	Exemplary predictions on the compensated sensor test dataset	90
6.10	Exemplary predictions on the compensated sensor test dataset with superior data quality	92
6.11	Lever arms of the barrier gate voltages on the potential of the SD	97
6.12	Experimental pinch-off measurement with corresponding GLF fits	98
6.13	Visualization of the calculation of the SD conductivity	99
B.1	Visualization of the convolutional kernels and feature maps of UNet-447	112

List of Tables

3.1	Most relevant noise sources in QD systems	28
4.1	Simulation evaluation results utilizing 1-precision and 1-recall	56
5.1	Parameter ranges for the simulated datasets and their distributions	60
5.2	Collected CT detection approaches	64
5.3	Training hyperparameters of the collected ML approaches	70
5.4	Size of the neural networks and their inference time, FLOPs, and MACs	71
5.5	Metrics calculated for all detectors across the three test datasets	72
6.1	Training hyperparameters of the U-Net networks	88
6.2	Metrics calculated for all approaches on the compensated sensor test set	89
6.3	Metrics calculated for all approaches on the compensated sensor test set with superior data quality	91
6.4	FLOPs, MACs, and an estimation for the required power for all previously discussed U-Net models	94
6.5	Required power for the local CT detection in the cryostat to enable the automated tuning of one million qubits in six hours	95
6.6	Overview of approaches and suitability for different scenarios	100

Abbreviations

2DEG	two-dimensional electron gas
2DHG	two-dimensional hole gas
ASIC	application-specific integrated circuit
AutoML	automated machine learning
CASENet	deep category-aware semantic edge detection
CDS	correlated double sampling
CHRNNet	cascaded and high-resolution network
CIM	constant interaction model
CMOS	complementary metal-oxide-semiconductor
CNN	convolutional neural network
CSD	charge stability diagram
CT	charge transition
DC	direct current
DFE	dynamic feature fusion
DFT	discrete Fourier transform
DICE	Dice similarity coefficient
DiffusionEdge	diffusion probabilistic model for crisp edge detection
DPM	diffusion probabilistic model
DQD	double quantum dot
EDTER	edge detection transformer
ED	edge drawing
FCN	fully convolutional network
FD-SOI	fully depleted silicon on insulator
FFT	fast Fourier transform
FLOP	floating point operation
FOD	first-order derivative
FPN	feature pyramid network
GCanny	generalized Canny
GLF	generalized logistic function
gPb	globalized probability of boundary
GPU	graphics processing unit
HIAT	hardware integrated averaging and thresholding
HM	Hubbard model
HOG	histogram of oriented gradients
HPO	hyperparameter optimization
IDT	inter-dot transition
KDE	kernel density estimation
k-NN	k-nearest neighbors

LDC	lightweight dense convolutional neural network
LDT	lead-to-dot transition
MAC	multiply-accumulate operation
MA-Net	multi-scale attention network
MedSegDiff	medical image segmentation with diffusion probabilistic model
MLE	maximum likelihood estimation
MLP	multilayer perceptron
ML	machine learning
MMViT-Seg	mini-mobile vision transformer for segmentation
MSG	minimal signal gradient
NAS	neural architecture search
NC	neuromorphic computing
OPA	operational amplifier
PCA	principal component analysis
PCB	printed circuit board
PhCon	phase congruency
PSD	power spectral density
PTQ	post-training quantization
QAT	quantization-aware training
QC	quantum computing
QD	quantum dot
QPC	quantum point contact
qubit	quantum bit
RB	ray-based
RBC	ray-based classification
ReLU	rectified linear unit
RF	radio frequency
RTN	random telegraph noise
SD	sensor dot
SET	single-electron transistor
SGNR	signal-gradient-to-noise ratio
SimCATS	Simulation of Charge Stability Diagrams for Automated Tuning Solutions
Si-MOS	silicon metal-oxide-semiconductor
SSM	state-space model
S-DICE	normalized surface Dice
SVM	support vector machine
TCT	total charge transition
TEED	tiny and efficient edge detector
TOPS	tera operations per second
ViT	vision transformer
VMamba	visual state-space model
VM-UNet	vision mamba UNet

Chapter 1

Introduction to Quantum Computing

1.1 Quantum Mechanics

QUANTUM MECHANICS is a fundamental theory in physics that describes the behavior of matter and energy at atomic and subatomic levels. In contrast to classical mechanics, which governs macroscopic objects, quantum mechanics elucidates a world in which particles exist in probabilistic states, can exhibit wave-particle duality, and are subject to phenomena such as superposition and entanglement.

The origins of quantum mechanics can be traced back to the early 20th century, primarily as a response to unresolved issues in classical physics. The blackbody radiation problem, which was addressed by Max Planck in 1900, marked a breakthrough in the field of quantum mechanics [1]. Planck proposed the hypothesis that electromagnetic energy could only be emitted in discrete units, or "quanta", rather than in a continuous flow by introducing the concept of quantization. This concept presents challenges to classical theories and constitutes the foundation for the development of quantum theory.

Building on Planck's work, Albert Einstein (1905) provided an explanation of the photoelectric effect by postulating that light is quantized into discrete particles, or photons [2]. This postulation introduced the duality of light, which behaves both as a wave and a particle, thus further diverging from the principles of classical mechanics. The formalization of quantum mechanics occurred in the 1920s, with significant contributions from physicists such as Niels Bohr. He developed the Bohr model of the atom, which incorporates quantized electron orbits to explain atomic spectra [3–5]. Werner Heisenberg's matrix mechanics (1925) [6], Paul Dirac's formulation [7], and Erwin Schrödinger's wave mechanics (1926) [8–11] unified the known concepts with the latter resulting in the Schrödinger equation, which governs the wavefunction of quantum systems.

The field of quantum mechanics experienced considerable advancement throughout the 20th century as the theory underwent a process of evolution and maturation. The Copenhagen interpretation, largely developed by Bohr and Heisenberg, emphasizes the probabilistic nature of quantum states and proposes that certain physical properties do not have a definite value until measurement occurs. The uncertainty principle, which Heisenberg postulated [12], became a foundational concept that states that certain pairs of properties (such as position and momentum) cannot be precisely known simultaneously. Later, the theory underwent further expansion with the advent of the quantum field theory (QFT), which integrated quantum

mechanics with special relativity to elucidate the quantization of fields, including electromagnetic fields. This resulted in the development of the standard model of particle physics, which unifies three of the four fundamental forces of nature: electromagnetism, the weak nuclear force, and the strong nuclear force [13].

In recent decades, quantum mechanics has gained increasing relevance due to advancements in quantum information theory and quantum computing (QC). Moreover, quantum mechanics experiments have been conducted to test the limits of quantum mechanics in systems that extend beyond the atomic scale. The realization of quantum entanglement and nonlocality has also enriched the philosophical discourse on the nature of reality and information in quantum systems. These advances have not only deepened our understanding of the quantum world but have also paved the way for practical applications in emerging technologies. One of the most promising research areas in this regard is QC, which has attracted significant attention in recent years.

1.2 Quantum Computing

It is anticipated that QC will provide significant computational advantages in several research areas, including medical drug development [14], encryption [15], physical simulations [16–19], optimization problems [20], and materials science [21, 22]. The expected exponential increase in processing speed compared to that of today’s supercomputers is referred to as quantum supremacy. The quantum computers in use today are capable of testing some quantum algorithms for limited problem sizes. Nevertheless, they have not yet attained the level of universality that would be necessary for them to be classified as universal systems. A universally programmable quantum computer is estimated to require at least one million physical quantum bits (qubits) [23, 24].

In 1996, David DiVincenzo established the following criteria [25] for a scalable quantum computer:

- identification of well-defined qubits,
- reliable state preparation,
- low decoherence,
- accurate quantum gate operations, and
- strong quantum measurements.

The physical realization of quantum computers has become one of the most rapidly evolving research areas, and several architectures have been developed. These implementations differ primarily in their underlying qubit realizations and computation methods. Two major categories of QC architectures are circuit-based quantum computers and quantum annealers. However, research is also active on other approaches such as topological QC.

1.2.1 Circuit-Based Quantum Computers

In the context of QC, a circuit-based quantum computer, also known as a gate-model quantum computer, employs **qubits** that are typically manipulated through a sequence of quantum gates to perform computations. Quantum gates represent the basic building blocks of quantum circuits, analogous to logic gates in classical circuits. They manipulate **qubits** by applying specific unitary transformations to their quantum states. Unlike their classical counterparts, quantum gates are reversible and operate on complex probability amplitudes, thereby enabling quantum parallelism.

Qubits can be implemented in a variety of physical systems, including superconducting architectures based on Josephson junctions [26, 27] (such as IBM's Osprey processor with 433 **qubits** [28]), ion traps [29], and quantum dots (**QDs**) [30, 31], in which the spin states of charge carriers confined in semiconductor nanostructures serve as **qubits**. In superconducting systems, quantum gates are typically realized using precisely shaped microwave pulses, while in **QD** systems, control is achieved via electric pulses, magnetic fields, or microwave pulses that manipulate the spin degree of freedom, with two-**qubit** gates often relying on the exchange interaction between neighboring dots. In contrast, ion trap quantum computers use ions suspended in electromagnetic fields as **qubits**, controlled by laser pulses to implement quantum gates. These systems offer high fidelity and long coherence times, yet present challenges in terms of scalability.

In particular, **QDs** offer the potential for monolithical integration within existing semiconductor technologies as ultimate goal for high scalability, making them promising candidates for future quantum computers. In either of the aforementioned realizations, quantum mechanical properties, such as superposition, entanglement, and quantum interference, are exploited to execute quantum algorithms.

The key challenges in these architectures are quantum error correction and decoherence mitigation because precise control over the **qubit** states is essential for reliable computation. These systems are ideal for implementing universal quantum algorithms, such as Shor's algorithm for factorization [15] and Grover's search algorithm [32].

1.2.2 Quantum Annealers

Quantum annealers, such as the D-Wave Advantage system [33], which comprises more than 5000 **qubits**, pursue a fundamentally different approach. In contrast to the manipulation of individual **qubits** through quantum gates, these devices employ quantum superposition and tunneling to explore the energy landscape of a problem and to find the ground state of a Hamiltonian, which encodes the solution. This process is highly suitable for combinatorial optimization tasks such as in logistics or materials science. Although quantum annealers are not universal quantum computers, they demonstrate exceptional performance in specific problem classes that involve finding global minima in complex systems [34].

1.2.3 Topological Quantum Computers

Another QC approach is based on topological quantum computers, a nascent field, that aims to utilize anyons – quasiparticles exhibiting nonabelian statistics – to construct error-resistant qubits, thereby enabling resilient quantum computations in the presence of noise [35].

Chapter 2

Tuning Semiconductor Quantum Dot Spin Qubits

THIS chapter provides an introduction to the basics of QDs in Section 2.1 and elucidates their application as spin qubits in Section 2.2. Then, it explores the necessity of tuning the qubit devices and analyzing the corresponding data. Consequently, Section 2.3 provides an overview of the process of tuning spin qubits, while Section 2.4 describes the field of data analysis techniques for the automation of the qubit tuning.

2.1 Quantum Dots

Quantum dots (QDs) are nanoscale semiconductor structures that exhibit quantum mechanical properties due to their size confinement in all three spatial dimensions. Often referred to as "artificial atoms", QDs have discrete energy levels similar to those of real atoms [36]. Their unique properties arise from the quantum confinement of charge carriers (electrons or holes), leading to size-dependent optical and electronic behaviors. This attribute renders QDs highly versatile for various applications, including QC, quantum optics, photovoltaics, and biomedical imaging.

In semiconductors, electron motion can be restricted to two dimensions in quantum wells or at interfaces between materials [37], by completely depleting the density of conduction electrons at low temperatures. Moreover, electrostatic tailoring of the potential landscape enables the restriction of electron motion to one or zero dimensions, as evidenced by studies on few-electron QDs [38, 39].

In bulk materials, electrons can occupy a continuum of energy levels; however, when the material size is reduced to the nanoscale, quantum mechanical effects become predominant. In QDs, electrons and holes are confined within a space smaller than their de Broglie wavelength, which forces charge carriers into discrete energy states, much like electrons in an atom. The energy difference between these states can be tuned by altering the size of the QD, resulting in wider energy gaps for smaller QDs. This tunability is crucial for applications in optoelectronics and QC. The theoretical description and experimental demonstration of quantum confinement in semiconductor QDs, which occurred in the 1980s, marked the commencement of extensive research into their optical and electronic properties [40].

2.1.1 Quantum Dots in Quantum Computing

Loss and DiVincenzo presented the initial proposal for using QDs in QC in 1998. They proposed that the spin of an electron in a QD could function as a qubit and suggested a method for implementing two-qubit gates through exchange interactions [30]. Since then, experimental demonstrations have shown that QDs can achieve coherent control of qubit states, although challenges remain in achieving long coherence times and scaling up to large numbers of qubits. The usage of QDs as qubits requires cooling to temperatures close to zero Kelvin to achieve a system with well-defined discrete energy levels and sufficient coherence. Apart from the spin qubits used in this work, several other types of semiconductor qubits have been developed, including charge qubits, in which the information is represented using different charge occupations, and hybrid variants.

2.1.2 Challenges

Semiconductor QDs have demonstrated immense potential, yet several challenges persist. In QC, achieving long coherence times for spin-based qubits remains a significant challenge. QDs are sensitive to their environment and decoherence due to interactions with nearby nuclear spins or charge noise¹ which limits their performance in quantum devices. A major obstacle is the tuning of gate-defined semiconductor QDs to form reliable qubits, which involves using electrostatic gates to define and control the potential landscape, allowing for the precise manipulation of electron or hole spins. However, achieving the necessary precision is difficult due to variations in material properties and imperfections in the fabrication process. Inconsistent quantum gate performance can lead to unreliable qubit states, making it challenging to perform coherent quantum operations. Furthermore, the necessity of cooling to near zero Kelvin using dilution refrigerators introduces additional challenges, and the connection of the control electronics to the individual qubits gives rise to a so-called wiring bottleneck. The reduction of wiring, for example, by inserting control electronics into the cryostat, poses further challenges, as the cooling capacity is limited and local heating negatively affects the performance of nearby qubits. Advances in material science and engineering are required to mitigate these effects and improve the scalability of QD-based systems. Enhancement of the uniformity of QDs and improvement of gate design are critical for achieving robust qubit performance.

¹Charge noise refers to fluctuations in the local electrostatic potential, typically arising from unintended trapping and emission of charges at defects within the heterostructure.

2.2 Semiconductor Spin Qubits

2.2.1 Qubit Realizations

Semiconductor spin **qubits** are typically composed of materials that contain isotopes with an even number of nuclear spins. This reduces the hyperfine interaction, which results in an unknown phase. Material stacks in Ga, Si, and Ge, as depicted in [Figure 2.1](#), are of particular interest and differ primarily in their band structure, disorder, and integration capabilities [41]. Moreover, the integration of some of these stacks into existing Si and Ge semiconductor technology offers the advantage of potential scalable applications in **QC**. The fabrication of semiconductor **QDs** can be achieved by methods such as colloidal synthesis, molecular beam epitaxy, or chemical vapor deposition.

GaAs/AlGaAs heterostructures exhibit an exceptional structural quality, allowing for very large electron mobility and a low critical density for conduction, making them essentially disorder-free [41]. However, the decoherence caused by hyperfine coupling to the nuclear spin represents a significant drawback for GaAs. Moreover, the integration of GaAs into the prevailing Si poses significant technical challenges due to structural incompatibilities [42], thus constraining its scalability potential [41].

Spin **qubits** in Si are implemented in silicon metal-oxide-semiconductor (**Si-MOS**) structures or in Si/SiGe heterostructures. The employment of **Si-MOS** structures facilitates the operation of **qubits** at relatively elevated temperatures of one Kelvin [43], thereby rendering them a compelling candidate for the ultimate goal of co-integration with complementary metal-oxide-semiconductor (**CMOS**)-based cryogenic control circuits. However, multi-**QD** systems pose significant challenges due to rather low mobility and the high critical density for conduction in **Si-MOS**, hindering their effective control. Conversely, significant advancements in the challenging quality aspect of Si/SiGe heterostructures have yielded high mobility, low critical density for conduction and extended quantum coherence. The current state of device yield is so high that the design and control of larger linear arrays of **QDs** is possible with this technology [44]. However, the challenge of low valley splitting² persists, albeit addressed through the incorporation of more intricate Ge concentration profiles.

Using holes instead of electrons (two-dimensional hole gas (**2DHG**) in [Figure 2.1](#)), Ge/SiGe heterostructures combine many advantages of Si and GaAs, while overcoming the limitations of both. The status of Ge as a **CMOS**-foundry material provides a valuable perspective on its scalability [41]. Additionally, Ge can be isotopically engineered for long quantum coherence. Furthermore, Ge/SiGe heterostructures exhibit low disorder, a high mobility, and a low effective mass. Together with the sizeable spin-orbit coupling, this results in large energy spacing, facilitating the straightforward fabrication of **QDs** with comprehensive electrical **qubit** control. Only an adverse adjustment of the spin-orbit coupling may deteriorate coherence.

²The phenomenon of low valley splitting in Si/SiGe heterostructures is attributed to atomic-scale imperfections present at the quantum well interface. These imperfections restrict the energy separation between valley states [45]. This phenomenon can result in **qubit** state mixing, reduced coherence, and challenges in spin initialization and control.

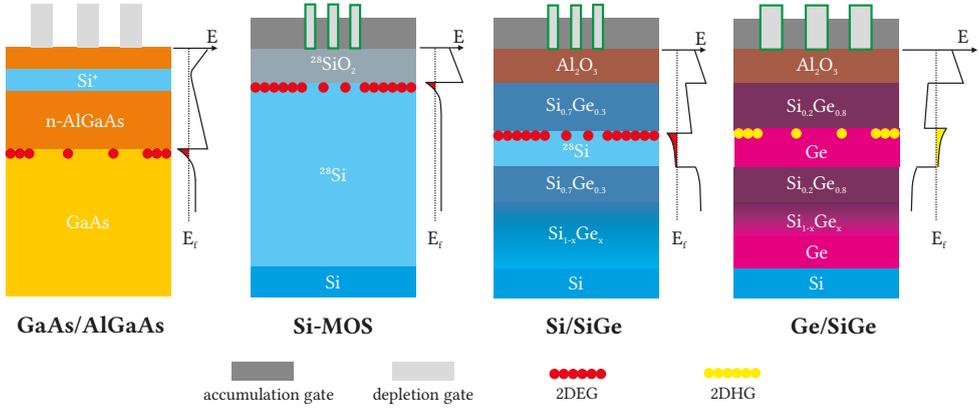


Figure 2.1: Material stacks and corresponding band diagrams for semiconductor spin qubits.

Several proposals for physical realizations of semiconductor spin qubit systems have been developed [46]:

- Loss-DiVincenzo spin qubits [30],
- donor spin qubits and Kane's proposal [47],
- singlet-triplet (ST_0 and ST_{\pm}) qubits [48, 49],
- exchange-only [50–52] and resonant-exchange qubits [53–55], and
- spin qubits with additional charge degrees of freedom [56–58].

Despite variations in the number of electrons utilized for each qubit, all are founded on the confinement of spin to isolated sites and the usage of the exchange interaction.

A singlet-triplet spin qubit is implemented through the use of two charge carriers confined within a double quantum dot (DQD). In this system, the qubit is represented by singlet and triplet states. These systems have shown to exhibit enhanced resilience against magnetic field noise.

The concrete construction of a quantum device necessitates meticulous engineering of the physical structures required for the creation, manipulation, and readout of qubits. In the case of semiconductor QD qubits, this process involves the development of sample layouts to confine and control electron or hole positions. Typically, a quantum well structure is grown to host a two-dimensional electron gas (2DEG). On top of the structure, metal gate electrodes are created by optical or electron beam lithographic techniques. These gates are later used to

- deplete the underlying 2DEG,
- form QDs,
- control the inter-dot coupling (by modulating tunnel barriers), and
- generally tune qubit properties.

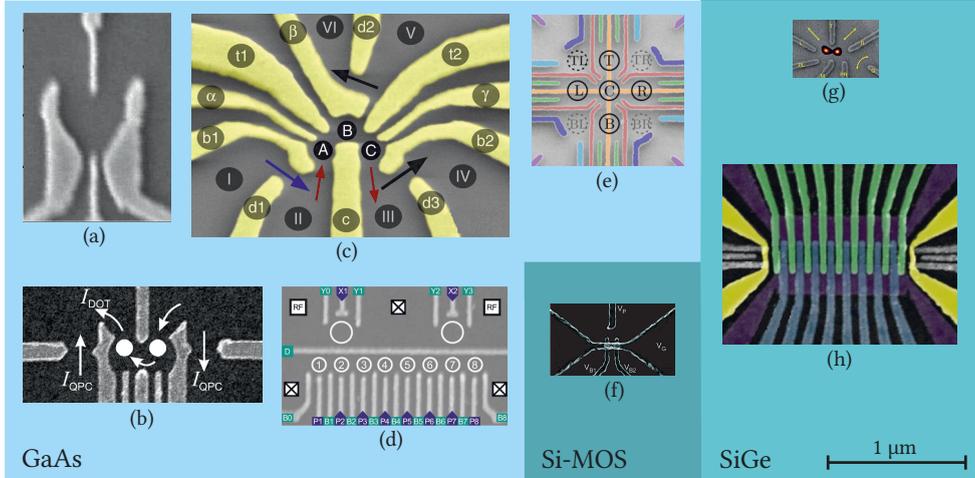


Figure 2.2: Examples of **qubit** samples of different materials and topologies. Images are sized to share common dimensional scales. (a)-(c) Single [59], double [60], and triple **QD** device [61]. (d)-(e) 8x1 [62] and 3x3 **QD** array [63]. (f) Single **QD** device [64]. (g) Double **QD** device [65]. (h) Spin **qubit** shuttle device (SQS or QuBus) [66].

The gates are often designated according to their intended function:

- screening gates shield **qubits** from undesired electric fields,
- depletion gates deplete the underlying **2DEG**,
- accumulation gates accumulate electrons in the underlying **2DEG**,
- plunger gates control the electrochemical potential of a **QD**, and
- barrier gates adjust the tunneling rates between **QDs** or to reservoirs.

Although the gates are not inherently limited to their primary function, their configuration is intentionally optimized to align with the specific objectives. The layout of the gates is a crucial factor in ensuring reliable **qubit** operation. A variety of different layouts were developed, as demonstrated in [Figure 2.2](#).

The present work focuses on the formation and tuning of **QDs**, with a particular emphasis on the gates that exert the most significant influence on this process, namely the plunger and barrier gates. Once the device is configured into a stable global state with a well-defined topology and a known number of charge islands, the barrier and plunger gates effectively control the **QDs**. [Figure 2.3](#) visualizes the potential landscape of a **DQD** formed with these gates. Barrier gates serve to create potential barriers that isolate specific regions, i.e. effectively "pinch-off" electron flow to define the boundaries of a **QD**. The voltage applied to the barrier gates adjusts the tunneling rates of charge carriers between the **QD** and adjacent reservoirs or other **QDs**, which is essential for quantum operations. Charge transitions between a lead (reservoir) and a **QD** are designated as lead-to-dot transitions (**LDTs**) and transitions between two **QDs** as

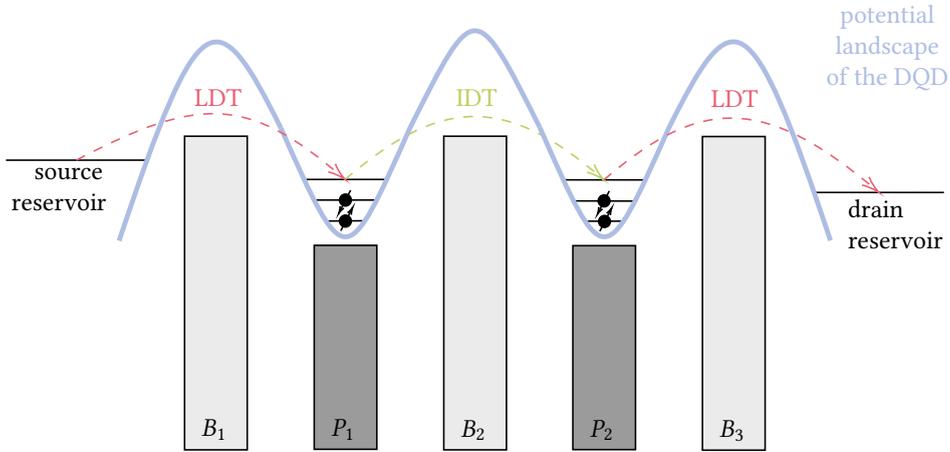


Figure 2.3: Visualization of the potential landscape for a DQD. Applying voltages to the barrier gates B_1 , B_2 , and B_3 creates the corresponding tunnel barriers in the DQD potential landscape. The voltages applied to plunger gates P_1 and P_2 control the potential of the two QDs, effectively changing the available energy levels for charge carriers. If an empty energy level below the source potential exists in the first QD, a charge carrier can tunnel into it. Charge carriers at energy levels above the drain potential can tunnel off the second QD. These transitions are called lead-to-dot transitions (LDTs). Additionally, charge carriers can tunnel between the two QDs in a so called inter-dot transition (IDT), if the energy levels are aligned, or if loading an electron from the source reservoir through the first QD into a lower energy level of the second QD.

inter-dot transitions (IDTs). In contrast, plunger gates are used to tune the electrochemical potential within the QD itself. Adjusting the voltage of a plunger gate enables fine control of the electron number and alignment of energy levels for desired quantum operations. This precise modulation is fundamental to the initialization, manipulation, and readout of qubit states. In summary, the interplay between the barrier and plunger gates is instrumental in confining and regulating charge carriers to achieve the single-electron (or hole) occupancy necessary for spin-based qubits [31].

2.2.2 Measurement Realizations

Accurate measurement of the QD properties is fundamental to operate QDs as qubits or to study quantum phenomena. Two primary methods, conductance measurements³ and radio

³Conductance measurements are often referred to as "transport measurements".

frequency (RF) reflectometry, are commonly employed; each is suitable for specific scenarios and sensitivities.

Conductance measurements directly measure the current through a QD as a function of the applied gate voltages. This invasive approach provides high-resolution insights into charge stability, Coulomb blockade, and energy-level spacing by monitoring electron transport under varying conditions.

The Coulomb blockade regime refers to a condition in which the addition of a single electron is energetically suppressed due to the strong electron-electron repulsion, also known as the Coulomb interaction. This phenomenon transpires when the QD is sufficiently small such that the addition of an extra electron necessitates a discrete charging energy, which cannot be surmounted at low bias voltages. Consequently, the flow of current through the device is suppressed, with the exception of specific gate voltages at which the energy levels are aligned. This enables the individual tunneling of electrons, a characteristic that is indicative of single-electron transport. However, this transport typically requires low-resistance leads and works optimally in systems in which a certain current flow (e.g., nanoamperes) is permissible.

RF reflectometry employs high-frequency circuits to detect changes in impedance caused by the QD's charge state. This technique allows noninvasive, high-bandwidth charge sensing, rendering it well-suited for scenarios in which direct current measurement is impractical, such as qubit readout or devices with minimal electron transport. It offers higher speed and sensitivity than conductance measurements and is particularly effective in multi-dot systems or when probing higher energy states.

For noninvasive spin qubit readout, additional techniques based on charge sensing and spin-to-charge conversion leverage a nearby quantum point contact (QPC) [67] or a sensor dot (SD) [68]. Both can be combined with RF measurement techniques [69–71].

A QPC is a one-dimensional constriction between two wide conducting areas with a width comparable to the electronic wavelength [72]. It exploits the fact that electron flow through this narrow channel occurs in discrete steps of $2e^2/h$ (so-called conductance quantum) [73]. The quantized conductance and high sensitivity to the electrostatic environment enable the measurement of the charge states of nearby QDs. Changes in the number of electrons in the QDs modify the local potential of the QPC, which is reflected in a change of its conductance. This effect is measured as a shift in current or voltage across the QPC.

A SD is a QD that physically implements a single-electron transistor (SET), enabling the readout of nearby QDs employed as qubits. Figure 2.4 illustrates the spatial configuration of a SD positioned in close proximity to a DQD, represented in a top-down view of an exemplary gate architecture. The SD is capacitively coupled to the QDs of interest to detect discrete changes in the charge configuration [71]. Its operation is based on Coulomb blockade, where the conductance of a SD oscillates with the addition of individual electrons. Changes in the charge state of the nearby QDs shift the SD's electrostatic potential, altering its conductance and the corresponding current. By tuning the SD close to the edge of a Coulomb blockade peak, small electrostatic perturbations from the target QDs induce quantifiable changes in the SD's conductance.

According to Barthel et al. [71], the sensitivity of a SD measurement with RF reflectometry is up to 30 times greater than that of a comparable QPC. Furthermore, the signal-to-noise ratio

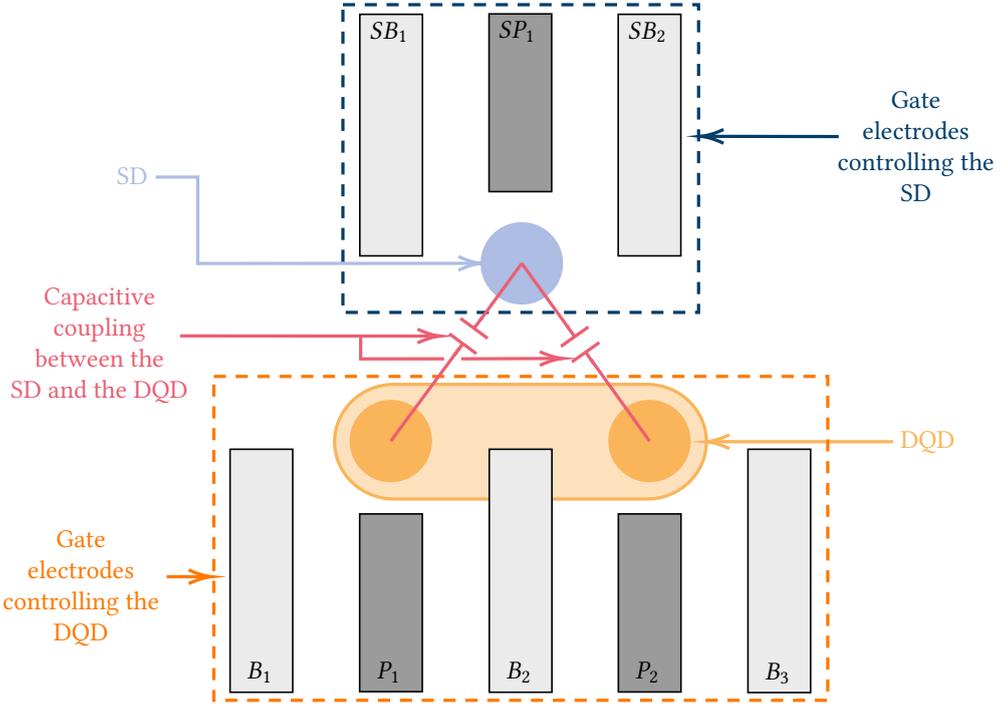


Figure 2.4: Spatial configuration of a SD positioned in close proximity to a DQD, represented in a top-down view of the gate architecture.

in single-shot reflectometry measurements is three times greater than that of QPC measurements.

The spin-to-charge conversion technique, introduced by Elzerman et al. [74], serves as foundational methodology for the quantification of the spin states of semiconductor qubits using charge sensors. The underlying principle of this technique entails the mapping of the spin state to a distinct charge configuration, which can subsequently be detected through the employment of highly sensitive charge sensors.

In practice, the method generally entails a spin-dependent tunneling process. To illustrate, in a singlet-triplet qubit, the spin state of the electron dictates whether it remains in the initial QD or tunnels to a nearby dot or reservoir. For a singlet state, the Pauli exclusion principle permits both electrons to occupy the same dot, whereas a triplet state forces one electron to tunnel. This spin-dependent movement generates a measurable change in the charge configuration, effectively translating the spin information into a charge signature. This enables rapid, single-shot qubit state detection, as demonstrated by Barthel et al. [68]. Similarly, another approach [75] exploits spin-dependent tunnel rates to determine the spin state from a timed occurrence

of a charge transition (CT). These readouts are crucial to scalable QC architectures and are thus indispensable for implementing quantum error correction protocols.

2.3 Qubit Tuning and Automation Approaches

The tuning of semiconductor QDs is essential for creating stable and reliable qubits, particularly as device complexity increases for QC applications. However, the tuning process is challenging due to the necessity of precision for controlling the electrostatic potentials, charge states, and inter-dot couplings [76]. Recent advancements in automation and machine learning (ML) have led to the proposal of approaches to some stages of the tuning process [77]. In the following, the different tuning stages are introduced in a generic, architecture-independent manner with recent automation approaches. The experimental tuning workflow and the principal measurement data underpinning this work are described in detail as needed in Chapter 3.

2.3.1 Turn-on and Initialization

The preliminary stage of tuning entails the evaluation of gate functionality and the determination of the initial gate voltages that establish the fundamental confinement potential for QDs within a semiconductor heterostructure. It is imperative to ensure that each dot can be formed and initialized individually prior to conducting more sophisticated tuning procedures.

2.3.2 Sensor Tuning

Sensor tuning is an essential process for achieving precise detection of charge states by configuring a nearby SD or QPC as a sensitive charge sensor typically measuring the conductance via the current. This tuning process requires precise gate voltage adjustment to optimize the sensitivity to single-electron transitions, a prerequisite for effective real-time monitoring in QC devices [78, 79]. Maintaining stable sensor operation continues to be a bottleneck in the tuning procedure [80].

2.3.3 Dot Regime Tuning

The objective of dot regime tuning is to form the desired type of dot, such as a single or double QD. ML models have streamlined this stage, with early studies demonstrating the neural networks' ability to identify suitable configurations rapidly and establish initial QD formations [81].

Furthermore, ML approaches, such as reinforcement learning, have accelerated the search for optimal gate configurations, improving the speed and efficiency of turn-on protocols. In

some cases, visual-based image processing approaches empower algorithms to process charge stability diagrams (CSDs) and locate the necessary initial configuration [82–84].

2.3.4 Charge State Detection and Single-Electron Regime

Following the initialization of a QD, the tuning process continues by refining gate voltages to reach the single-electron regime. This stage is crucial for reliable qubit operation, as single-electron occupancy enables controlled manipulation of quantum states. Autonomous tuning systems rely on ML models trained to detect specific charge states, with studies leveraging supervised learning techniques in single and double QD setups [81, 85–87].

In this field, the ray-based classification (RBC) framework proposed by Zwolak et al. [88, 89] merits particular attention. It harnesses ML in conjunction with one-dimensional measurements to reliably and efficiently identify QD charge states in complex QD systems. Subsequently, Ziegler et al. advanced this approach through the development of a physics-informed tuning framework, which combines ML classifiers with physics-based heuristics to automate and enhance the tuning of QD charge states [90].

2.3.5 Inter-Dot Coupling and Tunnel Barrier Fine Tuning

Establishing and adjusting inter-dot coupling through tunnel barrier tuning is essential for DQD qubits, which rely on precise exchange interactions. The inter-dot coupling must be calibrated with accuracy to control the entanglement and interaction strength; however, it exhibits a high degree of sensitivity to gate voltage fluctuations. To address this challenge, automated systems that monitor inter-dot tunneling transitions are being developed. These systems utilize adaptive ML frameworks automatically adjusting barrier gate voltages for reliable coupling [91, 92].

2.3.6 Further Fine-Tuning and Stabilization of Qubit Parameters

Following the attainment of fundamental charge and coupling states, the fine-tuning proceeds adjusting the qubit’s electrostatic environment to enhance coherence, minimize error rates, and define qubit-specific operational gate configurations. This stage encompasses the tuning of spin coherence times, the adjustment of resonance frequencies, the optimization of magnetic field gradients in spin qubits, and the localization of measurement and reload points. Autonomous fine-tuning algorithms address multiple tuning objectives to balance coherence and control fidelity, thereby reducing the need for human surveillance [80, 93].

2.4 Data Analysis Techniques

The **qubit** tuning process requires the continuous adaptation of device parameters to guide the **QD** system to the desired operational regime. The ability to analyze and interpret data is imperative for identifying the current state of the system and informing the adjustments required for optimal **qubit** operation.

QD devices exhibit complex and nonlinear behavior due to fabrication-induced variability, environmental noise, and control voltage drift. On the one hand, noise analysis techniques enable precise assessment of the measurement reliability in terms of the signal-to-noise ratio. On the other hand, classical pattern recognition techniques play an essential role in processing one- and two-dimensional **QD** measurements, enabling the identification of key features like **CTs** and tunnel couplings. However, the capabilities of classical techniques are limited when attempting to automate complex classification [81, 85, 94], measurement [95], and parameter optimization tasks [87], a field in which advanced **ML** approaches have demonstrated significant progress and advantages.

These advanced data analysis techniques enable dynamic feedback mechanisms, where parameters are updated iteratively to correct for deviations and ensure robust **qubit** operation. This capability is critical for scaling multi-**qubit QD** systems, bridging the gap between experimental challenges and the realization of functional quantum processors.

The following subsections introduce the concepts of various analysis techniques for noise estimation and pattern recognition.

2.4.1 Noise Analysis

Noise analysis remains a prominent subject in the realm of data analysis and processing, with an ongoing discourse surrounding its methodologies and applications. In the domain of signal processing, noise is defined as unknown and unwanted distortions that a signal may encounter during acquisition, storage, transmission, processing, or conversion [96].

Accurate characterization of noise facilitates the separation of meaningful information from extraneous disturbances, thereby enhancing the quality of data and optimizing subsequent processing tasks. The ability to estimate noise is indispensable in fields ranging from telecommunications [97] and imaging [98] to material science and quantum mechanics [99], where precise data interpretation is paramount. The development of noise estimation emerged from the necessity to enhance the comprehension and modeling of measurement uncertainties. Early contributions in the mid-20th century, including Wiener's theory of optimal filtering [100], placed emphasis on modeling noise as a stochastic process to improve signal reconstruction. This approach was complemented by research on spectral analysis methods, including the seminal contributions of Blackman and Tukey [101] on power spectral density (**PSD**) estimation. These foundational techniques laid the groundwork for extracting noise statistics directly from observed data, enabling both noise suppression and the development of robust models for data analysis.

Noise estimation has evolved in tandem with advancements in computational capabilities and data-driven technologies. By enabling a precise understanding of noise behavior, it fosters critical improvements in data reconstruction, uncertainty quantification, and system optimization. As a foundational element of signal processing, noise estimation maintains its significance in propelling the frontiers of scientific and technological innovation.

2.4.1.1 Parametric Noise Estimation

Parametric noise estimation approaches are predicated on the assumption that an underlying mathematical model for the noise exists, such as Gaussian, Poisson, or Laplacian distribution. The estimation of parameters such as mean, variance, or spectral density is achieved through the utilization of techniques like

- maximum likelihood estimation (MLE), which identifies parameters that maximize the likelihood of observing the measured data;
- method of moments, which calculates distribution parameters by equating the observed moments (mean, variance) to the theoretical moments of the assumed noise model;
- least squares estimation, which is a common method in regression analysis that seeks to minimize the sum of squared residuals, thereby inferring the noise variance; and
- principal component analysis (PCA), which entails the transformation of a dataset into a new coordinate system, wherein the dimensions (principal components) are arranged according to the extent of variance they encapsulate within the data. This procedure facilitates the segregation of noise (lower-variance components) from meaningful signal components (initial principal components).

Parametric methods are characterized by their computational efficiency; however, they are often vulnerable to nonstandard noise distributions.

2.4.1.2 Nonparametric Noise Estimation

Nonparametric approaches eschew assumptions concerning the nature of the noise distribution, instead employing direct analysis of data features. Examples include

- kernel density estimation (KDE), which estimates the probability density function of noise by placing smooth kernel functions (e.g., Gaussian kernels) at each data point and summing their contributions [102];
- empirical histogram-based methods, which divide data into bins and count occurrences within each bin, thereby providing a piecewise approximation of the noise distribution;
- Fourier-based spectral analysis, which analyzes the noise in signals in the frequency domain to identify dominant frequency components;
- wavelet-based methods, which decompose signals into distinct frequency bands, where high-frequency components are frequently associated with random noise [103]; and

- variance-based analysis, where techniques such as sliding window variance estimation calculate the local variance in small regions of an image to identify areas that are dominated by noise. This method is particularly effective in the context of adaptive denoising techniques.

Nonparametric methods have been shown to be highly effective in the analysis of nonstandard or complex noise patterns. However, these methods are often computationally intensive and require larger datasets to achieve the desired level of accuracy. They are particularly effective when the noise exhibits irregular structures or when the underlying distribution is unknown.

2.4.1.3 Noise Estimators

In the context of noise estimation for one-dimensional data, estimators frequently ascertain the standard deviation of the noise component of a signal. In the spatial domain, techniques such as local filtering and statistical analysis techniques [104–109] have been employed. However, contemporary approaches employ alternative methods to enhance estimation accuracy [110–114]. Patch-based approaches subdivide the data into homogeneous regions (patches, blocks, or superpixels) and incorporate filtering and statistical methods [115–119] and PCA [120, 121] for the estimation. The patch selection methods themselves are diverse, as evidenced by the works of [121–131]. Transform-based approaches entail the application of a transformation to the given data, with different approaches for noise identification being assumed to allow for the separation of noise in the transformed space. Noteworthy methods include the wavelet decomposition [132–136], the discrete cosine transform [137–142], as well as the Morrison noise reduction method [143]. However, extant state-of-the-art estimators frequently specialize in a particular application and data dimensionality. A more general insight into different noise estimators can be found in [111, 144–147].

2.4.2 Pattern Recognition Using Classical Processing Pipeline

Classical pattern recognition techniques emerged from the need to interpret visual data in various domains, including medical imaging, industrial inspection, and remote sensing. These methods, which predate the era of deep learning, typically employ mathematical analysis methods. They are designed to refine image information and extract features and patterns from images. The key steps include preprocessing, segmentation techniques, feature extraction, and semantic detection methods (classification / object detection), each of which addresses specific aspects of image analysis. The typical classical pattern recognition processing pipeline is illustrated in [Figure 2.5](#) in comparison to the deep learning approach.

2.4.2.1 Preprocessing

Preprocessing is of fundamental importance in pattern recognition and serves as a critical step to enhance the quality of raw data prior to further analysis. The genesis of this field can

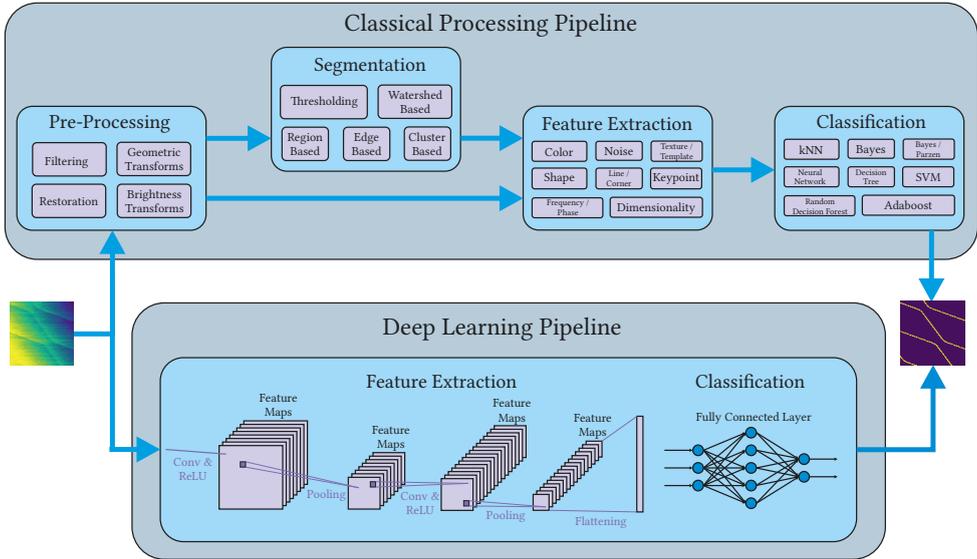


Figure 2.5: A comparison of the conventional classical pattern recognition pipeline and the deep learning approach reveals notable distinctions. In the classical pipeline, disparate algorithms are applied for the different stages, whereas in the deep learning context, all steps can be executed by a unified approach.

be traced back to early research in signal processing and image analysis, in which researchers identified the necessity to mitigate noise and enhance data fidelity.

The Fourier transform [148] and its discrete form discrete Fourier transform (DFT) with the algorithmic advancement fast Fourier transform (FFT) [149] are fundamental to preprocessing tasks, particularly in de-noising, image reconstruction, and frequency-domain analysis. Compared to convolution in the position space, the FFT can reduce the processing cost of filtering operations if the signal is larger. As a key preprocessing task, numerous approaches to suppress noise have been developed, including the Gaussian and median filter [150] as well as extensions designed to preserve structural information, such as the Wiener filter [100], the bilateral filter [151], the anisotropic diffusion [152], and the BM3D approach [153].

The field of preprocessing continues to evolve, ensuring data integrity and quality for accurate and robust pattern recognition.

2.4.2.2 Image Segmentation

Image segmentation is the process of dividing an image into continuous regions based on proximity linkage and intensity information. Region growing techniques are popular classical approaches, which exploit the homogeneity information of neighboring pixels or subre-

gions when applied in a hierarchical manner [154, 155]. A cornerstone of modern clustering methods is the k-means algorithm [156, 157]. It is based on the core concept of partitioning data into clusters to minimize within-group variance [158] but provides additional statistical insights that firmly established the method as a cornerstone. The edge based segmentation methods identify boundaries within images for the extraction of object shapes and spatial organization. Classical edge detection methods, including the Sobel operator [159] and the Canny edge detector [160], depend on gradient calculations to discern rapid intensity variations. Furthermore, the connected component analysis [161] and watershed algorithms [162] play significant roles in segmenting images based on pixel connectivity and intensity gradients.

2.4.2.3 Feature Extraction

Raw data are complex and challenging to process without extracting or selecting appropriate features. Feature extraction involves reducing the data in a way that the extracted features capture the essential characteristics of the original data, thereby facilitating more efficient processing and analysis. The reduction criterion generally aims to either enhance or sustain accuracy or simplify model complexity. In addition, it can contribute to preventing the model from overfitting. The subsequent overview outlines frequently employed features for pattern recognition in image processing:

- intensity (histogram, moments, coherence vector),
- edge (gradient-based, Gaussian difference-based, multiscale feature-based, structured learning-based) [163],
- line (Hough transform-based, local & gradient grouping-based, local & edge fitting-based) [164],
- corner (intensity-based, curvature-based, model/template-based) [165],
- local feature descriptors (keypoints) [166],
- shape (global contour-based, global region-based, structural contour-based, hybrid methods) [167], and
- texture (statistical-based, structural-based, model-based, spectral-based, graph-based, hybrid methods) [168].

Features are not necessarily uncorrelated. Therefore, the application of dimensionality reduction methods serves to reduce the complexity of high-dimensional data and enhance model interpretability. Dimensionality reduction includes methods like [PCA](#), linear discriminant analysis, autoencoders, t-distributed stochastic neighbor embedding, and independent component analysis.

2.4.2.4 Classification

Classification constitutes a foundational task in pattern recognition, wherein the objective is to assign data points (pixel-level classification or semantic segmentation) or a data entity to

predefined categories or classes based on their features. This necessity emerged with particular prominence in the mid-20th century, coinciding with the growth of automation in domains such as document categorization, speech recognition, and medical diagnostics.

Classical classification techniques encompass a range of statistical methods, algorithmic approaches, and neural network architectures. One foundational approach is the Naive Bayes classifier, which utilizes Bayes' theorem to estimate the probability of class membership under the simplifying assumption of feature independence. Despite its simplicity, the Naive Bayes classifier has proven to be effective in applications such as spam detection and text classification. Support vector machines (SVMs) address the problem of finding a hyperplane that maximally separates classes in a high-dimensional space. Since the proposal of using kernel tricks [169] and the "soft margin" variant [170], SVMs have gained prominence for their robustness in handling nonlinear decision boundaries. Another classical algorithm is the k-nearest neighbors (k-NN) method, a nonparametric technique that assigns class labels based on the majority vote among the nearest data points. Neural networks, particularly shallow networks, represent another class of classical classifiers. Early models, such as the perceptron [171], demonstrated the ability to separate linearly separable data using weighted sums of features. Despite being overshadowed by deep learning in recent years, these networks remain computationally efficient and interpretable for small-scale tasks.

This suite of classical methods continues to play a role in modern applications, particularly in scenarios where data is limited, interpretability is critical, or computational efficiency is crucial.

2.4.2.5 Object Detection

Object detection is the process of identifying and localizing particular objects within an image or scene. It addresses the dual challenges of classification and spatial localization and predicts bounding boxes around objects. The necessity for object detection became apparent with the emergence of computer vision applications in domains such as autonomous vehicles, industrial automation, and surveillance systems. The primary challenge has been to accurately detect and localize objects of varying scales and appearances in cluttered and dynamic environments.

Early approaches relied heavily on statistical pattern recognition and template matching techniques. Viola and Jones introduced one of the first real-time object detection frameworks, which uses cascaded classifiers and Haar features for face detection [172]. The sliding window approach, often paired with SVMs, was another foundational method. In this approach, the image is systematically scanned at different scales, and an SVM is employed to classify each window as containing an object or not. Feature-based methods, such as the histogram of oriented gradients (HOG) [173], became another cornerstone for object detection. These methods provide robust descriptions of object shapes and textures, improving detection accuracy.

Despite their computational efficiency and interpretability, classical object detection methods established the foundation for more sophisticated systems. Their continued utilization is predominantly in constrained environments where computational resources are scarce or the precision of deep learning techniques is deemed superfluous. Such environments include industrial inspection and embedded vision systems.

2.4.3 Pattern Recognition Using Deep Learning Pipeline

Despite the notable advancements and enduring relevance of classical pattern recognition techniques, particularly in terms of their computational efficiency and simplicity within specific applications, they are increasingly being supplemented or even replaced by deep learning methods. The superior performance of deep learning techniques in handling complex and high-dimensional data has led to a fundamental transformation in the field of pattern recognition.

Deep learning methods are capable of learning from data to identify complex patterns and structures, classify images, recognize objects, and restore images. The characteristics of these systems methods are defined by deep hierarchical neural networks, which contain multiple intermediate layers, or hidden layers, between the input and output layers (see [Figure 2.5](#)). The extensive internal structure is capable of automatically extracting abstract features from the data. In addition to classical neural network structures, special convolutional layers and layers for de-linearization and downsampling (pooling) operations are often employed to construct convolutional neural networks (CNNs).

The development of CNNs was profoundly influenced by the pioneering work of David Hubel and Torsten Wiesel in the 1960s. Their seminal investigations into the hierarchical structure of visual processing in the brain led to the identification of simple and complex cells in the visual cortex that respond to specific visual features such as edges and orientations [174, 175]. Their insights into feature detection laid the foundation for the layered architecture of CNNs, where convolutional layers extract patterns in a manner analogous to biological visual systems. Building on these principles, Kunihiko Fukushima developed the Neocognitron in 1980 [176], an early artificial neural network that mimics this hierarchical feature extraction. However, it lacked efficient learning mechanisms. The modern CNN framework was realized with LeNet-5 by Yann LeCun et al. in 1998 [177], integrating backpropagation for efficient weight optimization and applying CNNs to handwritten digit recognition.

The efficacy of deep learning methods in predicting is well-documented; however, there is a caveat regarding the practical implementation of these methods. The training of deep learning models necessitates substantial computational resources, including powerful graphics processing units (GPU) and extensive amounts of memory, which can impose considerable expenses and demands on time resources.

The phenomenon of overfitting arises when a model is excessively trained on a particular dataset and subsequently performs inadequately on new, unseen data. This phenomenon is a prevalent challenge in deep learning, particularly in the context of large neural networks,

and can be attributed to a variety of factors, including insufficient data, the complexity of the model, or the absence of regularization.

The interpretability of deep learning models, especially those with numerous layers, can be intricate, hindering the comprehension of their predictive processes and the identification of potential errors or biases. Furthermore, the efficacy of deep learning algorithms is contingent on the quality and quantity of the data on which they are trained. Consequently, data that is noisy, incomplete, or biased can adversely affect the performance of the model.

2.4.3.1 Applications

This section introduces ML-based contributions that address the tasks previously discussed in the context of classical pattern recognition in Section 2.4.2.

Early methods for image segmentation, like active contour models and watershed algorithms, were replaced by CNN-based approaches, like U-Net [178] for biomedical imaging and Mask R-CNN [179] for instance segmentation. Recent approaches incorporate transformers [180–186] or state-space models (SSMs) such as the visual state-space model (VMamba) [187–190] to further enhance detection abilities.

A significant milestone for image classification based on CNNs was marked by the introduction of AlexNet [191] in 2012, which employed deep architectures and GPU acceleration to achieve dominance in the ImageNet competition, thereby solidifying CNNs as the state-of-the-art for image classification tasks. Subsequent innovations, including GoogleNet [192], VGGNet [193], ResNet [194], MobileNet [195], and EfficientNet [196] have refined CNNs, addressing computational challenges and enhancing accuracy.

In object detection, modern methods like R-CNN [197] and YOLO [198] use deep learning to simultaneously predict object classes and their locations, significantly improving efficiency and accuracy.

In addition to the aforementioned applications, deep learning approaches have also been directly employed for feature extraction, including edges and contours [199–205], lines [206–211], corner and keypoints [212–216], and textures [217–222].

2.4.3.2 Model Optimization

The implementation of deep learning necessitates a high degree of technical expertise and comprehension. It also requires substantial engineering investment, including data preparation, feature engineering, and configuration of the training methods. This is particularly crucial for model generation, with the objective being to achieve optimal output with minimal model complexity based on the extracted features. Accordingly, the domain of automated machine learning (AutoML) [223] is an emerging research field that aims to address this issue. Significant research within this domain encompasses the automated hyperparameter

optimization (HPO) and neural architecture search (NAS), exhibiting a substantial overlap between the two, though the techniques for HPO and NAS are often substantially different.

The process of HPO entails the selection of optimal parameters for a given task, with the objective of maximizing the accuracy of the model. To address this challenge, various HPO algorithms [224] have been developed over time to automate the selection process, thereby facilitating more expeditious and efficient parameter optimization. The optimization procedure typically employs an iterative approach, wherein all parameter values are continuously adjusted to ascertain the optimal values.

The NAS method [225–227] is an automated approach that traverses the architecture search space for a given task and generates models with the capacity to compete with hand-crafted, state-of-the-art architectures. The primary focus of NAS is the optimization of architectural topology, which can exhibit greater complexity in comparison to the HPO approach.

2.5 Motivation

Scalable general purpose QC requires the automation of the complete tuning workflow. Despite advances, challenges persist in scaling these automated methods across increasingly complex QD arrays. A primary challenge is achieving consistency in device tuning because heterogeneity in material properties and environmental noise levels necessitates adaptable, robust tuning models. As QD qubit arrays continue to expand, the deployment of advanced ML architectures is expected to play a pivotal role in enabling scalable, high-fidelity tuning across diverse architectures and device platforms.

Presently, the majority of data processing, tuning, and control operations work on external room-temperature hardware and wiring, which introduce latency, thermal load, and noise, thereby limiting the velocity and fidelity of quantum operations. To address these challenges, integrating data processing within the cryostat environment is imperative, signifying a necessary paradigm shift. Disposing the "wiring bottleneck" [228–233] minimizes the length and complexity of signal pathways and reduces signal distortion, thermal crosstalk, and signal attenuation. Furthermore, the cryogenic integration enables localized real-time feedback for autonomous measurement routines, thereby significantly enhancing the reliability and adaptability of multi-qubit systems. Advanced cryogenic electronics, such as low-noise amplifiers, multiplexers, and on-chip ML units, have the potential to further decrease the overhead associated with transferring large datasets to room-temperature processors [229, 230].

Efforts to develop robust data analysis solutions are accompanied by parallel initiatives to reduce complexity, a critical step toward scalability. Achieving this objective necessitates a deliberate examination of various factors. One way involves the minimization of data and measurement time by substituting two-dimensional measurements with a reduced set of one-dimensional measurements [88, 89]. However, transitioning the data acquisition to a cryostat environment introduces challenges, including diminished measurement precision due to the demanding conditions. Consequently, an initial increase in the volume of measurement data

may be required to ensure robust and reliable analysis. Therefore, the selection of analysis techniques must be made with these considerations in mind.

The development of an efficient tuning framework requires a comprehensive model of the tasks in the different stages, along with their requirements and goals. Moreover, the dependencies of the various tasks must be modeled to ensure that the interfaces are compatible. This model facilitates the identification of crucial parts in the tuning workflow with high potential for data transmission reduction.

Chapter 3

Experimental Setup and Identification of Crucial Tuning Steps

THIS chapter is dedicated to the exposition of the experimental setup, accompanied by a thorough delineation of the hardware configuration and the GaAs samples that were utilized in the present study. It identifies crucial steps in the tuning process with significant potential for optimization, focusing on processes that are critical to qubit scalability. Additionally, it explicates the required measurements and data acquisition techniques, thereby providing a comprehensive framework for understanding the required data analysis tasks. While the research is specific to this experiment, the results are transferable to other architectures.

3.1 Qubit Experiment

The qubit experiment on which this work is based aims to form DQDs to use them as singlet-triplet spin qubits. It uses a GaAs sample similar to the one described by Volk et al. [62], with a gate layout designed for the formation of two DQDs and a corresponding SD each. Figure 3.1 illustrates the arrangement of the gates that form the regions for the SDs and QDs. The sample is cooled to millikelvin temperatures in a dilution cryostat and the signal lines are progressively thermalized along successive temperature stages, as illustrated in Figure 3.2. Botzem [234] and Cerfontaine [235] describe the overall experiment setup, while Figure 3.3 shows the schematic representation of the signal path.

The sample printed circuit board (PCB) contains the current sample with the gates for the SDs and QDs. In the direct current (DC) part, a digital-to-analog converter (DecaDAC) processes and generates the gate voltages. Then, the voltages are filtered by the breakout box before they enter the level cooled to 55 K. At this stage, 12 twisted pairs of DC lines (constantan looms) transmit the signals to the RC filter box, which operates at only 10 mK. Subsequent to the filtering process, the voltages reach the gates on the sample PCB.

The RF part implements a homodyne detection setup. A single tone generated by an RF source is sent via a coaxial cable and a circulator to the sample PCB where it is matched to the SD impedance by an LC tank circuit. The circulator directs the signal reflected at the sample to a second coaxial cable. Amplitude and phase of the reflected RF signal are sensitive to changes in the impedance of the sample, allowing the inference of charge dynamics. This signal is

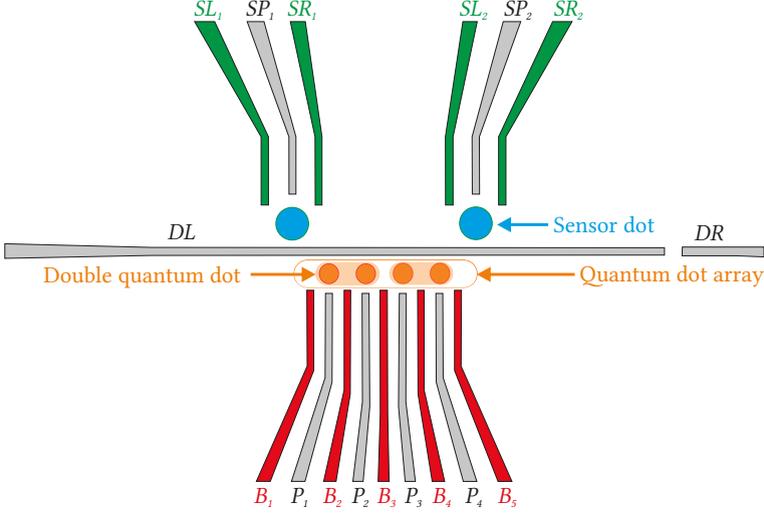


Figure 3.1: Arrangement of the DQD array and SD gates. Blue spots mark SD regions and orange spots QD areas. The gates SL_1 , SP_1 , and SR_1 form the left SD, the gates SL_2 , SP_2 , and SR_2 the right one. One DQD is formed by the gates B_1 , P_1 , B_2 , P_2 , and B_3 and another one by the gates B_3 , P_3 , B_4 , P_4 , and B_5 .

thermalized by a sapphire stripline and amplified at 3.5 K and at room temperature, where it is demodulated, low-pass filtered, and finally digitized using a PCI Express card.

Throughout this work, measurement values are reported in arbitrary units (a.u.), because the RF reflectometry constitutes relative changes and is not calibrated to an absolute scale.

As reported by Botzem [234] and Cerfontaine [235], charge noise and thermal noise are the most relevant noise sources for semiconductor QD systems (see Table 3.1). Dial et al. [236] state that charge noise exhibits pink characteristics. Quantitatively, the PSD of colored noise follows

$$S_{1/f} = \frac{S_0}{f^\alpha}, \quad (3.1)$$

where S_0 is a device-specific constant, f represents the frequency, and $\alpha \in [0, 2]$ specifies the spectral slope. An α of approximately one characterizes pink noise, which is also referred to as $1/f$ -noise.

Thermal noise, also referred to as Johnson-Nyquist noise, has a nearly Gaussian amplitude distribution if limited to a finite bandwidth [237]. In electronic devices, the constant, named white, PSD characteristic of thermal noise depends on the temperature and the resistance, following

$$S_V = 4k_B T R, \quad (3.2)$$

where k_B is Boltzmann's constant, T the absolute temperature, and R the resistance, which corresponds to the real part of the complex impedance.

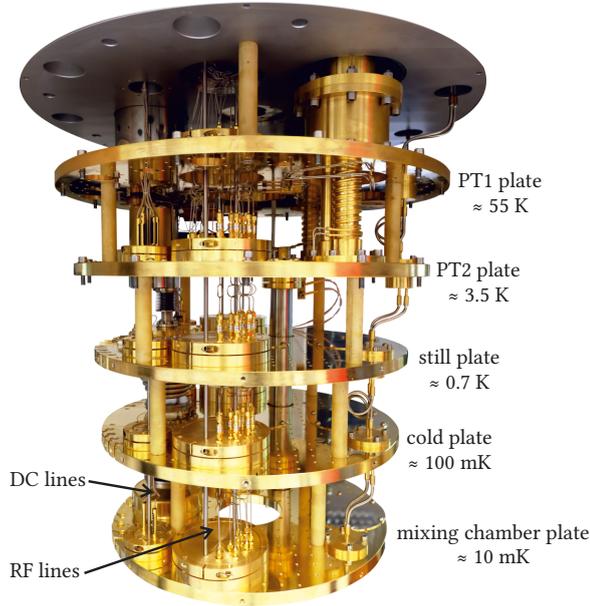


Figure 3.2: Cross-sectional view of the interior of a dilution refrigerator, illustrating the various thermal stages.

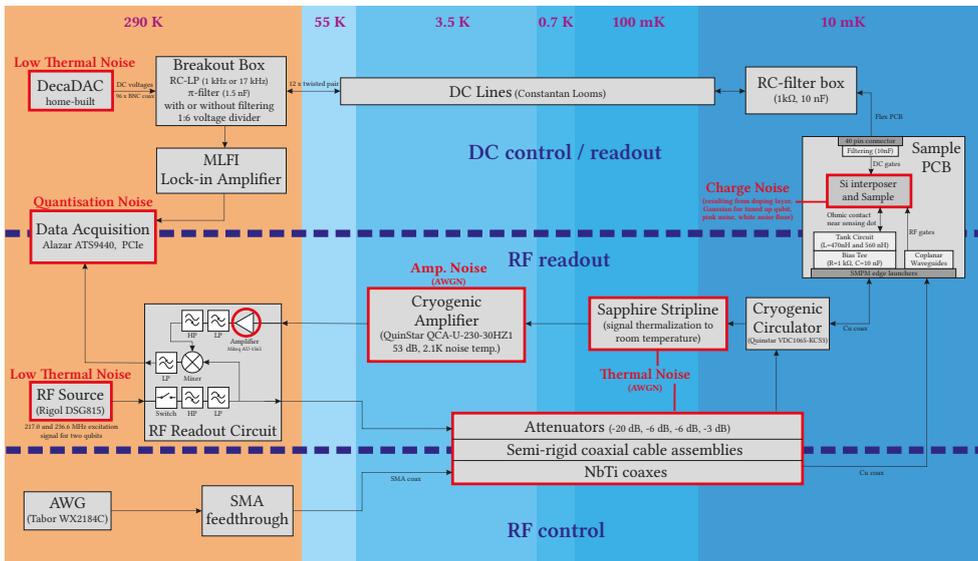


Figure 3.3: Diagram of the signal path.

Table 3.1: Most relevant noise sources in QD systems according to Botzem [234] and Cerfontaine [235].

Noise type	Noise source	Frequency characteristics	Probability distribution
Charge noise	Fluctuating charges in the heterostructure	Pink	Gaussian
Thermal noise	Thermal agitation of the charge carriers in the amplifier	White	Gaussian

3.2 General Tuning Workflow

Tuning a semiconductor QD is a multifaceted process that involves a series of steps, as previously outlined in Section 2.3 and visualized in Figure 3.4. Hereinafter, these steps are integrated into a comprehensive tuning workflow for the current experiment and assigned to different stages, as depicted in Figure 3.4. The operational sequence of the superordinated stages is depicted in Figure 3.5.

The tuning process begins with the bootstrapping, which involves the setup of the measurement equipment and an initial characterization of the gates (refer to Figure 3.6). This is achieved by testing the capability of each gate to pinch-off the underlying 2DEG-channel, beginning with the depletion gates (denoted as DL and DR in Figure 3.1) and progressing with the "finger" gates (specifically, SL_i , SP_i , SR_i , B_i , and P_i in Figure 3.1). The pinch-off values obtained from these measurements serve as a reliable starting point for adjusting the gate voltages in subsequent tuning steps. Following this, the tunability of the dot is assessed by determining the feasibility of dot formation. This is typically verified by observing Coulomb blockade signatures in conductance measurements.

The subsequent step involves the coarse tuning of the QDs (see Figure 3.7). The calibration of the SD is the initial step in the coarse tuning process (see Figure 3.8). These tuning steps typically rely on observations of the conductance characteristics of the SD. Initially, the formation of the SD is achieved by measuring a barrier-barrier conductance measurement wide-scan and selecting the region exhibiting Coulomb oscillations¹ (see sensor tuning in Figure 3.4), followed by the setup of the RF reflectometry. Subsequent measurements are conducted using RF reflectometry.

The Coulomb oscillations region is delimited prior to performing barrier-barrier narrow-scans to select a suitable Coulomb peak flank² (see sensor tuning in Figure 3.4). This is an iterative step, which consists of repeatedly performing measurements, detecting the Coulomb peak flanks, assessing their quality, and adapting the voltages for the subsequent scan. A Coulomb peak flank is considered to be of high quality if it is as steep as possible with low noise characteristics because this indicates an optimal sensitivity. Depending on the specific application, there may also be a preference for broad Coulomb peak flanks, which necessitates a trade-off with steepness (e.g. to have a wider suitable range for subsequent tuning steps).

¹Coulomb oscillations are voltage-dependent oscillations in conductance through a QD.

²Flank refers to the rising or falling slope of a Coulomb peak in a conductance trace.

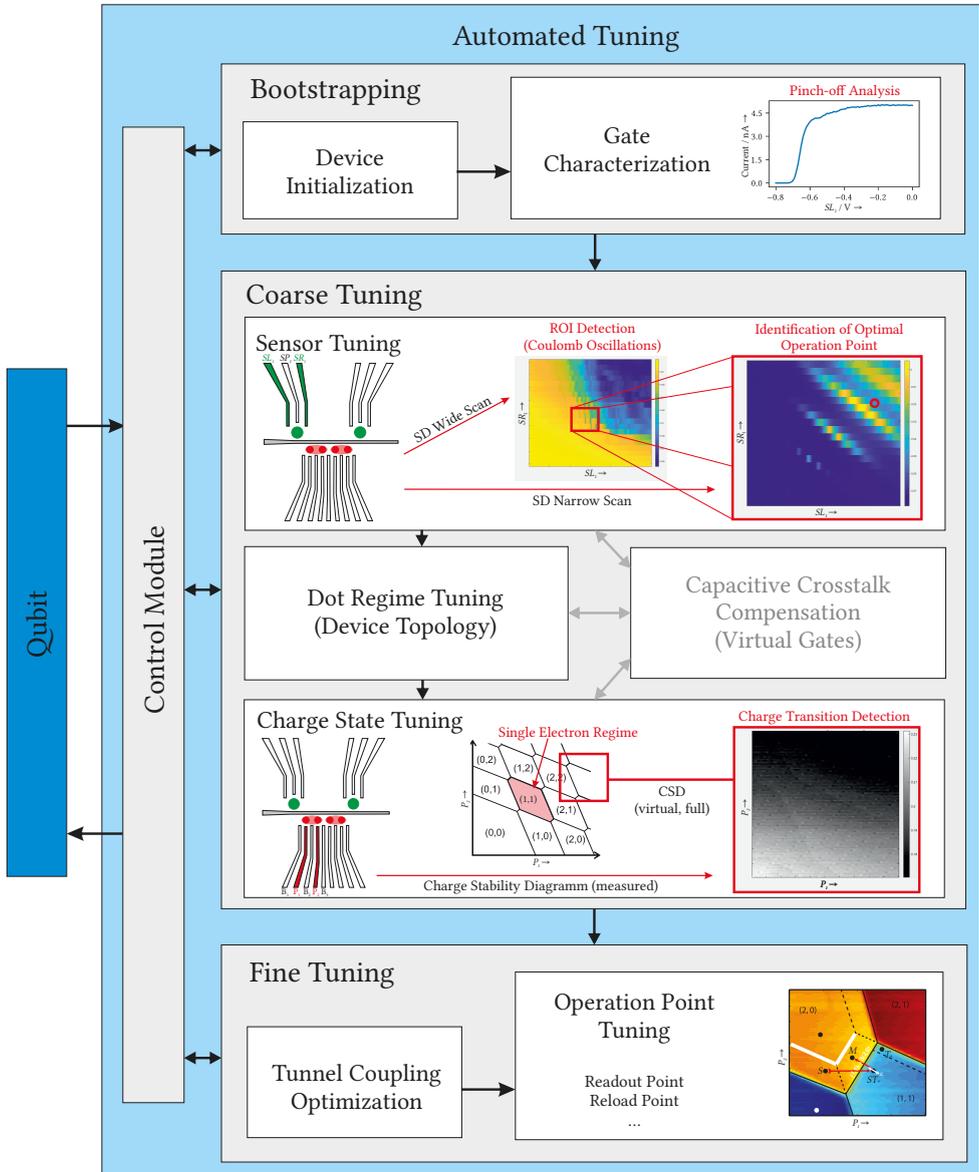


Figure 3.4: Overview of qubit tuning steps.

The tuned SD enables the coarse tuning of the computational QDs. Initially, the first QD of the array in Figure 3.1 is formed. This is accomplished by recording either barrier-barrier scans and/or plunger-barrier scans. In instances where the dot is formed using barrier-barrier scans, the plunger can be calibrated using one-dimensional plunger scans.

Subsequently, the precise number of electrons in the first array dot is adjusted to one, employing either barrier-barrier or plunger-barrier scans and observing CTs. At this stage, a cross-capacitance matrix can be established for the first array dot and the SD, expressed in the form of lever arms that quantify the capacitive coupling between the various gate electrodes and dot potentials. This cross-capacitance matrix facilitates the compensation of undesired effects on other potentials in the device that arise during the application of voltage changes to a gate with the objective of adapting a specific potential. The compensation is achieved using so called "virtual gates", which are a linear combination of multiple physical gates. It is imperative to note that this cross-capacitance matrix undergoes continuous adaptation during the tuning process.

In the present case, the further QDs are added iteratively by first forming the next neighbored dot in the array using the newly added plunger and barrier gates, adjusting to the last electron, and updating the cross-capacitance matrix. Plunger-plunger scans are employed to reach the desired number of electrons in the DQDs. The cross-capacitance matrix is updated in the following manner: For each gate that is either currently in use or will be utilized in the subsequent step, the (relative) lever arm toward the previously used plungers (at least the last 1-2) is measured by sweeping a plunger gate and determining the shift in the CT of the corresponding QD if another gate is stepped.

Finally, when all QDs of the array are in the single electron regime, the fine tuning procedure is initiated [76]. This step entails the characterization and optimization of several DQD properties to prepare reliable qubit operations. Initially, the inter-dot tunnel couplings are adjusted by modifying the barriers between two QDs and the lead-to-dot tunnel couplings by altering the barriers between the leads and their neighboring QDs. Then, different points and transitions have to be located in the DQD voltage space, including:

- the measurement point (subject to spin blockade),
- the singlet reload point (enables to reload a DQD in the singlet state),
- the T_+ reload point (enables to reload a DQD in the T_+ state), and
- the ST_+ transition.

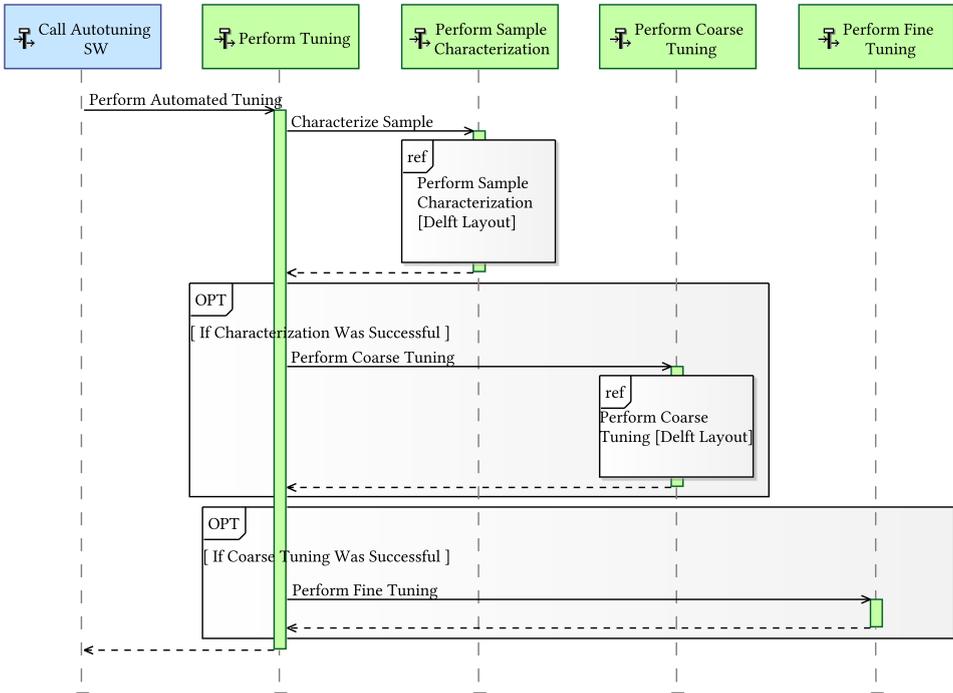


Figure 3.5: Sequence diagram for the highest level automated qubit tuning functionalities.

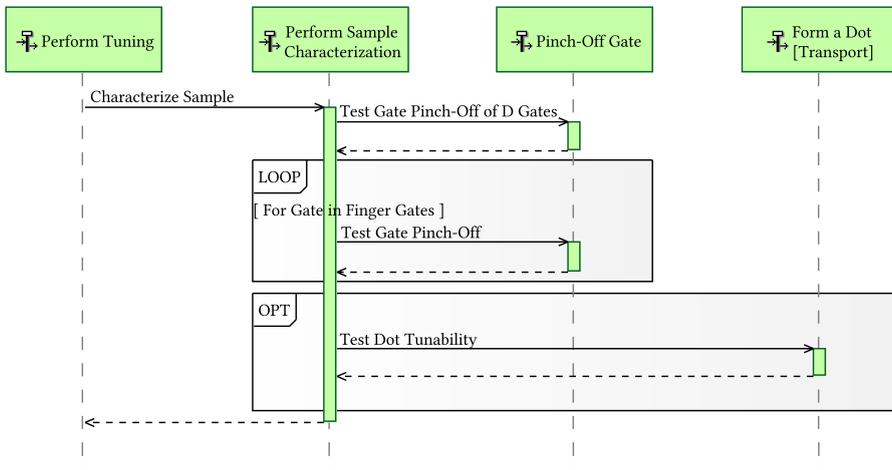


Figure 3.6: Sequence diagram for the sample characterization (referenced in Figure 3.5).

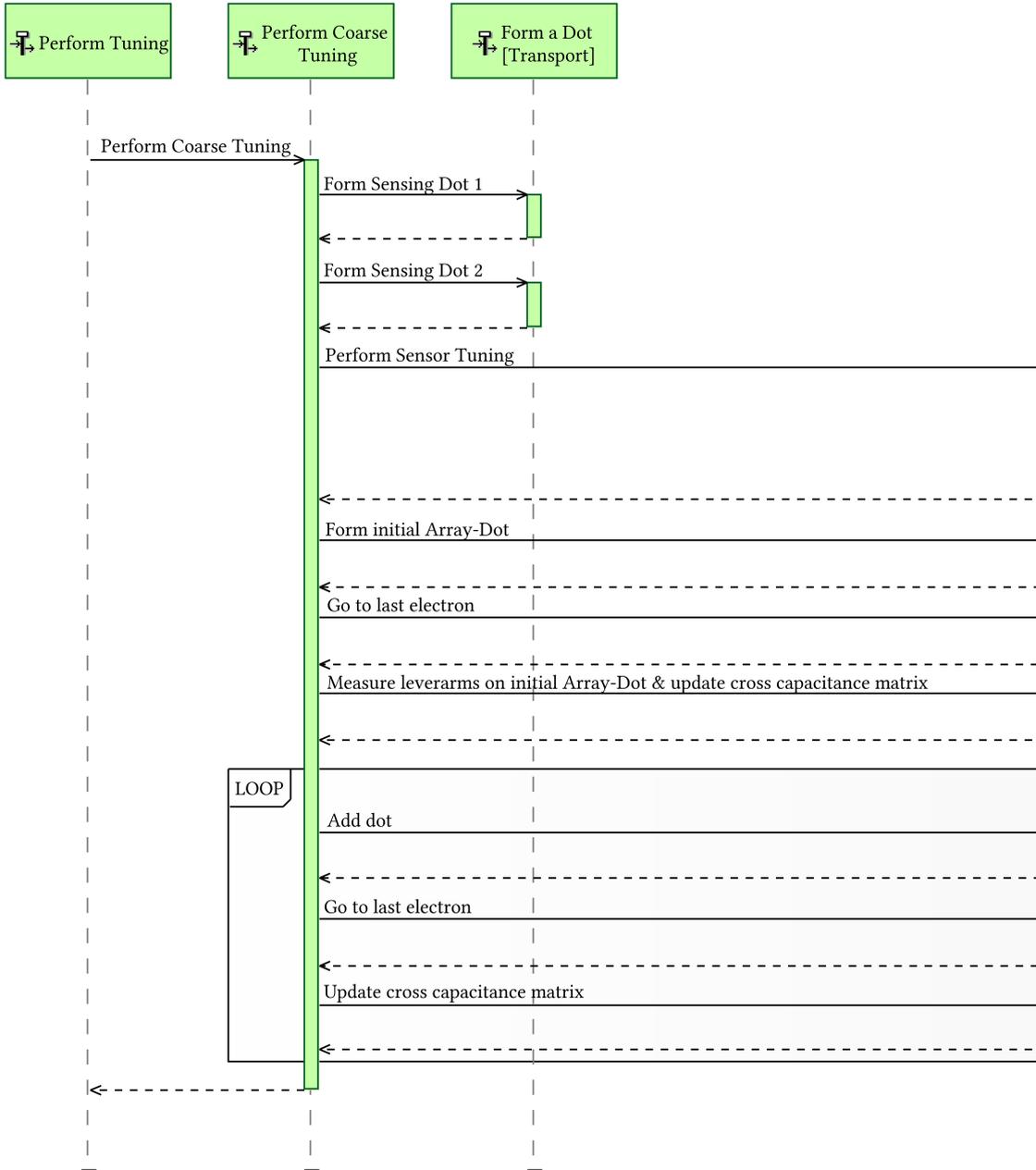


Figure 3.7: Sequence diagram for the coarse tuning, which is referenced in Figure 3.5.

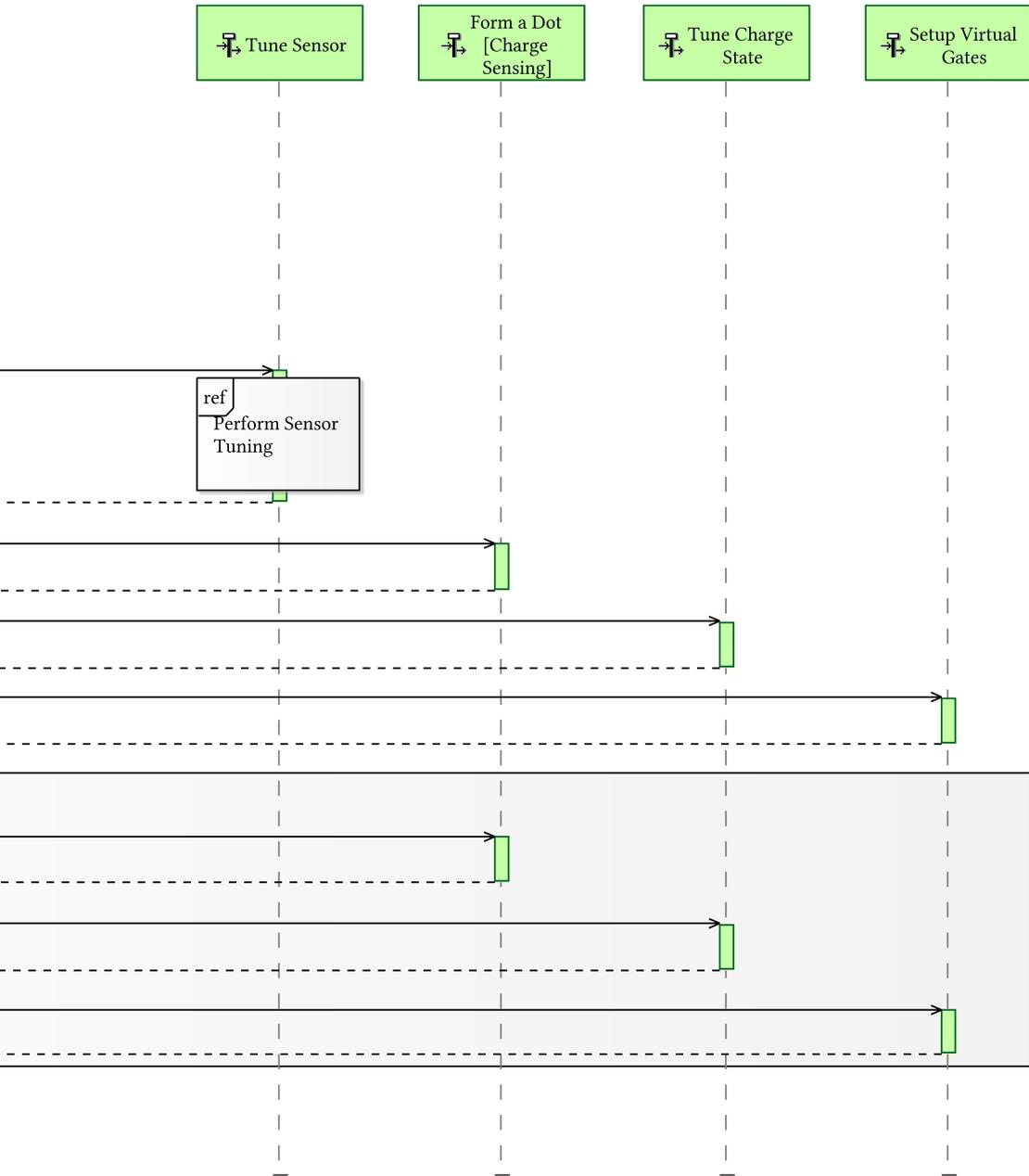


Figure 3.7 (continued)

3 Experimental Setup and Identification of Crucial Tuning Steps

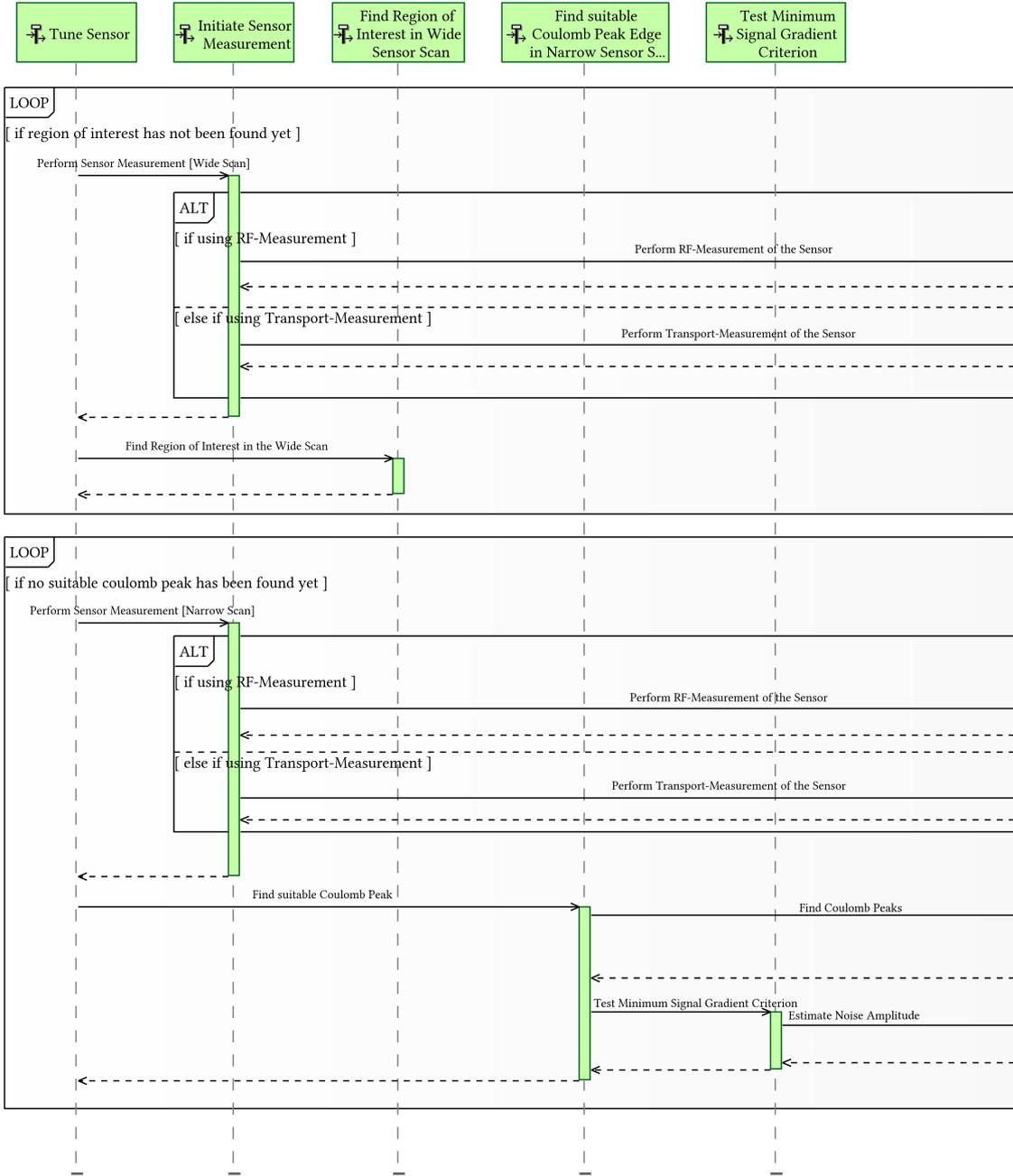


Figure 3.8: Sequence diagram for the sensor tuning, which is referenced in Figure 3.7.

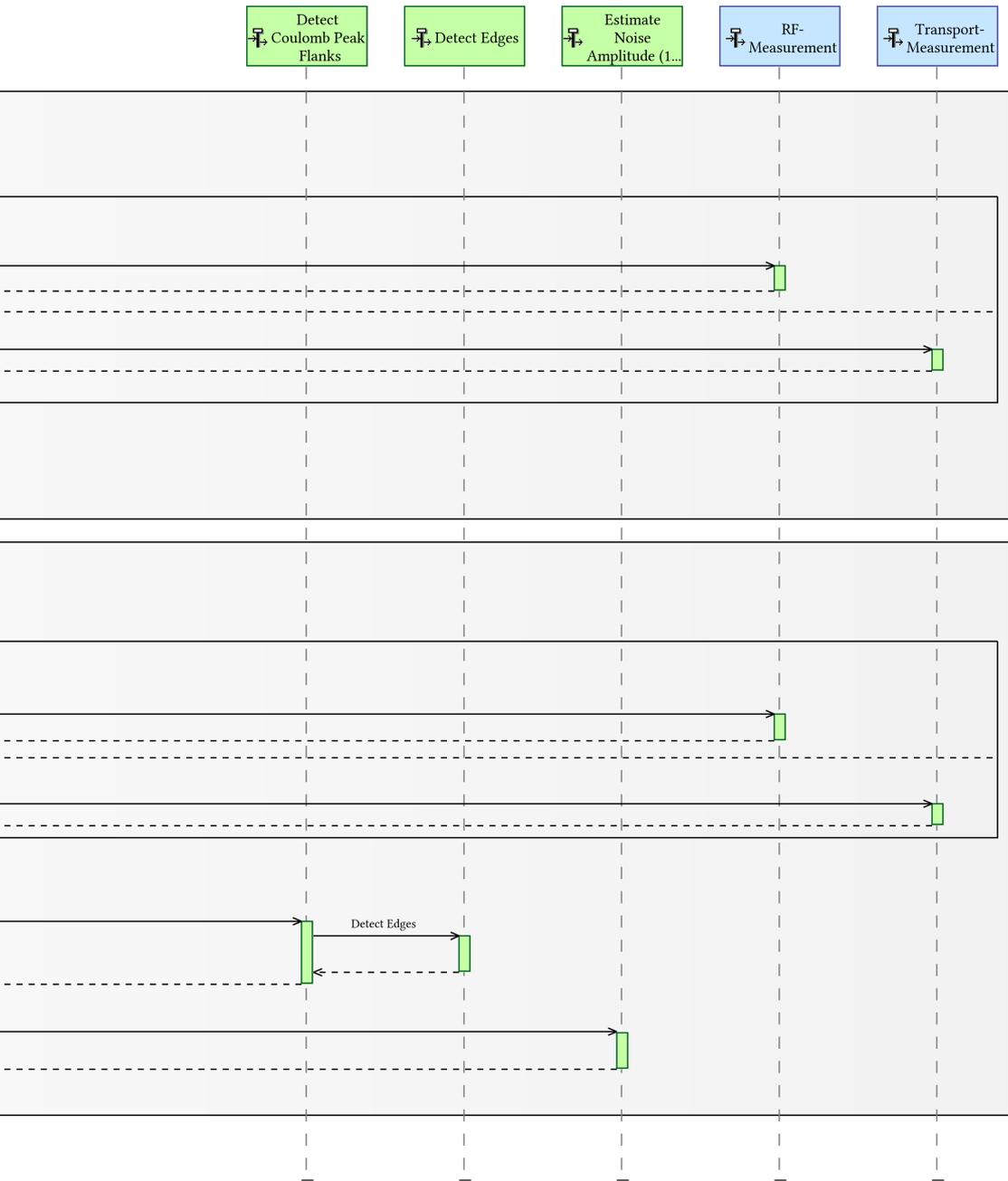


Figure 3.8 (continued)

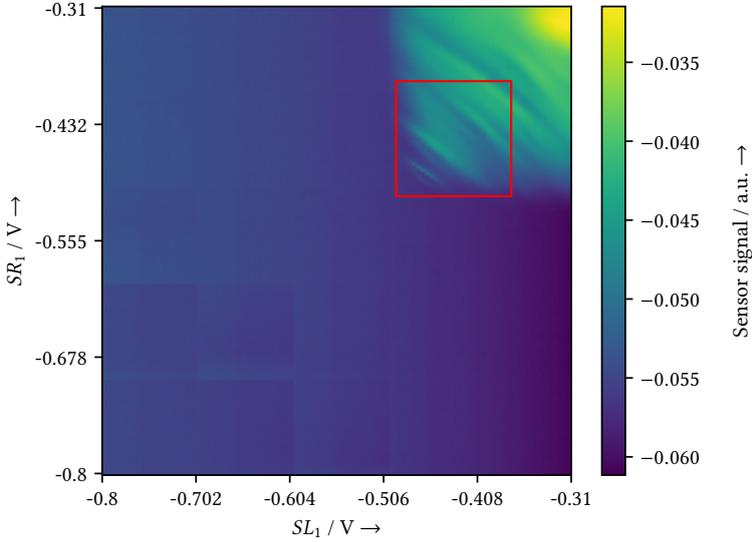


Figure 3.9: Wide scan data³ of a SD with a red box marking the region-of-interest that contains the Coulomb oscillations. In this region a SD has been formed and can be further adjusted.

3.3 Crucial Steps for the Automation of Sensor Dot Tuning

Tuning a SD for charge detection involves several critical steps, as depicted in the previous section, to ensure high sensitivity and reliability. First, the device must be configured to establish a stable Coulomb blockade regime, achieved by adjusting the gate voltages to control the tunnel barriers and dot occupancy. Next, the working point of the SD is fine-tuned near a SD conductance peak.

Precise tuning ensures the optimal performance of the SD for resolving single-electron transitions in nearby QDs. Typically, these tuning steps rely on observations of the conductance characteristics of the SD. Therefore, it is crucial to analyze such data automatically. The selection of the most sensitive operating point considering the signal gradient-to-noise ratio was identified as a particularly important step in this context. This step is essential and an enabling factor for scalable QC because it determines the quality of all subsequent measurements, which in turn influence the necessary complexity of analysis techniques.

The investigation and development of suitable approaches require large amounts of labeled data. The availability of experimentally measured data is limited, and measurement and la-

³Measurement provided by Simon Humpohl and Tobias Hangleiter from the Quantum Technology Group of RWTH Aachen.

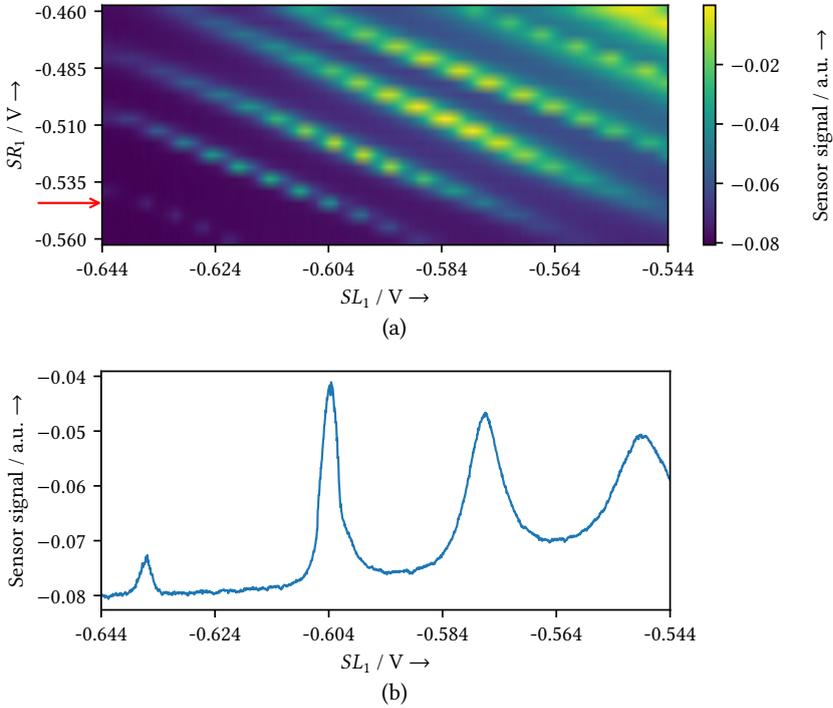


Figure 3.10: Example narrow scan³ from SD measurements. (a) SD narrow scan data. (b) Row data of the narrow scan in (a) with $SR_1 = -0.545$ V (marked by the red arrow).

being are time-intensive. In addition, the limitations in the control over distortions hinder the comprehensive investigation of the requirements and constraints of different approaches. Consequently, a high-quality simulation is eligible, enabling the production of a sufficient amount of labeled data with strict control over the different parameters.

3.4 Characteristical Scans for the Selection of Suitable Coulomb Peaks

The conductance $G_{sd}(SL, SR)$ of a SD, represented in a SD scan, is predominantly governed by the two barrier gate voltages, designated as SL_i and SR_i , as illustrated in the gate layout of Figure 3.1. The exemplary scan depicted in Figure 3.9 encompasses a wide voltage range, with the objective of identifying the region exhibiting Coulomb oscillations, as delineated by the red box. A fine or narrow scan in this region (illustrated in Figure 3.10a) sufficiently resolves the oscillations to determine an optimal SD working point.

The experimental setup and the time reduction of the measurement result in a stepwise change in the voltage on one axis. Concurrently, the other voltage continuously ramps up within a defined voltage interval, and an integration process generates the measurement data. In the discussed experiments, a narrow scan typically delivers 10 or 20 rows of data each consisting of 1,664 data points. This distinct anisotropic two-dimensional data resolution necessitates row-wise data analysis to avoid artifacts. [Figure 3.10b](#) shows a one-dimensional data example of a single row.

3.5 Crucial Steps for the Automation of Spin Qubit Tuning

As delineated in [Section 3.2](#), the process of tuning a [DQD](#) to function as a singlet-triplet spin qubit encompasses multiple stages. Establishing a well-defined charge configuration is essential here, as it provides the necessary starting point for subsequent tuning procedures. This objective is typically accomplished by means of observing [CTs](#) through the use of two-dimensional [CSDs](#).

Due to the varying and often mediocre data quality, the analysis is considered a complex task. Additionally, the strengths of distortions cannot be controlled, making it impossible to comprehensively investigate the requirements and limits of different analysis approaches. Therefore, an appropriate [CSD](#) simulation is imperative for enabling the development and benchmarking of suitable algorithms.

Additionally, the quantity of measurements and the transfer of the full [CSDs](#) to the room-temperature domain result in scalability challenges, characterized by limited bandwidth and inadequate heat dissipation. At this stage, the essential information is the occurrence of the [CTs](#) as a binary value. Consequently, the efficient detection of this information in the cryostat is considered a pivotal factor in reducing the volume of data to be transmitted.

The integration into the cryostat necessitates the minimization of the complexity of dedicated [CT](#) detection approaches, given limitations such as spatial constraints and dissipated heat.

3.6 Charge Stability Diagrams

[CSDs](#) are graphical representations that map the stable charge configurations of [QDs](#) as functions of gate voltages. These diagrams are instrumental in the characterization of [QDs](#) and the determination of their charge occupancy under varying electrostatic conditions. These insights are essential for understanding the formation and tunability of [QDs](#), which is crucial for the utilization of [QDs](#) as spin-based qubits [39].

To measure a [CSD](#), the device is typically tuned into a regime where discrete charge states emerge due to strong Coulomb interactions. The measurement of the [CSD](#) involves the sweeping of two gate voltages (typically two plunger gates, e.g. P_1 and P_2 in [Figure 3.1](#), which con-

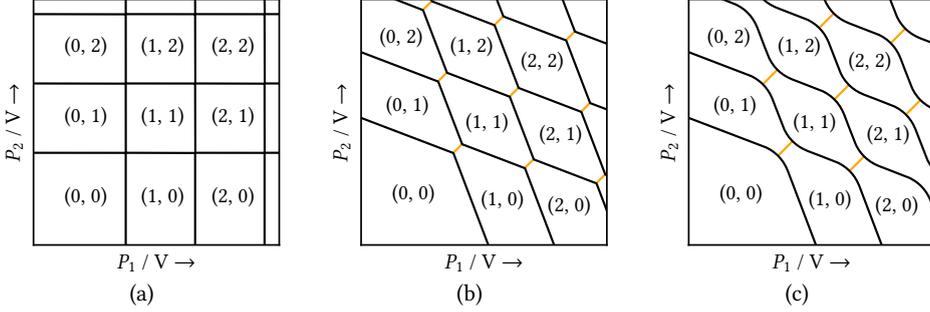


Figure 3.11: Visualization of the characteristic structure of CSDs, adapted from the work of Hanson et al. [31]. Black lines indicate LDTs and orange lines IDTs. Electron occupations of the two QDs are labeled (N_1, N_2) . The structure in (a) is not affected by any coupling effects, while (b) takes cross capacitance effects between P_1 and QD 2 as well as P_2 and QD 1 into account. In (c), the effect of the tunnel coupling is included, which leads to a rounding at the triple points.

trol the electrochemical potentials of adjacent QDs), while monitoring the conductance of a nearby SD functioning as a charge sensor (see Section 2.2.2).

The structure of CSDs is illustrated in Figure 3.11. Assuming that the electrochemical potential of the two QDs were entirely independent of each other, the lines corresponding to LDTs would exhibit a precise linear configuration, as depicted in Figure 3.11a.

However, due to existence of a capacitive coupling between the dots, the electron occupation of one dot has a direct effect on the electrostatic energy of the other dot. This phenomenon gives rise to inclined LDTs and a splitting of the cross point into two triple points. As a result, the characteristic honeycomb pattern in Figure 3.11b emerges. The effect described herein is further propelled by the direct coupling of P_1 to QD 2 and vice versa of P_2 to QD 1. Concurrently, the inderdot capacitance governs the spatial separation of the triple points within the CSD.

In the presence of a low source-drain bias, the electron transport through the DQD is constrained to triple points, which energetically allow tunneling through all adjacent CTs. Therefore, measuring the current through the DQD is not suitable to resolve the whole structure.

In contrast, noninvasive charge sensing techniques, which are characterized by their exceptional sensitivity, allow to identify any alteration in electron occupation. This property enables the revelation of the complete set of CT lines, encompassing both LDTs and IDTs, which are depicted in Figure 3.11b as black and oranges lines respectively.

The CSD structure in Figure 3.11b is consistent with the characteristics observed in the low tunnel coupling regime. With increased tunnel coupling, the electrons are no longer fully localized in a single QD; instead, they occupy molecular-like orbitals that span both dots [39]. This phenomenon leads to rounding of the LDTs at the triple points, as illustrated in Figure 3.11c.

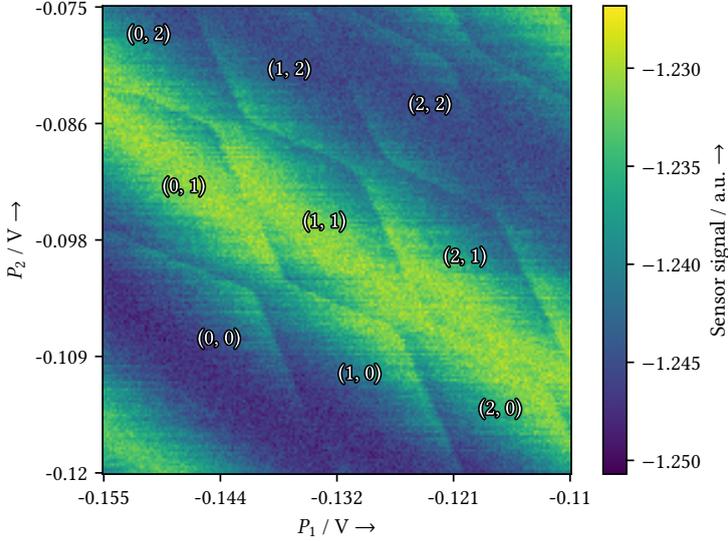


Figure 3.12: Example of a CSD over a large voltage range with the characteristic honeycomb-like pattern. Electron occupations of the two QDs are labeled (N_1, N_2) .

Figure 3.12 exemplarily illustrates the appearance of a CSD using a SD. The previously described characteristic structure becomes visible as a jump in the conductance of the SD, caused by the capacitive coupling of the SD to the DQD. In the lower left corner, the DQD is completely depleted. The identification of this region enables the retrieval of the charge configuration for other regions through the counting of CTs.

3.7 Conclusion

CSDs play an important role in the qubit initialization and manipulation, providing a visual framework for optimizing tunnel couplings, charge configurations, and valley splitting in QDs. The precise interpretation of CSDs is essential for ensuring the reproducibility and scalability of qubit devices in large-scale quantum processors. Due to this fact and with respect to the higher complexity of the detection, this step is identified as most crucial task and therefore prioritized in the following work.

Chapter 4

Simulation of Charge Stability Diagrams for Automated Tuning Solutions (**SimCATS**)

DEVELOPMENT of tuning algorithms based on **ML** and classical methods, quality assessment of a solution, and fair comparison of different approaches benefit from publicly available datasets. Simulations can generate the necessary number of datasets along with the corresponding ground truth data for these tasks.

Simulating the transport of semiconductor **QD** systems on classical computers is a demanding task that incorporates phenomena such as Coulomb blockade [238–240], Pauli spin blockade [241], or **SD** response [60]. Specific models ease the calculation of **CSDs** [242] that are fundamental for spin-based quantum computation regarding **qubit** manipulation and information readout.

The constant interaction model (**CIM**) [31, 39, 61, 243] describes the electronic states of **QDs** and parametrizes the onsite and intersite Coulomb interactions as a network of capacitors, resistors, and leaky capacitors¹ between dots, gates, and leads. While this model satisfactorily explains many aspects of experiments, it is important to note that it does not incorporate quantum effects that deform the modeled **CSDs**, sometimes substantially.

Theories capable of considering both classical and quantum effects are necessary to understand the quantum aspects of **CSDs** and improve their usefulness. The Thomas-Fermi capacitive model [81] employs the Thomas-Fermi approximation [244] to calculate the electron density of the islands and derive an estimate of inverse capacitive elements for a capacitance model in a given potential profile. It also models the electron transport using a Markov chain among charge states incorporating single electron tunneling rates between islands or to contacts calculated under the Wentzel-Kramers-Brillouin tunnel probability [245].

The quantum-mechanical two-level model [246–250] derives the tunneling of a single electron between two dots from the probability crossover of the two eigenstates, and Zhang et al. [251] investigated its influence on the **CSD**.

Gaudreau et al. [252] and Korkusinski et al. [253] fundamentally demonstrated the capability of Hubbard models (**HMs**) to describe **CSDs** for triple-**QD** systems. Subsequently, Yang et al. [254], Das Sarma et al. [255], and Wang et al. [256] proposed a generalized **HM** as the quantum generalization of the classical **CIM**, incorporating quantum effects such as spin exchange,

¹A leaky capacitor is modeled as an ideal capacitor in parallel with a resistor.

pair-hopping, and occupation-modulated hopping. However, the generalized **HM** focuses on the electronic interaction within the **QD** system itself while disregarding environmental factors contributing to the decoherence including the hyperfine interaction with the nuclear spin bath [257, 258], the coupling to background impurities [259, 260], and the phonon modes [261, 262].

The models frequently find their application in **ML** approaches for automated measurements [95, 263, 264], in tuning of **QDs** devices [81, 265, 266] and in providing reference datasets for tuning and testing purposes [86, 267]. Additionally, a group of simulators focuses on the simulation and design of semiconductor-based information devices including the generation of **CSDs** [268–272]. This group of simulators employs a range of methods, including the Schrödinger equation, the nonlinear Poisson equation, master equations (Pauli, Redfield, Lindblad), first- and second-order von Neumann approaches, the real-time diagrammatic approach, the configuration interaction method, and many-body solvers.

Finally, some simulators focus exclusively on the **CSD** simulation. The **SIMON** simulator [273] is a notable example, as it utilizes the **CIM** and the Monte Carlo method for the master equation to implement tunnel junctions in the context of single-electron tunnel devices and circuits. Although the Python-based **QTT** framework [274] was designed for the tuning and calibration of **QDs** and spin **qubits**, it offers simulation functions for the calculation of **CSDs** based on the **CIM**. The system parameters are used to establish the Hamiltonian, calculate the eigenvalues, determine the occupation numbers, and derive the **CSD**. In addition to the **CIM**-based simulation, the **QuDiPy** project [275] implements the **HM** of Das Sarma et al. [255] to generate simulated **CSDs**.

While physical models have enabled the acquisition of valuable insights, they are encumbered by substantial limitations when employed in the context of algorithm development. The computational demands of these models impede the rapid generation of datasets, and their reliance on fully-understood physics limits their capacity to reproduce all the features observed in experiments. Moreover, the necessity of precise device-specific parameters – often difficult or impractical to obtain – limits their usability. These challenges underscore the necessity for alternative simulation approaches that prioritize empirical realism, versatility, and efficiency over exhaustive physical modeling, particularly to support scalable, data-driven methods.

In this work, a generic framework for simulating **CSDs** is proposed, integrating the necessary functionalities to mimic experimental data. In contrast to the aforementioned approaches, this framework employs a geometric model for the versatile simulation of the ideal **CSD**² that does not require knowledge of the physical device parameters. This facilitates the efficient reconstruction of measurement data solely based on parameters describing the geometry observed in previously recorded data.

²In this context, ideal refers to simulated undisturbed (ground truth) data.

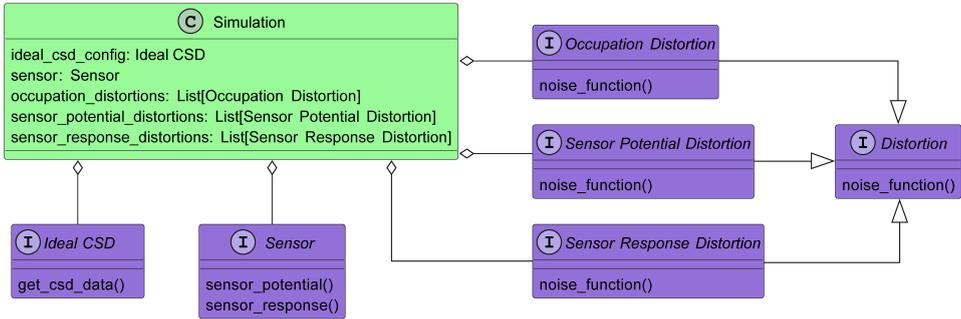


Figure 4.1: Interfaces of the simulation class of the simulation framework `SimCATS`. The Python package includes standard implementations.

4.1 Simulation Model

The simulation model integrates the simulation of the occupation probabilities (Section 4.1.1), the sensor response (Section 4.1.2), and multiple types of distortions (Section 4.1.3) into a unified framework, thereby providing a comprehensive simulation of CSDs. The conceptualization of the framework is designed to emulate the experimental setup to the greatest extent possible. Specifically, the design prioritizes integrating flexibility to support diverse dot regime and charge state tuning experiments. Consequently, the framework enables users

- to perform two- and one-dimensional measurements,
- to measure in different directions with consideration for the time-dependent nature of specific distortion types,
- to switch to diverse sensor configurations, such as for multi-sensor samples, and
- to deactivate distortions selectively.

All components of the simulation are interchangeable and defined through straightforward interfaces, as illustrated in Figure 4.1.

4.1.1 Dot Occupation Model

The electron occupation of the dots constitutes the fundamental basis for the CSD simulation. The presented occupation model is purely geometric and, in contrast to available physical models, provides the flexibility and simplicity to support the different honeycomb shapes observed in DQD measurements. The fundamental idea is to describe a CSD as a series of total charge transitions (TCTs), representing the borders between regions containing a fixed number of electrons in the system. As illustrated in Figure 4.2, this representation involves the separation of regions containing $i - 1$ and i electrons by the total charge transition tct_i , $i = 1, \dots, n$.

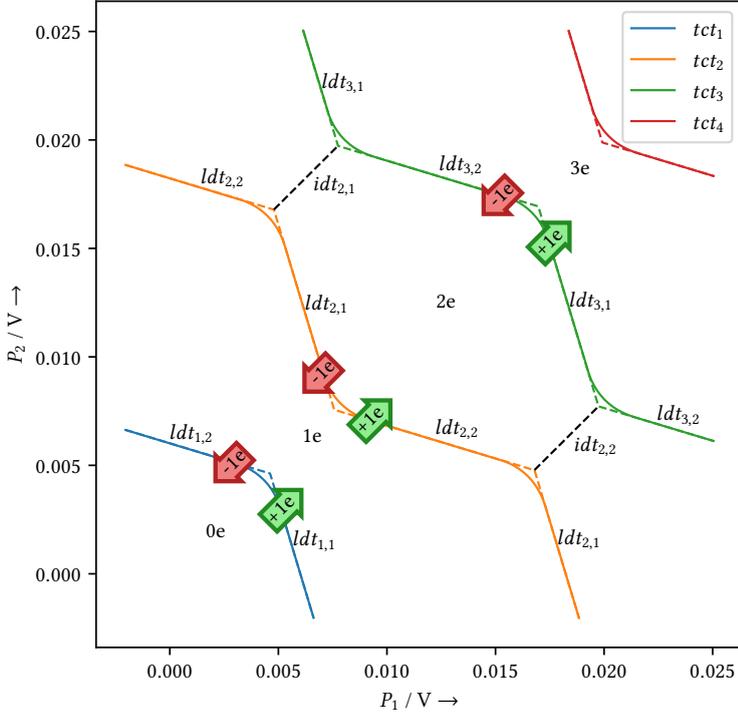


Figure 4.2: Representation of the separation between regions with fixed numbers of electrons using TCTs. The solid lines represent TCTs affected by inter-dot tunnel coupling, while the dashed lines indicate the LDTs without tunnel coupling.

The orientation of the LDTs in the two-dimensional voltage space is contingent on the capacitive coupling of several gates. Nevertheless, the lead-to-dot transition $ldt_{i,j}$ associated with tct_i is predominantly influenced by the plunger gate P_j of quantum dot qd_j or, in the case of employing virtual gates, it is exclusively affected by the virtual plunger gate P_j^v . For the representation in Figure 4.2, the slope of $ldt_{i,1}$ is in the interval $[-\infty, -1)$ ($-\infty$ for virtual gates) and the slope of $ldt_{i,2}$ is in the interval $(-1, 0]$ (0 for virtual gates). To ensure definiteness of the slopes in the mathematical sense, a parametric representation of the TCTs in a voltage space (V'_{P_1}, V'_{P_2}) is proposed. This space originates from (V_{P_1}, V_{P_2}) through an affine transformation corresponding to a 45° rotation. This results in slope intervals of $[-1, 0]$ for $ldt_{i,1}$ and $(0, 1]$ for $ldt_{i,2}$. Depending on the inter-dot tunnel coupling, the TCT exhibits curves at triple points, where $ldt_{i,1}$, $ldt_{i,2}$, and an IDT intersect. Consequently, the composition of a TCT consisting of linear segments and Bézier curves, visualized in Figure 4.3, ensures twice continuous differentiability.

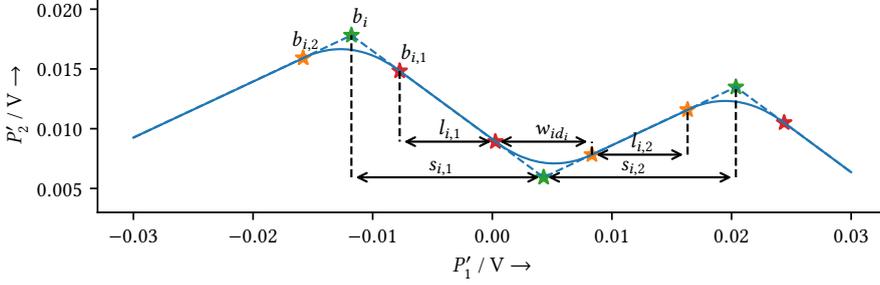


Figure 4.3: One-dimensional TCT description (rotated). The blue dashed line indicates the TCT devoid of tunnel coupling, while the blue solid line signifies the TCT with tunnel coupling.

The subsequent parameters delineate tct_i :

1. $s_{i,j}$: V'_{P_1} -intercept of $ldt_{i,j}$, ($j = 1, 2$),
2. $m_{i,j}$: slope of $ldt_{i,j}$, ($j = 1, 2$),
3. $b_{i,2}$: Bézier anchor on $ldt_{i,2}$ that marks the starting point of the curved segment of the first triple point of tct_i ,
4. $b_{i,1}$: Bézier anchor on $ldt_{i,1}$ that marks the end point of the curved segment of the first triple point of tct_i .

Utilizing this set of parameters, enables the complete construction of a TCT through the repetition of linear segments, the Bézier curve, and its 180° rotation. Moreover, the intersection of the two LDTs constitutes the required center Bézier anchor b_i :

$$P'_{1b_i} = \frac{P'_{2b_{i,2}} - P'_{2b_{i,1}} + P'_{1b_{i,1}} \cdot m_{i,1} - P'_{1b_{i,2}} \cdot m_{i,2}}{m_{i,1} - m_{i,2}} \quad (4.1)$$

$$P'_{2b_i} = P'_{2b_{i,2}} + m_{i,2} \cdot (P'_{1b_i} - P'_{1b_{i,2}}). \quad (4.2)$$

The V'_{P_1} -intercept of only the linear part is

$$l_{i,j} = s_{i,j} - 2 \cdot |(P'_{1b_{i,j}} - P'_{1b_i})|. \quad (4.3)$$

Depending on the identifier i of the TCT, the number of Bézier curves and triple points is limited to

$$n_t = 2 \cdot i - 1. \quad (4.4)$$

Existing TCTs allow the calculation of the electron occupation. In the area between tct_i and tct_{i+1} , there is a total of i electrons in the system. Their distribution to the two QDs qd_1 and qd_2 is determined as follows:

1. The connecting vector from the triple point of tct_i to the opposite triple point of tct_{i+1} represents the IDT $idt_{i,k}$, $k \in \{1, \dots, i\}$.
2. Across each $idt_{i,k}$, a sigmoid function orthogonal to it approximates the Fermi distribution.
3. The superposition³ of all sigmoid functions represents the electron occupation of qd_1 .
4. The difference between the total number of charges i and the occupation of qd_1 results in the occupation of qd_2 .

4.1.2 Sensor Model

The sensor response or sensor signal is calculated at each point in the CSD using the simulated occupation information and the sensor characteristic distinguished by the selected Coulomb peak.

Besides the required capacitive coupling of the SD to the DQD, the SD also cross-couples to the plunger gates of the DQD. In the context of CSDs, the former enables the observation of electron occupation changes as edges, while the latter manifests as undesired value shifts within the honeycombs⁴. It is imperative that the simulation incorporates both of these components.

The following model is proposed for the sensor response S [276]:

$$\mu_{SD} = \sum_{j=1}^2 [\alpha_j \cdot N_j + \beta_j \cdot V_{P_j}] + \mu_{SD,0} \quad (4.5)$$

$$S(\mu_{SD}) = S_{off} + a \cdot \frac{\gamma^2}{\gamma^2 + (\mu_{SD} - \mu_0)^2},$$

where j is the index of the corresponding plunger gate. In this model, the electrochemical potential of the SD, represented by μ_{SD} , is influenced by the number of electrons N_j in the dots and the voltages V_{P_j} applied to the plunger gates together with their corresponding lever arms α and β . Effectively, α influences the sharpness of the edges and β the drifts within the honeycombs. It is important to note that both effects are counteractive: the former is negative, while the latter is positive. Furthermore, the initial potential $\mu_{SD,0}$ of the SD contributes to the overall potential.

A simplified Lorentzian approximation is used to model $S(\mu_{SD})$ [243] (see Figure 4.4a). The parameter γ defines the width of the Lorentzian, and μ_0 specifies the potential at the peak. As

³Here, superposition denotes mathematical function combination.

⁴The undesired coupling between the DQD plunger gates and the SD can be compensated by using virtual gates.

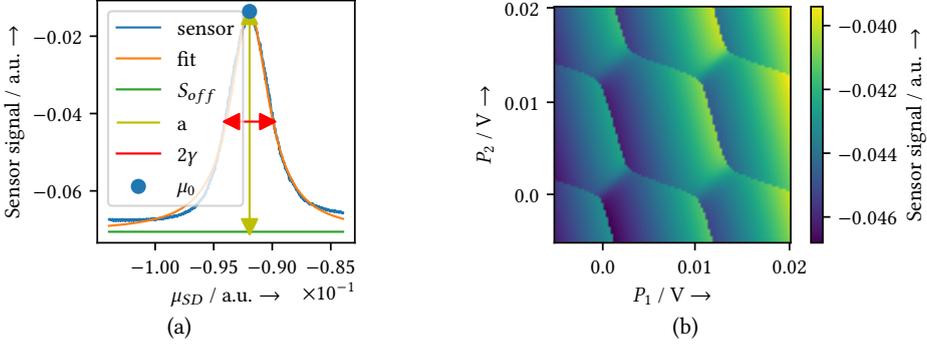


Figure 4.4: Simulation of the sensor response. (a) The measured sensor response is shown in blue, with a Lorentzian fit shown in orange. (b) An example of the sensor response simulated on the left flank of the peak is shown. This results in rising value shifts with rising voltages in the CSD.

linear filters transform the sensor response in the experimental setup, a scaling factor a and an offset S_{off} parametrize the Lorentzian. Figure 4.4b shows an example of a simulated CSD that includes the cross-coupling effects.

4.1.3 Distortions Model

The simulation of realistic CSDs requires the consideration of occurring distortions [265, 267, 276]. In the ensuing discussion, the identified distortion stages are defined, the relevant distortion types are assigned to these, and their sources, simulation, and the required parameters as they typically appear in measurement setups similar to the one used in this work (see Section 3.1) are described. It is further assumed that the samples and their layout are sufficiently reliable for scalability, i.e., the presence of a spurious QD under the barrier gates or intensively moving QDs are negligible. This is a prerequisite for the construction of functional quantum computers. Under the assumption that the measurement is performed at a slow enough rate, effects such as latching can be neglected here.

4.1.3.1 Distortion Categories

The following is proposed: the CSD distortions should be assigned to three categories based on their impact point in the signal path, due to the nonlinear dependencies observed in the measurements:

- cat. 1: Occupation distortions,
- cat. 2: SD potential distortions, and
- cat. 3: SD response distortions.

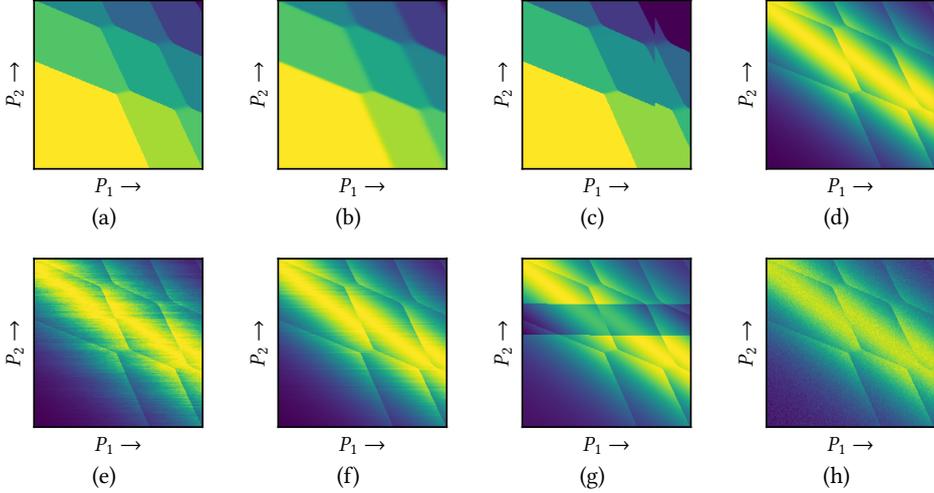


Figure 4.5: Examples of the simulated sensor response and distortions. Distortions affecting the sensor (cat. 2 and 3) are visualized in combination with a sensor response. (a) Ideal CSD, (b) ideal CSD with occupation transition blurring (cat. 1), (c) ideal CSD with dot jumps (cat. 1), (d) ideal CSD with sensor response, (e) ideal CSD with sensor response and pink noise (cat. 2), (f) ideal CSD with sensor response and RTN (cat. 2), (g) ideal CSD with sensor response and RTN (cat. 3), and (h) ideal CSD with sensor response and white noise (cat. 3).

Distortions of the first category alter the simulated occupations of the DQD. The category includes dot jumps and temperature broadening, the latter leading to occupation transition blurring.

The category of SD potential distortions includes both the random telegraph noise (RTN) and pink noise. It is imperative to differentiate these from undesired effects on the nonlinear SD response, primarily white noise and RTN.

4.1.3.2 Dot Jumps

The occurrence of dot jumps is attributable to deterministic charge-trapping effects on the QDs, arising from fabrication-related imperfections. These imperfections manifest as displacements within the occupation structure of a CSD [267]. These jumps are simulated by shifting a block of columns horizontally, as illustrated in Figure 4.5c, or a block of rows vertically. The occurrence of these jumps is simulated by a geometric distribution of the jump extension, while a Poisson distribution of the jump amplitude determines their intensity [267].

4.1.3.3 Occupation Transition Blurring

The thermal occupation of states in the reservoir results in a broadening of the [LDTs](#), which follows the Fermi-Dirac distribution under the assumption that the density of states is constant [277]. This effect is simulated by applying a one-dimensional Fermi-Dirac filter kernel along the measurement direction⁵.

4.1.3.4 Pink Noise

Pink noise, also referred to as $1/f$ or flicker noise, has been observed in a wide range of electronic devices and is attributed to the inherent heterogeneity of electronic components, such as oxide traps or lattice dislocations [278]. Its [PSD](#) is inversely proportional to the frequency and manifests as stripes in line-wise measured [CSDs](#).

The algorithmic generation of pink noise is described by Timmer and König [279] and implemented in the Python module `colorednoise` [280]. In the simulation, pink noise is applied to the sensor potential, increasing its visibility in regions with high gradients due to the nonlinear sensor response. This phenomenon aligns with the observations reported in experimental data [79].

4.1.3.5 Random Telegraph Noise

As stated in the literature, [RTN](#) is characterized by its random switching between two or multiple discrete levels [278], a phenomenon also referred to as burst noise. This dynamic behavior arises from time-dependent random capture or emission processes of charge carriers, induced by oxide traps [281]. The [PSD](#) of this noise is proportional to $1/f^2$, at frequencies that are significantly larger than the corner frequency. In line-wise measured [CSDs](#), [RTN](#) manifests as stripes with a well-defined temporal start and end (see [Figure 4.5f](#)).

The occurrence of bursts is simulated using a geometric distribution for their extension and a normal distribution for their amplitude [267].

Jumps predominantly manifest as noise on the sensor potential, but also occur in the sensor response. Consequently, it is proposed to consider the additional inclusion of [RTN](#) in distortion category 3.

4.1.3.6 White Noise

In the system under consideration, white noise, characterized by a constant [PSD](#), originates from thermal and shot noise. Thermal noise is attributed to the thermal agitation of charge carriers in an electrical conductor [282], while shot noise is traced back to discrete charges in the current flow and does not relate to a operating temperature of the system [283].

In measured data, the amplitude distribution of thermal noise is nearly Gaussian, while that

⁵For the presented results the Fermi-Dirac filter was still approximated by a Gaussian filter kernel.

of shot noise is Poissonian. However, given the closeness of the normal distribution to the Poisson distribution, the simulation combines both noise types into a single Gaussian distribution with standard deviation σ_w . Additionally, it is assumed that they solely accrue after the sensor, as their dominant components result from the amplification of the sensor signal.

4.2 Parameter Extraction

The parameters for simulating CSDs are exemplarily extracted from the GaAs sample⁶, which is described in Section 3.1. The extraction approach is independent of the specific sample employed.

4.2.1 Extraction of Occupation Data Parameters

The extraction of parameters describing the structure of CSDs can be achieved through the utilization of previously recorded measurements.

Each TCT under consideration requires the parameters delineated in Section 4.1.1. The extraction process is facilitated by manual labeling of the TCTs. It is noteworthy that the preservation of parameters for all TCTs is not a prerequisite. Instead, the establishment of a transformation rule that generates the subsequent TCT from a preceding one is feasible. This transformation rule can be implemented through shifting of the TCT and adjusting of w_{id} based on observed relations.

4.2.2 Extraction of Sensor Parameters

Measured sensor scans used for the extraction of parameters describe the sensor response as a function of the voltage applied to the plunger gate of the corresponding SD. Furthermore, the proportionality of the electrochemical sensor potential to the SD plunger gate is employed in the subsequent analysis.

Fitting the Lorentzian to the experimental sensor scan ascertains the parameters S_{off} , a , γ , and μ_0 in the sensor model (Section 4.1.2). The subsequent analysis, focused on the determination of the parameters α_j and β_j , is constrained to regions within the CSD that exhibit an overall rising value $S(V_{P_1}, V_{P_2})$, corresponding to the left side in the fit $S(\mu_{SD})$. This restriction is attributed to the irreversible uniqueness of the Lorentzian. Subsequently, the inverse fit function, denoted by $\mu_{SD}(S)$, is employed to estimate the electrochemical sensor potential $\mu_{SD}(V_{P_1}, V_{P_2})$. Within the confines of the honeycomb regions,

$$\beta_j = \left(\frac{\partial}{\partial V_{P_j}} \mu_{SD} \right)_{N_1, N_2 = const} \quad (4.6)$$

is determined.

⁶The determined parameters are provided with the SimCATS package (`default_configs["GaAs_v1"]`) [284].

Conclusively, within the respective **LDT** areas of the two dots,

$$\alpha_i = \left(\frac{\Delta(\mu_{SD}(V_{P_i}) - \beta_i \cdot V_{P_i})}{\Delta N_j} \right)_{N_{j=i} \neq \text{const}, N_{j \neq i} = \text{const}} \quad (4.7)$$

is calculated.

4.2.3 Extraction of Distortion Parameters

The distortion parameters are determined from a variety of scans. For white noise and **RTN** of category 3 measured **CSDs** are used; for pink noise and **RTN** of category 2, sensor scans are used. However, dot jumps are characterized manually in measured **CSDs**.

4.2.3.1 Dot Jumps

Automatic detection of dot jumps remains unattainable at this juncture. Consequently, the amplitude and extension of the jumps are manually extracted. However, given the absence of a discernible jump back in the **CSDs**, the intensity of the jumps remains the sole variable that can be ascertained. Therefore, the extension is presumed to exceed the measured voltage space of the experimental **CSDs**. Furthermore, the parameters for the two swept gates are extracted independently, as their values are subject to variation depending on the specific sample.

4.2.3.2 Pink Noise

The intensity of pink noise in the sensor potential can be determined using the **PSD**. To this end, two-dimensional sensor scans with a high resolution on the abscissa and a low resolution on the ordinate are examined [79].

However, it is first necessary to estimate the sensor potential. Therefore, for each measured row, a sum of Lorentzians is fitted. If successful⁷ for each measured gate voltage, the corresponding sensor potential is determined by computing the inverted Lorentzian of the measurement value.

The **PSD** of the estimated potential can then be determined for each row and averaged across different rows to obtain a more accurate approximation. The fit from Section 4.2.3.4 is employed to calculate the intensities of the white and pink noise components, as the calculated electrochemical potential incorporates the white noise from the sensor response. However, it should be noted that the white noise parameters utilized for the simulation were extracted directly from the sensor response.

⁷Evaluated manually, cf. [79].

Assessments of the estimation method with simulated data reveal that it generally overestimates the noise by a constant factor that is contingent on the Coulomb oscillation characteristics of the sample.

4.2.3.3 Random Telegraph Noise

The automatic detection of the **RTN** and its separation from the pink noise in **CSDs** are currently problematic, because both result in similar features. Therefore, sensor scans are manually investigated for **RTN** of category 2, with the extensions of bursts being determined directly and the amplitudes from the jump being ascertained in the calculated electrochemical sensor potential. The mean of the extensions and the mean and empirical standard deviation of the amplitudes constitute the **RTN** parameters. Translating the extension of bursts into the **CSD** domain during the simulation requires consideration of the measurement time per voltage due to the time-dependent stochastic nature of the **RTN**. The median of the **CSD** measurement time is used as the default value.

For **RTN** of category 3, the extension and amplitude parameters are manually extracted directly in **CSDs**. The extension corresponds to the residence time and the amplitude corresponds to the observed offset, as illustrated in [Figure 4.5g](#).

4.2.3.4 White Noise

In the **PSD** of a **CSD**, white noise is predominant in the highest frequencies, while other noise types and the signal itself prevail in lower frequencies. The **PSD** of white noise is given by

$$PSD_w = c_w \cdot \sigma_w^2. \quad (4.8)$$

However, as the ratio between pink and white noise varies, a fixed corner frequency for the noise separation cannot be determined. Thus, the sum

$$PSD_{w,p} = c_w \cdot \sigma_w^2 + \frac{c_p}{f} \cdot \sigma_p^2 \quad (4.9)$$

of pink and white noise is fitted to the highest frequencies of the **PSD** computed by Welch's method [285]⁸ from `SciPy` [286]. The fit is applied to the average **PSD** of all rows to provide a better approximation.

The method was tested with simulated data to study its accuracy, and it was found to work well for $\sigma_w > 0.001$ and tend to underestimate the intensity of white noise otherwise.

⁸ $c_p = 0.1$ and $c_w = 2$ for the Welch method

4.3 Evaluation

For the purpose of assessment of the implemented simulation, a set of generated CSDs is first subjected to a visual evaluation⁹ to ascertain the fidelity, diversity, and plausibility. Subsequently, the equivalency of the simulation to experimental data is ascertained. Finally, the performance of the model is compared to simulations based on physical models.

4.3.1 Visual Assessment

As illustrated in Figure 4.6a a series of two-dimensional simulations were conducted. With varying electron occupation, these simulations revealed notable alterations in the structures of the honeycombs, which correspond to the observations made in experimental studies. Additionally, the distortions and sensor responses exhibited in the simulations showed significant similarity to the measurements obtained in experimental trials, further validating the efficacy of the simulation approach. One-dimensional simulations, illustrated in Figure 4.6b, were validated against experimental data in the same manner, complementing the two-dimensional results. These results confirm that the one-dimensional simulations also appropriately capture the essential features observed experimentally.

4.3.2 Metrics for Generative Models

The issue of comparing the statistical distributions of generated and measured data emerges in the context of ML when evaluating generative models. However, many of the metrics employed in that field, such as the inception score [287] and the Fréchet inception distance [288], are not applicable in this scenario. This is due to the fact that their classification method must be trained specifically for the application in question or is pre-trained on natural images. Furthermore, a sample-based metric facilitates a more comprehensive analysis of the deficiencies of the simulation model. Consequently, the α -precision and β -recall metrics are employed to assess the fidelity and diversity of the generated datasets [289]. Specifically, the α -precision metric quantifies the probability that a generated sample falls within in the α -support of the measured data, while the β -recall metric denotes the proportion of measured samples that reside within the β -support of the generated data, in conjunction with a designated k -neighborhood¹⁰. The range of both metrics is from zero to one, with higher values denoting more closely matched distributions of measured and generated data.

To compute the metrics, the CSDs x_i are first embedded into a feature hypersphere. Therefore,

⁹The configuration `default_configs["GaAs_v1"]` provided in the SimCATS project [284] was utilized, in conjunction with varying sensors from the study of Fleitmann [276].

¹⁰ α - or β -support is the minimum volume (sphere) subset of the whole set that supports a probability mass of α or β .

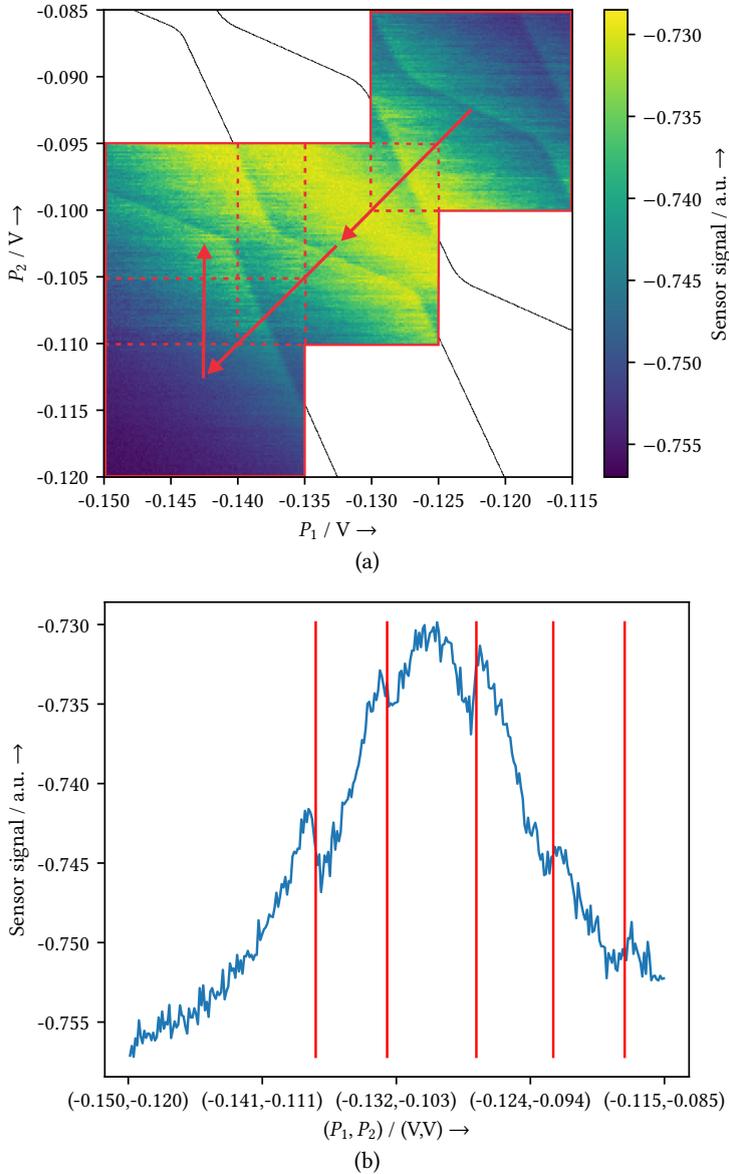


Figure 4.6: Example of simulation results. (a) Simulation of an iterative CSD measurement sequence during tuning. Red boxes illustrate the individually measured CSDs, with the dotted part indicating their overlap and the red arrows their chronological order. (b) A diagonal one-dimensional scan in the voltage space of (a) with red lines indicating the LDTs.

an own adaptation of the MNIST_LeNet neural network ϕ implemented by Ruff [290] is used. The training process is designed to minimize the loss [291]

$$L = R^2 + \frac{1}{vn} \sum_{i=1}^n \max\{0, \|\phi(x_i) - c\|^2 - R^2\}, \quad (4.10)$$

where R is the radius of the hypersphere, c represents the center, v denotes a balancing factor, and n is the number of data points. A corresponding framework for training neural networks [290] is available on GitHub.

For completeness, details on hyperparameter selection and metric computation are available in the work of Fleitmann [276], however they are not pivotal to the present analysis.

4.3.2.1 Preprocessing and Training

Prior to the application of the neural network, a normalization process is initiated to ensure the alignment of the disparate value ranges of the CSDs.

The network is trained with a random selection of 50% of the available 484 experimental data, with the remaining 50% constituting the test set. To increase the quantity of the training dataset by augmentation, rotations, flips, and random brightness and contrast changes are applied to the CSDs using the Python package `albumentations` [292].

4.3.2.2 Results

In order to investigate the results, the simulation should exclusively incorporate CSDs represented in the experimental data. Thus, the parameters α and β are set to 1, as no further outliers are expected. Then, it is tested whether the network can embed CSDs into a feature hypersphere that sufficiently distinguishes CSDs from non-CSD data. Therefore, the training set is benchmarked against the retained experimental test set. The comparison demonstrates high precision and recall values, indicating a reasonable mapping. Following the recommendation by Alaa et al. [289], the value of k for the k -nearest-neighbor region is chosen as the smallest value that achieves 100% recall, which in this case corresponds to nine. This choice balances sensitivity to the distribution of the data with robustness against noise, providing a reliable measure of the performance of the simulation. Furthermore, it is examined whether the network embeds non-CSD data into the same hypersphere. Tests with MNIST data¹¹ [293] attain high precision but very low recall, indicating that they reside in a non coinciding subspace. Consequently, the network has been demonstrated to effectively map CSD data into a designated feature hypersphere. The employed metrics are deemed adequate for the purpose of evaluating the equivalency of experimental and simulated CSDs.

The metrics result in a high precision and a recall of 67.5%. Contrary to the simulated data, it must be considered that the experimental data do not cover the entire voltage space homogeneously, but rather prefer particular regions due to the experimenter's experience. A

¹¹The MNIST dataset consists of images containing handwritten digits.

Table 4.1: Evaluation results utilizing 1-precision and 1-recall. The listed datasets are compared to the experimental data test set. The datasets MNIST and *SimCATS* contain the same number of images as the experimental test set. Additionally, results for expanded MNIST and *SimCATS* datasets, which are approximately 10 times the size of the test set, are provided.

Dataset	1-precision [%]	1-recall (k=9) [%]
Experimental training set	98.8	100.0
MNIST	100.0	2.5
MNIST expanded	100.0	3.3
<i>SimCATS</i>	99.6	67.5
<i>SimCATS</i> expanded	99.8	79.0

comparison with approximately 10 times the amount of data increases the recall to 79.0%, which supports the hypothesis. Conversely, utilizing 10 times more MNIST data than experimental *CSDs* does not result in a substantial enhancement of the recall.

In summary, the simulated data align strongly with the experimental data and map their distribution to a significant extent. Nevertheless, the results indicate that the simulation does not yet represent all experimental *CSDs*. As some of the available experimental datasets include anomalies from postprocessing steps not represented in the simulation, an even higher coverage is expected for unprocessed data.

4.3.3 Performance Analysis

The geometric simulation approach is benchmarked against two typical physical simulations regarding the execution time per simulation using an Intel Xeon w5-2455X - 3.19 GHz. For the comparison, the parameters of the different approaches are configured so that the simulated area covers similar structures. Then, simulations of different resolutions are performed and the execution time is averaged over 50 runs each.

Figure 4.7 visualizes the results. It is evident from the figure that the execution time of the physical models increases quadratically with the resolution per axis. Furthermore, the *HM*, which incorporates quantum effects such as tunnel coupling, exhibits a significantly higher execution time of up to 27.9 s (at a resolution of 500×500 pixels) compared to 16.1 s for the *CIM*. In comparison, the execution time for the geometric model is consistently lower with at most 0.4 s. More precisely, for resolutions in $[50, 500]$, the execution time is in $[19, 418]$ ms with a memory requirement in $[44.4, 90.0]$ MB. The moderate increase in execution time is attributed to the fact that the computation of *TCTs* is the primary determining factor here, whereas the calculation of pixels is considerably faster.

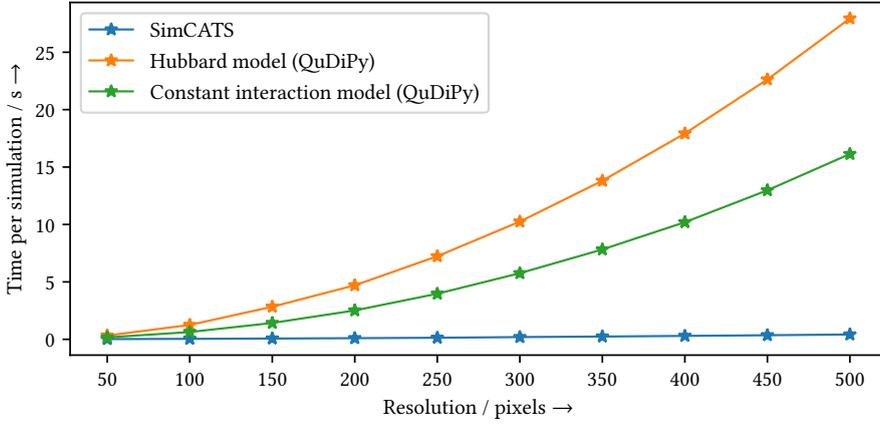


Figure 4.7: Average execution time of different simulation approaches in dependence on the resolution (in pixels per axis). The time is measured as the computation per CSD (without distortions) subsequent to initialization. The physical simulation models implemented in QuDiPy [275] show a quadratic time performance, while SimCATS exhibits minimal dependence on image resolution.

4.4 Outcome and Perspective

A methodology for simulating CSD data has been proposed, incorporating the most relevant effects observed in measurements. Firstly, the model was defined, including the ideal data generation, the sensor response, and the distortions. The TCT-based geometric representation of CSD structures enables the ideal data simulation for DQDs, independent of the sample material and layout. The sensor model incorporates the representation of observed Coulomb peaks as simplified Lorentzians and the dependence of the sensor potential on the lever arms of the DQD gate voltages and occupation. A typical measurement signal path was considered, and the distortions were assigned to three newly proposed categories, to account for the non-linear dependencies observed in the measurements. Their source, simulation, and parameters were subsequently defined. Then, the various model parameters included in the SimCATS project [284] were extracted from the measured GaAs data. Finally, the simulation results were presented and the ability of the simulation to mimic a diverse set of measured data was analyzed.

The simulation framework is intended to be used for tuning algorithm development and benchmarking. It generates realistic data, facilitates the rapid generation of large datasets with known ground truth, and enables fair comparability of diverse approaches from different sites. It also offers complete control over the strength of the distortions and the sensitivity of the sensor, thereby enabling the determination of minimum measurement requirements for the success of an algorithm. Furthermore, using the provided interfaces depicted in Figure 4.1,

the open-source Python framework is designed for simple extension and adaptation to allow the establishment of a standard framework that drives the development of automated tuning solutions.

In the future, the simulation of *CSDs* could be expanded to include additional effects and their impact on the data. For instance, computing wider *CSD* scans¹² has the potential to encompass the varying lengths of *LDTs*. The correlation between consecutively recorded *CSDs* or multiple *DQDs* measured concurrently exerts a negligible influence; however, its inclusion is permissible if deemed necessary. Another open question pertains to the relationship between the geometric *TCT* parameters and the parameters of physical models. With this knowledge, the fast simulation can directly adapt the geometric parameters to changes in the system, thereby facilitating the development of more complex tuning routines.

Nevertheless, algorithms that have been trained solely on simulated data must be tested in experiments, and parameter sets from experiments of other sites are necessary to develop and benchmark robust technology-independent algorithms.

¹²For applications that analyze only small voltage ranges, this is not considered to be necessary.

Chapter 5

Automated Charge Transition Detection in Quantum Dot Charge Stability Diagrams

TUNING the number of electrons in QDs is imperative for the creation of gate-defined semiconductor qubits. Two-dimensional CSDs reveal a change in the number of electrons, hereafter denoted as CT, as an edge in the pixel information (usually in floating-point representation). It is imperative that CTs are detected with a high degree of reliability to realize the automation of the tuning process in its entirety. In the context of scalability, the necessity to overcome bandwidth limitations demands the implementation of an integrated, low-complexity solution that operates within the stringent spatial and heat dissipation constraints, as outlined in Section 3.5.

To this end, a comprehensive analysis of detection approaches was conducted, encompassing classical and ML methods. The present study focuses first on identifying the most effective detection candidates, even if they involve higher algorithmic complexity, deferring the optimization and minimization of the complexity to subsequent and future investigations. The approaches were evaluated through the utilization of the simulation framework SimCATS [294] (see Chapter 4) to generate training and test data. Furthermore, the applicability of these approaches to experimental data was investigated, aiming to ascertain their effectiveness and reliability.

5.1 Datasets

The generation of datasets for training and evaluating CT detection approaches is performed by the simulation framework SimCATS¹ [294]. Hand-labeled experimental data from the Quantum Technology Group of RWTH Aachen are used to assess the generalization ability.

Some of the considered ML approaches require image resolutions divisible by a power of two, e.g., because the resolution is halved in each encoder step and doubled in the decoder. For the models under investigation, a resolution divisible by 32 was required, given that there are at most five such steps. The initially available experimental data from the GaAs sample described in Section 3.1 have a resolution of 100×100 pixels ($30 \text{ mV} \times 30 \text{ mV}$), and therefore the closest

¹Using the Python package SimCATS-Datasets [295] for dataset handling.

Table 5.1: Parameter ranges for the simulated datasets and their distributions (uniform $U(a, b)$, normal $\mathcal{N}(\mu, \sigma^2)$ or exponential $\exp(\lambda)$). The entries without a specified distribution were used solely to verify the correspondence between the sampled parameters. The parameter names in parentheses refer to the variables used in Chapter 4 and Figure 5.1. The TCT parameters are specified in the rotated voltage space (V'_{p_1}, V'_{p_2}). In addition to the distortions mentioned here, dot jumps were also applied as occupation distortion, while RTN was employed as sensor potential and sensor response distortion. The parameters for these distortions were obtained from the original SimCATS configuration `default_configs["GaAs_v1"]` [284].

	Parameter	Minimum	Maximum	Distribution
TCTs	V'_{p_1} -intercept of $ldt_{i,j}$, ($j=1, 2$) ($s_{i,j}$) [V]	$1.0 \cdot 10^{-2}$	$2.5 \cdot 10^{-2}$	$\mathcal{N}(\mu, \sigma^2)$
	Slope of $ldt_{i,1}$ ($m_{i,1}$)	$-4.4 \cdot 10^{-1}$	$-8.0 \cdot 10^{-2}$	$\mathcal{N}(\mu, \sigma^2)$
	Slope of $ldt_{i,2}$ ($m_{i,2}$)	$2.1 \cdot 10^{-1}$	$5.5 \cdot 10^{-1}$	$\mathcal{N}(\mu, \sigma^2)$
	Angle between $ldt_{i,1}$ and $ldt_{i,2}$ (θ_{ld_i}) [rad]	$4.4 \cdot 10^{-1}$	1.7	
	Angle between idt_i and $ldt_{i,2}$ (θ_{ld_i}) [rad]	$5.8 \cdot 10^{-1}$	1.4	$U(a, b)$
	Length of idt_i (s_{id_i}) [V]	$2.6 \cdot 10^{-3}$	$9.9 \cdot 10^{-3}$	$U(a, b)$
	Width of idt_i (w_{id_i}) [V]	$4.3 \cdot 10^{-4}$	$8.1 \cdot 10^{-3}$	$U(a, b)$
	Relation of s_{id_i} to w_{id_i}	$8.7 \cdot 10^{-1}$	9.1	
Sensor	Number of Coulomb peaks	3	6	$U(a, b)$
	Lorentzian scaling factor (a)	$2.2 \cdot 10^{-2}$	$1.9 \cdot 10^{-1}$	$\exp(\lambda)$
	Lorentzian width (γ)	$9.6 \cdot 10^{-4}$	$3.0 \cdot 10^{-3}$	$U(a, b)$
	Lever arm dot 1 (α_1)	$-8.0 \cdot 10^{-4}$	$-9.6 \cdot 10^{-5}$	$U(a, b)$
	Lever arm dot 2 (α_2)	$-5.2 \cdot 10^{-4}$	$-6.3 \cdot 10^{-5}$	$U(a, b)$
	Lever arm gate 1 (β_1)	$2.8 \cdot 10^{-2}$	$1.5 \cdot 10^{-1}$	$U(a, b)$
	Lever arm gate 2 (β_2)	$1.4 \cdot 10^{-2}$	$3.0 \cdot 10^{-1}$	$U(a, b)$
	Relation of β_1 to β_2	$5.0 \cdot 10^{-1}$	2.0	
Distortions	Fermi-Dirac transition blurring <code>sigma</code>	$7.5 \cdot 10^{-5}$	$6.0 \cdot 10^{-4}$	$\mathcal{N}(\mu, \sigma^2)$
	Pink noise <code>sigma</code>	$1.0 \cdot 10^{-10}$	$5.0 \cdot 10^{-4}$	$\exp(\lambda)$
	White noise <code>sigma</code>	$1.0 \cdot 10^{-10}$	$5.0 \cdot 10^{-4}$	$\exp(\lambda)$

resolution of 96×96 pixels ($28.8 \text{ mV} \times 28.8 \text{ mV}$) has been selected to test the networks later on experimental data².

The parameter ranges for the SimCATS simulations were obtained from the data of the GaAs sample as described in Section 4.2. The objective was to generate a diverse dataset to enhance the generalizability of the networks. To that end, the generation procedure randomly selected all model parameters from the extracted ranges listed in Table 5.1. While sampling most values from a uniform distribution, some parameters that describe the structure of the TCTs were sampled from a normal distribution because this matches the observations in the experimental data. In addition to the parameters described in Section 4.1.1, the following parameters for the LDTs and IDTs are supplied:

- the relation between the slopes of $ldt_{i,1}$ and $ldt_{i,2}$ (θ_{ld_i}),
- the relation between the slopes of idt_i and $ldt_{i,2}$ (θ_{ld_i}),
- the length of idt_i (s_{id_i}),

²The experimental GaAs data were reduced by removing the first and last two rows and columns.

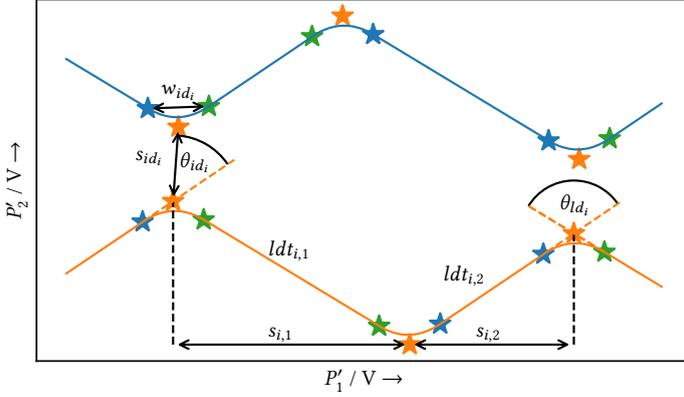


Figure 5.1: Visualization of parameters used to define the TCTs for the geometric simulation in SimCATS.

- the width of idt_i (w_{id_i})³, and
- the relation between s_{id_i} and w_{id_i} .

Figure 5.1 visualizes the various parameters of the used TCT model, with the TCTs displayed in the 45°-rotated voltage space (V'_{p_1}, V'_{p_2}) (see Section 4.1.1). In anticipation of future improvements in sample quality, it was decided to include lower noise levels more often than the very high levels that are sometimes observed in the measurements. Therefore, an exponential distribution is used for these values.

For normal distributions, the mean μ is set to the center and the standard deviation σ is set to a sixth of the range to best represent the distribution in the interval. Furthermore, the scale for the exponential rate is selected to reach the 99% quantile at 60% of the sampling range. Generally, parameters outside the given ranges were resampled.

It is explicitly permitted that the relationship between various parameters is not necessarily physically reasonable. Thus, it is intended to avoid overfitting to the specific qubit sample and improve the generalizability of the models.

The simulated train/validation/test dataset featured 10,000/1,000/1,000 different TCT and sensor configurations, each with 100 CSDs generated with randomly sampled distortion strengths. The two experimental test datasets consisted of 439 (plunger, plunger)-GaAs-CSDs and 81 (plunger, barrier)-SiGe⁴-CSDs. While the GaAs dataset is used to assess the generalization capability of the approaches from simulated to experimental data, the SiGe dataset is employed to evaluate the transferability to a sample of a different technology.

As illustrated in Figure 5.2 the GaAs sample data exhibit DQD features, while the SiGe data

³The width of idt_i defines the rounding at the triple points, which is controlled by the distance between the Bézier anchors $b_{i,j}$, ($j=1,2$) (see Section 4.1.1).

⁴The sample is similar to the one described by Struck et al. [296].

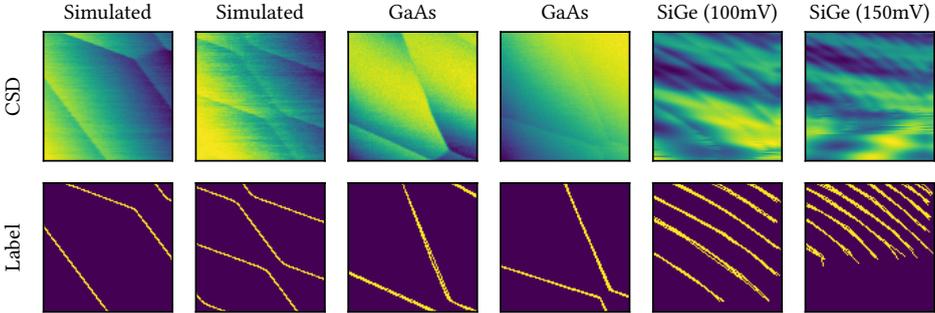


Figure 5.2: Examples for CSDs of the final evaluation datasets and their corresponding ground truth label masks.

feature single QD features. The resolution of the SiGe data is 96×96 pixels, corresponding to $100 \text{ mV} \times 100 \text{ mV}$ or $150 \text{ mV} \times 150 \text{ mV}$. The selection of the SiGe voltage ranges was done to attain a comparable balance of microscopic and macroscopic features as observed in the GaAs data. However, it should be noted that distinct properties are still discernible. Specifically, the CTs appear more blurred, and the spacing between them is reduced, resulting in a higher number of CTs per image. Nearly all SiGe-CSDs exhibit visible CTs, as the Quantum Technology Group of RWTH Aachen specifically recorded them for this evaluation.

While the ground truth CT masks for the simulated datasets are available automatically, masks for the experimental data were manually generated. Consequently, a group of five researchers manually labeled all images, and subsequently combined the labels into a single binary mask to enhance the representation of weak lines. It is noteworthy that the SiGe data were more challenging to label, resulting in broader line segments for the combined manual labels (see Figure 5.2).

5.2 Metrics and Evaluation Methods

The Dice similarity coefficient (DICE) metric was utilized to compare the results of the various approaches and to gauge their quality:

$$DICE = \frac{2|X \cap Y|}{|X| + |Y|}. \quad (5.1)$$

This metric quantifies the similarity between the predicted segmentation mask X and the ground truth segmentation mask Y . A potential drawback of the DICE is its reliance on pixel-precise segmentation, which can be overly stringent, even for minor misalignments. Given the manual labeling of the experimental data, pixel-precise segmentation was not expected. The normalized surface Dice (S-DICE) metric [297] addresses this challenge by introducing a

class-specific threshold for a segmentation deviation. Therefore, the **S-DICE** was additionally used.

For the deep learning approaches, the inference time of the models was measured using the CUDA API, as analysis speed is a crucial factor for a scalable tuning solution. The inference time specifies the calculation time required by the neural network for prediction and does not include the time required to transfer data into GPU memory, which is identical for all approaches. The Python package Calflops [298] was utilized to calculate the floating point operations (FLOPs) and multiply-accumulate operations (MACs) of the networks. These values offer a reasonable estimate of the feasibility of cryogenic implementation on dedicated hardware, as they are independent of the hardware used. Furthermore, the number of trainable parameters was compared, indicating the expected implementation size.

5.3 Charge Transition Detection Approaches

In the study conducted by Scherer [299], various traditional edge and line detection methodologies were assessed on a curated set of experimental data. The evaluation in the present work incorporated a selection of the most effective classical edge detection approaches, in addition to the deep learning methods that are introduced here. The supplementary Table 5.2 provides a comprehensive overview of the utilized approaches, their respective publication years, and their underlying code basis.

5.3.1 Classical Approaches

Canny

A widely used classical edge detector is the algorithm proposed by Canny [160]. This algorithm computes the gradient magnitude and orientation at each pixel and applies nonmaximum suppression and subsequent hysteresis thresholding to determine the edge pixels.

CannyPF

Lu et al. [321] proposed a parameter-free version of the Canny edge detector (CannyPF) that was included in the CannyLines line detection method. This method adaptively selects thresholds based on the distribution of gradient magnitude values of the image pixels.

Generalized Canny (GCanny)

As other local image features may be more robust at detecting CTs than the gradient, a generalized version of the Canny edge detector was developed. It uses a general feature map to replace the gradient magnitude and an optional orientation map to replace the gradient orientation as its input. After nonmaximum suppression, it performs an additional step to connect small gaps in the resulting image. These connections are the result of checking for specific patterns of edge pixels within a window measuring 4×4 pixels. Finally, the GCanny creates a binary edge map using either binary thresholding or hysteresis thresholding. In this

Table 5.2: Collected approaches, their category, publication year, and code basis.

Detector Category	Detector Name	Year	Code Basis
Convolutional	CASENet	2017	[300]
	CHRNet	2023	[301]
	DeepLabV3+	2018	[302]
	DFF	2019	[300]
	FPN	2016	[302]
	LDC	2022	[303]
	LinkNet	2017	[302]
	TEED	2023	[304]
	U-Net	2015	[305]
UNet++	2018	[302]	
Transformer	CrackFormer	2023	[306]
	EDTER	2022	[307]
	MA-Net	2020	[302]
	MMViT-Seg	2023	[308]
	SegFormer	2021	[309]
	Segmenter	2021	[310]
	Swin-Unet	2021	[311]
TransUNet	2021	[312]	
SSM	VM-UNet	2024	[313]
Diffusion	DiffusionEdge	2024	[314]
	MedSegDiff-V2	2023	[315]
Classical	Canny	1986	[316]
	CannyPF	2015	[317]
	ED	2011	[318]
	gPb+GCanny	2008	[319]
	PhCon+GCanny	1999	[320]

work, the features phase congruency and globalized probability of boundary [322] were used to determine the edge strength.

The idea of **phase congruency (PhCon)** rests upon the local energy model proposed by Morrone and Owens [323], which posits that image features become discernible if the Fourier components are maximally in phase. In this work, phase congruency is computed using an enhanced algorithm proposed by Kovese [324], which utilizes log-Gabor wavelets in lieu of Fourier components.

The **globalized probability of boundary (gPb)** approach proposed by Maire et al. [322] involves the integration of diverse local image features to generate a map of boundary probabilities, with subsequent refinement through the consideration of global feature distribution.

Edge drawing (ED)

ED [325] is designed to identify edges by ascertaining the anchor points that are most likely to correspond to edge pixels. This determination is made based on the gradient magnitude of the pixels in question. Once the anchor points have been determined, the **ED** approach establishes links between them.

5.3.2 Machine Learning Approaches

5.3.2.1 Convolution-Based Approaches

A prevalent **ML** approach for image processing involves the utilization of **CNNs**. Networks engineered for edge detection and segmentation tasks have been employed in this work to detect **CTs**.

Cascaded and high-resolution network (CHRNet)

CHRNet [326] is a **CNN** that detects edges using multi-scale representations of the image while preserving the high resolution of the output map. It achieves this by concatenating the output of a convolutional block with the result of the previous block. Additionally, it uses batch normalization layers with an active affine parameter as an erosion operation for the homogeneous region in the image, and generates the output of the network by fusing the outputs of each block.

Lightweight dense convolutional neural network (LDC)

A number of approaches have been developed that prioritize minimizing the size of neural networks, thereby facilitating faster prediction and reducing hardware requirements. One such approach is the **LDC** [327], which integrates aspects of the advanced architectures DexiNed [328] and CATS [329], yet is notably smaller due to some modifications. With approximately 0.7 million parameters, it requires less than 4% of the parameters of DexiNed. Despite its reduced size, **LDC** achieves competitive quality compared to more complex systems. As with DexiNed, the architecture of the **LDC** consists of layers structured in blocks. An ablation study [327] comparing the **LDC** network with three or four blocks demonstrated that the **LDC** with three blocks has approximately 75% fewer parameters but delivers valuable results. For the task of detecting the presence of **CTs**, the versions of **LDC** were trained with either three blocks (**LDC-B3**) or four blocks (**LDC-B4**).

Tiny and efficient edge detector (TEED)

TEED [330] is another lightweight **CNN**, which was developed with the objective of simplicity, efficiency, and generalization. The authors of **TEED** assert that, with a parameter size (58k) of less than 0.2% of contemporary models, it is a remarkably compact network.

Deep category-aware semantic edge detection (CASENet)

The architecture of **CASENet** [331] is based on ResNet and a skip-layer architecture in which category-wise edge activations at the top convolutional layer merge with the corresponding bottom layer features. It uses fixed fusion weights and bases the decision result primarily on high-level features.

Dynamic feature fusion (DFF)

The **DFF** [332] model is based on the **CASENet** architecture, but it employs a feature extractor that normalizes the magnitude scales of multi-level features and adaptive fusion weights for

different locations of multi-level feature maps. This results in finer edges.

U-Net

A task closely related to category-aware edge detection is semantic segmentation, which involves the classification of objects in an image into different categories. The fully convolutional network (FCN) architecture U-Net [178] is a prominent example in this field. It comprises a contracting and an expansive path.

In addition to the standard U-Net, smaller versions of U-Net architectures with fewer layers and channels for convolution were investigated in this work. Small U-Nets are particularly promising candidates for hardware implementation due to their relatively simple architecture. The tiny version **UNet-38k** has three encoder and decoder layers (four in the standard U-Net), begins with six output channels for the first convolutional layer (64 in the standard U-Net), and uses bilinear upsampling.

UNet++

UNet++ [333] is a more advanced version of U-Net, that uses deep supervision to segment medical images. The main difference to U-Net is the use of nested and dense skip connections to reduce the semantic gap between the feature maps of the encoder and decoder.

Feature pyramid network (FPN)

FPN [334] constructs feature pyramids to detect objects at different scales at marginal extra cost. Therefore, it employs a top-down architecture with lateral connections to build high-level semantic feature maps at all scales.

DeepLabV3+

Another semantic segmentation approach is DeepLabV3+ [335], which combines spatial pyramid pooling with an encoder-decoder structure, resulting in refined segmentation results along object boundaries compared to the predecessor DeepLabV3 [336]. DeepLab approaches [337] typically implement dilated filters for "atrous convolutions" and atrous spatial pyramid pooling to robustly segment objects at multiple scales.

LinkNet

LinkNet [338] aims for efficient semantic segmentation by using ResNet18 as a light encoder and bypassing the input of each encoder layer to the output of the corresponding decoder. Thus, the decoder requires fewer parameters because it shares the knowledge learned by the encoder in each layer.

5.3.2.2 Transformer-Based Approaches

Another type of a neural network is the transformer [339], which possesses an underlying attention mechanism. While the original design of the transformer was for processing sequential data, such as text, a variant for image processing named vision transformer (ViT) was developed [340]. However, some models do not directly incorporate a ViT but rather use

special attention blocks, like the [MA-Net](#) model.

Multi-scale attention network (MA-Net)

The [MA-Net](#) model [341] employs a self-attention mechanism to integrate local features with global dependencies in an adaptive manner. Consequently, position-wise and multi-scale fusion attention blocks capture the spatial dependencies between pixels in an overall view and the channel dependencies between feature maps.

Segmenter

The [Segmenter](#) [181] utilizes a [ViT](#) as the encoder and implements two decoder variants for semantic segmentation. The decoder is selected as either an ordinary linear layer or a novel mask transformer, which is a transformer encoder comprising multiple layers. The results that are presented subsequently in this work employ the latter option due to its superior performance on the validation dataset.

SegFormer

[SegFormer](#) [182] employs a modified [ViT](#) as an encoder, characterized by a hierarchical structure that eschews positional embedding. The decoder constitutes a lightweight multilayer perceptron ([MLP](#)), which aggregates information from diverse spatial resolution features stemming from the hierarchical structure, thereby integrating local and global attention.

Edge detection transformer (EDTER)

The transformer-based encoder and convolution-based decoder in [EDTER](#) [183] extract precise and sharp object boundaries and meaningful edges. [EDTER](#) concurrently exploits the complete contextual information of the image and detailed local cues by operating in two stages. The initial stage ([EDTER-Global](#)) utilizes the encoder component of a [ViT](#) with coarse image patches, while the subsequent stage employs a modified local [ViT](#) encoder with finer image patches. Both stages apply a bi-directional multi-level aggregation decoder, and a feature fusion module combines their results prior to being fed into a final decision head.

Mini-mobile vision transformer for segmentation (MMViT-Seg)

The lightweight [MMViT-Seg](#) [184] model features a two-path encoder subnetwork design that effectively captures the global dependence of image features and low-layer spatial details. It employs convolutional and [MobileViT](#) blocks, along with a multi-query attention module, to fuse multi-scale features from different levels in the decoder sub-network.

CrackFormer

[CrackFormer](#) [185] combines a [SegNet](#)-like [342] encoder-decoder with self-attention. It replaces all convolutional layers (except the first and last) with self-attention blocks and implements a feature fusion module that combines the features from each encoder-decoder stage using self-attention. Each self-attention block consists of two convolutional layers, followed by a batch normalization layer and a rectified linear unit ([ReLU](#)) activation function, with a self-attention layer positioned in between these layers.

TransUNet

TransUNet [180] combines the general concepts of the U-Net architecture with a transformer. Although U-Net is effective in local feature detection, its ability to model long-range dependencies is weak. In contrast, transformers have an innate global self-attention mechanism but only limited localization capabilities due to insufficient low-level details. The TransUNet model integrates these two architectures using a hybrid CNN-transformer for the encoder and convolutional layers in the decoder, leveraging the strengths of both to overcome the weaknesses of the other.

Swin-Unet

Swin-Unet [186] also combines the U-shaped architecture and skip-connections of the traditional U-Net with a pure transformer encoder architecture. The proposed method utilizes swin transformers [343], a variant of ViTs [340], for the encoder and a swin transformer-based decoder with patch-expanding layers to up-sample features during the expansive path.

5.3.2.3 State-Space-Model-Based Approaches

SSMs present an alternative to transformers for processing long sequences [344].

Vision mamba UNet (VM-UNet)

An SSM is employed in VM-UNet [188], which shares architectural similarities with Swin-Unet, incorporating the advantages of U-Net and nonconvolutional layers. However, it does not build upon ViTs (more specifically on swin transformers) but rather upon VMambas, a recent architecture that integrates a global receptive field with linear complexity.

5.3.2.4 Diffusion-Based Approaches

Diffusion models are hallmarked by learning both forward and reverse diffusion, and they are frequently employed in the domains of image denoising, image inpainting, and image generation. Typically, forward diffusion involves the application of Gaussian noise to the original image, while reverse diffusion inverts the diffusion process, thereby resulting in the reconstruction of the the original image.

Diffusion probabilistic model for crisp edge detection (DiffusionEdge)

DiffusionEdge [205] is a diffusion probabilistic model (DPM) designed for edge detection. The authors of DiffusionEdge contended that it circumvents the need for extensive computational resources while maintaining optimal performance by leveraging a DPM operating on the latent space. This approach facilitates the application of the standard cross-entropy loss, enabling optimization of parameters within the latent space. Additionally, they adapted a decoupled architecture to expedite the denoising process and proposed an adaptive Fourier filter to adjust the latent features of specific frequencies. This combination should result in very accurate and crisp edge maps.

Medical image segmentation with diffusion probabilistic model (MedSegDiff)

[MedSegDiff](#) [345] is a *DPM* for medical image segmentation. It employs a modified U-Net with conditional encoding and a feature frequency parser in the reverse diffusion stage. The dynamic conditional strategy enables stepwise attention, and the feature frequency parser eliminates the high-frequency noise introduced by the former.

[MedSegDiff-V2](#) [346] enhances [MedSegDiff](#) by integrating vision transformer mechanisms into the *DPM*. It employs convolutional U-Nets as feature extractors but utilizes a transformer to combine features.

5.4 Training and Optimization

Where applicable, the networks were trained using the optimizers, schedulers, loss functions, and hyperparameters of the original publications. Furthermore, the networks were trained using a combination of the `AdamW` optimizer and the `OneCycleLR` scheduler (implemented in `torch.optim` [347]), which is the de facto state-of-the-art superconvergence method proposed by Smith and Topin [348]. In this case, the loss function consists of a combination of `BCEWithLogitsLoss` (implemented in `torch.nn` [347]) and `DICE` loss. Subsequently, the training results of the networks were evaluated on the validation dataset described in [Section 5.1](#) and the best training checkpoint was selected for final comparison. [Table 5.3](#) provides an overview of the hyperparameters employed to train the final checkpoint for comparison. Classical approaches, with the exception of `gPb+GCanny`, were optimized using differential evolution implemented in `SciPy` [286]. For `gPb+GCanny`, a simple grid search was used because only one parameter was optimized.

5.5 Evaluation

The collected approaches from [Section 5.3](#) were evaluated on the test sets described in [Section 5.1](#) using the metrics from [Section 5.2](#).

[Table 5.4](#) specifies the number of parameters, the inference times, the `FLOPs`, and the `MACs` of all *ML* models. A thorough examination reveals that merely four candidates possess parameters counts less than one million. Notably, `UNet-38k` and `TEED` stand out as particularly diminutive models, with parameter counts of 38,041 and 58,622, respectively. Both of these models are primarily based on convolutions and have a relatively simple network structure, which is considered beneficial for hardware implementation.

The inference times for single images (batch size 1) differed significantly between the approach categories. The fastest approaches were from the convolution-based category and the slowest were from the diffusion-based category. In particular, `DiffusionEdge` and `MedSegDiff-V2` are highly time-consuming and, therefore, are less applicable to scalable tuning solutions. It is noteworthy that tiny networks do not fully utilize the `GPU`, and enabling more predictions

Table 5.3: Training hyperparameters of the collected ML approaches, which are sorted by detector name. The optimizers and schedulers refer to the implementations in the `torch.optim` package [347]. For `DiffusionEdge`, parameters were not supplied because the original parameters were used.

Detector	Training		Name	Optimizer		Scheduler
	Epochs	Batch Size		Learning Rate	Weight Decay	
CASENet	5	16	Adam	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	OneCycleLR
CHRNet	5	16	Adam	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	OneCycleLR
CrackFormer	5	64	AdamW	$1.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-3}$	OneCycleLR
DeepLabV3+	5	16	Adam	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	OneCycleLR
DFF	5	16	Adam	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	OneCycleLR
DiffusionEdge	-	-	-	-	-	-
EDTER	1	16	AdamW	$5.0 \cdot 10^{-4}$	$3.0 \cdot 10^{-3}$	OneCycleLR
EDTER-Global	1	16	AdamW	$5.0 \cdot 10^{-4}$	$3.0 \cdot 10^{-1}$	OneCycleLR
FPN	4	64	AdamW	$1.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-3}$	OneCycleLR
LDC-B3	5	64	AdamW	$1.0 \cdot 10^{-2}$	$1.0 \cdot 10^{-3}$	OneCycleLR
LDC-B4	5	64	AdamW	$1.0 \cdot 10^{-2}$	$1.0 \cdot 10^{-3}$	OneCycleLR
LinkNet	10	16	Adam	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	OneCycleLR
MA-Net	5	16	Adam	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	OneCycleLR
MedSegDiff-V2	1	32	AdamW	$1.0 \cdot 10^{-4}$	0	LinearLR
MMViT-Seg	5	16	Adam	$1.0 \cdot 10^{-5}$	$1.0 \cdot 10^{-5}$	OneCycleLR
SegFormer	1	256	AdamW	$3.0 \cdot 10^{-3}$	$3.0 \cdot 10^{-1}$	OneCycleLR
Segmenter	5	32	AdamW	$3.0 \cdot 10^{-3}$	$3.0 \cdot 10^{-3}$	OneCycleLR
Swin-Unet	10	256	AdamW	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-2}$	OneCycleLR
TEED	5	8	AdamW	$1.0 \cdot 10^{-2}$	$2.0 \cdot 10^{-3}$	OneCycleLR
TransUNet	4	64	AdamW	$1.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-3}$	OneCycleLR
U-Net	5	64	AdamW	$2.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-4}$	OneCycleLR
UNet-38k	5	64	AdamW	$2.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-4}$	OneCycleLR
UNet++	5	64	AdamW	$1.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-3}$	OneCycleLR
VM-UNet	5	32	AdamW	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-2}$	OneCycleLR

when multiple such models run in parallel is possible. In a scalable, fully automated tuning setup, a CT detector could analyze the CSDs of multiple different qubits at the same time. Therefore, inference times were recorded for a batch of 64 images.

The numbers of FLOPs and MACs is contingent on the number of parameters and the architecture. As depicted in Table 5.4, compact CNNs require fewer FLOPs and MACs compared to other architectures or convolutional networks with more parameters.

Table 5.5 summarizes the achieved metrics, sorted by the DICE (simulated data) or the S-DICE (experimental data), with a corresponding visualization provided in Figure 5.3. The ML approaches have a clear advantage over classical methods for all test datasets. With regard to the simulated data, numerous approaches achieved substantial success, with U-Net-based architectures demonstrating the highest performance (top 5). Notably, the small UNet-38k architecture attained a DICE score above 0.9. Among the classical methods, only the PhCon+GCanny approach showed any practical application, achieving an S-DICE score of 0.62. It is noteworthy that the global component of EDTER demonstrated superior performance (DICE score of 0.76) in comparison to the entire algorithm (DICE score of 0.75). This is attributable to the

Table 5.4: Size of the neural networks and their inference time, FLOPs, and MACs. The inference times were measured on an NVIDIA RTX A5000. FLOPs and MACs could not be calculated for all networks. Networks are ordered by number of parameters.

Name	Detector Model	Parameters ·10 ⁶	Inference [ms]		GFLOPs /image	GMACs /image
			Batch Size 1	Batch Size 64		
UNet-38k	Convolutional	0.038	0.956	3.295	0.077	0.038
TEED	Convolutional	0.059	1.118	4.070	0.270	0.134
LDC-B3	Convolutional	0.156	1.269	5.155	0.559	0.276
LDC-B4	Convolutional	0.674	2.247	6.813	0.954	0.472
MMViT-Seg	Transformer	1.013	23.810	66.641	0.721	0.354
CHRNet	Convolutional	1.450	2.545	19.167	4.085	2.037
SegFormer	Transformer	4.446	5.157	15.429	5.897	2.946
CrackFormer	Transformer	4.961	21.825	123.081	6.323	3.080
Segmenter	Transformer	6.455	4.971	3775.265	14.517	7.235
LinkNet	Convolutional	11.658	2.190	6.360	1.420	0.706
U-Net	Convolutional	17.262	1.747	36.417	11.237	5.612
CASENet	Convolutional	21.793	2.409	23.293	16.220	8.100
DFF	Convolutional	21.799	2.683	24.545	16.226	8.103
FPN	Convolutional	23.149	3.514	20.702	3.805	1.899
UNet++	Convolutional	26.072	4.637	25.619	10.298	5.141
Swin-Unet	Transformer	27.154	5.994	32.387	2.160	1.074
VM-Unet	State-space	27.424	7.432	99.503	-	-
MA-Net	Transformer	31.777	4.337	14.145	4.628	2.309
DeepLabV3+	Convolutional	45.663	6.779	22.873	7.834	3.906
MedSegDiff-V2	Diffusion	95.723	1726.027	41106.613	-	-
TransUNet	Transformer	105.153	13.828	74.393	21.016	10.489
DiffusionEdge	Diffusion	297.583	357.617	1246.634	-	-
EDTER-Global	Transformer	322.386	16.215	12851.895	254.208	127.008
EDTER	Transformer	416.964	37.759	25147.160	373.765	186.685

adaptation of the global patch size (from 16 to 8) to the low resolution (96×96 pixels) of the measurements. Consequently, the cues of the local stage (patch size of 4) became too akin to the ones of the global stage, resulting in a loss of global information.

The findings derived from the GaAs dataset provide a good gauge for assessing the generalization ability of the different approaches. The simulated data originate from GaAs parameter ranges; thus, similar scores are expected for the experimental GaAs data for the promising methods. Due to the uncertainty of manual labels, the *S-DICE* score was used for comparison. Once again, U-Net-based architectures scored the best (four out of the top five). Remarkably, DeepLabV3+ demonstrated an inability to perform pixel-perfect segmentations, as evidenced by its significantly higher *S-DICE* score of 0.93 compared to its *DICE* score of 0.77 on the simulated data. The proposed UNet-38k model anew attained commendable outcomes with an *S-DICE* score exceeding 0.89. Among the classical approaches, only PhCon+GCanny yielded moderate results with a *S-DICE* score of 0.68.

The edge detection capabilities for an entirely different qubit sample architecture and material were analyzed with the SiGe dataset. This evaluation serves to probe the ability of the approaches to generalize across heterogeneous device platforms. As expected, the mean *S-DICE*

Table 5.5: Metrics calculated for all detectors across the three test datasets. The simulated data results were sorted by the **DICE** score and the experimental data results (GaAs & SiGe) were sorted by the **S-DICE** score, with a threshold of two. This procedure was adopted due to the inaccuracies in the manual labels, which limit the statistical significance of the **DICE** score.

Simulated Data			GaAs Data			SiGe Data		
Detector	DICE	S-DICE	Detector	DICE	S-DICE	Detector	DICE	S-DICE
VM-UNet	0.949	0.985	Swin-UNet	0.683	0.935	Segmenter	0.560	0.860
U-Net	0.948	0.984	VM-UNet	0.680	0.935	Swin-UNet	0.530	0.852
UNet++	0.947	0.982	U-Net	0.684	0.931	LDC-B3	0.542	0.846
Swin-UNet	0.946	0.984	DeepLabV3+	0.688	0.930	LDC-B4	0.522	0.827
TransUNet	0.945	0.982	TransUNet	0.679	0.928	TransUNet	0.485	0.803
CrackFormer	0.942	0.981	Segmenter	0.687	0.925	VM-UNet	0.495	0.793
MA-Net	0.938	0.979	CrackFormer	0.674	0.922	UNet++	0.494	0.790
LinkNet	0.936	0.981	FPN	0.680	0.914	EDTER-Global	0.533	0.785
CASENet	0.926	0.977	CHRNet	0.670	0.913	CASENet	0.452	0.761
DFE	0.923	0.976	UNet++	0.669	0.913	TEED	0.466	0.738
MMViT-Seg	0.913	0.972	LDC-B4	0.660	0.902	EDTER	0.482	0.735
LDC-B4	0.912	0.962	MMViT-Seg	0.654	0.893	UNet-38k	0.427	0.717
LDC-B3	0.912	0.962	UNet-38k	0.640	0.890	DFE	0.435	0.693
UNet-38k	0.908	0.964	LDC-B3	0.644	0.888	CrackFormer	0.404	0.661
CHRNet	0.889	0.972	MA-Net	0.655	0.887	MA-Net	0.398	0.661
MedSegDiff-V2	0.866	0.944	LinkNet	0.623	0.870	LinkNet	0.398	0.659
TEED	0.851	0.924	SegFormer	0.571	0.862	MMViT-Seg	0.402	0.645
FPN	0.775	0.978	MedSegDiff-V2	0.608	0.855	SegFormer	0.371	0.638
Segmenter	0.773	0.976	DFE	0.586	0.834	MedSegDiff-V2	0.381	0.634
DeepLabV3+	0.769	0.980	TEED	0.606	0.829	CHRNet	0.359	0.599
EDTER-Global	0.755	0.876	CASENet	0.566	0.815	DiffusionEdge	0.367	0.595
EDTER	0.748	0.874	EDTER	0.545	0.792	U-Net	0.335	0.594
DiffusionEdge	0.727	0.865	EDTER-Global	0.560	0.778	DeepLabV3+	0.326	0.576
SegFormer	0.501	0.880	DiffusionEdge	0.411	0.753	FPN	0.306	0.535
PhCon+GCanny	0.317	0.619	PhCon+GCanny	0.420	0.678	PhCon+GCanny	0.224	0.485
CannyPF	0.224	0.428	CannyPF	0.179	0.391	gPb+GCanny	0.012	0.102
Canny	0.146	0.371	Canny	0.156	0.386	ED	0.022	0.094
ED	0.142	0.391	ED	0.132	0.368	Canny	0.017	0.081
gPb+GCanny	0.120	0.209	gPb+GCanny	0.171	0.250	CannyPF	0.013	0.068

score of the dataset is lower, because of the fairly different feature properties described in Section 5.1. Surprisingly, U-Net-based architectures no longer perform best; in particular, the original U-Net architecture is deteriorating with an **S-DICE** score of 0.59. This finding suggests that the model is overfitted to double-dot features that are not present in the SiGe data. Concurrently, Segmenter and **LDC** demonstrated higher robustness compared to their competitors, with **S-DICE** scores of 0.86 and 0.85. Once more, competitive results for the UNet-38k (**S-DICE** score above 0.71) were observed, indicating that overfitting was not present, presumably due to its reduced capacity to learn more complex structures. It is noteworthy that all classical approaches attained unsatisfactory results with **S-DICE** scores below 0.49 on the SiGe data due to the presence of blurrier transitions.

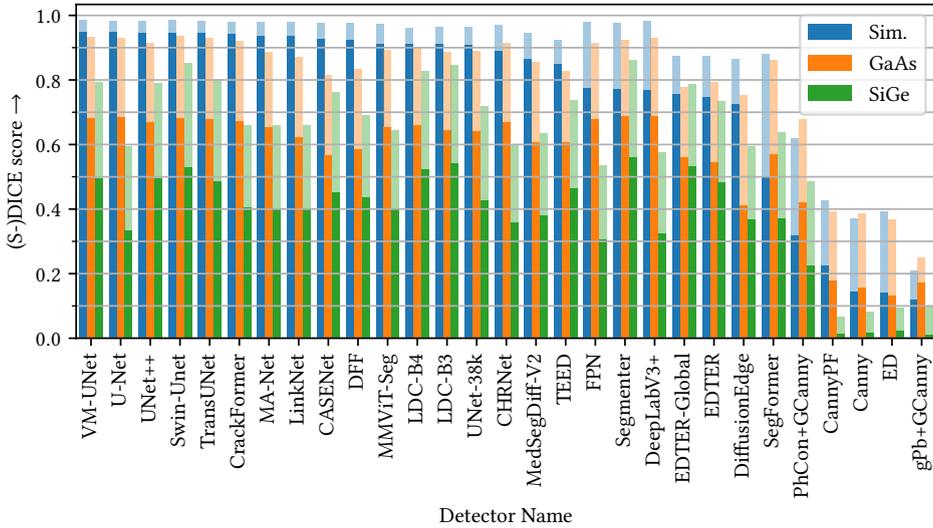


Figure 5.3: Bar plot visualizing the results from Table 5.5. For each detector, the solid portion of the bar represents the DICE score, while the semi-transparent extension indicates the corresponding S-DICE score. The detectors are arranged in accordance with their DICE score on simulated data.

In summary, numerous approaches have been demonstrated to possess favorable overall analysis capabilities. Among the classical approaches, **PhCon+GCanny** is the only one that yielded moderate results. The most effective convolution-based approach appears to be U-Net, which exhibited substandard performance on the SiGe data attributed to overfitting. The scaled-down UNet-38k approach, notable for its surprising robustness and efficiency, did not encounter issues with the SiGe data. Swin-Unet emerged as the leading approach among the analyzed transformer models. The **SSM** approach VM-Unet also detected **CT**s robustly. However, diffusion networks are not well-suited for the application due to their complexity, which hinders an energy-efficient hardware implementation, and their suboptimal performance metrics in the analysis.

Figure 5.4 illustrates exemplary predictions of the best representatives of the respective approach categories. Although **PhCon+GCanny** yields beneficial predictions for certain images, it is highly vulnerable to distortions and the characteristics of the **CT** features. The **ML** approaches exhibit minimal variation in their simulated and GaAs data predictions, with discernible differences only emerging when generalizing to the SiGe data. For instance, U-Net exhibits notable deficiencies in its prediction of contiguous edge structures, while UNet-38k demonstrates better performance and Swin-Unet exhibits best results.

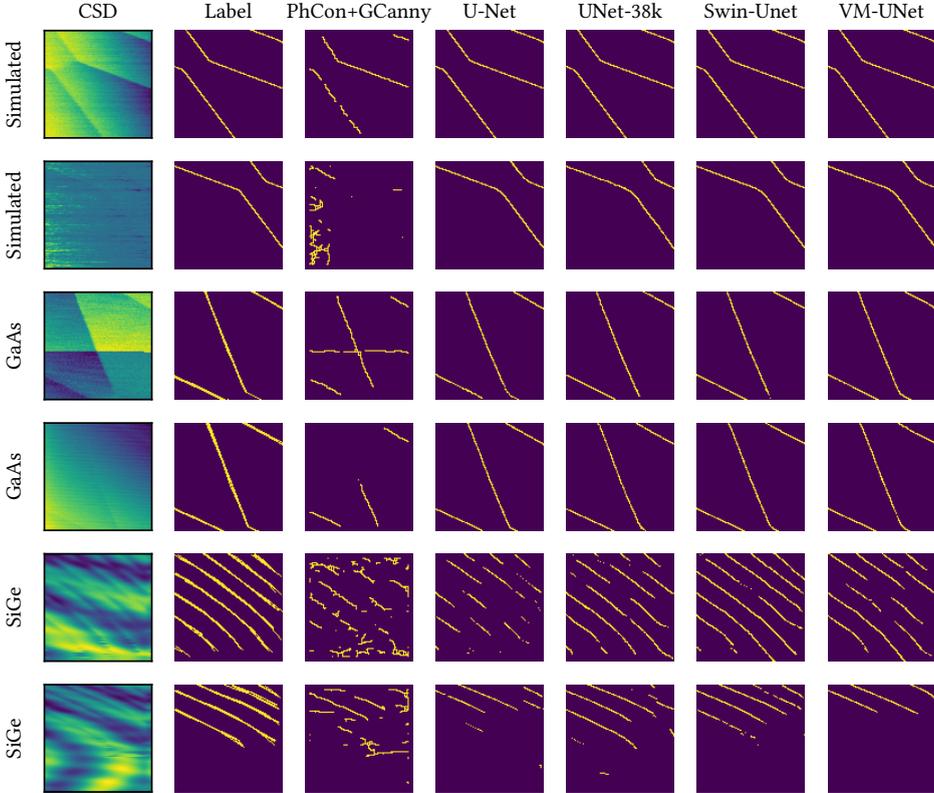


Figure 5.4: Exemplary predictions on the test datasets.

5.6 Outcome and Perspective

In this study, a range of classical and ML methods for detecting CTs in CSDs were evaluated using both simulated data from the SimCATS framework and experimentally measured CSDs. The primary objective was to identify suitable approach categories for robust detection with the potential for future qubit-near hardware implementation.

The ML-based approaches were subdivided into convolution-based, transformer-based, SSM-based, and diffusion-based approaches to analyze the abilities of different architectures. The analysis of the detection metrics⁵ has shown that ML-based approaches are far superior to their classical competitors. All investigated ML architecture categories featured valuable candidates except for the diffusion-based approaches. The results demonstrate that approaches trained on the simulated dataset can generalize to experimental data.

⁵The simulated test dataset used in this study is available for benchmarking purposes [349].

Convolution-based architectures are recommended for hardware implementation due to their low complexity and convincing detection results. Furthermore, the possibility of creating smaller versions of networks that still have sufficient detection ability, as demonstrated with UNet-38k, is emphasized.

It is recommended that subsequent research endeavors investigate the viability of convolution-based networks that exhibit reduced complexity. However, experimentation with smaller variants of U-Net indicated that a further reduction in size can rapidly result in substantial quality losses. Consequently, the potential for reduction in network size is closely associated with the intricacy abatement of the analysis task. Therefore, further investigations are necessary to ascertain the extent to which improvements in SD tuning and sensor compensation lead to further minimization potential.

Chapter 6

Sensitivity and Efficiency Refinement with Improved Sensor Tuning

THE quality of QD measurements is contingent upon the sensitivity of proximate charge sensors, which are implemented in the present work using SDs. Sufficient sensor sensitivity facilitates precise feature extraction, thereby enabling accurate qubit state readout and efficient QD tuning. Chapter 5 discussed the feasibility of the CT analysis with the assumption that the SD is not necessarily optimally tuned. Although many approaches have achieved excellent detection quality, the complexity reduction remains a major challenge. Therefore, it must be investigated how much the complexity of the CT analysis can be reduced by different sensor tuning optimizations.

First, Section 6.1 discusses the automated selection of the Coulomb oscillation area. Next, Section 6.2 provides a comprehensive overview of the noise-sensitive selection of a working point. Then, Section 6.3 explores the reduction in complexity of the CT detection algorithms under the assumption of a compensated SD. Section 6.4 provides a complementary evaluation of the energy efficiency of scaled-down ML networks. Finally, Section 6.5 presents a developed simulation of realistic SD data to guide future advancements.

6.1 Automated Selection of the Coulomb Peak Oscillation Area

The identification of the Coulomb oscillation regime constitutes a preparatory step in the process of SD tuning. This procedure entails the sweeping of the gate voltages of the SD, concomitant with the monitoring of its conductance, to enable the detection of periodic oscillations. The presence of a well-defined oscillation pattern signifies that the sensor has been successfully positioned within the appropriate regime for charge sensing, thereby conferring the requisite sensitivity to resolve QD CTs with a high degree of fidelity.

For the devices under consideration in this work, these steps include the division of the sample into a sensing and computation region and the manipulation of SD gates to a state where Coulomb oscillations can be discerned in barrier-barrier scans. While the latter tuning step is merely preparatory for the noise-sensitive sensor tuning and is not directly critical for the

final sensitivity, it should also be automated to enable a scalable system. Consequently, several promising approaches comprising traditional and ML methods have been proposed [78, 95, 350, 351]. However, these approaches still require improvement, particularly the training of deep learning methods, which necessitates substantial datasets and, contingent on the learning method, additional ground truth labels. Subsequent to the tuning of the Coulomb oscillation regime, the SD fine-tuning selects the exact operation point to optimize the sensor response by balancing sensitivity and stability for robust readout.

6.2 Noise Sensitive Coulomb Peak Selection

In sensor scans (see Section 3.4), the measured signal is proportional to the conductance, denoted by $G_m(SL, SR)$, and comprises the undisturbed SD signal overlaid with some noise. As part of the autotuning development, a primary focus is to identify regions with sufficient signal-gradient-to-noise ratio (SGNR), to ensure optimal sensitivity. The SGNR quantifies the freedom from interference of the sensor sensitivity with respect to the SD potential, which is proportional to a gate voltage. For instance, given the barrier voltage SL its calculation is possible by

$$SGNR = \frac{\overline{\left| \frac{dG_m}{dSL} \right|}}{\sigma_n}, \quad (6.1)$$

where σ_n is the standard deviation of the noise and \bar{x} denotes the mean value of x . It reflects how effectively small changes in the input can be distinguished in the presence of noise, making it a critical metric for sensors relying on differential measurements.

To facilitate understanding, the subsequent subsections distill the essential facts of the works by Hader et al. [79, 352].

6.2.1 Noise Analysis Results

To evaluate noise estimators for the given application, the nature of the existing noise must be analyzed, to construct generated data. In the context of this analysis, random fluctuations and telegraph noise [353] were not considered, as they must be addressed by dedicated algorithms for the rejection of Coulomb peak flanks. A fully automatic approach was employed for the noise analysis (see Figure 6.1).

The investigations resulted in a simplified generic model for the random variable of the measured signal G_m that consists of an idealistic noise-free signal G_{sd} at the SD and additive noise n . The noise n is composed of gradient-dependent noise n_{grad} , which impacts the signal before the dot, and signal-independent noise n_{const} , which affects the signal after the dot:

$$\begin{aligned} G_m(SL, SR) &= G_{sd}(SL, SR) + n \\ &= G_{sd}(SL, SR) + n_{grad} + n_{const}. \end{aligned} \quad (6.2)$$

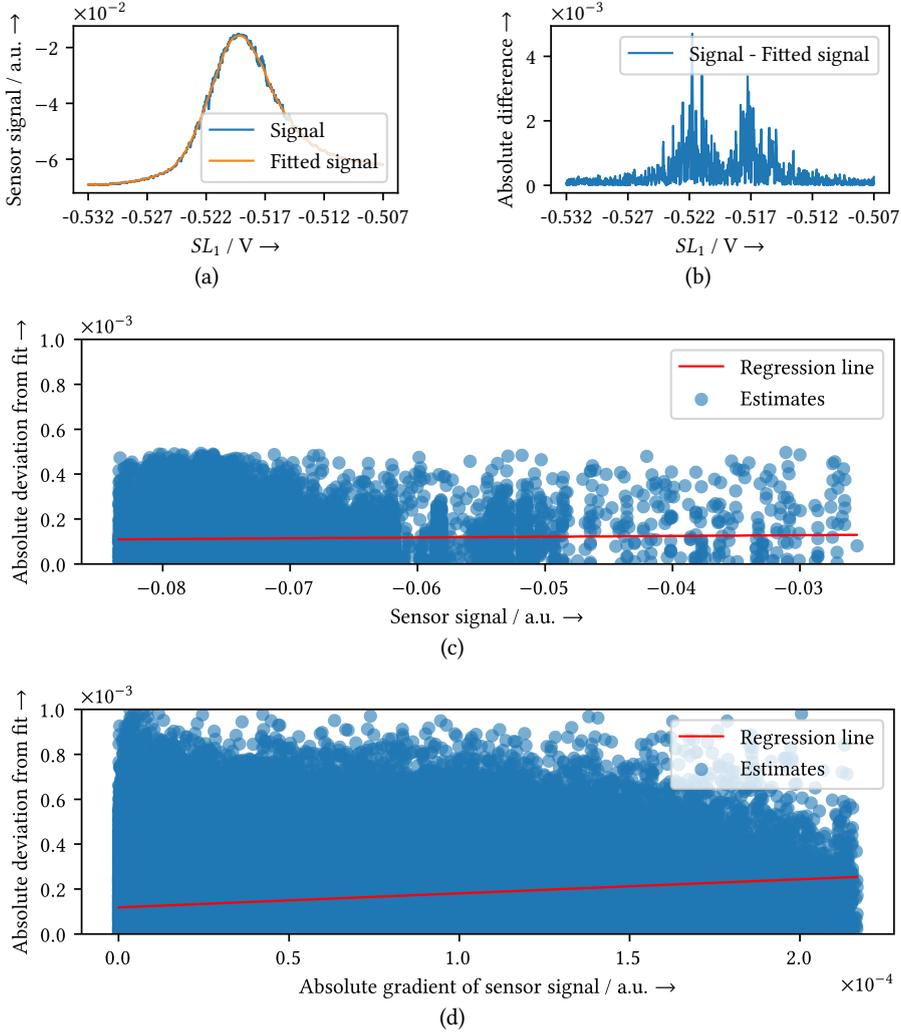


Figure 6.1: Example of the fully automatic approach for the noise analysis. (a) Initially, a one-dimensional Gaussian smoothing was performed, followed by a spline interpolation. (b) The absolute difference between the approximated and original signal suggests that the noise is dependent on the signal regime. A local noise analysis using the methods of Chen et al. [121] and Donoho [133] revealed the dependence of the noise on the sensor signal, as depicted in (c) for low-gradient regions and in (d) for high-gradient regions. The regression lines exhibited an increase in noise of 18.33% and 115.52%, respectively, indicating a substantial local noise in the flanks. Among others, this originates from distortions directly affecting the gate voltages or occurring in the sample, amplified parametrically by the sensor.

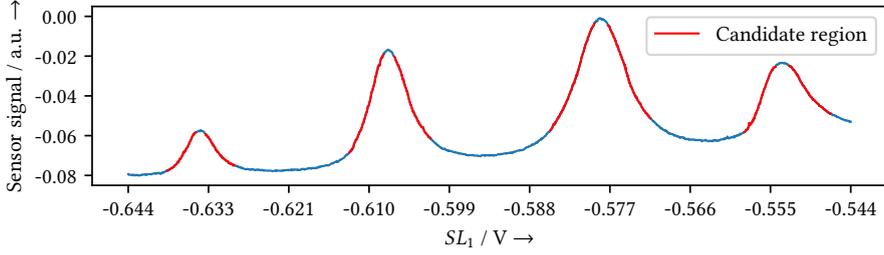


Figure 6.2: Example of candidate regions for a sensitive SD operating point. Candidate regions are colored red in the sensor signal.

In the present experiment, n_{grad} substantially consists of charge noise and n_{const} primarily emerges from thermal noise. The standard deviation of the total noise in the measurement data, σ_n , is composed of σ_{const} and $\hat{\sigma}_{grad}$:

$$\begin{aligned} \sigma_n &= \sqrt{\hat{\sigma}_{grad}^2 + \sigma_{const}^2} \\ \hat{\sigma}_{grad} &= \sigma_{grad} \cdot \frac{\partial}{\partial SL} G_{sd}(SL, SR). \end{aligned} \quad (6.3)$$

6.2.2 Flank Candidate Selection and MSG Criterion

The regions that are deemed suitable for a sensitive SD are delineated by the flanks of the Coulomb peaks. The example depicted in Figure 6.2 provides a visual representation of the candidate regions in the signal.

The establishment of qualified regions for an optimal operating point is contingent upon the tuning step under consideration. These regions are determined by a composition of the following requirements:

- a steep slope to achieve the strongest possible response and sensitivity [77, 234],
- an adequate linearity of the slope to ensure a proportional sensor signal within the region and ease signal interpretation,
- an extensive voltage range to ensure maximum flexibility when the working point undergoes shifts, and
- a sufficient SGNR.

The selection of the operation point is sensitive to the last requirement, thus necessitating precise noise estimation. Consequently, the proposed minimal signal gradient (MSG) criterion

$$\begin{aligned}
 \text{SGNR} &\geq \epsilon \cdot \frac{\Delta V}{V_{\min}} \\
 \text{MSG} &= \epsilon \cdot \sigma_n \cdot \frac{\Delta V}{V_{\min}} \leq \left| \frac{dG_m}{dSL} \right|
 \end{aligned}
 \tag{6.4}$$

with

- ϵ denoting the desired **SGNR** factor,
- σ_n denoting the estimated standard deviation of the noise,
- ΔV denoting the voltage sample distance, and
- V_{\min} denoting the minimal required voltage resolution of the **SD**,

ensures a sufficient **SGNR** in a qualified region. In [Figure 6.3](#) a visualization of the **MSG**'s composition is provided. For a Gaussian distribution, 99.73% of the noise falls within the range of 3σ . A value of 10 is selected for ϵ , resulting in an additional security distance of $4\sigma_n$ between two measurement points. Consequently, if the gradient in a qualified region is consistently greater than the **MSG**, the signal change for each V_{\min} is at least ϵ times as large as σ_n .

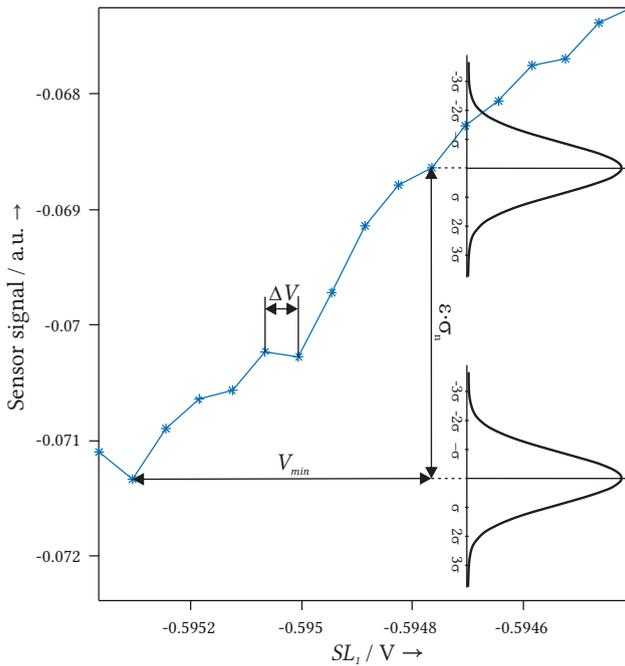


Figure 6.3: Visualization of the **MSG** criterion (Eq. (6.4)).

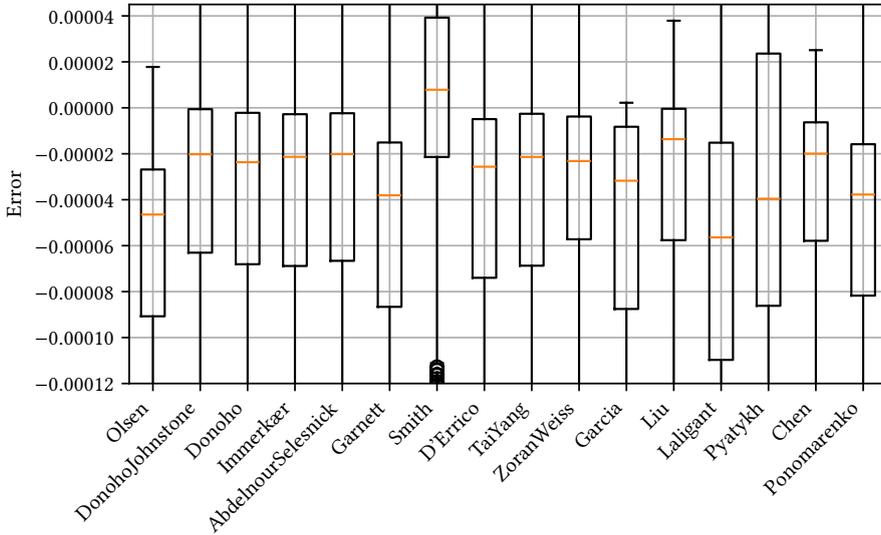


Figure 6.4: Boxplots of the errors on the evaluation dataset with the noise model described in Section 6.2.1. It is important to note that the plot is confined to the boundaries of the boxes. The nomenclature for the estimators aligns with that of the study by Hader et al. [79].

6.2.3 Suggested Noise Estimator and Application to Measured Data

The estimation quality of several noise estimators was evaluated in the flank candidate regions of fitted measurement data with added noise, based on the aforementioned noise model. As shown in Figure 6.4, in most cases the estimation error of the noise estimators is negative, indicating a tendency to underestimate the noise. Subsequent analysis revealed that this underestimation of the noise is systematic, and consequently, it can be approximately corrected with a compensation term.

Due to the absence of ground truth, a quantitative analysis of the estimation errors in the measured data is not feasible. However, the impact of the estimators on the detection of qualified regions can be evaluated. This evaluation was performed with the *Chen* (with and without underestimation compensation) and *Pyatykh* estimators to examine their scatter and underestimation influence. The *Chen* approach exhibited the lowest dispersion, yet it also demonstrated an underestimation tendency, while the *Pyatykh* estimator demonstrated the highest dispersion but no underestimation.

The flanks form diagonal wavefronts across multiple rows (see Figure 6.5). Regions that were rejected (red) were identified in areas where these wavefronts diminished due to lower gradients falling short of the *MSG* criterion. It was evident that this phenomenon exhibited less reg-

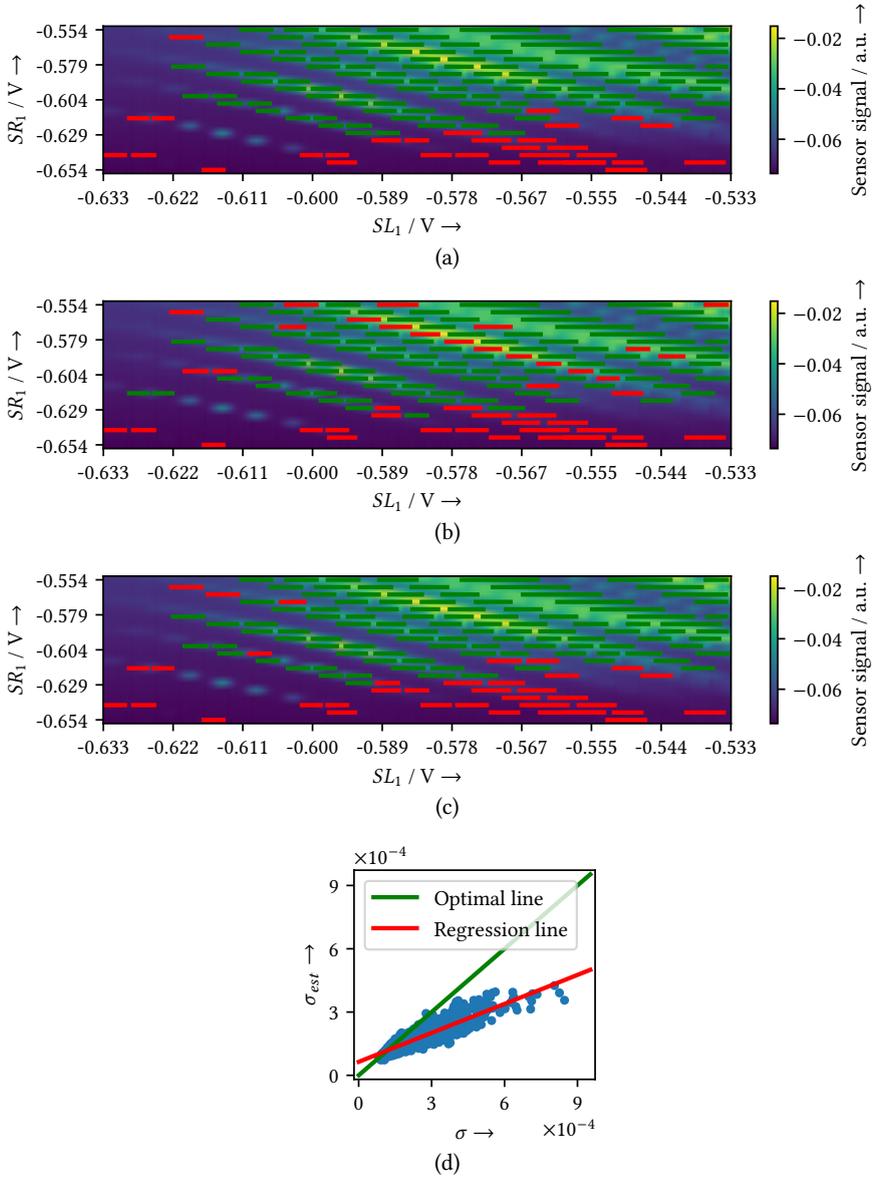


Figure 6.5: Results of the investigation regarding the sensitivity of the quality of results to broader dispersion of the estimation: (a) *Chen* approach, (b) *Pyatykh* approach, and (c) *Chen* approach with approximated compensation. Qualified regions are marked green, and rejected regions are marked red. (d) Regression line in *Chen*'s scatter plot for the results on the evaluation dataset.

ularity for an estimator with broader dispersion, such as the *Pyatykh* estimator (Figure 6.5b), than for one with smaller dispersion, like the *Chen* estimator (Figure 6.5a). A diminished signal-to-noise ratio, attributable to a persistent noise floor, amplified this phenomenon in regions of low wavefront amplitude.

To assess the influence of the underestimation, the *Chen* estimator was corrected by an approximated linear compensation derived from the regression line in the examinations on generated data (refer to Figure 6.5d). The results of this analysis, illustrated in Figure 6.5c, revealed no random behavior; only a negligible amount is observed in the weak parts of the wavefronts. Consequently, it has been demonstrated that the estimation dispersion is of greater pertinence for the region identification than the extent of underestimation. In light of this, when approximately compensating for the underestimation, it is proposed to employ the *Chen* estimator.

The presented work has enabled the automated selection of Coulomb peaks with a high degree of sensitivity. However, the incorporation of these components into a comprehensive automation framework remains an experiment-specific task and is therefore left open. In the pursuit of the integration, future research endeavors must take into account several aspects. Depending on the specific application scenario, such as tuning or readout, a flank selector must optimize the balance between the width and steepness of the flanks. Moreover, the selector is tasked with mapping flanks to wavefronts, identifying wavefront irregularities, and evaluating the neighborhood to select the most stable operation area.

Additionally, compensating for the SD drift can enhance the quality of the readout data, thereby facilitating the automation of subsequent tuning steps.

6.3 Charge Transition Detection Complexity Reduction Using Compensated Sensor

The objective of this study is to determine the extent to which the compensation of the SD leads to further minimization potential, thereby validating the additional effort required for compensation using virtual gates as described in Section 3.2. A fundamental objective is to assess the efficacy of such compensation in facilitating the effective application of ray-based (RB)¹ analysis. Therefore, the investigation encompasses both two-dimensional and RB U-Net variants and compares them with simple classical alternatives. To enable the comparison with the two-dimensional approaches, the RB candidates are applied line-wise to the two-dimensional CSDs.

The results from Section 5.5 suggest that an approach must achieve an S-DICE score of 0.95 on simulated data to ensure sufficient detection quality on experimental data.

Under the hypothesis that future qubit samples will achieve superior overall data quality and consistency, a separate investigation on the expected improvement of the analysis procedure is conducted.

¹RB analysis means, that single rays of one-dimensional data are analyzed instead of two-dimensional CSDs.

6.3.1 Compensated Dataset Generation and Optimization of Approaches

A preliminary study is conducted using the *SimCATS* simulation in which the *SD* does not yet require explicit compensation. Rather, an almost perfectly compensated *SD* is assumed². Furthermore, the *SD* is reset for each *CSD*, ensuring that the measurement initiates at the steepest point of the rising sensor slope. Apart from this modification, the training, validation, and test datasets are generated with the same parameters as in *Section 5.1*. *Figure 6.6* delineates the effects of the sensor compensation on the data. This data enables an initial estimation of the potential for reduction that can be expected through the implementation of a *SD* compensation. In the compensated measurements, the sensor signal exhibits a step characteristic with *CTs* manifesting as discrete jumps.

An additional dataset was generated for the investigation of the impact of the superior data quality, wherein the parameter ranges for the noise and the coupling of the *SD* to the *DQD* are constrained to the superior quality-affecting 10% of the original range. It is noteworthy that, apart from sensor compensation, this data quality has already been demonstrated with contemporary samples, as the ranges are extracted from experimental data.

Initially, U-Net and UNet-38k are trained on the new compensated dataset to assess whether the prediction quality increases with compensated data. Subsequently, the U-Net is successively reduced further while observing the prediction quality on the validation dataset. In addition to reducing the size, the training hyperparameters of the different variants are optimized for the new dataset.

Two smaller versions, one and two orders of magnitude smaller than UNet-38k, are selected for the final evaluation on the test dataset. UNet-4k comprises two encoder and decoder layers, initiates with four output channels for the initial convolutional layer, and employs bilinear upsampling. In contrast, UNet-447 features a single encoder and decoder layer, commences with two output channels for the initial convolutional layer, and does not utilize bilinear upsampling. Furthermore, the analysis encompasses the consideration of the *RB* versions of the two smallest variants, which feature only 1481 and 179 parameters, respectively.

In addition to the U-Net models, the parameters of the most effective classical approach *PhCon+GCanny* (see *Section 5.5*) are optimized for the new dataset. Furthermore, a first-order derivative (*FOD*) filter is used as a reference for a simple one-dimensional filtering approach. The *FOD* filter is combined with a normalization of the data before applying the filter and a subsequent thresholding of the filter output.

Finally, a straightforward hardware integrated averaging and thresholding (*HIAT*) methodology is considered, which is based on the schematic in *Figure 6.7*. The proposed circuit [354] utilizes a transimpedance amplifier, comprising an operational amplifier (*OPA*) and a feedback resistance. While sweeping the voltages of the *DQD* plunger gates, the output voltage V_{out} is sampled. The sampling capacitors, in conjunction with the output resistance of the transimpedance amplifier and the on-resistance of the switches, function as a low-pass filter, thereby implementing an averaging process. The correlated double sampling (*CDS*) process

² $\beta_1, \beta_2 \in [0.0001, 0.0005]$

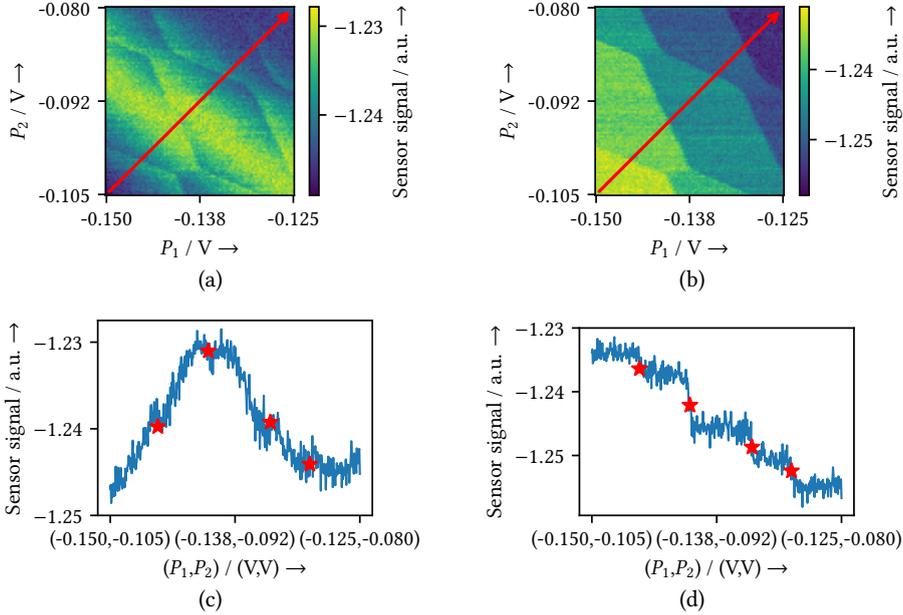


Figure 6.6: Visualization of the effects of sensor compensation on the CSD data using SimCATS. (a) Simulated CSD without and (b) the same CSD with sensor compensation. One-dimensional simulations along the red arrows in (a) and (b) are shown in (c) and (d), respectively. Evidently, the compensated data is more conducive to low-complexity analysis techniques, as the transitions between disparate levels, highlighted by red stars, are more readily discernible in the global space. It is noteworthy, that in the provided example, the sensor abides in a distinct Coulomb peak, a phenomenon that is not guaranteed for larger sweep ranges.

is executed in an alternating nonoverlapping manner, followed by a comparator. The timing diagram in Figure 6.8 provides a visual representation of the aforementioned. This approach might facilitate the detection of the CTs while traversing the DQD plunger gate voltages, obviating the need for digitizing the intermediate measurement data. Consequently, the circuit is regarded as a contender for facilitating scalable detection with an integration adjacent to the qubits.

The final training parameters for the U-Net models, which were used for both the compensated and the compensated superior quality dataset, are delineated in Table 6.1. It is noteworthy that the learning rate and weight decay parameters of U-Net and UNet-38k remain identical to those employed for the dataset with the uncompensated sensor. Nevertheless, it was necessary to reduce the learning rates for the smaller versions. This finding suggests a correlation between the model size and the learning rates. Furthermore, an increased batch size proved advantageous for smaller models, with the exception of the RB variants. The weight decay

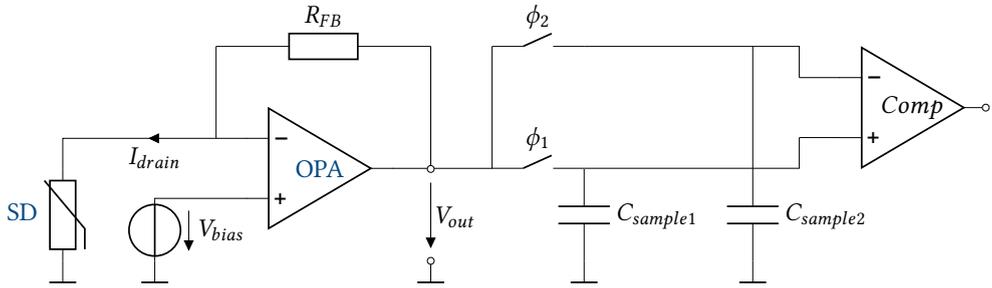


Figure 6.7: Schematic of the HIAT.

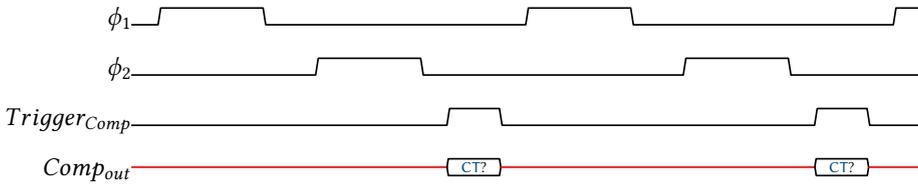


Figure 6.8: Timing diagram of the HIAT.

parameter was fixed due to its stable performance across network sizes, with the exception of UNet-1k-RB, which necessitated an alternative setting. Moreover, the number of epochs was modified.

6.3.2 Results with Sensor Compensation

The selected approaches were evaluated on the compensated data test set using the same methods as described in Section 5.2.

As demonstrated in Table 6.2, the metrics for both the two-dimensional and the RB approaches are arranged in descending order according to the S-DICE score. It is important to note that, even with a further reduction in size, the methods based on ML demonstrate a substantial advantage over the best-performing classical approach. It can be observed that the DICE and S-DICE scores of U-Net and UNet-38k exhibited an increase in comparison to the results obtained without sensor compensation that were detailed in Table 5.5. Consequently, the sensor compensation demonstrates to improve the detection quality of the networks. The smaller versions achieved remarkable strong results, particularly UNet-4k, which nearly equaled the results of the two larger variants. The smallest two-dimensional version (UNet-447) still achieved an adequate detection quality, especially in terms of the S-DICE score, which is considered to be the more important metric, as a pixel-perfect detection is not necessarily required. The application of the compensated sensor dataset demonstrates to enhance the

Table 6.1: Training hyperparameters of the U-Net networks. The optimizers and schedulers of the **ML** approaches refer to the implementations in the `torch.optim` package [347].

Model	Training		Name	Optimizer		Scheduler
	Epochs	Batch Size		Learning Rate	Weight Decay	
U-Net	5	64	AdamW	$2.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-4}$	OneCycleLR
UNet-38k	5	64	AdamW	$2.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-4}$	OneCycleLR
UNet-4k	3	128	AdamW	$1.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-4}$	OneCycleLR
UNet-447	4	128	AdamW	$1.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-4}$	OneCycleLR
UNet-1k-RB	2	64	AdamW	$1.0 \cdot 10^{-1}$	$1.0 \cdot 10^{-3}$	OneCycleLR
UNet-179-RB	5	64	AdamW	$1.0 \cdot 10^{-2}$	$1.0 \cdot 10^{-4}$	OneCycleLR

detection capabilities of both UNet-4k and UNet-447, which did not exhibit promising capabilities on uncompensated data. The detection abilities of **PhCon+GCanny** are also enhanced, yet they remain inferior to the **ML** models, particularly with regard to the **DICE** score.

The accuracy of the **RB** approaches is generally inferior to that of the two-dimensional approaches. A notable exception is UNet-1k-RB which attained an **S-DICE** score of 0.8903. This result indicates that UNet-1k-RB performs better than the most effective classical two-dimensional approach. Even the most rudimentary network UNet-179-RB generated superior outcomes in comparison to the classical **RB** approaches. It is noteworthy, that the **DICE** score of UNet-179-RB remained superior to that of the most effective classical two-dimensional approach (**PhCon+GCanny**). This observation suggests that **ML** methods generally possess a stronger capacity for highly accurate **CT** detection. The **FOD** filter and **HIAT** achieved **S-DICE** scores below 0.8 and **DICE** scores below 0.4. Consequently, they are not considered to be sufficient for a robust automated tuning, as they demonstrate a substantial shortfall relative to the target threshold of an **S-DICE** score of 0.95.

Figure 6.9 illustrates the predictions of the aforementioned approaches on exemplary simulated test data with compensation. U-Net and UNet-38k have been excluded from the comparison, because their prediction is not distinguishable from UNet-4k, which produces visually excellent predictions. It is noteworthy that UNet-447 maintains its excellent performance, with the exception of the most severely noise-contaminated scenario that still retains some **CTs**, where its performance declines. The inability of **PhCon+GCanny** to detect **CTs** in certain **CSDs**, is likely attributable to its comparatively limited flexibility in comparison to a **ML** network. This limitation impedes the capacity to respond in a manner that is adaptable to varying circumstances.

Among the various **RB** approaches, UNet-1k-RB yielded the best results. However, its deficiencies relative to the two-dimensional approaches become particularly evident at elevated noise levels. The smallest **ML** network UNet-179-RB achieved similar results as UNet-1k-RB, but exhibited random false detections. The results obtained from the utilization of **HIAT** were visually different to those yielded by the other detectors. This phenomenon is attributed to the implementation of nonoverlapping windows, which results in a coarser discretization. The effect described herein leads to a reduction of the **DICE** score; however, it has no effect on the **S-DICE** score due to a sufficiently large distance threshold. As evidenced by the visual results

Table 6.2: Metrics calculated for all approaches on the compensated sensor test set. The results are separated into two-dimensional and RB approaches, each sorted according to the S-DICE score.

Category	Detector Name	DICE	S-DICE
2D	U-Net	0.9780	0.9977
	UNet-38k	0.9644	0.9944
	UNet-4k	0.9483	0.9884
	UNet-447	0.8917	0.9474
	PhCon+GCanny	0.4991	0.8673
RB	UNet-1k-RB	0.7182	0.8903
	UNet-179-RB	0.6322	0.8229
	FOD filter	0.3289	0.7472
	HIAT	0.2155	0.7400

of HIAT, the detection probability is found to exhibit a notable dependence on the relative direction of the measurement direction in relation to the CT direction. This phenomenon can be attributed to the fact, that the averaging time and the corresponding amplitude threshold are not robust across all conditions. Furthermore, the performance of HIAT was found to be contingent upon the noise level, a factor that gives rise to a considerable number of false detections. This incidence was also observed in the context of pure noise, in the absence of underlying CTs. The results of the FOD filter manifested a visual resemblance to the outcomes of HIAT, exhibiting analogous deficiencies. However, its performance at higher noise levels appeared to be marginally superior, though it exhibited a comparable rate of false detections.

6.3.3 Results with Sensor Compensation and Superior Data Quality

The results obtained under the hypothesis of superior data quality for the approaches under consideration are shown in Table 6.3. Notably, all two-dimensional U-Net variants attained an optimal S-DICE score, indicating their effectiveness. It is evident that PhCon+GCanny achieved an S-DICE score of 0.916, which, while falling short of the targeted value of 0.95, exhibits a substantial enhancement over the outcome observed on the initial dataset. However, its DICE score showed a slight decrease, suggesting that there has been no advancement in the realm of pixel-perfect detection.

In the group of the RB approaches, UNet-1k-RB attained a substantial enhancement in performance, achieving an S-DICE score of 0.9926 and a DICE score of 0.8662. Consequently, it is the sole RB approach that fulfills the objective of an S-DICE score of 0.95. Although the target was not met, UNet-179-RB, HIAT, and the FOD filter demonstrated a substantial enhancement in their performance. It is noteworthy that for the superior quality dataset, these three approaches achieved a similar S-DICE score. Nevertheless, UNet-179-RB continued to demonstrate a superior DICE score, thereby underscoring its strength in detecting with high precision. The declining S-DICE score of UNet-447 and UNet-179-RB compared to the larger variants implies that a further size reduction of the U-Net architecture is not advisable.

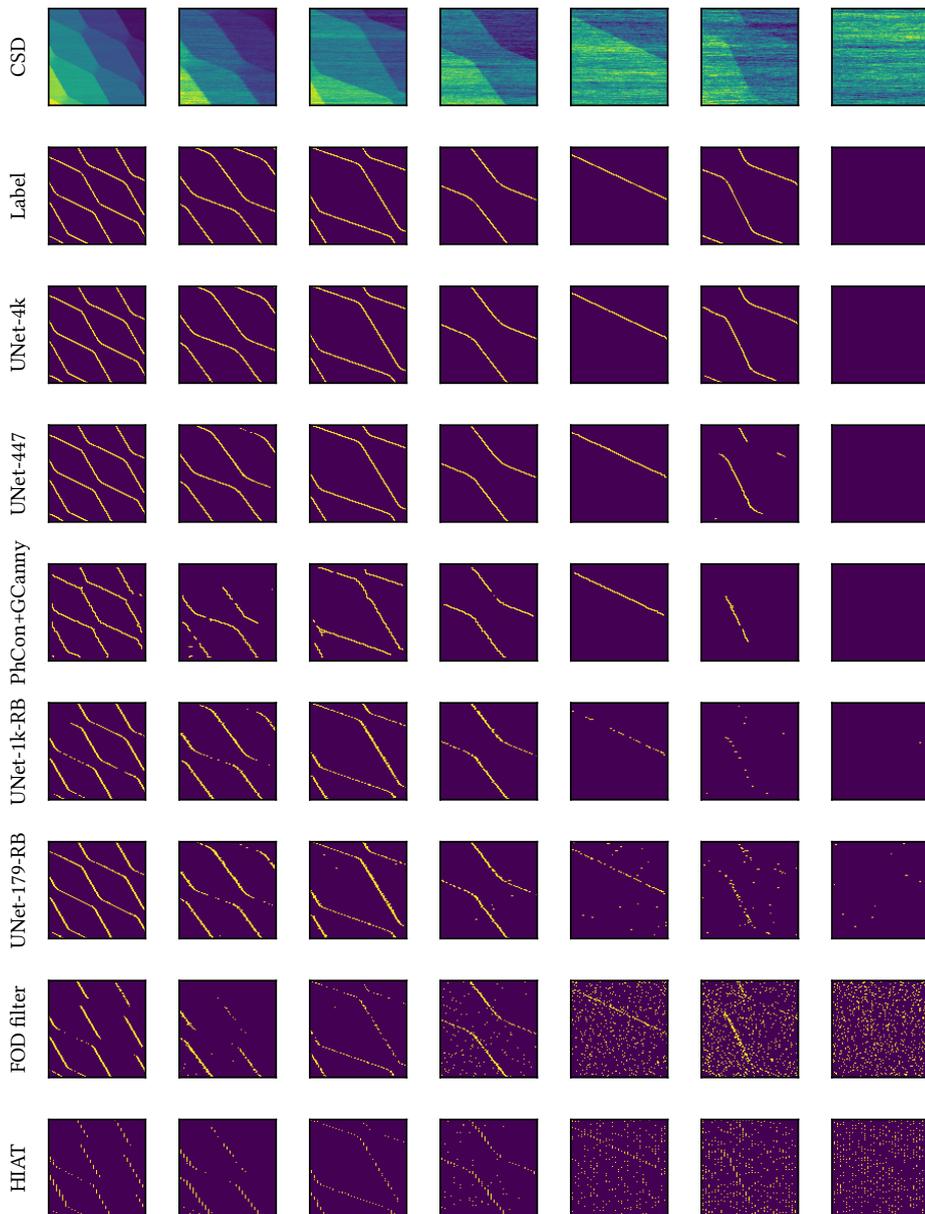


Figure 6.9: Exemplary predictions on the compensated sensor test dataset. The examples are sorted by the observed noise level.

Table 6.3: Metrics calculated for all approaches on the compensated sensor test set with the parameter ranges for the noise and the coupling of the SD to the DQD restricted to the top 10% of the original range. The results are separated into two-dimensional and RB approaches, each sorted according to the S-DICE score.

Category	Detector Name	DICE	S-DICE
2D	U-Net	0.9935	0.9999
	UNet-38k	0.9906	0.9998
	UNet-4k	0.9805	0.9979
	UNet-447	0.9558	0.9884
	PhCon+GCanny	0.4886	0.9160
RB	UNet-1k-RB	0.8662	0.9926
	UNet-179-RB	0.7571	0.8971
	FOD filter	0.4429	0.8914
	HIAT	0.3256	0.8861

Examples for the predictions on the dataset with superior measurement quality are illustrated in Figure 6.10. Notably, the two-dimensional U-Net variants show excellent results. Only UNet-447 exhibits a failure to detect negligible components of a CT in the upper right corner of the first CSD. Even with superior measurement quality, PhCon+GCanny consistently exhibits deficiencies in detecting CTs within specific CSDs.

A comparative analysis of the RB approaches reveals that UNet-1k-RB and UNet-179-RB both demonstrate excellent results that are qualitatively equivalent to those of the two-dimensional variants. Nevertheless, random false detections still occur in UNet-179-RB, albeit with markedly reduced incidence. Both networks visually showed superior results than those of the two-dimensional PhCon+GCanny. It is evident that HIAT continues to demonstrate the dependence of the detection probability on the relative direction of the measurement direction in relation to the CT direction. Furthermore, even at lower noise levels, false detections still occurred, although not as frequently as in the initial investigation. The results of the FOD filter again were consistent with those of HIAT, yet the FOD filter demonstrated a higher rate of CT misses. However, it is noteworthy that the examples hardly showed random false detections for the FOD filter.

6.3.4 Suggested Detection Methods

The investigation has demonstrated, that the SD compensation has the potential to markedly diminish the complexity of the analysis. Moreover, the efficacy of the two-dimensional analysis has been shown to be significantly superior. In the first investigation, employing identical noise levels and SD sensitivity as those utilized in Chapter 5, solely two-dimensional approaches yielded sufficiently robust results that satisfied the stipulated requirement of an S-DICE score of 0.95. In the course of the second investigation, with the additional assumption of a superior data quality, the RB U-Net variant UNet-1k-RB also fulfilled this requirement. However, it is noteworthy that a two-dimensional variant of even smaller scale has achieved this objective, necessitating less computational effort to attain a comparable outcome.

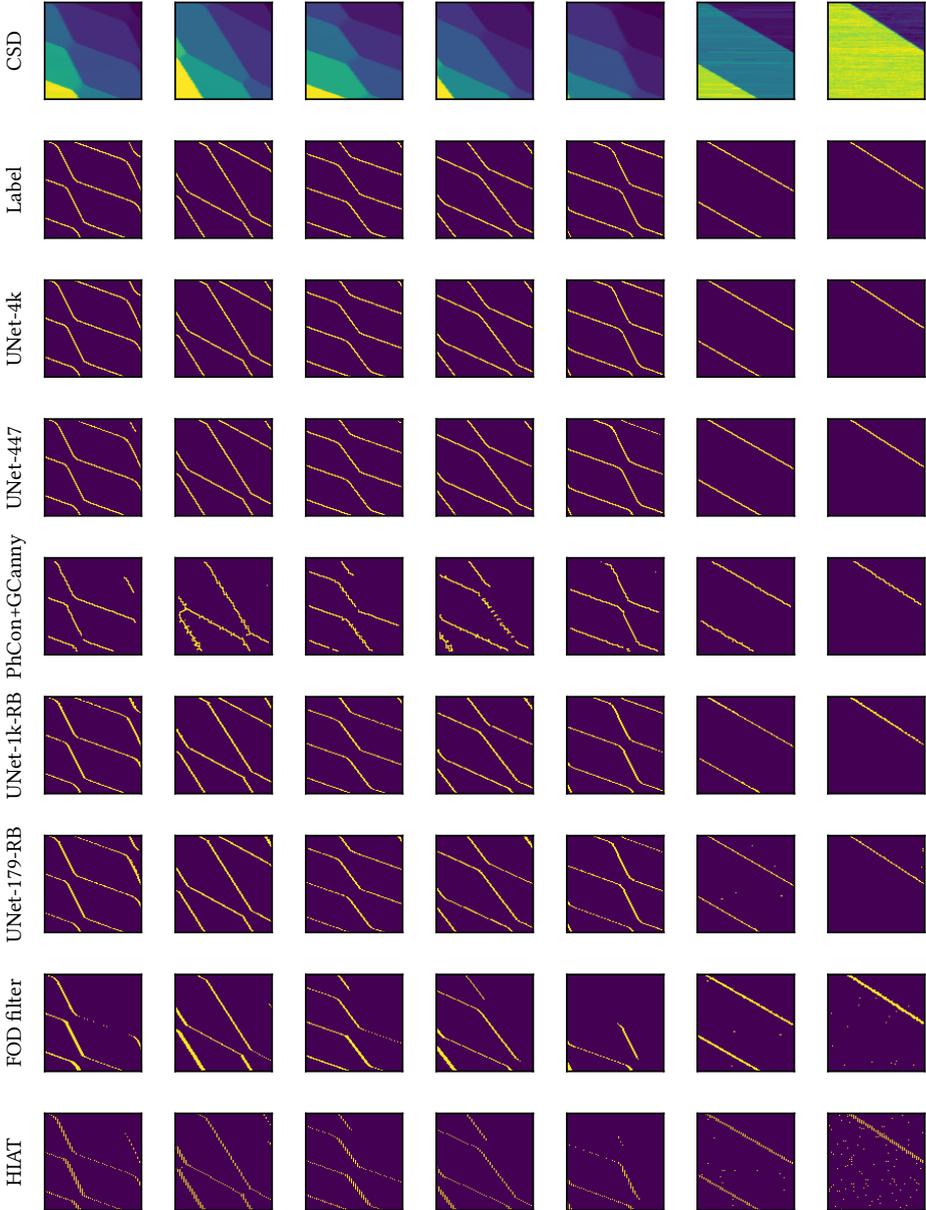


Figure 6.10: Exemplary predictions on the compensated sensor test dataset with the parameter ranges for the noise and the coupling of the SD to the DQD being restricted to the top 10% of the original range. The examples are sorted by the amplitude of the observed CTs.

Furthermore, the classical two-dimensional **PhCon+GCanny**, the one-dimensional **FOD** filter, and the proposed candidate for a hardware integrated averaging and thresholding (**HIAT**) did not attain the stipulated **S-DICE** score. This finding serves to underscore the effectiveness of **ML**-based techniques for the task at hand. Nonetheless, a robust application of **HIAT** could be achieved, if the data quality is further optimized and two different averaging times are used in conjunction. The latter would allow for flexibility with regard to the relation between the direction of the measurement and the **CT**.

In summary, the combination of **SD** compensation and the consistent achievement of the highest data quality of contemporary measurements enables **RB ML** approaches like **UNet-1k-RB** to successfully detect **CTs**. If the superior data quality cannot be consistently attained, the implementation of the two-dimensional **UNet-447** is recommended. Should future **qubit** measurements attain optimal data quality excelling the superior quality considered here, the prospect of achieving a completely hardware-integrated robust detection becomes a distinct possibility.

6.4 Energy Efficiency of Scaled Machine Learning Models

In the domain of **QC**, where control and readout electronics are required to function within the extreme constraints of cryogenic environments when scaling up to universal quantum computation, the electrical power consumption of **ML** models constitutes a pivotal consideration. In dilution refrigerators, where quantum processors function at millikelvin temperatures, power dissipation leads to heating that might critically degrade **qubit** coherence and fidelity. This underscores the necessity of efficient co-processing, particularly for tasks such as **qubit** state discrimination, error correction, and calibration.

The energy efficiency of a model is closely tied to its size, and the model size can be reduced based on the assumptions about the quality of the measurement data made in the previous sections. **Table 6.4** depicts the resulting **FLOPs** for the different network sizes. The number of **FLOPs** is directly proportional to the size and the number of **CSDs** that are analyzed concurrently. Consequently, the energy efficiencies that are calculated in the following can be extrapolated to higher numbers of **qubits**. Furthermore, it is hypothesized that the resolution of the **CSDs** can be further reduced, as the size was initially chosen to closely match the experimental data.

Specialized hardware accelerators [355], including **ML** application-specific integrated circuits (**ASICs**), have exhibited energy efficiencies reaching up to 50.62 tera operations per second (**TOPS**) per watt for U-Net architectures employing TSMC's 40 nm **CMOS** technology [356]. As asserted by Teo et al. [356], the implementation of pipelining within each processing element effectively reduces the required chip area while enhancing the computational efficiency. Earlier approaches made use of techniques such as dynamic-voltage-accuracy-frequency scaling implemented in a 28 nm fully depleted silicon on insulator (**FD-SOI**) technology to improve the efficiency to 10 **TOPS** per watt [357].

In order to maximize computational efficiency, contemporary hardware accelerators frequent-

Table 6.4: FLOPs, MACs, and an estimation for the required power to sustain a throughput of one CSD (96×96 pixels) inference per second for all previously discussed U-Net models. The estimation is calculated using Eq. (6.7).

Category	Model	Parameters	FLOPs	MACs	Power [W]
2D	U-Net	17,261,825	$1.124 \cdot 10^{10}$	$5.612 \cdot 10^9$	$2.220 \cdot 10^{-4}$
	UNet-38k	38,041	$7.701 \cdot 10^7$	$3.788 \cdot 10^7$	$1.521 \cdot 10^{-6}$
	UNet-4k	4,193	$2.404 \cdot 10^7$	$1.165 \cdot 10^7$	$4.749 \cdot 10^{-7}$
	UNet-447	447	$4.471 \cdot 10^6$	$2.083 \cdot 10^6$	$8.832 \cdot 10^{-8}$
RB	UNet-1k-RB	1,481	$1.374 \cdot 10^7$	$6.341 \cdot 10^6$	$2.715 \cdot 10^{-7}$
	UNet-179-RB	179	$2.207 \cdot 10^6$	$9.216 \cdot 10^5$	$4.360 \cdot 10^{-8}$

ly leverage quantized neural networks. In these networks both parameters and activations are encoded using reduced-precision integer representations, often comprising only a few bits. The quantization is achieved through the implementation of either post-training quantization (PTQ) or quantization-aware training (QAT). In the case of PTQ, the initial training phase involves a floating-point network. This is followed by the quantization of weights and activations. Conversely, in the QAT approach, the model undergoes training in a manner that is specifically designed to be quantization-aware. The latter option necessitates a greater investment of engineering resources, as it involves adapting the training pipeline. Nevertheless, this approach has the potential to yield superior outcomes. In most cases, the accuracy of a network quantized to 8-bit integers using PTQ does not decline by more than 1% compared to the floating-point variant [358]. The utilization of a properly configured QAT pipeline has been demonstrated to mitigate accuracy loss to a considerable extent, while maintaining performance even under aggressive 4-bit quantization [358]. In certain instances, quantized networks have been shown to outperform their floating-point counterparts [358–360]. Initial experimentation with the proposed U-Net models revealed no substantial decline in accuracy with QAT, when employing quantization to 8-bit integer representations. The hypothesis posits that the precision of integer representation can be further reduced.

The subsequent calculations are based on the U-Net accelerator efficiency

$$E_{unet} = \frac{N_{ops}}{P \cdot t} = 50.62 \left[\frac{TOPS}{W} \right] \quad (6.5)$$

reported by Teo et al. [356], where N_{ops} is the number of operations, P is the power, t is the time, and TOPS are the tera operations per second. For the power consumption analysis of the CT detection, the number of operations is given by

$$N_{ops} = N_{ops/csd} \cdot n_{csd}, \quad (6.6)$$

with $N_{ops/csd}$ being the number of operations per CSD and n_{csd} the number of CSDs. Consequently, the required power for a tuning procedure with a time requirement t_{tune} can be calculated as

$$P_{tune} = \frac{N_{ops/csd} \cdot n_{csd}}{E_{unet} \cdot t_{tune}}. \quad (6.7)$$

Table 6.5: Required power for the local CT detection in the cryostat to achieve a sufficient throughput to enable the automated tuning of one million qubits in six hours calculated using Eq. (6.7). The RB approaches are expected to require six rays instead of a two-dimensional CSD.

Category	Model	Power [W]
2D	U-Net	$1.028 \cdot 10^{-1}$
	UNet-38k	$7.043 \cdot 10^{-4}$
	UNet-4k	$2.199 \cdot 10^{-4}$
	UNet-447	$4.089 \cdot 10^{-5}$
RB	UNet-1k-RB	$7.855 \cdot 10^{-6}$
	UNet-179-RB	$1.262 \cdot 10^{-6}$

Table 6.4 provides the resulting power associated with the throughput of one CSD inference per second. It is noteworthy that UNet-447 necessitates approximately 88.3 nW, while UNet-1k-RB requires 271.5 nW in the case of 96×96 pixel CSDs or 2.8 nW in the case of 96 pixel RB measurements.

A conservative assessment indicates that the charge state tuning procedure requires a maximum of 10 CSD measurements per qubit. In addition, the results in the study of Zwolak et al. [89] indicated that six rays are sufficient to replace a full two-dimensional CSD. Thus, for a million qubits, at most 10 million CSD- or 60 million RB-measurements are required. As charge state tuning is confined to the initial setup phase, its time constraints are less stringent than those of subsequent tuning procedures which must be repeated periodically. Therefore, a duration of six hours is considered to be an acceptable time for a system of one million qubits, particularly given that drifts of a full charge carrier in a QD take longer than a day and only require slight re-tuning. By reducing the data to CT information, the volume becomes sufficiently compact to enable rapid transmission to room-temperature control hardware. On this hardware, specialized algorithms can autonomously compute the requisite voltage adjustments for each qubit. Leveraging a pipelined architecture, this optimization step can be executed in a concurrent manner to the acquisition of data from subsequent qubits in the cryostat. Therefore, this approach serves to eliminate latency and ensure the continuity of high-throughput system calibration.

The resulting power requirements calculated under the aforementioned assumptions are delineated in Table 6.5. With just $7.86 \mu\text{W}$, UNet-1k-RB is a promising candidate for an integration in the 100 mK stage, as contemporary high-end cryostats like the Bluefors XLDsl offer a cooling power of approximately 1 mW [361]. This assertion is equally applicable to UNet-447, which necessitates $40.89 \mu\text{W}$. In the event that other control electronics require augmented cooling power, it is possible to relocate the CT detection to the 4 K stage, which offers more cooling power.

The previously mentioned benchmarks indicate that tiny ML models have the potential to operate effectively in cryogenic environments, thereby establishing a foundation for the development of scalable quantum-classical hybrid systems.

6.5 Unified Sensor and CSD Simulation Framework

The extensive development and testing of algorithms for the aforementioned SD tuning steps and the investigation of an effectual compensation require a substantial amount of labeled data. Therefore, a replicative model for simulating barrier-barrier sensor scans (see Section 3.4 for an introduction to these scans) has been developed. This model was developed to extend the SD simulation in SimCATS and propel future research endeavors. The realization of the model by Papajewski [362] mimics the features of sensor scans phenomenologically. Utilizing a replicative model offers the advantage, that it does not necessitate the development of a comprehensive physical model, which can be more intricate to calculate and might not be feasible for all the phenomena evident in sensor scans.

In the simplified simulation, the SD is modeled using three resistors in series. The SD parameter of interest is the conductivity across the series of resistors. If the SD is measured using RF reflectometry, the measured value is proportional to the conductance.

The simulation consists of four steps:

1. calculation of the potential of the SD μ_{SD} and the SD-barriers μ_{B_i} , $i = 1, 2$,
2. distortion of the potentials,
3. application of the sensor and barrier functions and calculation of the total sensor response, and
4. distortion of the sensor response.

Figure 6.11 provides a visual representation of the lever arms of the two barrier voltages V_{B_1} and V_{B_2} on the potentials of the SD system. These lever arms are instrumental in calculating the potentials

$$\begin{pmatrix} \mu_{B_1} & \mu_{SD} & \mu_{B_2} \end{pmatrix} = \begin{pmatrix} V_{B_1} & V_{B_2} \end{pmatrix} \cdot \begin{pmatrix} \alpha_{V_{B_1}, \mu_{B_1}} & \alpha_{V_{B_1}, \mu_{SD}} & \alpha_{V_{B_1}, \mu_{B_2}} \\ \alpha_{V_{B_2}, \mu_{B_1}} & \alpha_{V_{B_2}, \mu_{SD}} & \alpha_{V_{B_2}, \mu_{B_2}} \end{pmatrix}. \quad (6.8)$$

The next step in the process involves the application of the sensor and barrier functions to the potentials, which have previously undergone distortion by RTN and pink noise. The sensor function delineates the Coulomb peak behavior of the SD, while the barrier functions model the pinch-off curves of the corresponding barrier gate. Simplified Lorentzians [243] in superposition represent the Coulomb oscillations in the sensor function. A final sigmoid term is overlaid to model the transition to direct current flow as tunnel barriers vanish. The barrier functions are modeled by generalized logistic functions (GLFs)³ [363]:

$$glf(\mu_{B_i}) = a_l + \frac{a_r - a_l}{(c + \epsilon \cdot e^{-\gamma \cdot (\mu_{B_i} - \mu_{off})})^{\frac{1}{r}}}. \quad (6.9)$$

³GLFs extend sigmoid functions and offer the capability to approximate possible asymmetry in the barrier functions.

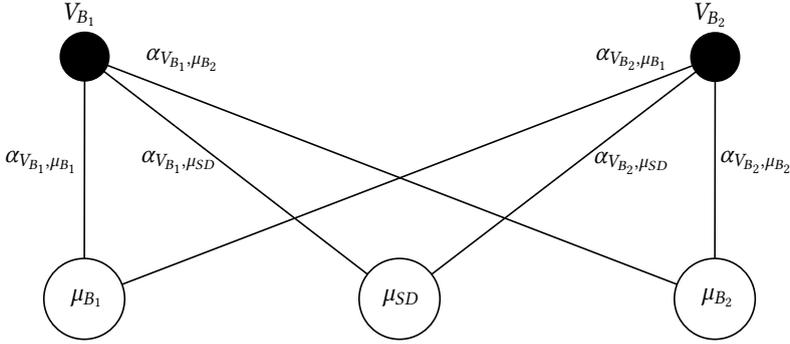


Figure 6.11: Lever arms of the voltages V_{B_1} and V_{B_2} applied to the two SD barrier gates B_1 and B_2 (SL_i and SR_i in Figure 3.1) on the potential of the SD μ_{SD} and its barriers μ_{B_i} , $i = 1, 2$.

Herein,

- a_l is the left horizontal asymptote that indicates the lower limit,
- a_r is the right horizontal asymptote that indicates the maximum of the function (if $c = 1$),
- ϵ is linked to the initial value of the function at $\mu_{B_i} = \mu_{off}$ and influences the early growth⁴,
- γ defines the general growth rate,
- r influences the range in which maximum growth occurs, and
- c is set to 1, which simplifies the function and reduces the number of parameters in the present case.

Figure 6.12 illustrates GLFs fitted to a pinch-off measurement.

Subsequently, the results of the sensor function and the barrier functions are combined to calculate the total sensor response, which is proportional to the conductivity G_{total} across the series of resistors

$$G_{total} = \frac{G_{B_1} \cdot G_{SD} \cdot G_{B_2}}{(G_{B_2} \cdot G_{SD}) + (G_{B_1} \cdot G_{B_2}) + (G_{B_1} \cdot G_{SD})}. \quad (6.10)$$

Afterward, the sensor response is distorted with white noise, which originates from amplifications of the sensor signal. Figure 6.13 illustrates the process of constructing ideal SD data. The interplay of the barriers with the SD itself leads to the characteristic fading out of the Coulomb peaks with closing barriers (see Figure 6.13j).

Using this model it is possible to develop and especially test further sensor tuning approaches. Additionally it enhances the CSD simulation, as the model is an implementation of the sensor interface in the SimCATS framework. Therefore, it is possible to iterate between simulating CSDs and sensor scans, e.g. to continuously improve the sensor while moving through the

⁴Higher values lead to a faster growth in the beginning.

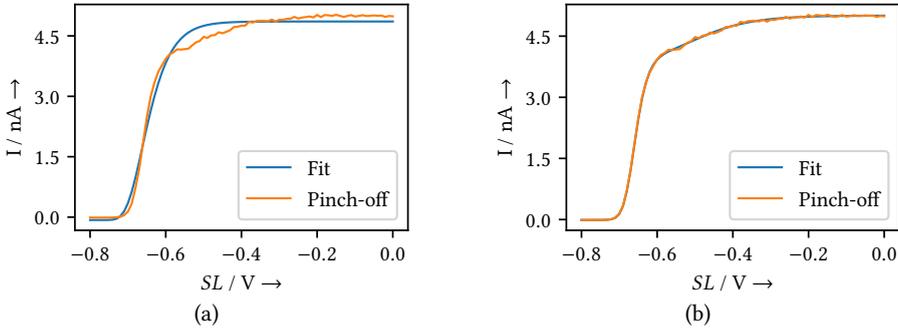


Figure 6.12: Experimental pinch-off measurement with corresponding GLF fits. (a) Single GLF fit. The initial rate of growth exhibited a greater magnitude than the subsequent rate of growth, thereby substantiating the efficacy of the GLF in modeling the asymmetry associated with the barrier pinch-off. (b) Multi GLF fit, consisting of two GLFs. The model demonstrates an enhanced capacity to simulate the pinch-off process with greater precision by employing a combination of multiple GLFs.

CSD space.

The extraction of suitable simulation parameters and the evaluation of the generated data is depicted in the work of Papajewski [362].

6.6 Outcome and Perspective

The investigations regarding the sensitivity and efficiency refinement with improved SD tuning have demonstrated that the most crucial factor for the CT analysis is an appropriate SD compensation and reliably achieving superior data quality. In such conditions, a scalable CT detection can be integrated into the cryostat adjacent to the qubits, potentially at the 100 mK stage. This close integration reduces wiring to minimize thermal conductance between temperature stages, shortens the signal path to mitigate noise, and enables faster feedback loops due to fewer parasitic elements. Table 6.6 provides a final assessment of the suitability of the various CT approaches for different data quality. The evaluation encompasses the S-DICE score, the visually assessed detection quality, and the requisite power for a scalable million qubit system.

In instances where a reliable SD compensation is not applied, UNet-38k emerges as the sole contender that allows an integrated solution, albeit not at the 100 mK stage. In the context of a compensated SD, UNet-447 emerges as the preferred solution, which can be integrated at the 100 mK stage.

Despite the potential for utilization with SD compensation at contemporary data quality, RB

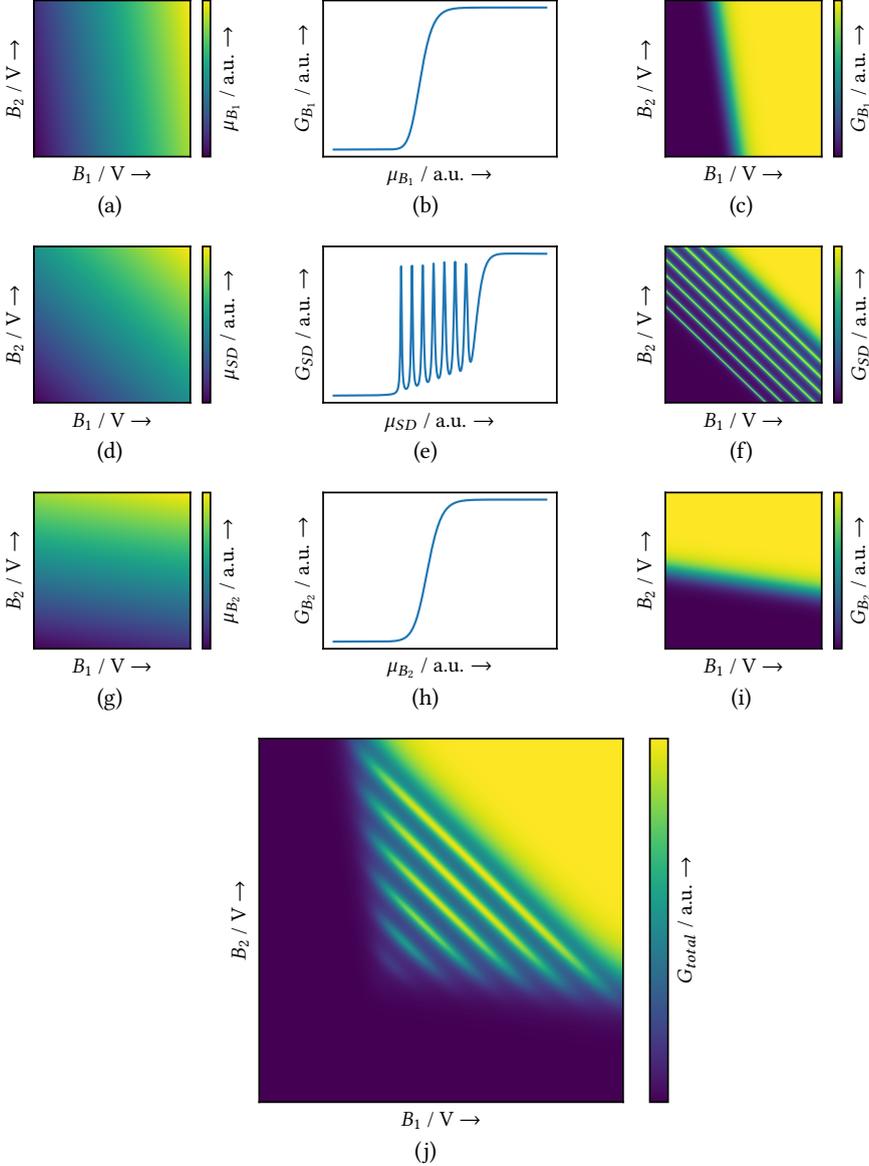


Figure 6.13: Visualization of the calculation of G_{total} . (a) The potential μ_{B_1} , to which the barrier function shown in (b) is applied, resulting in (c) the conductivity G_{B_1} . (d) The potential μ_{SD} , to which the sensor function shown in (e) is applied, resulting in (f) the conductivity G_{SD} . (g) The potential μ_{B_2} , to which the barrier function shown in (h) is applied, resulting in (i) the conductivity G_{B_2} . (j) G_{total} , calculated via Eq. (6.10).

Table 6.6: Overview of approaches and suitability for different scenarios. The validation is based on the *S-DICE* scores, the visual assessment, and the power calculations. The colors visualize the suitability, with yellow meaning possible but restricted, e.g. not on the 100 mK but only on the 4K stage.

Category	Model	Power	Data Quality		
			uncompensated, contemporary	compensated <i>SD</i>	compensated <i>SD</i> , superior data quality
2D	U-Net				
	UNet-38k				
	UNet-4k				
	UNet-447				
	PhCon+GCanny				
RB	UNet-1k-RB				
	UNet-179-RB				
	FOD filter				
	HIAT				

approaches are only rendered reliable when data quality is consistently superior⁵. This once again corroborates the hypothesis that two-dimensional analysis is undoubtedly easier, as even larger *RB ML* networks did not achieve a competitive result compared to their smaller two-dimensional competitors. However, when considering the case of *SD* compensation in combination with superior data quality, the UNet-1k-*RB* approach emerges as the preferred option.

When extending the *HIAT* approach to utilize two different averaging times in conjunction and further optimizing the data quality (see Section 6.3.4), this candidate could also potentially perform reliably and offer optimal energy efficiency. However, it should be noted that this approach offers extremely limited flexibility to respond dynamically to different conditions.

Furthermore, the extension of *SimCATS* described in Section 6.5 enables the comprehensive investigation, development, and benchmarking of automated *SD* tuning algorithms. Additionally, this allows the simulation of the sequential switching between *SD* and *DQD* tuning steps. For instance, this can be employed to reassure highest *SD* sensitivity after several *CSD* measurements. Thus, a more comprehensive algorithm development becomes feasible.

⁵Consistently achieving the quality of the best measurements from contemporary *qubit* experiments.

Chapter 7

Conclusion and Outlook

Conclusion

THIS thesis initially provided a comprehensive survey of the fundamentals of QC, with a particular emphasis on the implementation of spin qubits in semiconductor QDs. Subsequently as its main subject, the present work addressed the challenge of tuning QD systems to achieve high-fidelity qubit operation. The primary focus was on the complexity of developing a scalable and automated charge state tuning. At the core of this discourse lies the pivotal role of data analysis, particularly in the context of noise characterization and pattern recognition, as a foundational element of any viable tuning strategy. A thorough investigation was conducted into both conventional algorithms and modern deep learning approaches.

Crucially, the work underscored that scalable tuning architectures demand tight integration with the quantum hardware itself. This integration is imperative due to the projected impossibility of transferring complete measurement datasets to room temperature for processing on a large scale.

A discussion and modeling of the general tuning procedure was conducted, which resulted in the identification of the most crucial challenges to achieving a scalable system. The primary challenges pertain to the establishment and retention of optimal SD sensitivity, as well as the tuning of the QD charge state. In this study, the central focus was on the charge state tuning based on CSD measurements, which is considered the more demanding task.

The development and evaluation of robust tuning algorithms is critically contingent upon access to an substantial amount of labeled datasets. However, the process of manually annotating experimental data is inherently laborious. Moreover, experimental conditions, such as noise levels, cannot be systematically controlled to probe algorithmic limits. In order to surmount the aforementioned constraints, the present work introduces a simulation-based approach that facilitates comprehensive benchmarking of CT detection strategies. Specifically, the open-source framework SimCATS was developed to generate realistic CSD data under a wide range of conditions. SimCATS integrates a novel modular and extensible distortion model, applying distortions at three distinct, experimentally motivated stages: (1) the physical charge occupation of the DQD, (2) the electrostatic potential of the SD, and (3) the final measurement response of the sensor. The model demonstrates a high degree of similarity with the actual measurement pipeline. Coupled with a flexible, high-performance geometric

model for the charge occupation, [SimCATS](#) enables the efficient generation of realistic [CSD](#) datasets. The empirical realism of the simulated data was quantitatively validated using metrics commonly applied to generative [ML](#) models. Experimental measurements served as the reference point for this validation.

A comprehensive evaluation of [CT](#) detection strategies was conducted using a large simulated dataset. This dataset was generated through randomized parameter variations within experimentally derived ranges from a GaAs-based system. The study encompassed five classical algorithms and 24 [ML](#)-based approaches, including convolutional, transformer, state-space, and diffusion models.

Following the adaptation to the task and the parameter optimization, all candidates were first benchmarked on simulated test sets and subsequently evaluated on two manually labeled experimental datasets. One of these datasets was acquired from a GaAs [DQD](#) and the other was acquired from a SiGe single [QD](#). The employment of classical methods has demonstrated an inability to achieve reliable detection in the presence of conventional distortion levels and sensor sensitivities. In contrast, several [ML](#) models exhibited robust performance on simulated data, with numerous models maintaining high detection accuracy when applied to experimental GaAs data, from which the simulation parameters were derived. A decline in performance was evident when applying the models to the SiGe dataset, which is devoid of the honeycomb structure that is a hallmark of [DQD](#) systems. However, the detection quality maintained a comparatively high level, indicating the high capacity to adapt to disparate feature distributions with a moderate refinement of the model weights utilizing target-specific experimental data. Diffusion-based networks were excluded from consideration for scalable deployment among the tested architectures due to their excessive computational overhead and lack of a clear performance advantage.

Furthermore, an exploratory analysis of network compression was conducted using the U-Net architecture, a fully convolutional model that is particularly well-suited for image segmentation tasks. The selection of this architecture was made on the basis of the inherent suitability of convolutional networks for efficient hardware implementation, particularly in contexts characterized by resource constraints or the requirement for embedded environments. The resulting UNet-38k variant demonstrated a high level of accuracy while concurrently exhibiting a substantial reduction in parameter count, reaching a mere 0.22% of the original value. This underscores the potential of U-Net for future applications in domains requiring low power dissipation or compact on-chip integration.

Subsequently, an investigation was conducted to determine whether further reduction in complexity of the [CT](#) detection could be achieved. The rationale for this investigation stems from the observation that, while UNet-38k demonstrates commendable performance, its size still presents a challenge to its scalability for implementation alongside the [qubit](#) sample. Earlier attempts at more aggressive model compression resulted in a rapid deterioration of detection accuracy, underscoring the necessity to enhance signal quality. In this context, achieving high-quality sensor performance – particularly in terms of high sensitivity and effective compensation – emerged as a central factor. In order to enhance sensitivity, it is necessary to

automate the tuning of the **SD** in a robust manner. This process must be given particular emphasis in the noise-sensitive selection of a stable Coulomb peak flank. Concurrently, sensor compensation has been shown to induce a fundamental alteration in the signal characteristics that is hypothesized to substantially simplify **CT** detection.

A comprehensive evaluation was conducted to determine the impact of sensor compensation on **CT** detection performance and the feasibility of further model compression. In addition to **ML**-based approaches, the most effective classical method from prior studies was re-evaluated. The study also incorporated the analysis of **RB** variants, employing U-Net models, a one-dimensional **FOD** filter, and a hardware integrated averaging and thresholding (**HIAT**) approach. The objective of this extension was to determine whether the compensation enables a reliable **RB** analysis. Using simulated compensated data generated with the **SimCATS** framework, U-Net architectures were systematically reduced in size until the performance on the validation datasets was insufficient. Two compressed variants – one and two orders of magnitude smaller than UNet-38k – were selected for final evaluation, including their corresponding **RB** variants.

The final comparison confirmed that sensor compensation significantly reduces the analytical complexity of the detection task. This was demonstrated by the fact that even the smallest two-dimensional U-Net variant achieved the target accuracy determined from the previous investigations. Conversely, both the classical approach and the **RB** approaches fell short of the desired level of accuracy. This finding once again underscores the notion that **RB** analysis is undoubtedly more challenging.

In order to further explore the role of data quality, an additional investigation restricted the simulation parameters to the superior 10% of the original distortion and sensitivity range. The objective of this investigation is to ascertain whether the **RB** analysis, particularly the proposed **HIAT**, attains reliability if, in addition to **SD** compensation, the measurements consistently attain the highest quality observed in contemporary measurements. Under these conditions, a small **RB** U-Net variant attains the necessary level of accuracy, while the **FOD** filter and **HIAT** continue to exhibit deficiencies. Nevertheless, possible improvements to **HIAT** were proposed, which might enable its reliable utilization, contingent upon the concomitant enhancement toward optimal data quality.

Subsequent to these investigations, the required power of the various U-Net variants in the context of a scalable tuning for a one million **qubit** system was estimated. These estimations were based on the efficiency metrics of dedicated U-Net hardware accelerators, as reported in recent studies. The results of the study indicated that a scalable implementation at the 100 mK stage is indeed feasible, provided that adequate compensation is applied to the **SD**.

In order to further propel the future development of **SD** tuning algorithms, a more sophisticated sensor simulation was proposed, which extends the existing **SimCATS** framework.

Outlook

FUTURE research endeavors should concentrate on determining the acceptable data transmission rates from cryogenic environments to room temperature. This encompasses the physical realization of the data links and the associated thermal budget, particularly the impact of heat dissipation on system stability. The transmission rates will be a decisive factor regarding the proportion of analysis and data compression tasks which should be performed in the cryostat to reduce the required bandwidth. Ultimately this enables a quantitative comparison of the trade-offs between local processing accuracy and power requirements versus the demands and implications of high-bandwidth data transmission.

Furthermore, the deployment of a hardware demonstrator for the most promising **ML**-based recognition models would facilitate the acquisition of critical insights into real-world performance and thermal behavior under cryogenic conditions.

In this regard, neuromorphic computing (**NC**) architectures present a compelling alternative, offering the potential for ultra-low-power implementations. **NC** is a computational paradigm that emulates the architecture and temporal dynamics of biological neural systems to enable energy-efficient, event-driven information processing. Especially promising are hardware-level implementations such as memristor-based crossbar arrays, which offer high integration density and energy-efficient operation [364].

It is imperative that research on additional required recognition tasks prioritizes the identification of versatile approaches with the potential for wide-ranging applicability to a multitude of tasks. To illustrate, one set of tiny U-Net model weights could be trained for **SD** tuning, while an alternative set could be trained for **QD** charge state tuning. Consequently, in scenarios where targeted tasks are performed sequentially, the specialized hardware that efficiently implements the network can be reused. With this approach only the weights of the network would have to be updated between the operations. Depending on the implementation, the storage of different versions of the weights in the hardware implementation could facilitate rapid switching and reduce the power budget required for loading parameters into the memory. Therefore, the research endeavors could employ techniques such as **NAS** [365, 366] and **HPO** [367] to find the optimal versatile network for a wide variety of tasks.

Another focal point for future research endeavors should be the establishment of a robust **SD** compensation and the maintenance of a high degree of sensitivity through the implementation of optimal automated **SD** tuning. This is considered to play a pivotal role in enabling low-complexity analysis for all measurements subsequent to tuning up the **SD**.

Advancement in this domain necessitates the establishment of high-fidelity electromagnetic field simulations of **qubit** samples to precisely ascertain the lever arms of the various gate electrodes on the potential landscape of the **QD** system. Complementary algorithmic frameworks should be developed to validate and refine these simulated parameters through the analysis of experimental data. The statistics obtained from these algorithms will also facilitate the

enhancement of the simulations, thereby potentially obviating the necessity for subsequent corrections.

Finally, dedicated hardware could be used to translate virtual gate operations into optimized combinations of physical gate voltages in real time.

Another significant topic for future research is the comprehensive advancement of high-level tuning algorithms, which will likely remain confined to room-temperature infrastructure. Given the previously delineated data and measurement reduction techniques that facilitate a transfer to room temperature but entail information loss, the decision-making process for tuning progressively intricate QD architectures will necessitate enhanced algorithmic effort. In order to address this issue, it is necessary to employ high-level calibration strategies that will likely require the integration of advanced ML models. These models have already demonstrated their capacity to manage complex, high-dimensional tasks with exceptional performance.

Appendix **A**

Basic Condensed Training Routine for Machine Learning Networks

The following presents a basic, condensed training routine for ML networks. The reproduction of the training requires (1) a functional Python installation, (2) the reference U-Net implementation of Milesi [305], and (3) the auxiliary libraries PyTorch [347] (tested with version 2.7.0), and SimCATS-Datasets [295] (tested with version 2.5.0).

Listing A.1 adapts Milesi's original training loop to train a U-Net model on SimCATS CSDs. Prior to execution, amend line 7 so that it points to the local clone of the Pytorch-UNet repository [305]. Furthermore, a training dataset must be generated in advance. This can be done using the notebook `dataset_gen[random_variations].ipynb`, which is included with the SimCATS-Datasets package [295]. The path to the dataset must be provided in line 22.

```
1 from pathlib import Path
2 from simcats_datasets.loading.pytorch import SimcatsDataset
3 import torch
4 from torch.utils.data import DataLoader
5 from tqdm import tqdm
6 import sys
7 sys.path.append("./Pytorch-UNet")
8 from utils.dice_score import dice_loss
9 from unet.unet_model import UNet
10
11 model = UNet(n_channels=1, n_classes=1, bilinear=True)
12 # define training hyperparameters
13 learning_rate = 0.2
14 weight_decay = 0.0001
15 batch_size = 64
16 epochs = 5
17 device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
18 # create directory to store network checkpoints
19 dir_checkpoints = Path("./checkpoints")
20 dir_checkpoints.mkdir(parents=True, exist_ok=True)
21 # setup the dataset and dataloader
22 dataset = SimcatsDataset(h5_path="dataset.h5",
23                          load_ground_truth="load_tct_masks",
24                          data_preprocessors=["standardization", "add_newaxis"],
25                          ground_truth_preprocessors=["only_two_classes"],
26                          preload=True)
```

```

27 dataloader = DataLoader(dataset, shuffle=True, batch_size=batch_size, pin_memory=True)
28 # set the model to the efficient channel_last memory format
29 model = model.to(memory_format=torch.channels_last)
30 # create and configure the optimizer and the scheduler
31 optimizer = torch.optim.AdamW(lr=learning_rate, weight_decay=weight_decay, foreach=True,
    params=model.parameters())
32 scheduler = torch.optim.lr_scheduler.OneCycleLR(optimizer=optimizer, max_lr=learning_rate,
    epochs=epochs, steps_per_epoch=len(dataloader))
33 # define the loss function
34 criterion = torch.nn.BCEWithLogitsLoss()
35 def loss_function(masks_pred: torch.Tensor, true_masks: torch.Tensor) -> torch.Tensor:
36     """Calculates the loss using a combination of BCEWithLogitsLoss and Dice
37     Args:
38         masks_pred (torch.Tensor): Predicted masks
39         true_masks (torch.Tensor): Ground truth masks
40     Returns:
41         torch.Tensor: Loss tensor
42     """
43     loss = criterion(masks_pred.squeeze(1), true_masks.float())
44     loss += dice_loss(torch.nn.functional.sigmoid(masks_pred.squeeze(1)), true_masks.float(),
        multiclass=False)
45     return loss
46 # send the model to the device used for the training
47 model.to(device=device)
48 model.train()
49
50 # actual training loop
51 for epoch in range(1, epochs + 1):
52     epoch_loss = 0
53     with tqdm(total=len(dataset), desc=f"Epoch {epoch}/{epochs}", unit="CSD") as pbar:
54         for batch in dataloader:
55             csds, true_masks = batch["csd"], batch["ground_truth"]
56             # send CSDs and ground truth masks to the device used for the training
57             csds = csds.to(device=device, memory_format=torch.channels_last)
58             true_masks = true_masks.to(device=device)
59             # perform predictions (forward pass) and calculate loss
60             masks_pred = model(csds)
61             loss = loss_function(masks_pred, true_masks)
62             # backward pass and parameter update
63             optimizer.zero_grad(set_to_none=True)
64             loss.backward()
65             torch.nn.utils.clip_grad_norm_(model.parameters(), 1.0)
66             optimizer.step()
67             scheduler.step()
68             # update the progress bar
69             pbar.update(csds.shape[0])
70             epoch_loss += loss.item()
71             pbar.set_postfix(**{"loss (batch)": loss.item()})
72     # save the model weights as checkpoint after each training epoch
73     state_dict = model.state_dict()
74     torch.save(state_dict, Path.joinpath(dir_checkpoints, f"checkpoint_epoch{epoch}.pth"))

```

Listing A.1: Basic condensed training routine for the U-Net model using SimCATS data.

Appendix B

Visual Analysis of Feature Maps and Kernels

Visualizing the learned kernels and the corresponding feature maps produced during inference allows for a more thorough analysis of the network’s internal operations. For particularly simple or compact architectures, this analysis can reveal insights into how the network extracts and processes information. This information can inform the design of more robust classical algorithms. [Listing B.1](#) shows the code used to visualize both the convolutional kernels and the resulting feature maps of the UNet-447 model¹ used in this study. The execution requires the same packages and setup as outlined in [Appendix A](#) and an adjustment of the paths in lines 9 and 47.

[Figure B.1](#) depicts the convolutional kernels of UNet-447 and exemplarily illustrates the feature maps produced when inferring a [CSD](#) with sensor compensation. Each blue block represents a network block, and the white blocks indicate the internal layers (using the names from `torch.nn` [347]). Note that, in the implemented version, each `ReLU` activation is preceded by a batch normalization layer (`BatchNorm2d`), which is not shown in the figure for visual clarity. While this is a common modification in modern U-Net variants, it is not part of the original architecture proposed by Ronneberger et al. [178]. The arrows denote the direction of data flow. The input [CSD](#) first passes through the input block, where two successive convolutions extract low-level features. In the subsequent down block, the image resolution is halved using a max pooling operation, followed by another double convolution to capture features at a coarser scale. The up block processes the output of the down block and the output of the input block. The latter is being forwarded through a skip connection (grey arrow). To align their resolutions, the output of the down block is upsampled via a transposed convolution operator. Then, it is concatenated with the output of the input block (see grey block in [Figure B.1](#)) before undergoing an additional pair of convolutions. Finally, the output block applies a 1×1 convolution that yields the network’s prediction. This prediction is later thresholded to obtain the binary mask.

Inspection of the convolutional kernels and the resulting feature maps reveals that individual kernels specialize in detecting distinct structures. For instance, some respond preferentially to edges of a particular orientation. However, the precise semantic content of each kernel is not always evident and may only become interpretable when evaluated on [CSDs](#) with particular features, such as varying noise levels or specific structures.

¹The implementation of the model is not included here but can be derived by adapting the reference implementation by Milesi [305].

```

1 import torch
2 import math
3 from torchvision import utils
4 import matplotlib.pyplot as plt
5 from simcats_datasets.loading.pytorch import SimcatsDataset
6 from hunet.unet_models import UNet_flexible
7
8 # define path to previously trained checkpoint
9 checkpoint = r"./checkpoints_UNet-447/checkpoint_epoch4.pth",
10 device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
11 # create U-Net model and load trained weights
12 model = UNet_flexible(n_channels=1, n_classes=1, bilinear=False, n_layers=1,
13                       out_channels_inconv=2)
14 model.to(device=device)
15 state_dict = torch.load(checkpoint, map_location=device)
16 model.load_state_dict(state_dict)
17 model.eval()
18
19 # define a helper function to visualize tensors in a grid
20 def visualize_tensor(tensor: torch.Tensor, ch: int = 0, allkernels: bool = False, title: str =
21 None) -> None:
22     """Function to visualize a PyTorch tensor. Usually used for visualizing the filters of a
23     convolutional layer.
24     Args:
25     tensor (torch.Tensor): PyTorch tensor to be visualized
26     ch (int, optional): Index of the channel that will be visualized. Defaults to 0.
27     allkernels (bool, optional): If True visualize all channels. Defaults to False.
28     title (str, optional): Title of the plot. Defaults to None.
29     """
30     n, c, h, w = tensor.shape
31     if allkernels:
32         tensor = tensor.view(n * c, -1, w, h)
33     elif c != 3:
34         tensor = tensor[:, ch, :, :].unsqueeze(dim=1)
35     grid = utils.make_grid(tensor, nrow=4, normalize=True, padding=1, pad_value=1)
36     plt.figure(figsize=(4, math.ceil(tensor.shape[0] // 4) + 1))
37     plt.imshow(grid.cpu().numpy().transpose((1, 2, 0)))
38     if title is not None:
39         plt.title(title)
40     plt.axis("off")
41     plt.ioff()
42     plt.tight_layout()
43     plt.show()
44 # plot kernel filters of each layer
45 for name, param in model.named_parameters():
46     if name[-6:] == "weight" and param.dim() > 1:
47         visualize_tensor(tensor=param, ch=0, allkernels=True, title=name)
48
49 # load a CSD
50 dataset = SimcatsDataset(h5_path="dataset.h5",
51                          load_ground_truth="load_tct_masks",
52                          data_preprocessors=["standardization", "add_newaxis", "add_newaxis"],

```

```

50         ground_truth_preprocessors=["only_two_classes"],
51         preload=False)
52 csd_id = 0
53 csd, true_mask = dataset[csd_id]["csd"], dataset[csd_id]["ground_truth"]
54 # perform the prediction (manual workflow of the "forward" function)
55 csd = csd.to(device=device)
56 output_inc = model.inc(csd)
57 output_down1 = model.down1(output_inc)
58 output_up1 = model.up1(output_down1, output_inc)
59 output_outc = model.outc(output_up1)
60 mask_pred = torch.sigmoid(output_outc) > 0.5
61 # plot feature maps of each block
62 fig, ax = plt.subplots(6, 4, figsize=(12, 18))
63 ax[0, 0].imshow(csd.detach().cpu()[0,0], origin="lower", interpolation="none", aspect="auto")
64 ax[0, 0].set_title(f"Input CSD data {csd_id}")
65 for i in range(output_inc.shape[1]):
66     ax[1, i].imshow(output_inc.detach().cpu()[0, i, :, :], origin="lower", interpolation="none",
67                    aspect="auto")
68     ax[1, i].set_title(f"inc feature map {i+1}")
69 for i in range(output_down1.shape[1]):
70     ax[2, i].imshow(output_down1.detach().cpu()[0, i, :, :], origin="lower", interpolation="none",
71                    aspect="auto")
72     ax[2, i].set_title(f"down1 feature map {i+1}")
73 for i in range(output_up1.shape[1]):
74     ax[3, i].imshow(output_up1.detach().cpu()[0, i, :, :], origin="lower", interpolation="none",
75                    aspect="auto")
76     ax[3, i].set_title(f"up1 feature map {i+1}")
77 for i in range(output_outc.shape[1]):
78     ax[4, i].imshow(output_outc.detach().cpu()[0, i, :, :], origin="lower", interpolation="none",
79                    aspect="auto")
80     ax[4, i].set_title(f"outc feature map {i+1}")
81 ax[5, 0].imshow(mask_pred.detach().cpu()[0,0], origin="lower", interpolation="none", aspect="auto")
82 ax[5, 0].set_title("Output (thresholded)")
83 plt.tight_layout()
84 plt.show()

```

Listing B.1: Python code for the visualization of convolutional kernels and feature maps.

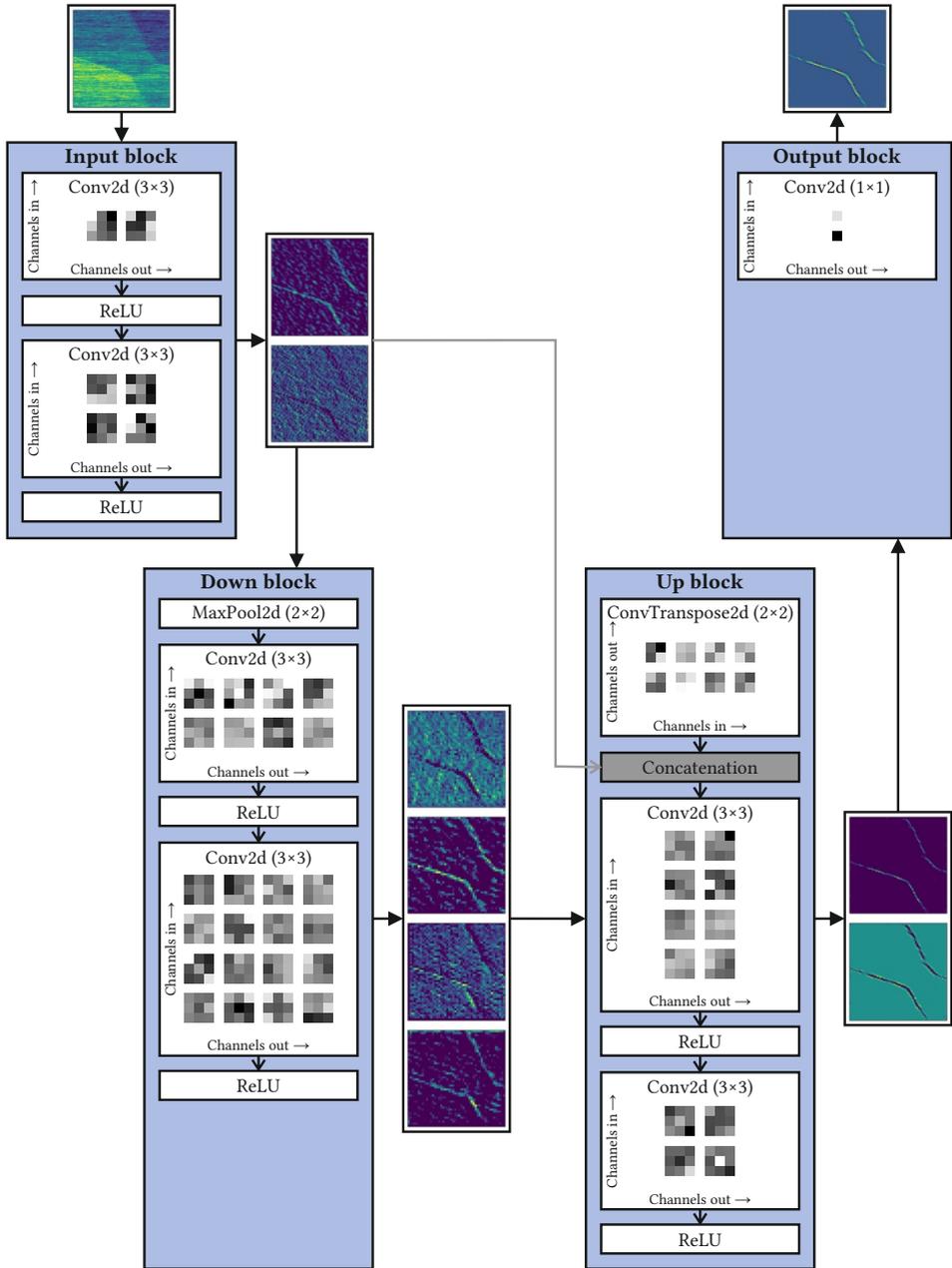


Figure B.1: Visualization of the convolutional kernels of UNet-447 and feature maps produced when inferring a CSD with sensor compensation.

Bibliography

- [1] Max Karl Ernst Ludwig Planck. “Zur Theorie des Gesetzes der Energieverteilung im Normalspectrum”. German. In: *Verhandlungen der Deutschen Physikalischen Gesellschaft* 2 (1900), p. 237.
- [2] A. Einstein. “Über einen die Erzeugung und Verwandlung des Lichtes betreffenden heuristischen Gesichtspunkt”. German. In: *Annalen der Physik* 322.6 (1905), pp. 132–148. ISSN: 1521-3889. DOI: 10.1002/andp.19053220607.
- [3] Niels Bohr. “On the Constitution of Atoms and Molecules, Part I”. In: *Philosophical Magazine* (1913).
- [4] Niels Bohr. “On the Constitution of Atoms and Molecules, Part II Systems Containing Only a Single Nucleus”. In: *Philosophical Magazine* (1913).
- [5] Niels Bohr. “On the Constitution of Atoms and Molecules, Part III Systems containing several nuclei”. In: *Philosophical Magazine* (1913).
- [6] W. Heisenberg. “Über quantentheoretische Umdeutung kinematischer und mechanischer Beziehungen.” German. In: *Zeitschrift für Physik* 33.1 (Dec. 1925), pp. 879–893. ISSN: 0044-3328. DOI: 10.1007/BF01328377.
- [7] Paul Adrien Maurice Dirac and Ralph Howard Fowler. “The quantum theory of the electron”. In: *Proceedings of the Royal Society of London. Series A, Containing Papers of a Mathematical and Physical Character* 117.778 (1928), pp. 610–624. DOI: 10.1098/rspa.1928.0023.
- [8] Erwin Schrödinger. “Quantisierung als Eigenwertproblem (Erste Mitteilung)”. In: *Annalen der Physik* (1926).
- [9] Erwin Schrödinger. “Quantisierung als Eigenwertproblem (Zweite Mitteilung)”. In: *Annalen der Physik* (1926).
- [10] Erwin Schrödinger. “Quantisierung als Eigenwertproblem (Dritte Mitteilung)”. In: *Annalen der Physik* (1926).
- [11] E. Schrödinger. “An Undulatory Theory of the Mechanics of Atoms and Molecules”. In: *Physical Review* 28.6 (Dec. 1926), pp. 1049–1070. DOI: 10.1103/PhysRev.28.1049.
- [12] W. Heisenberg. “Über den anschaulichen Inhalt der quantentheoretischen Kinematik und Mechanik”. German. In: *Zeitschrift für Physik* 43.3 (Mar. 1927), pp. 172–198. ISSN: 0044-3328. DOI: 10.1007/BF01397280.
- [13] Steven Weinberg. *The Quantum Theory of Fields*. Cambridge University Press, 1995. ISBN: 978-1-139-64416-7. DOI: 10.1017/CBO9781139644167.

- [14] Yudong Cao et al. “Quantum Chemistry in the Age of Quantum Computing”. In: *Chemical Reviews* 119.19 (Oct. 2019), pp. 10856–10915. ISSN: 0009-2665. DOI: 10.1021/acs.chemrev.8b00803.
- [15] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* 26.5 (1997), pp. 1484–1509. DOI: 10.1137/S0097539795293172.
- [16] Richard P. Feynman. “Simulating physics with computers”. In: *International Journal of Theoretical Physics* 21.6 (June 1982), pp. 467–488. ISSN: 1572-9575. DOI: 10.1007/BF02650179.
- [17] Emily Grumbling and Mark Horowitz, eds. *Quantum Computing: Progress and Prospects*. Washington, D.C.: National Academies Press, Mar. 2019. ISBN: 978-0-309-47969-1. DOI: 10.17226/25196.
- [18] Andrew M. Childs and Wim van Dam. “Quantum algorithms for algebraic problems”. In: *Reviews of Modern Physics* 82.1 (Jan. 2010), pp. 1–52. DOI: 10.1103/RevModPhys.82.1.
- [19] David P. DiVincenzo. “Quantum Computation”. In: *Science* 270.5234 (Oct. 1995), pp. 255–261. DOI: 10.1126/science.270.5234.255.
- [20] Amira Abbas et al. “Challenges and Opportunities in Quantum Optimization”. In: *Nature Reviews Physics* 6.12 (Dec. 2024), pp. 718–735. ISSN: 2522-5820.
- [21] He Ma, Marco Govoni, and Giulia Galli. “Quantum Simulations of Materials on Near-Term Quantum Computers”. In: *npj Computational Materials* 6.1 (July 2, 2020), pp. 1–8. ISSN: 2057-3960.
- [22] Philip Ball. “Materials Innovation from Quantum to Global”. In: *Nature Materials* 21.9 (Sept. 2022), pp. 962–967. ISSN: 1476-4660.
- [23] Rodney Van Meter and Dominic Horsman. “A blueprint for building a quantum computer”. In: *Commun. ACM* 56.10 (2013), pp. 84–93. ISSN: 0001-0782. DOI: 10.1145/2494568.
- [24] Joe O’Gorman and Earl T. Campbell. “Quantum computation with realistic magic-state factories”. In: *Physical Review A* 95.3 (Mar. 2017). DOI: 10.1103/PhysRevA.95.032338.
- [25] D. P. Divincenzo. “Topics in Quantum Computers”. In: *Mesoscopic Electron Transport*. Ed. by Lydia L. Sohn, Leo P. Kouwenhoven, and Gerd Schön. Dordrecht: Springer Netherlands, 1997, pp. 657–677. ISBN: 978-94-015-8839-3. DOI: 10.1007/978-94-015-8839-3_18.
- [26] B. D. Josephson. “Possible new effects in superconductive tunnelling”. In: *Physics Letters* 1.7 (July 1962), pp. 251–253. ISSN: 0031-9163. DOI: 10.1016/0031-9163(62)91369-0.
- [27] Yuriy Makhlin, Gerd Schön, and Alexander Shnirman. “Quantum-state engineering with Josephson-junction devices”. In: *Reviews of Modern Physics* 73.2 (May 2001), pp. 357–400. DOI: 10.1103/RevModPhys.73.357.

- [28] IBM. *IBM Unveils 400 Qubit-Plus Quantum Processor and Next-Generation IBM Quantum System Two*. 2022. URL: <https://newsroom.ibm.com/2022-11-09-IBM-Unveils-400-Qubit-Plus-Quantum-Processor-and-Next-Generation-IBM-Quantum-System-Two> (visited on 08/29/2024).
- [29] S. A. Moses et al. “A Race-Track Trapped-Ion Quantum Processor”. In: *Phys. Rev. X* 13 (4 Dec. 2023). DOI: 10.1103/PhysRevX.13.041052.
- [30] Daniel Loss and David P. DiVincenzo. “Quantum computation with quantum dots”. In: *Physical Review A* 57.1 (Jan. 1998), pp. 120–126. DOI: 10.1103/PhysRevA.57.120.
- [31] R. Hanson, L. P. Kouwenhoven, J. R. Petta, S. Tarucha, and L. M. K. Vandersypen. “Spins in few-electron quantum dots”. In: *Reviews of Modern Physics* 79.4 (2007), pp. 1217–1265. DOI: 10.1103/RevModPhys.79.1217.
- [32] Lov K. Grover. “A fast quantum mechanical algorithm for database search”. In: *Proceedings of the twenty-eighth annual ACM symposium on Theory of Computing*. STOC '96. New York, NY, USA, July 1996, pp. 212–219. ISBN: 978-0-89791-785-8. DOI: 10.1145/237814.237866.
- [33] D-Wave Quantum Inc. *The Most Connected and Powerful Quantum Computer Built for Business*. 2024. URL: <https://www.dwavesys.com/solutions-and-products/systems/> (visited on 08/29/2024).
- [34] Catherine C. McGeoch and Cong Wang. “Experimental evaluation of an adiabatic quantum system for combinatorial optimization”. In: *Proceedings of the ACM International Conference on Computing Frontiers*. CF '13. New York, NY, USA, 2013, pp. 1–11. ISBN: 978-1-4503-2053-5. DOI: 10.1145/2482767.2482797.
- [35] Michael Freedman, Alexei Kitaev, Michael Larsen, and Zhenghan Wang. “Topological quantum computation”. In: *Bulletin of the American Mathematical Society* 40.1 (2003), pp. 31–38. ISSN: 0273-0979, 1088-9485. DOI: 10.1090/S0273-0979-02-00964-3.
- [36] M. A. Kastner. “The single-electron transistor”. In: *Reviews of Modern Physics* 64.3 (July 1992), pp. 849–858. DOI: 10.1103/RevModPhys.64.849.
- [37] Tsuneya Ando, Alan B. Fowler, and Frank Stern. “Electronic properties of two-dimensional systems”. In: *Reviews of Modern Physics* 54.2 (Apr. 1982), pp. 437–672. DOI: 10.1103/RevModPhys.54.437.
- [38] L. P. Kouwenhoven, D. G. Austing, and S. Tarucha. “Few-electron quantum dots”. In: *Reports on Progress in Physics* 64.6 (June 2001), p. 701. ISSN: 0034-4885. DOI: 10.1088/0034-4885/64/6/201.
- [39] W. G. van der Wiel, S. De Franceschi, J. M. Elzerman, T. Fujisawa, S. Tarucha, and L. P. Kouwenhoven. “Electron transport through double quantum dots”. In: *Reviews of Modern Physics* 75.1 (Dec. 2002), pp. 1–22. ISSN: 0034-6861, 1539-0756. DOI: 10.1103/RevModPhys.75.1.
- [40] A. I. Ekimov and A. A. Onushchenko. “Quantum size effect in three-dimensional microscopic semiconductor crystals”. In: *Soviet Journal of Experimental and Theoretical Physics Letters* 34 (Sept. 1981), p. 345.

- [41] G. Scappucci, P. J. Taylor, J. R. Williams, T. Ginley, and S. Law. “Crystalline materials for quantum computing: Semiconductor heterostructures and topological insulators exemplars”. In: *MRS Bulletin* 46.7 (July 2021), pp. 596–606. ISSN: 1938-1425. DOI: 10.1557/s43577-021-00147-8.
- [42] Bernardette Kunert, Yves Mols, Marina Baryshniskova, Niamh Waldron, Andreas Schulze, and Robert Langer. “How to control defect formation in monolithic III/V hetero-epitaxy on (100) Si? A critical review on current approaches”. In: *Semiconductor Science and Technology* 33.9 (Aug. 2018). DOI: 10.1088/1361-6641/aad655.
- [43] Jonathan Y. Huang et al. “High-fidelity spin qubit operation and algorithmic initialization above 1 K”. In: *Nature* 627.8005 (Mar. 2024), pp. 772–777. ISSN: 1476-4687. DOI: 10.1038/s41586-024-07160-2.
- [44] D. M. Zajac, T. M. Hazard, X. Mi, E. Nielsen, and J. R. Petta. “Scalable Gate Architecture for a One-Dimensional Array of Semiconductor Spin Qubits”. In: *Phys. Rev. Appl.* 6 (5 Nov. 2016). DOI: 10.1103/PhysRevApplied.6.054013.
- [45] Samuel F. Neyens et al. “The critical role of substrate disorder in valley splitting in Si quantum wells”. In: *Applied Physics Letters* 112.24 (June 2018). ISSN: 0003-6951. DOI: 10.1063/1.5033447.
- [46] Guido Burkard, Thaddeus D. Ladd, Andrew Pan, John M. Nichol, and Jason R. Petta. “Semiconductor spin qubits”. In: *Reviews of Modern Physics* 95.2 (June 2023). DOI: 10.1103/RevModPhys.95.025003.
- [47] B. E. Kane. “A silicon-based nuclear spin quantum computer”. In: *Nature* 393.6681 (May 1998), pp. 133–137. ISSN: 1476-4687. DOI: 10.1038/30156.
- [48] Jeremy Levy. “Universal Quantum Computation with Spin- $1/2$ Pairs and Heisenberg Exchange”. In: *Physical Review Letters* 89.14 (Sept. 2002). DOI: 10.1103/PhysRevLett.89.147902.
- [49] J. R. Petta et al. “Coherent Manipulation of Coupled Electron Spins in Semiconductor Quantum Dots”. In: *Science* 309.5744 (Sept. 2005), pp. 2180–2184. DOI: 10.1126/science.1116955.
- [50] D. Bacon, J. Kempe, D. A. Lidar, and K. B. Whaley. “Universal Fault-Tolerant Quantum Computation on Decoherence-Free Subspaces”. In: *Physical Review Letters* 85.8 (Aug. 2000), pp. 1758–1761. DOI: 10.1103/PhysRevLett.85.1758.
- [51] D. P. DiVincenzo, D. Bacon, J. Kempe, G. Burkard, and K. B. Whaley. “Universal quantum computation with the exchange interaction”. In: *Nature* 408.6810 (Nov. 2000), pp. 339–342. ISSN: 1476-4687. DOI: 10.1038/35042541.
- [52] J. Kempe, D. Bacon, D. A. Lidar, and K. B. Whaley. “Theory of decoherence-free fault-tolerant universal quantum computation”. In: *Physical Review A* 63.4 (Mar. 2001). DOI: 10.1103/PhysRevA.63.042307.
- [53] J. Medford et al. “Self-consistent measurement and state tomography of an exchange-only spin qubit”. In: *Nature Nanotechnology* 8.9 (Sept. 2013), pp. 654–659. ISSN: 1748-3395. DOI: 10.1038/nnano.2013.168.

-
- [54] J. Medford et al. “Quantum-Dot-Based Resonant Exchange Qubit”. In: *Physical Review Letters* 111.5 (July 2013). DOI: 10.1103/PhysRevLett.111.050501.
- [55] J. M. Taylor, V. Srinivasa, and J. Medford. “Electrically Protected Resonant Exchange Qubits in Triple Quantum Dots”. In: *Physical Review Letters* 111.5 (July 2013). DOI: 10.1103/PhysRevLett.111.050502.
- [56] Zhan Shi et al. “Fast Hybrid Silicon Double-Quantum-Dot Qubit”. In: *Phys. Rev. Lett.* 108 (14 Apr. 2012). DOI: 10.1103/PhysRevLett.108.140503.
- [57] Maximilian Russ, J. R. Petta, and Guido Burkard. “Quadrupolar Exchange-Only Spin Qubit”. In: *Phys. Rev. Lett.* 121 (17 Oct. 2018). DOI: 10.1103/PhysRevLett.121.177701.
- [58] M. Benito et al. “Electric-field control and noise protection of the flopping-mode spin qubit”. In: *Phys. Rev. B* 100 (12 Sept. 2019). DOI: 10.1103/PhysRevB.100.125430.
- [59] M. Ciorga et al. “Addition Spectrum of a Lateral Dot from Coulomb and Spin-Blockade Spectroscopy”. In: *Physical Review B* 61.24 (June 15, 2000).
- [60] J. M. Elzerman et al. “Few-Electron Quantum Dot Circuit with Integrated Charge Read Out”. In: *Physical Review B* 67.16 (Apr. 2003). DOI: 10.1103/PhysRevB.67.161308.
- [61] D. Schröer et al. “Electrostatically Defined Serial Triple Quantum Dot Charged with Few Electrons”. In: *Physical Review B* 76.7 (Aug. 2007). DOI: 10.1103/PhysRevB.76.075306.
- [62] C. Volk et al. “Loading a quantum-dot based Qubyte register”. In: *npj Quantum Information* 5.1 (Apr. 2019), pp. 1–8. ISSN: 2056-6387. DOI: 10.1038/s41534-019-0146-y.
- [63] Pierre-André Mortemousque et al. “Enhanced Spin Coherence While Displacing Electron in a Two-Dimensional Array of Quantum Dots”. In: *PRX Quantum* 2.3 (Aug. 20, 2021).
- [64] Susan J. Angus, Andrew J. Ferguson, Andrew S. Dzurak, and Robert G. Clark. “Gate-Defined Quantum Dots in Intrinsic Silicon”. In: *Nano Letters* 7.7 (July 1, 2007), pp. 2051–2055. ISSN: 1530-6984.
- [65] B. M. Maune et al. “Coherent Singlet-Triplet Oscillations in a Silicon-Based Double Quantum Dot”. In: *Nature* 481.7381 (Jan. 2012), pp. 344–347. ISSN: 1476-4687.
- [66] Ran Xue et al. “Si/SiGe QuBus for Single Electron Information-Processing Devices with Memory and Micron-Scale Connectivity Function”. In: *Nature Communications* 15.1 (Mar. 14, 2024), p. 2296. ISSN: 2041-1723.
- [67] M. Field et al. “Measurements of Coulomb blockade with a noninvasive voltage probe”. In: *Physical Review Letters* 70.9 (Mar. 1993), pp. 1311–1314. DOI: 10.1103/PhysRevLett.70.1311.
- [68] C. Barthel, D. J. Reilly, C. M. Marcus, M. P. Hanson, and A. C. Gossard. “Rapid Single-Shot Measurement of a Singlet-Triplet Qubit”. In: *Physical Review Letters* 103.16 (Oct. 2009). DOI: 10.1103/PhysRevLett.103.160503.

- [69] D. J. Reilly, C. M. Marcus, M. P. Hanson, and A. C. Gossard. “Fast single-charge sensing with a rf quantum point contact”. In: *Applied Physics Letters* 91.16 (Oct. 2007). ISSN: 0003-6951. DOI: 10.1063/1.2794995.
- [70] R. J. Schoelkopf, P. Wahlgren, A. A. Kozhevnikov, P. Delsing, and D. E. Prober. “The Radio-Frequency Single-Electron Transistor (RF-SET): A Fast and Ultrasensitive Electrometer”. In: *Science* 280.5367 (May 1998), pp. 1238–1242. DOI: 10.1126/science.280.5367.1238.
- [71] C. Barthel et al. “Fast sensing of double-dot charge arrangement and spin state with a radio-frequency sensor quantum dot”. In: *Physical Review B* 81.16 (Apr. 2010). DOI: 10.1103/PhysRevB.81.161308.
- [72] Henk van Houten and Carlo Beenakker. “Quantum Point Contacts”. In: *Physics Today* 49.7 (July 1996), pp. 22–27. ISSN: 0031-9228. DOI: 10.1063/1.881503.
- [73] B. J. van Wees et al. “Quantized conductance of point contacts in a two-dimensional electron gas”. In: *Physical Review Letters* 60.9 (Feb. 1988), pp. 848–850. DOI: 10.1103/PhysRevLett.60.848.
- [74] J. M. Elzerman, R. Hanson, L. H. Willems van Beveren, B. Witkamp, L. M. K. Vandersypen, and L. P. Kouwenhoven. “Single-shot read-out of an individual electron spin in a quantum dot”. In: *Nature* 430.6998 (July 2004), pp. 431–435. ISSN: 1476-4687. DOI: 10.1038/nature02693.
- [75] R. Hanson et al. “Single-Shot Readout of Electron Spin States in a Quantum Dot Using Spin-Dependent Tunnel Rates”. In: *Physical Review Letters* 94.19 (May 2005). DOI: 10.1103/PhysRevLett.94.196802.
- [76] Tim Botzem et al. “Tuning Methods for Semiconductor Spin Qubits”. In: *Physical Review Applied* 10.5 (Nov. 2018). ISSN: 2331-7019. DOI: 10.1103/PhysRevApplied.10.054026.
- [77] Justyna P. Zwolak and Jacob M. Taylor. “Colloquium: Advances in automation of quantum dot devices control”. In: *Reviews of Modern Physics* 95.1 (Feb. 2023). DOI: 10.1103/RevModPhys.95.011006.
- [78] T. A. Baart, P. T. Eendebak, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen. “Computer-automated tuning of semiconductor double quantum dots into the single-electron regime”. In: *Applied Physics Letters* 108.21 (May 2016). ISSN: 0003-6951. DOI: 10.1063/1.4952624.
- [79] Fabian Hader et al. “On Noise-Sensitive Automatic Tuning of Gate-Defined Sensor Dots”. In: *IEEE Transactions on Quantum Engineering* 4 (2023), pp. 1–18. DOI: 10.1109/TQE.2023.3255743.
- [80] Julian D. Teske et al. “A Machine Learning Approach for Automated Fine-Tuning of Semiconductor Spin Qubits”. In: *Applied Physics Letters* 114.13 (Apr. 2019). ISSN: 0003-6951, 1077-3118. DOI: 10.1063/1.5088412.

-
- [81] Sandesh S. Kalantre et al. “Machine Learning techniques for state recognition and auto-tuning in quantum dots”. In: *npj Quantum Information* 5.1 (Dec. 2019). ISSN: 2056-6387. DOI: 10.1038/s41534-018-0118-7.
- [82] Cara Patricia Monical, Phillip J. Lewis, Abrielle Agron, Kurt W. Larson, and Andrew Mounce. *Image Processing Algorithms for Tuning Quantum Devices and Nitrogen-Vacancy Imaging*. Tech. rep. SAND-2019-11786. Sandia National Lab. (SNL-NM), Albuquerque, NM (United States), Sept. 2020. DOI: 10.2172/1662017.
- [83] Yui Muto et al. “Visual explanations of machine learning model estimating charge states in quantum dots”. In: *APL Machine Learning* 2.2 (Apr. 2024). ISSN: 2770-9019. DOI: 10.1063/5.0193621.
- [84] B. Severin et al. “Cross-architecture tuning of silicon and SiGe-based quantum devices using machine learning”. In: *Scientific Reports* 14.1 (July 2024). ISSN: 2045-2322. DOI: 10.1038/s41598-024-67787-z.
- [85] J. Darulová et al. “Autonomous Tuning and Charge-State Detection of Gate-Defined Quantum Dots”. In: *Physical Review Applied* 13.5 (May 2020). DOI: 10.1103/PhysRevApplied.13.054005.
- [86] Justyna P. Zwolak, Sandesh S. Kalantre, Xingyao Wu, Stephen Ragole, and Jacob M. Taylor. “QFlow lite dataset: A machine-learning approach to the charge states in quantum dot experiments”. In: *PLOS ONE* 13.10 (Oct. 2018). ISSN: 1932-6203. DOI: 10.1371/journal.pone.0205844.
- [87] H. Moon et al. “Machine learning enables completely automatic tuning of a quantum device faster than human experts”. In: *Nature Communications* 11.1 (Aug. 2020), p. 4161. ISSN: 2041-1723. DOI: 10.1038/s41467-020-17835-9.
- [88] Justyna P. Zwolak, Sandesh S. Kalantre, Thomas McJunkin, Brian J. Weber, and Jacob M. Taylor. “Ray-based classification framework for high-dimensional data”. In: (Oct. 2020). DOI: 10.48550/arXiv.2010.00500.
- [89] Justyna P. Zwolak et al. “Ray-based framework for state identification in quantum dot devices”. In: *PRX Quantum* 2.2 (June 2021). ISSN: 2691-3399. DOI: 10.1103/PRXQuantum.2.020335.
- [90] Joshua Ziegler, Florian Luthi, Mick Ramsey, Felix Borjans, Guoji Zheng, and Justyna P. Zwolak. “Tuning Arrays with Rays: Physics-Informed Tuning of Quantum Dot Charge States”. In: *Physical Review Applied* 20.3 (Sept. 2023). DOI: 10.1103/PhysRevApplied.20.034067.
- [91] C. J. van Diepen et al. “Automated tuning of inter-dot tunnel coupling in double quantum dots”. In: *Applied Physics Letters* 113.3 (2018). ISSN: 0003-6951. DOI: 10.1063/1.5031034.
- [92] Jonas Schuff et al. “Fully autonomous tuning of a spin qubit”. In: (Feb. 2024). DOI: 10.48550/arXiv.2402.03931.

- [93] A. R. Mills et al. “Computer-automated tuning procedures for semiconductor quantum dot arrays”. In: *Applied Physics Letters* 115.11 (Sept. 2019). ISSN: 0003-6951. DOI: 10 . 1063/1.5121444.
- [94] Justyna P. Zwolak et al. “Autotuning of Double-Dot Devices In Situ with Machine Learning”. In: *Physical Review Applied* 13.3 (Mar. 2020). DOI: 10 . 1103 / PhysRevApplied.13.034075.
- [95] D. T. Lennon et al. “Efficiently measuring a quantum device using machine learning”. In: *npj Quantum Information* 5.1 (Sept. 2019), pp. 1–8. ISSN: 2056-6387. DOI: 10 . 1038 / s41534-019-0193-4.
- [96] Vyacheslav P. Tuzlukov. *Signal Processing Noise*. First. The Electrical Engineering and Applied Signal Processing Series. Boca Raton: CRC Press, 2002. ISBN: 978-0-8493-1025-6.
- [97] Abid Muhammad Khan, Varun Jeoti, MZU Rehman, and MT Jilani. “Noise power estimation for broadcasting OFDM systems”. In: *2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*. Apr. 2017, pp. 1–6. DOI: 10 . 1109 / CCECE . 2017 . 7946746.
- [98] Ajay Kumar Boyat and Brijendra Kumar Joshi. *A Review Paper: Noise Models in Digital Image Processing*. May 2015. DOI: 10 . 48550 / arXiv . 1505 . 03489.
- [99] HeeBong Yang and Na Young Kim. “Material-Inherent Noise Sources in Quantum Information Architecture”. In: *Materials* 16.7 (2023). ISSN: 1996-1944. DOI: 10 . 3390 / ma16072561.
- [100] Norbert Wiener. *Extrapolation, Interpolation, and Smoothing of Stationary Time Series: With Engineering Applications*. The MIT Press, Aug. 1949. ISBN: 978-0-262-25719-0. DOI: 10 . 7551 / mitpress / 2946 . 001 . 0001.
- [101] R. B. Blackman and J. W. Tukey. “The measurement of power spectra from the point of view of communications engineering Part I”. In: *The Bell System Technical Journal* 37.1 (Jan. 1958), pp. 185–282. ISSN: 0005-8580. DOI: 10 . 1002 / j . 1538 - 7305 . 1958 . tb03874 . x.
- [102] B. W. Silverman. *Density estimation for statistics and data analysis*. Chapman & Hall/CRC, 1998. ISBN: 978-0-412-24620-3.
- [103] Stéphane Mallat. “A Wavelet Tour of Signal Processing (Third Edition)”. In: ed. by Stéphane Mallat. Boston: Academic Press, Jan. 2009. ISBN: 978-0-12-374370-1. DOI: 10 . 1016 / B978 - 0 - 12 - 374370 - 1 . X0001 - 8.
- [104] S. I. Olsen. “Estimation of Noise in Images: An Evaluation”. In: *CVGIP: Graphical Models and Image Processing* 55.4 (1993), pp. 319–323. ISSN: 1049-9652. DOI: 10 . 1006 / cgif . 1993 . 1022.
- [105] John Immerkær. “Fast Noise Variance Estimation”. In: *Computer Vision and Image Understanding* 64.2 (Sept. 1996), pp. 300–302. ISSN: 10773142. DOI: 10 . 1006 / cviu . 1996 . 0060.

- [106] K. Rank, M. Lendl, and R. Unbehauen. "Estimation of Image Noise Variance". In: *IEEE Proceedings - Vision, Image and Signal Processing* 146.2 (Apr. 1999), pp. 80–84. ISSN: 1350-245X. DOI: 10.1049/ip-vis:19990238.
- [107] B. R. Corner, R. M. Narayanan, and S. E. Reichenbach. "Noise Estimation in Remote Sensing Imagery Using Data Masking". In: *International Journal of Remote Sensing* 24.4 (Jan. 2003), pp. 689–702. ISSN: 0143-1161, 1366-5901. DOI: 10.1080/01431160210164271.
- [108] R. Garnett, T. Huegerich, C. Chui, and Wenjie He. "A Universal Noise Removal Algorithm with an Impulse Detector". In: *IEEE Transactions on Image Processing* 14.11 (Nov. 2005), pp. 1747–1754. ISSN: 1057-7149. DOI: 10.1109/TIP.2005.857261.
- [109] Tuan-Anh Nguyen and Min-Cheol Hong. "Filtering-Based Noise Estimation for Denoising the Image Degraded by Gaussian Noise". In: *Advances in Image and Video Technology*. Ed. by Yo-Sung Ho. Vol. 7088. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 157–167. ISBN: 978-3-642-25346-1. DOI: 10.1007/978-3-642-25346-1_15.
- [110] John D'Errico. *Estimatenoise*. 2007. URL: <https://www.mathworks.com/matlabcentral/fileexchange/16683-estimatenoise> (visited on 06/05/2021).
- [111] Wei Liu and Weisi Lin. "Additive White Gaussian Noise Level Estimation in SVD Domain for Images". In: *IEEE Transactions on Image Processing* 22.3 (Mar. 2013), pp. 872–883. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2012.2219544.
- [112] Olivier Laligant, Frederic Truchetet, and Eric Fauvet. "Noise Estimation From Digital Step-Model Signal". In: *IEEE Transactions on Image Processing* 22.12 (Dec. 2013), pp. 5158–5167. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2013.2282123.
- [113] S. Mostafa Mousavi and Charles A. Langston. "Adaptive Noise Estimation and Suppression for Improving Microseismic Event Detection". In: *Journal of Applied Geophysics* 132 (Sept. 2016), pp. 116–124. ISSN: 09269851. DOI: 10.1016/j.jappgeo.2016.06.008.
- [114] N. ur Rehman, B. Khan, and K. Naveed. "Data-Driven Multivariate Signal Denoising Using Mahalanobis Distance". In: *IEEE Signal Processing Letters* 26.9 (Sept. 2019), pp. 1408–1412. ISSN: 1558-2361. DOI: 10.1109/LSP.2019.2932715.
- [115] A. Bosco, A. Bruna, G. Messina, and G. Spampinato. "Fast Method for Noise Level Estimation and Integrated Noise Reduction". In: *IEEE Transactions on Consumer Electronics* 51.3 (Aug. 2005), pp. 1028–1033. ISSN: 0098-3063. DOI: 10.1109/TCE.2005.1510518.
- [116] Shen-Chuan Tai and Shih-Ming Yang. "A Fast Method for Image Noise Estimation Using Laplacian Operator and Adaptive Edge Detection". In: *2008 3rd International Symposium on Communications, Control and Signal Processing*. St. Julian's, Malta: IEEE, Mar. 2008, pp. 1077–1081. ISBN: 978-1-4244-1687-5. DOI: 10.1109/ISCCSP.2008.4537384.

- [117] Shih-Ming Yang. “Fast and Reliable Image-Noise Estimation Using a Hybrid Approach”. In: *Journal of Electronic Imaging* 19.3 (July 2010). ISSN: 1017-9909. DOI: 10.1117/1.3476329.
- [118] Uwe Schmidt, Kevin Schelten, and Stefan Roth. “Bayesian Deblurring with Integrated Noise Estimation”. In: *CVPR 2011*. Colorado Springs, CO, USA: IEEE, June 2011, pp. 2625–2632. ISBN: 978-1-4577-0394-2. DOI: 10.1109/CVPR.2011.5995653.
- [119] Camille Sutour, Charles-Alban Deledalle, and Jean-François Aujol. “Estimation of the Noise Level Function Based on a Nonparametric Detection of Homogeneous Image Regions”. In: *SIAM Journal on Imaging Sciences* 8.4 (Jan. 2015), pp. 2622–2661. ISSN: 1936-4954. DOI: 10.1137/15M1012682.
- [120] Stanislav Pyatykh and Jurgen Hesser. “Image Sensor Noise Parameter Estimation by Variance Stabilization and Normality Assessment”. In: *IEEE Transactions on Image Processing* 23.9 (Sept. 2014), pp. 3990–3998. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2014.2339194.
- [121] Guangyong Chen, Fengyuan Zhu, and Pheng Ann Heng. “An Efficient Statistical Method for Image Noise Level Estimation”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile, Dec. 2015, pp. 477–485. ISBN: 978-1-4673-8391-2. DOI: 10.1109/ICCV.2015.62.
- [122] J.S. Lee and K. Hoppel. “Noise Modeling and Estimation of Remotely-Sensed Images”. In: *12th Canadian Symposium on Remote Sensing Geoscience and Remote Sensing Symposium*, vol. 2. 1989, pp. 1005–1008. DOI: 10.1109/IGARSS.1989.579061.
- [123] A. Amer and E. Dubois. “Fast and Reliable Structure-Oriented Video Noise Estimation”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 15.1 (Jan. 2005), pp. 113–118. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2004.837017.
- [124] Dong-Hyuk Shin, Rae-Hong Park, Seungjoon Yang, and Jae-Han Jung. “Block-Based Noise Estimation Using Adaptive Gaussian Filtering”. In: *IEEE Transactions on Consumer Electronics* 51.1 (Feb. 2005), pp. 218–226. ISSN: 1558-4127. DOI: 10.1109/TCE.2005.1405723.
- [125] Xinhao Liu, Masayuki Tanaka, and Masatoshi Okutomi. “Noise Level Estimation Using Weak Textured Patches of a Single Noisy Image”. In: *2012 19th IEEE International Conference on Image Processing*. Orlando, FL, USA, Sept. 2012, pp. 665–668. ISBN: 978-1-4673-2532-5. DOI: 10.1109/ICIP.2012.6466947.
- [126] Xinhao Liu, Masayuki Tanaka, and Masatoshi Okutomi. “Single-Image Noise Level Estimation for Blind Denoising”. In: *IEEE Transactions on Image Processing* 22.12 (Dec. 2013), pp. 5226–5237. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2013.2283400.
- [127] S. Pyatykh, J. Hesser, and Lei Zheng. “Image Noise Level Estimation by Principal Component Analysis”. In: *IEEE Transactions on Image Processing* 22.2 (Feb. 2013), pp. 687–699. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2012.2221728.

- [128] Meisam Rakhshanfar and Maria A. Amer. "Estimation of Gaussian, Poissonian-Gaussian, and Processed Visual Noise and Its Level Function". In: *IEEE Transactions on Image Processing* 25.9 (Sept. 2016), pp. 4172–4185. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2016.2588320.
- [129] Miguel Colom and Antoni Buades. "Analysis and Extension of the PCA Method, Estimating a Noise Curve from a Single Image". In: *Image Processing On Line* 5 (Dec. 2016), pp. 365–390. ISSN: 2105-1232. DOI: 10.5201/ipo1.2016.124.
- [130] Ping Jiang and Jian-zhou Zhang. "Fast and Reliable Noise Level Estimation Based on Local Statistic". In: *Pattern Recognition Letters* 78 (July 2016), pp. 8–13. ISSN: 01678655. DOI: 10.1016/j.patrec.2016.03.026.
- [131] Zhuang Fang and Xuming Yi. "A Novel Natural Image Noise Level Estimation Based on Flat Patches and Local Statistics". In: *Multimedia Tools and Applications* 78.13 (July 2019), pp. 17337–17358. ISSN: 1380-7501, 1573-7721. DOI: 10.1007/s11042-018-7137-4.
- [132] David L Donoho and Iain M Johnstone. "Ideal Spatial Adaptation by Wavelet Shrinkage". In: *Biometrika* 81.3 (Sept. 1994), pp. 425–455. ISSN: 1464-3510. DOI: 10.1093/biomet/81.3.425.
- [133] D. L. Donoho. "De-Noising by Soft-Thresholding". In: *IEEE Transactions on Information Theory* 41.3 (May 1995), pp. 613–627. ISSN: 1557-9654. DOI: 10.1109/18.382009.
- [134] A.F. Abdelnour and I.W. Selesnick. "Design of 2-Band Orthogonal near-Symmetric CQF". In: *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*. Vol. 6. Salt Lake City, UT, USA, 2001, pp. 3693–3696. ISBN: 978-0-7803-7041-8. DOI: 10.1109/ICASSP.2001.940644.
- [135] A. De Stefano, P.R. White, and W.B. Collis. "Training Methods for Image Noise Level Estimation on Wavelet Components". In: *EURASIP Journal on Advances in Signal Processing* 2004.16 (Dec. 2004). ISSN: 1687-6180. DOI: 10.1155/S1110865704401218.
- [136] Khuram Naveed, Bisma Shaukat, and Naveed ur Rehman. "Signal Denoising Based on Dual Tree Complex Wavelet Transform and Goodness of Fit Test". In: *2017 22nd International Conference on Digital Signal Processing (DSP)*. London: IEEE, Aug. 2017, pp. 1–5. ISBN: 978-1-5386-1895-0. DOI: 10.1109/ICDSP.2017.8096067.
- [137] Nikolay N. Ponomarenko, Vladimir V. Lukin, Sergey K. Abramov, Karen O. Egiazarian, and Jaakko T. Astola. "Blind Evaluation of Additive Noise Variance in Textured Images by Nonlinear Processing of Block DCT Coefficients". In: *Image Processing: Algorithms and Systems II*. Vol. 5014. SPIE, May 2003, pp. 178–189. DOI: 10.1117/12.477717.
- [138] Daniel Zoran and Yair Weiss. "Scale Invariance and Noise in Natural Images". In: *2009 IEEE 12th International Conference on Computer Vision*. Kyoto, Sept. 2009, pp. 2209–2216. ISBN: 978-1-4244-4420-5. DOI: 10.1109/ICCV.2009.5459476.
- [139] Damien Garcia. "Robust Smoothing of Gridded Data in One and Higher Dimensions with Missing Values". In: *Computational Statistics & Data Analysis* 54.4 (Apr. 2010), pp. 1167–1178. ISSN: 01679473. DOI: 10.1016/j.csda.2009.09.020.

- [140] Miguel Colom and Antoni Buades. “Analysis and Extension of the Ponomarenko et al. Method, Estimating a Noise Curve from a Single Image”. In: *Image Processing On Line* 3 (July 2013), pp. 173–197. ISSN: 2105-1232. DOI: 10.5201/ipo1.2013.45.
- [141] Marc Lebrun, Miguel Colom, and Jean-Michel Morel. “The Noise Clinic: A Blind Image Denoising Algorithm”. In: *Image Processing On Line* 5 (Jan. 2015), pp. 1–54. ISSN: 2105-1232. DOI: 10.5201/ipo1.2015.125.
- [142] Mykola Ponomarenko, Nikolay Gapon, Viacheslav Voronin, and Karen Egiazarian. “Blind estimation of white Gaussian noise variance in highly textured images”. In: *Electronic Imaging* 30.13 (2018), pp. 382-1–382-1. DOI: 10.2352/ISSN.2470-1173.2018.13.IPAS-382.
- [143] Eric F Smith. “Development of an Image Noise Estimation Method and a Sub-Imaging Based Wiener Method”. PhD thesis. New Orleans: University of New Orleans, Theses and Dissertations, 1052, Dec. 2006.
- [144] Guomin Luo and Daming Zhang. “Wavelet Denoising”. In: *Advances in Wavelet Theory and Their Applications in Engineering, Physics and Technology*. Ed. by Dumitru Baleanu. InTech, 2012, pp. 59–80. ISBN: 978-953-51-0494-0. DOI: 10.5772/37424.
- [145] Rakshita Pandya and Kshitij Pathak. “Survey on Noise Estimation and Removal Methods through SVM”. In: *International Journal of Computer Applications* 86.9 (Jan. 2014), pp. 25–32. ISSN: 09758887. DOI: 10.5120/15014-3297.
- [146] Wei Liu. “Additive White Gaussian Noise Level Estimation Based on Block SVD”. In: *2014 IEEE Workshop on Electronics, Computer and Applications*. Ottawa, ON, Canada, May 2014, pp. 960–963. ISBN: 978-1-4799-4565-8. DOI: 10.1109/IWECA.2014.6845781.
- [147] Asem Khmag, Abd Rahman Ramli, S. A. R. Al-haddad, and Noraziahtulhidayu Kamarudin. “Natural Image Noise Level Estimation Based on Local Statistics for Blind Noise Reduction”. In: *The Visual Computer* 34.4 (Apr. 2018), pp. 575–587. ISSN: 0178-2789, 1432-2315. DOI: 10.1007/s00371-017-1362-0.
- [148] Jean-Baptiste-Joseph Fourier. *Théorie analytique de la chaleur*. French. Paris: F. Didot, 1822.
- [149] James W. Cooley and John W. Tukey. “An Algorithm for the Machine Calculation of Complex Fourier Series”. In: *Mathematics of Computation* 19.90 (1965), pp. 297–301. ISSN: 0025-5718. DOI: 10.2307/2003354.
- [150] T. Huang, G. Yang, and G. Tang. “A fast two-dimensional median filtering algorithm”. In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 27.1 (Feb. 1979), pp. 13–18. ISSN: 0096-3518. DOI: 10.1109/TASSP.1979.1163188.
- [151] C. Tomasi and R. Manduchi. “Bilateral filtering for gray and color images”. In: *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*. Jan. 1998, pp. 839–846. DOI: 10.1109/ICCV.1998.710815.

-
- [152] Pietro Perona and Jitendra Malik. “Scale-Space and Edge Detection Using Anisotropic Diffusion”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 12.7 (July 1990), pp. 629–639. DOI: 10.1109/34.56205.
- [153] K. Dabov, A. Foi, V. Katkounnik, and K. Egiazarian. “Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering”. In: *IEEE Transactions on Image Processing* 16.8 (Aug. 2007), pp. 2080–2095. ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2007.901238.
- [154] Steven L. Horowitz and Theodosios Pavlidis. “Picture Segmentation by a Tree Traversal Algorithm”. In: *J. ACM* 23.2 (Apr. 1, 1976), pp. 368–388. ISSN: 0004-5411. DOI: 10.1145/321941.321956.
- [155] Volker Rehrmann and Lutz Priebe. “Fast and Robust Segmentation of Natural Color Scenes”. In: *Computer Vision ACCV’98*. Ed. by Roland Chin and Ting-Chuen Pong. Red. by Gerhard Goos, Juris Hartmanis, and Jan Van Leeuwen. Vol. 1351. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 598–606. ISBN: 978-3-540-69669-8.
- [156] S. Lloyd. “Least squares quantization in PCM”. In: *IEEE Transactions on Information Theory* 28.2 (Mar. 1982), pp. 129–137. ISSN: 1557-9654. DOI: 10.1109/TIT.1982.1056489.
- [157] J. MacQueen. “Some methods for classification and analysis of multivariate observations”. In: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*. Vol. 5.1. University of California Press, Jan. 1967, pp. 281–298.
- [158] Hugo Steinhaus. “Sur la division des corps matériels en parties”. In: *Bulletin L’Académie Polonaise des Science* 4 (1956), pp. 801–804.
- [159] Irwin Sobel and Gary M. Feldman. “A 3x3 Isotropic Gradient Operator for Image Processing”. presented at the Stanford Artificial Intelligence Project (SAIL). 1968.
- [160] John Canny. “A Computational Approach to Edge Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI-8.6* (Nov. 1986), pp. 679–698. ISSN: 1939-3539. DOI: 10.1109/TPAMI.1986.4767851.
- [161] Azriel Rosenfeld and John L. Pfaltz. “Sequential Operations in Digital Picture Processing”. In: *J. ACM* 13.4 (Oct. 1, 1966), pp. 471–494. ISSN: 0004-5411. DOI: 10.1145/321356.321357.
- [162] H. Digabel and Christian Lantuéjoul. “Iterative algorithms”. In: *Proc. 2nd European Symp. Quantitative Analysis of Microstructures in Material Science, Biology and Medicine*. Vol. 19. 7. Riederer Verlag, 1978.
- [163] Junfeng Jing, Shenjuan Liu, Gang Wang, Weichuan Zhang, and Changming Sun. “Recent Advances on Image Edge Detection: A Comprehensive Review”. In: *Neurocomputing* 503 (Sept. 2022), pp. 259–271. ISSN: 09252312. DOI: 10.1016/j.neucom.2022.06.083.

- [164] Xinyu Lin, Yingjie Zhou, Yipeng Liu, and Ce Zhu. “A Comprehensive Review of Image Line Segment Detection and Description: Taxonomies, Comparisons, and Challenges”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46.12 (2024), pp. 8074–8093. DOI: 10.1109/TPAMI.2024.3400881.
- [165] Junqing Wang and Weichuan Zhang. “A Survey of Corner Detection Methods”. In: *Proceedings of the 2018 2nd International Conference on Electrical Engineering and Automation (ICEEA 2018)*. Chengdu, China: Atlantis Press, 2018. ISBN: 978-94-6252-497-2. DOI: 10.2991/iceea-18.2018.47.
- [166] Steffen Gauglitz, Tobias Höllerer, and Matthew Turk. “Evaluation of Interest Point Detectors and Feature Descriptors for Visual Tracking”. In: *International Journal of Computer Vision* 94.3 (Sept. 1, 2011), pp. 335–360. ISSN: 1573-1405. DOI: 10.1007/s11263-011-0431-5.
- [167] Dengsheng Zhang and Guojun Lu. “Review of Shape Representation and Description Techniques”. In: *Pattern Recognition* 37.1 (Jan. 1, 2004), pp. 1–19. ISSN: 0031-3203. DOI: 10.1016/j.patcog.2003.07.008.
- [168] Anne Humeau-Heurtier. “Texture Feature Extraction Methods: A Survey”. In: *IEEE Access* 7 (2019), pp. 8975–9000. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2890743.
- [169] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. “A training algorithm for optimal margin classifiers”. In: *Proceedings of the fifth annual workshop on Computational learning theory*. COLT '92. New York, NY, USA: Association for Computing Machinery, July 1992, pp. 144–152. ISBN: 978-0-89791-497-0. DOI: 10.1145/130385.130401.
- [170] Corinna Cortes and Vladimir Vapnik. “Support-vector networks”. In: *Machine Learning* 20.3 (Sept. 1995), pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/BF00994018.
- [171] F. Rosenblatt. “The perceptron: A probabilistic model for information storage and organization in the brain”. In: *Psychological Review* 65.6 (1958), pp. 386–408. ISSN: 1939-1471. DOI: 10.1037/h0042519.
- [172] P. Viola and M. Jones. “Rapid object detection using a boosted cascade of simple features”. In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*. Vol. 1. Dec. 2001, pp. I–I. DOI: 10.1109/CVPR.2001.990517.
- [173] Robert K. McConnell. “Method of and apparatus for pattern recognition”. US4567610A. Jan. 1986.
- [174] D. H. Hubel and T. N. Wiesel. “Receptive fields, binocular interaction and functional architecture in the cat’s visual cortex”. In: *The Journal of Physiology* 160.1 (Jan. 1962), pp. 106–154.2. ISSN: 0022-3751.
- [175] D. H. Hubel and T. N. Wiesel. “Receptive fields and functional architecture of monkey striate cortex”. In: *The Journal of Physiology* 195.1 (Mar. 1968), pp. 215–243. ISSN: 0022-3751. DOI: 10.1113/jphysiol.1968.sp008455.

- [176] Kuniyuki Fukushima. “Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”. In: *Biological Cybernetics* 36.4 (Apr. 1980), pp. 193–202. ISSN: 1432-0770. DOI: 10.1007/BF00344251.
- [177] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (Nov. 1998), pp. 2278–2324. ISSN: 1558-2256. DOI: 10.1109/5.726791.
- [178] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention MICCAI 2015*. Ed. by Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2015, pp. 234–241. ISBN: 978-3-319-24574-4. DOI: 10.1007/978-3-319-24574-4_28.
- [179] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. “Mask R-CNN”. In: *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.
- [180] Jieneng Chen et al. “TransUNet: Transformers Make Strong Encoders for Medical Image Segmentation”. In: (Feb. 2021). DOI: 10.48550/arXiv.2102.04306.
- [181] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. “Segformer: Transformer for Semantic Segmentation”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada, Oct. 2021, pp. 7242–7252. ISBN: 978-1-66542-812-5. DOI: 10.1109/ICCV48922.2021.00717.
- [182] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. “SegFormer: simple and efficient design for semantic segmentation with transformers”. In: *Proceedings of the 35th International Conference on Neural Information Processing Systems*. NIPS ’21. Red Hook, NY, USA: Curran Associates Inc., 2021. ISBN: 978-1-71384-539-3.
- [183] Mengyang Pu, Yaping Huang, Yuming Liu, Qingji Guan, and Haibin Ling. “EDTER: Edge Detection with Transformer”. In: *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 1392–1402. DOI: 10.1109/CVPR52688.2022.00146.
- [184] Yuan Yang, Lin Zhang, Lei Ren, and Xiaohan Wang. “MMViT-Seg: A lightweight transformer and CNN fusion network for COVID-19 segmentation”. In: *Computer Methods and Programs in Biomedicine* 230 (2023). ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2023.107348.
- [185] Huajun Liu, Jing Yang, Xiangyu Miao, Christoph Mertz, and Hui Kong. “CrackFormer Network for Pavement Crack Segmentation”. In: *IEEE Transactions on Intelligent Transportation Systems* 24.9 (2023), pp. 9240–9252. DOI: 10.1109/TITS.2023.3266776.
- [186] Hu Cao et al. “Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation”. In: *Computer Vision – ECCV 2022 Workshops*. Ed. by Leonid Karlinsky, Tomer Michaeli, and Ko Nishino. Cham: Springer Nature Switzerland, 2023, pp. 205–218. ISBN: 978-3-031-25066-8. DOI: 10.1007/978-3-031-25066-8_9.

- [187] Yue Liu et al. “VMamba: Visual State Space Model”. In: (Apr. 2024). DOI: 10.48550/arXiv.2401.10166.
- [188] Jiacheng Ruan and Suncheng Xiang. “VM-UNet: Vision Mamba UNet for Medical Image Segmentation”. In: (Feb. 2024). DOI: 10.48550/arXiv.2402.02491.
- [189] Zhaohu Xing, Tian Ye, Yijun Yang, Guang Liu, and Lei Zhu. “SegMamba: Long-Range Sequential Modeling Mamba for 3D Medical Image Segmentation”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*. Ed. by Marius George Linguraru et al. Cham: Springer Nature Switzerland, 2024, pp. 578–588. ISBN: 978-3-031-72111-3.
- [190] Jun Ma, Feifei Li, and Bo Wang. “U-Mamba: Enhancing Long-range Dependency for Biomedical Image Segmentation”. In: (Jan. 2024). DOI: 10.48550/arXiv.2401.04722.
- [191] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 25. Curran Associates, Inc., 2012.
- [192] Christian Szegedy et al. “Going deeper with convolutions”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA, June 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.
- [193] K Simonyan and A Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: Computational and Biological Learning Society, Apr. 2015, pp. 1–14.
- [194] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
- [195] Andrew G. Howard et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. In: (Apr. 2017). DOI: 10.48550/arXiv.1704.04861.
- [196] Mingxing Tan and Quoc Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. June 2019, pp. 6105–6114.
- [197] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Region-Based Convolutional Networks for Accurate Object Detection and Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38.1 (Jan. 2016), pp. 142–158. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2015.2437384.
- [198] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. “You Only Look Once: Unified, Real-Time Object Detection”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 779–788. DOI: 10.1109/CVPR.2016.91.
- [199] Wei Shen, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. “DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2015, pp. 3982–3991. DOI: 10.1109/CVPR.2015.7299024.

- [200] Iasonas Kokkinos. “Pushing the Boundaries of Boundary Detection Using Deep Learning”. In: *Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)*. Jan. 22, 2016.
- [201] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. “Convolutional Oriented Boundaries”. In: *Computer Vision ECCV 2016*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 580–596. ISBN: 978-3-319-46448-0. DOI: 10.1007/978-3-319-46448-0_35.
- [202] Ruoxi Deng, Chunhua Shen, Shengjun Liu, Huibing Wang, and Xinru Liu. “Learning to Predict Crisp Boundaries”. In: *Computer Vision ECCV 2018*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 570–586. ISBN: 978-3-030-01231-1. DOI: 10.1007/978-3-030-01231-1_35.
- [203] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. “Focal Loss for Dense Object Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42.2 (Feb. 1, 2020), pp. 318–327. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2018.2858826.
- [204] Zhuo Su et al. “Pixel Difference Networks for Efficient Edge Detection”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 5097–5107. DOI: 10.1109/ICCV48922.2021.00507.
- [205] Yunfan Ye, Kai Xu, Yuhang Huang, Renjiao Yi, and Zhiping Cai. “DiffusionEdge: diffusion probabilistic model for crisp edge detection”. In: *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence. AAAI’24/IAAI’24/EAAI’24*. AAAI Press, 2024. ISBN: 978-1-57735-887-9. DOI: 10.1609/aaai.v38i7.28490.
- [206] Kun Huang, Yifan Wang, Zihan Zhou, Tianjiao Ding, Shenghua Gao, and Yi Ma. “Learning to Parse Wireframes in Images of Man-Made Environments”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 626–635. DOI: 10.1109/CVPR.2018.00072.
- [207] Yichao Zhou, Haozhi Qi, and Yi Ma. “End-to-End Wireframe Parsing”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2019, pp. 962–971. DOI: 10.1109/ICCV.2019.00105.
- [208] Nan Xue et al. “Holistically-Attracted Wireframe Parsing”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2785–2794. DOI: 10.1109/CVPR42600.2020.00286.
- [209] Kai Zhao, Qi Han, Chang-Bin Zhang, Jun Xu, and Ming-Ming Cheng. “Deep Hough Transform for Semantic Line Detection”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.9 (2022), pp. 4793–4806. DOI: 10.1109/TPAMI.2021.3077129.

- [210] Yifan Xu, Weijian Xu, David Cheung, and Zhuowen Tu. “Line Segment Detection Using Transformers without Edges”. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021, pp. 4255–4264. DOI: 10.1109/CVPR46437.2021.00424.
- [211] Geonmo Gu, Byungsoo Ko, SeoungHyun Go, Sung-Hyun Lee, Jingeun Lee, and Minchul Shin. “Towards Light-Weight and Real-Time Line Segment Detection”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 36.1 (June 2022), pp. 726–734. DOI: 10.1609/aaai.v36i1.19953.
- [212] Karel Lenc and Andrea Vedaldi. “Learning Covariant Feature Detectors”. In: *Computer Vision ECCV 2016 Workshops*. Ed. by Gang Hua and Hervé Jégou. Cham: Springer International Publishing, 2016, pp. 100–117. ISBN: 978-3-319-49409-8. DOI: 10.1007/978-3-319-49409-8_11.
- [213] Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. “SuperPoint: Self-Supervised Interest Point Detection and Description”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2018. DOI: 10.1109/CVPRW.2018.00060.
- [214] Mihai Dusmanu et al. “D2-Net: A Trainable CNN for Joint Description and Detection of Local Features”. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 8084–8093. DOI: 10.1109/CVPR.2019.00828.
- [215] Zixin Luo et al. “ASLFeat: Learning Local Features of Accurate Shape and Localization”. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 6588–6597. DOI: 10.1109/CVPR42600.2020.00662.
- [216] Axel Barroso-Laguna and Krystian Mikolajczyk. “Key.Net: Keypoint Detection by Handcrafted and Learned CNN Filters Revisited”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.01 (Jan. 1, 2023), pp. 698–711. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2022.3145820.
- [217] Tsung-Han Chan, Kui Jia, Shenghua Gao, Jiwen Lu, Zinan Zeng, and Yi Ma. “PCANet: A Simple Deep Learning Baseline for Image Classification?” In: *IEEE Transactions on Image Processing* 24.12 (2015), pp. 5017–5032. DOI: 10.1109/TIP.2015.2475625.
- [218] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. “Bilinear CNN Models for Fine-Grained Visual Recognition”. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 1449–1457. DOI: 10.1109/ICCV.2015.170.
- [219] Felix Juefei-Xu, Vishnu Naresh Boddeti, and Marios Savvides. “Local Binary Convolutional Neural Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 4284–4293. DOI: 10.1109/CVPR.2017.456.
- [220] Hang Zhang, Jia Xue, and Kristin Dana. “Deep TEN: Texture Encoding Network”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2896–2905. DOI: 10.1109/CVPR.2017.309.

- [221] Relja Arandjelovi, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. “NetVLAD: CNN Architecture for Weakly Supervised Place Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.6 (2018), pp. 1437–1451. DOI: 10.1109/TPAMI.2017.2711011.
- [222] Joshua Peeples, Weihuang Xu, and Alina Zare. “Histogram Layers for Texture Analysis”. In: *IEEE Transactions on Artificial Intelligence* 3.4 (2022), pp. 541–552. DOI: 10.1109/TAI.2021.3135804.
- [223] Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren, eds. *Automated Machine Learning: Methods, Systems, Challenges*. The Springer Series on Challenges in Machine Learning. Cham, 2019. ISBN: 978-3-030-05317-8. DOI: 10.1007/978-3-030-05318-5.
- [224] Mohaimenul Azam Khan Raiaan et al. “A Systematic Review of Hyperparameter Optimization Techniques in Convolutional Neural Networks”. In: *Decision Analytics Journal* 11 (June 1, 2024). ISSN: 2772-6622. DOI: 10.1016/j.dajour.2024.100470.
- [225] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. “Neural Architecture Search: A Survey”. In: *Journal of Machine Learning Research* 20 (Jan. 2019), pp. 1–21. ISSN: 1532-4435.
- [226] Jeon-Seong Kang, JinKyu Kang, Jung-Jun Kim, Kwang-Woo Jeon, Hyun-Joon Chung, and Byung-Hoon Park. “Neural Architecture Search Survey: A Computer Vision Perspective”. In: *Sensors (Basel, Switzerland)* 23.3 (Feb. 3, 2023), p. 1713. ISSN: 1424-8220. DOI: 10.3390/s23031713.
- [227] Matt Poyser and Toby P. Breckon. “Neural Architecture Search: A Contemporary Literature Review for Computer Vision Applications”. In: *Pattern Recognition* 147 (Mar. 1, 2024). ISSN: 0031-3203. DOI: 10.1016/j.patcog.2023.110052.
- [228] L. M. K. Vandersypen et al. “Interfacing spin qubits in quantum dots and donorshot, dense, and coherent”. In: *npj Quantum Information* 3.1 (Sept. 2017), pp. 1–10. ISSN: 2056-6387. DOI: 10.1038/s41534-017-0038-y.
- [229] Bishnu Patra et al. “Cryo-CMOS Circuits and Systems for Quantum Computing Applications”. In: *IEEE Journal of Solid-State Circuits* 53.1 (Jan. 2018), pp. 309–321. ISSN: 1558-173X. DOI: 10.1109/JSSC.2017.2737549.
- [230] S. J. Pauka et al. “A cryogenic CMOS chip for generating control signals for multiple qubits”. In: *Nature Electronics* 4.1 (Jan. 2021), pp. 64–70. ISSN: 2520-1131. DOI: 10.1038/s41928-020-00528-y.
- [231] Lotte Geck, Andre Kruth, Hendrik Bluhm, Stefan van Waasen, and Stefan Heinen. “Control electronics for semiconductor spin qubits”. In: *Quantum Science and Technology* 5.1 (Dec. 2019). ISSN: 2058-9565. DOI: 10.1088/2058-9565/ab5e07.
- [232] J. M. Hornibrook et al. “Frequency multiplexing for readout of spin qubits”. In: *Applied Physics Letters* 104.10 (Mar. 2014). ISSN: 0003-6951. DOI: 10.1063/1.4868107.

- [233] Andrea Ruffino, Tsung-Yeh Yang, John Michniewicz, Yatao Peng, Edoardo Charbon, and Miguel Fernando Gonzalez-Zalba. “A cryo-CMOS chip that integrates silicon quantum dots and multiplexed dispersive readout electronics”. In: *Nature Electronics* 5.1 (Jan. 2022), pp. 53–59. ISSN: 2520-1131. DOI: 10.1038/s41928-021-00687-6.
- [234] Tim Botzem. “Coherence and high fidelity control of two-electron spin qubits in GaAs quantum dots”. PhD thesis. Aachen: RWTH Aachen, Jan. 2017. DOI: 10.18154/RWTH-2017-04410.
- [235] Pascal Cerfontaine. “High-Fidelity Single- and Two-Qubit Gates for Two-Electron Spin Qubits”. PhD thesis. Aachen: RWTH Aachen University, July 2019. DOI: 10.18154/RWTH-2019-09348.
- [236] O. E. Dial, M. D. Shulman, S. P. Harvey, H. Bluhm, V. Umansky, and A. Yacoby. “Charge Noise Spectroscopy Using Coherent Exchange Oscillations in a Singlet-Triplet Qubit”. In: *Physical Review Letters* 110.14 (Apr. 2013). ISSN: 0031-9007, 1079-7114. DOI: 10.1103/PhysRevLett.110.146804.
- [237] John R. Barry, Edward A. Lee, and David G. Messerschmitt. *Digital Communication*. Third. Boston, MA: Springer US, 2004. ISBN: 978-1-4615-0227-2. DOI: 10.1007/978-1-4615-0227-2.
- [238] C. W. J. Beenakker. “Theory of Coulomb-blockade Oscillations in the Conductance of a Quantum Dot”. In: *Physical Review B* 44.4 (July 1991), pp. 1646–1656. DOI: 10.1103/PhysRevB.44.1646.
- [239] Hermann Grabert and Michel H. Devoret, eds. *Single Charge Tunneling*. Vol. 294. NATO ASI Series. Boston, MA: Springer US, 1992. ISBN: 978-1-4757-2166-9. DOI: 10.1007/978-1-4757-2166-9.
- [240] C. Livermore, C. H. Crouch, R. M. Westervelt, K. L. Campman, and A. C. Gossard. “The Coulomb Blockade in Coupled Quantum Dots”. In: *Science* 274.5291 (Nov. 1996), pp. 1332–1335. DOI: 10.1126/science.274.5291.1332.
- [241] K. Ono, D. G. Austing, Y. Tokura, and S. Tarucha. “Current Rectification by Pauli Exclusion in a Weakly Coupled Double Quantum Dot System”. In: *Science* 297.5585 (Aug. 2002), pp. 1313–1317. DOI: 10.1126/science.1070958.
- [242] F. Hofmann et al. “Single Electron Switching in a Parallel Quantum Dot”. In: *Physical Review B* 51.19 (May 1995), pp. 13872–13875. DOI: 10.1103/PhysRevB.51.13872.
- [243] Leo P. Kouwenhoven, Charles M. Marcus, Paul L. McEuen, Seigo Tarucha, Robert M. Westervelt, and Ned S. Wingreen. “Electron Transport in Quantum Dots”. In: *Mesoscopic Electron Transport*. Ed. by Lydia L. Sohn, Leo P. Kouwenhoven, and Gerd Schön. Dordrecht: Springer Netherlands, 1997, pp. 105–214. ISBN: 978-94-015-8839-3. DOI: 10.1007/978-94-015-8839-3_4.
- [244] N. H. March. “Origins – The Thomas-Fermi Theory”. In: *Theory of the Inhomogeneous Electron Gas*. Ed. by S. Lundqvist and N. H. March. Physics of Solids and Liquids. Boston, MA: Springer US, 1983, pp. 1–77. ISBN: 978-1-4899-0415-7. DOI: 10.1007/978-1-4899-0415-7_1.

- [245] E. Merzbacher. *Quantum Mechanics*. Wiley, 1998. ISBN: 978-0-471-88702-7.
- [246] L. DiCarlo et al. “Differential Charge Sensing and Charge Delocalization in a Tunable Double Quantum Dot”. In: *Physical Review Letters* 92.22 (June 2004). DOI: 10.1103/PhysRevLett.92.226801.
- [247] J. R. Petta, A. C. Johnson, C. M. Marcus, M. P. Hanson, and A. C. Gossard. “Manipulation of a Single Charge in a Double Quantum Dot”. In: *Physical Review Letters* 93.18 (Oct. 2004). DOI: 10.1103/PhysRevLett.93.186802.
- [248] T. Hatano, M. Stopa, and S. Tarucha. “Single-Electron Delocalization in Hybrid Vertical-Lateral Double Quantum Dots”. In: *Science* 309.5732 (July 2005), pp. 268–271. DOI: 10.1126/science.1111205.
- [249] A. K. Hüttel, S. Ludwig, H. Lorenz, K. Eberl, and J. P. Kotthaus. “Direct Control of the Tunnel Splitting in a One-Electron Double Quantum Dot”. In: *Physical Review B* 72.8 (Aug. 2005). DOI: 10.1103/PhysRevB.72.081310.
- [250] M. Pioro-Ladrière et al. “Charge Sensing of an Artificial $\{\mathrm{H}\}_{2}^{\dagger}$ Molecule in Lateral Quantum Dots”. In: *Physical Review B* 72.12 (Sept. 2005). DOI: 10.1103/PhysRevB.72.125307.
- [251] L.-X. Zhang, D. V. Melnikov, and J.-P. Leburton. “Exchange Interaction and Stability Diagram of Coupled Quantum Dots in Magnetic Fields”. In: *Physical Review B* 74.20 (Nov. 2006). DOI: 10.1103/PhysRevB.74.205306.
- [252] L. Gaudreau et al. “Stability Diagram of a Few-Electron Triple Dot”. In: *Physical Review Letters* 97.3 (July 2006). DOI: 10.1103/PhysRevLett.97.036807.
- [253] Marek Korkusinski, Irene Puerto Gimenez, Pawel Hawrylak, Louis Gaudreau, Sergei A. Studenikin, and Andrew S. Sachrajda. “Topological Hund’s Rules and the Electronic Properties of a Triple Lateral Quantum Dot Molecule”. In: *Physical Review B* 75.11 (Mar. 2007). DOI: 10.1103/PhysRevB.75.115301.
- [254] Shuo Yang, Xin Wang, and S. Das Sarma. “Generic Hubbard Model Description of Semiconductor Quantum-Dot Spin Qubits”. In: *Physical Review B* 83.16 (Apr. 2011). DOI: 10.1103/PhysRevB.83.161301.
- [255] S. Das Sarma, Xin Wang, and Shuo Yang. “Hubbard Model Description of Silicon Spin Qubits: Charge Stability Diagram and Tunnel Coupling in Si Double Quantum Dots”. In: *Physical Review B* 83.23 (June 2011). DOI: 10.1103/PhysRevB.83.235314.
- [256] Xin Wang, Shuo Yang, and S. Das Sarma. “Quantum Theory of the Charge-Stability Diagram of Semiconductor Double-Quantum-Dot Systems”. In: *Physical Review B* 84.11 (Sept. 2011). DOI: 10.1103/PhysRevB.84.115301.
- [257] W. M. Witzel and S. Das Sarma. “Quantum Theory for Electron Spin Decoherence Induced by Nuclear Spin Dynamics in Semiconductor Quantum Computer Architectures: Spectral Diffusion of Localized Electron Spins in the Nuclear Solid-State Environment”. In: *Physical Review B* 74.3 (July 2006). DOI: 10.1103/PhysRevB.74.035322.

- [258] J. M. Taylor, J. R. Petta, A. C. Johnson, A. Yacoby, C. M. Marcus, and M. D. Lukin. “Relaxation, Dephasing, and Quantum Control of Electron Spins in Double Quantum Dots”. In: *Physical Review B* 76.3 (July 2007). DOI: 10.1103/PhysRevB.76.035315.
- [259] Irene Puerto Gimenez, Chang-Yu Hsieh, Marek Korkusinski, and Pawel Hawrylak. “Charged-Impurity-Induced Dephasing of a Voltage-Controlled Coded Qubit Based on Electron Spin in a Triple Quantum Dot”. In: *Physical Review B* 79.20 (May 2009). DOI: 10.1103/PhysRevB.79.205311.
- [260] Nga T. T. Nguyen and S. Das Sarma. “Impurity Effects on Semiconductor Quantum Bits in Coupled Quantum Dots”. In: *Physical Review B* 83.23 (June 2011). DOI: 10.1103/PhysRevB.83.235322.
- [261] Markus J. Storz, Udo Hartmann, Sigmund Kohler, and Frank K. Wilhelm. “Intrinsic Phonon Decoherence and Quantum Gates in Coupled Lateral Quantum-Dot Charge Qubits”. In: *Physical Review B* 72.23 (Dec. 2005). DOI: 10.1103/PhysRevB.72.235321.
- [262] V. N. Stavrou and Xuedong Hu. “Charge Decoherence in Laterally Coupled Quantum Dots Due to Electron-Phonon Interactions”. In: *Physical Review B* 72.7 (Aug. 2005). DOI: 10.1103/PhysRevB.72.075362.
- [263] Giovanni A. Oakes, Jingyu Duan, John J. L. Morton, Alpha Lee, Charles G. Smith, and M. Fernando Gonzalez Zalba. “Automatic Virtual Voltage Extraction of a 2x2 Array of Quantum Dots with Machine Learning”. In: (May 2021). DOI: 10.48550/arXiv.2012.03685.
- [264] Weijie Li et al. “Charge States, Triple Points, and Quadruple Points in an InAs Nanowire Triple Quantum Dot Revealed by an Integrated Charge Sensor”. In: *Advanced Quantum Technologies* 6.5 (2023). ISSN: 2511-9044. DOI: 10.1002/qute.202200158.
- [265] J Darulová, M Troyer, and M C Cassidy. “Evaluation of synthetic and experimental training data in supervised machine learning applied to charge-state detection of quantum dots”. In: *Machine Learning: Science and Technology* 2.4 (Sept. 2021). DOI: 10.1088/2632-2153/ac104c.
- [266] Oswin Krause, Anasua Chatterjee, Ferdinand Kuemmeth, and Evert Van Nieuwenburg. “Learning Coulomb Diamonds in Large Quantum Dot Arrays”. In: *SciPost Physics* 13.4 (Oct. 2022). ISSN: 2542-4653. DOI: 10.21468/SciPostPhys.13.4.084.
- [267] Joshua Ziegler et al. “Toward Robust Autotuning of Noisy Quantum dot Devices”. In: *Physical Review Applied* 17.2 (Feb. 2022). DOI: 10.1103/PhysRevApplied.17.024069.
- [268] Stefan Birner et al. “Nextnano: General Purpose 3-D Simulations”. In: *IEEE Transactions on Electron Devices* 54.9 (Sept. 2007), pp. 2137–2142. ISSN: 1557-9646. DOI: 10.1109/TED.2007.902871.
- [269] Gediminas Kiranskas, Jonas Nyvold Pedersen, Olov Karlström, Martin Leijnse, and Andreas Wacker. “QmeQ 1.0: An Open-Source Python Package for Calculations of Transport through Quantum Dot Devices”. In: *Computer Physics Communications* 221 (Dec. 2017), pp. 317–342. ISSN: 0010-4655. DOI: 10.1016/j.cpc.2017.07.024.

- [270] Tsutomu Ikegami, Koichi Fukuda, Junichi Hattori, Hidehiro Asai, and Hiroyuki Ota. “A TCAD Device Simulator for Exotic Materials and Its Application to a Negative-Capacitance FET”. In: *Journal of Computational Electronics* 18.2 (June 2019), pp. 534–542. ISSN: 1572-8137. DOI: 10.1007/s10825-019-01313-7.
- [271] Hidehiro Asai et al. “Development of Integrated Device Simulator for Quantum Bit Design: Self-consistent Calculation for Quantum Transport and Qubit Operation”. In: *2021 5th IEEE Electron Devices Technology & Manufacturing Conference (EDTM)*. Apr. 2021, pp. 1–3. DOI: 10.1109/EDTM50988.2021.9420978.
- [272] Félix Beaudoin et al. “Robust Technology Computer-Aided Design of Gated Quantum Dots at Cryogenic Temperature”. In: *Applied Physics Letters* 120.26 (June 2022). ISSN: 0003-6951. DOI: 10.1063/5.0097202.
- [273] C. Wasshuber, H. Kosina, and S. Selberherr. “SIMON-A Simulator for Single-Electron Tunnel Devices and Circuits”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 16.9 (Sept. 1997), pp. 937–944. ISSN: 1937-4151. DOI: 10.1109/43.658562.
- [274] Pieter Eendebak. *Qtt Documentation*. QuTech. Delft, Jan. 2023. URL: https://qtt.readthedocs.io/_/downloads/en/dev/pdf (visited on 05/05/2023).
- [275] Khromets, Bohdan. “Heterostructure development and quantum control for semiconductor qubits”. MA thesis. University of Waterloo, 2022.
- [276] Sarah Fleitmann. *Characterization of Distortions in Charge Stability Diagrams and their Simulation in Modeled Data*. Tech. rep. Forschungszentrum Jülich GmbH Zentralbibliothek, Verlag, 2023. DOI: 10.34734/FZJ-2023-05704.
- [277] D. Maradan et al. “GaAs Quantum Dot Thermometry Using Direct Transport and Charge Sensing”. In: *Journal of Low Temperature Physics* 175.5 (June 2014), pp. 784–798. ISSN: 1573-7357. DOI: 10.1007/s10909-014-1169-6.
- [278] S. Yuan et al. “The Characterization of Electronic Noise in the Charge Transport through Single-Molecule Junctions”. In: *Small Methods* 5.3 (Mar. 2021). ISSN: 2366-9608, 2366-9608. DOI: 10.1002/smt.202001064.
- [279] J. Timmer and M. König. “On generating power law noise.” In: *Astronomy and Astrophysics* 300 (Aug. 1995), pp. 707–710. ISSN: 0004-6361.
- [280] F. Patzelt. *colorednoise.py*. 2017. URL: <https://github.com/felixpatzelt/colorednoise> (visited on 06/07/2022).
- [281] X. Chen et al. “Modeling Random Telegraph Noise as a Randomness Source and its Application in True Random Number Generation”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 35.9 (Sept. 2016), pp. 1435–1448. ISSN: 0278-0070, 1937-4151. DOI: 10.1109/TCAD.2015.2511074.
- [282] J. B. Johnson. “Thermal Agitation of Electricity in Conductors”. In: *Phys. Rev.* 32 (1 July 1928), pp. 97–109. DOI: 10.1103/PhysRev.32.97.
- [283] W. Schottky. “Über spontane Stromschwankungen in verschiedenen Elektrizitätsleitern”. German. In: (Jan. 1918). DOI: 10.1002/andp.19183622304.

- [284] Fabian Hader. *f-hader/SimCATS: SimCATS v1.2.0*. Version v1.2.0. Sept. 2024. DOI: 10.5281/zenodo.13805234.
- [285] P. Welch. “The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms”. In: *IEEE Transactions on Audio and Electroacoustics* 15.2 (1967), pp. 70–73. DOI: 10.1109/TAU.1967.1161901.
- [286] P. Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2.
- [287] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. “Improved techniques for training GANs”. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems. NIPS’16*. Barcelona, Spain: Curran Associates Inc., 2016, pp. 2234–2242. ISBN: 978-1-5108-3881-9.
- [288] M. Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. “GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium”. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS’17*. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6629–6640. ISBN: 978-1-5108-6096-4.
- [289] Ahmed Alaa, Boris Van Breugel, Evgeny S. Saveliev, and Mihaela van der Schaar. “How Faithful is your Synthetic Data? Sample-level Metrics for Evaluating and Auditing Generative Models”. In: *Proceedings of the 39th International Conference on Machine Learning*. Ed. by Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato. Vol. 162. Proceedings of Machine Learning Research. July 2022, pp. 290–306.
- [290] L. Ruff. *PyTorch Implementation of Deep SVDD*. 2018. URL: <https://github.com/lukasruff/Deep-SVDD-PyTorch> (visited on 08/12/2022).
- [291] L. Ruff et al. “Deep One-Class Classification”. In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. July 2018, pp. 4393–4402.
- [292] Alexander Buslaev, Vladimir I. Iglovikov, Eugene Khvedchenya, Alex Parinov, Mikhail Druzhinin, and Alexandr A. Kalinin. “Albumentations: Fast and Flexible Image Augmentations”. In: *Information* 11.2 (Feb. 2020), p. 125. ISSN: 2078-2489. DOI: 10.3390/info11020125.
- [293] Li Deng. “The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of the Web]”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142. DOI: 10.1109/MSP.2012.2211477.
- [294] Fabian Hader, Sarah Fleitmann, Jan Vogelbruch, Lotte Geck, and Stefan van Waasen. “Simulation of Charge Stability Diagrams for Automated Tuning Solutions (SimCATS)”. In: *IEEE Transactions on Quantum Engineering* 5 (2024), pp. 1–14. DOI: 10.1109/TQE.2024.3445967.

-
- [295] Hader Fabian. *f-hader/SimCATS-Datasets: SimCATS-Datasets v2.5.0*. Version v2.5.0. Oct. 2024. DOI: 10.5281/zenodo.13909180.
- [296] Tom Struck et al. “Spin-EPR-pair separation by conveyor-mode single electron shuttling in Si/SiGe”. In: *Nature Communications* 15.1 (Feb. 2024), p. 1325. ISSN: 2041-1723. DOI: 10.1038/s41467-024-45583-7.
- [297] Silvia Seidlitz et al. “Robust deep learning-based semantic organ segmentation in hyperspectral images”. In: *Medical Image Analysis* 80 (Aug. 2022). ISSN: 1361-8415. DOI: 10.1016/j.media.2022.102488.
- [298] Xiaoju Ye. *calflows: a FLOPs and Params calculate tool for neural networks in pytorch framework*. 2023. URL: <https://github.com/MrYxJ/calculate-flops.pytorch> (visited on 02/13/2024).
- [299] Benedikt Scherer. *Evaluation of Line Detection Methods for the Analysis of Charge Stability Diagrams*. Tech. rep. 4441. Jülich, 2023. DOI: 10.34734/FZJ-2023-02705.
- [300] *DFF and CASENet source code*. URL: <https://github.com/Lavender105/DFF> (visited on 02/26/2024).
- [301] *CHRNet source code*. URL: <https://github.com/elharroussomar/Refined-Edge-Detection-With-Cascaded-and-High-Resolution-Convolutional-Network> (visited on 03/22/2024).
- [302] Pavel Iakubovskii. *Segmentation Models Pytorch*. 2019. URL: https://github.com/qubvel/segmentation_models.pytorch (visited on 07/03/2024).
- [303] *LDC source code*. URL: <https://github.com/xavysp/LDC> (visited on 03/25/2024).
- [304] *TEED source code*. URL: <https://github.com/xavysp/TEED> (visited on 05/16/2024).
- [305] Alexandre Milesi. *U-Net source code*. URL: <https://github.com/milesial/Pytorch-UNet> (visited on 01/25/2024).
- [306] *CrackFormer source code*. URL: <https://github.com/LouisNUST/CrackFormer-II> (visited on 03/22/2024).
- [307] *EDTER source code*. URL: <https://github.com/MengyangPu/EDTER> (visited on 01/20/2024).
- [308] *MMViT-Seg source code*. URL: <https://github.com/yywbkn/MMViT-Segmentation> (visited on 04/17/2024).
- [309] *SegFormer source code*. URL: <https://github.com/NVlabs/SegFormer> (visited on 01/22/2024).
- [310] *Segmenter source code*. URL: <https://github.com/rstrudel/segmenter> (visited on 02/26/2024).
- [311] *Swin-Unet source code*. URL: <https://github.com/HuCaoFighting/Swin-Unet> (visited on 02/15/2024).
- [312] *TransUNet source code*. URL: <https://github.com/Beckschen/TransUNet> (visited on 02/22/2024).

- [313] *VM-UNet source code*. URL: <https://github.com/JCruan519/VM-UNet> (visited on 02/13/2024).
- [314] *DiffusionEdge source code*. URL: <https://github.com/GuHuangAI/DiffusionEdge> (visited on 02/26/2024).
- [315] *MedSegDiff-V2 source code*. URL: <https://github.com/MedicineToken/MedSegDiff> (visited on 04/22/2024).
- [316] *Canny edge detector software interface*. URL: <https://scikit-image.org/docs/stable/api/skimage.feature.html#skimage.feature.canny> (visited on 07/29/2024).
- [317] *CannyPF software interface*. URL: <https://cvrs.whu.edu.cn/cannylines/#ct1> (visited on 07/29/2024).
- [318] *Edge Drawing software interface*. URL: <https://github.com/shaojunluo/EDLinePython> (visited on 07/29/2024).
- [319] *gPb software interface*. URL: <https://github.com/HiDiYANG/gPb-GSoC> (visited on 07/29/2024).
- [320] *Phase Congruency software interface*. URL: <https://github.com/peterkovesi/ImagePhaseCongruency.jl> (visited on 07/29/2024).
- [321] Xiaohu Lu, Jian Yao, Kai Li, and Li Li. “CannyLines: A parameter-free line segment detector”. In: *2015 IEEE International Conference on Image Processing (ICIP)*. 2015, pp. 507–511. DOI: 10.1109/ICIP.2015.7350850.
- [322] Michael Maire, Pablo Arbelaez, Charless Fowlkes, and Jitendra Malik. “Using contours to detect and localize junctions in natural images”. In: *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on* (June 2008). DOI: 10.1109/CVPR.2008.4587420.
- [323] M.C. Morrone and R.A. Owens. “Feature detection from local energy”. In: *Pattern Recognition Letters* 6.5 (1987), pp. 303–313. ISSN: 0167-8655. DOI: 10.1016/0167-8655(87)90013-4.
- [324] Peter Kovesei. “Image Features from Phase Congruency”. In: *Videre: Journal of Computer Vision Research* 1.3 (1999), pp. 1–26.
- [325] Cuneyt Akinlar and Cihan Topal. “EDLines: A real-time line segment detector with a false detection control”. In: *Pattern Recognition Letters* 32.13 (2011), pp. 1633–1642. ISSN: 0167-8655. DOI: 10.1016/j.patrec.2011.06.001.
- [326] Omar Elharrouss, Youssef Hmamouche, Assia Kamal Idrissi, Btissam El Khamlichi, and Amal El Fallah-Seghrouchni. “Refined edge detection with cascaded and high-resolution convolutional network”. In: *Pattern Recognition* 138 (2023). ISSN: 0031-3203. DOI: 10.1016/j.patcog.2023.109361.
- [327] Xavier Soria, Gonzalo Pomboza-Junez, and Angel Domingo Sappa. “LDC: Lightweight Dense CNN for Edge Detection”. In: *IEEE Access* 10 (2022), pp. 68281–68290. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3186344.

- [328] Xavier Soria, Angel Sappa, Patricio Humanante, and Arash Akbarinia. “Dense extreme inception network for edge detection”. In: *Pattern Recognition* 139 (2023). ISSN: 0031-3203. DOI: 10.1016/j.patcog.2023.109461.
- [329] Linxi Huan, Nan Xue, Xianwei Zheng, Wei He, Jianya Gong, and Gui-Song Xia. “Unmixing Convolutional Features for Crisp Edge Detection”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 44.10_Part_1 (Oct. 2022), pp. 6602–6609. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2021.3084197.
- [330] Xavier Soria, Yachuan Li, Mohammad Rouhani, and Angel D. Sappa. “Tiny and Efficient Model for the Edge Detection Generalization”. In: *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*. 2023, pp. 1356–1365. DOI: 10.1109/ICCVW60793.2023.00147.
- [331] Zhiding Yu, Chen Feng, Ming-Yu Liu, and Srikumar Ramalingam. “CASNet: Deep Category-Aware Semantic Edge Detection”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1761–1770. DOI: 10.1109/CVPR.2017.191.
- [332] Yuan Hu, Yunpeng Chen, Xiang Li, and Jiashi Feng. “Dynamic feature fusion for semantic edge detection”. In: *Proceedings of the 28th International Joint Conference on Artificial Intelligence. IJCAI’19*. Macao, China: AAAI Press, 2019, pp. 782–788. ISBN: 978-0-9992411-4-1.
- [333] Zongwei Zhou, Md Mahfuzur Rahman Siddiquee, Nima Tajbakhsh, and Jianming Liang. “UNet++: A Nested U-Net Architecture for Medical Image Segmentation”. In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*. Ed. by Danail Stoyanov et al. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 3–11. ISBN: 978-3-030-00889-5. DOI: 10.1007/978-3-030-00889-5_1.
- [334] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. “Feature Pyramid Networks for Object Detection”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, 2017, pp. 936–944. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.106.
- [335] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. “Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation”. In: *Computer Vision – ECCV 2018*. Ed. by Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss. Cham: Springer International Publishing, 2018, pp. 833–851. ISBN: 978-3-030-01234-2. DOI: 10.1007/978-3-030-01234-2_49.
- [336] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. “Rethinking Atrous Convolution for Semantic Image Segmentation”. In: (2017). DOI: 10.48550/arXiv.1706.05587.

- [337] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. “DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40.4 (2018), pp. 834–848. DOI: 10.1109/TPAMI.2017.2699184.
- [338] Abhishek Chaurasia and Eugenio Culurciello. “LinkNet: Exploiting encoder representations for efficient semantic segmentation”. In: *2017 IEEE Visual Communications and Image Processing (VCIP)*. Dec. 2017. DOI: 10.1109/vcip.2017.8305148.
- [339] Ashish Vaswani et al. “Attention is all you need”. In: NIPS’17. Long Beach, California, USA: Curran Associates Inc., 2017, pp. 6000–6010. ISBN: 978-1-5108-6096-4.
- [340] Alexey Dosovitskiy et al. “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale”. In: *ICLR* (2021).
- [341] Tongle Fan, Guanglei Wang, Yan Li, and Hongrui Wang. “MA-Net: A Multi-Scale Attention Network for Liver and Tumor Segmentation”. In: *IEEE Access* 8 (2020), pp. 179656–179665. DOI: 10.1109/ACCESS.2020.3025372.
- [342] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39.12 (2017), pp. 2481–2495. DOI: 10.1109/TPAMI.2016.2644615.
- [343] Ze Liu et al. “Swin Transformer: Hierarchical Vision Transformer using Shifted Windows”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 9992–10002. DOI: 10.1109/ICCV48922.2021.00986.
- [344] Albert Gu, Karan Goel, and Christopher Ré. “Efficiently Modeling Long Sequences with Structured State Spaces”. In: (2022). DOI: 10.48550/arXiv.2111.00396.
- [345] Junde Wu et al. “MedSegDiff: Medical Image Segmentation with Diffusion Probabilistic Model”. In: *Medical Imaging with Deep Learning*. Ed. by Ipek Oguz et al. Vol. 227. Proceedings of Machine Learning Research. July 2024, pp. 1623–1639.
- [346] Junde Wu, Wei Ji, Huazhu Fu, Min Xu, Yueming Jin, and Yanwu Xu. “MedSegDiff-V2: diffusion-based medical image segmentation with transformer”. In: *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI’24/IAAI’24/EAAI’24. AAAI Press, 2024. ISBN: 978-1-57735-887-9. DOI: 10.1609/aaai.v38i6.28418.
- [347] Adam Paszke et al. “PyTorch: an imperative style, high-performance deep learning library”. In: *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2019.
- [348] Leslie N. Smith and Nicholay Topin. “Super-convergence: very fast training of neural networks using large learning rates”. In: *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*. Ed. by Tien Pham. Vol. 11006. SPIE, 2019. DOI: 10.1117/12.2520589.

- [349] Fabian Hader. *SimCATS_GaAs_v1_random_variations_v2*. Zenodo, Oct. 2024. DOI: 10.5281/zenodo.13903285.
- [350] V. Nguyen et al. “Deep Reinforcement Learning for Efficient Measurement of Quantum Devices”. In: *npj Quantum Information* 7.1 (June 18, 2021), pp. 1–9. ISSN: 2056-6387. DOI: 10.1038/s41534-021-00434-x.
- [351] Shyam Nandan Reddy Uppuluru. “Automated Coarse Tuning of Quantum Sensor Dots”. MA thesis. Göttingen: Georg-August-Universität, Institute of Computer Science, Oct. 2021.
- [352] Fabian Hader. “Noise Analysis and Estimation in Sensor Dot Fine Scans for Automated Tuning of Gate Defined Quantum Dots”. MA thesis. Jülich, Germany: FH Aachen, University of Applied Sciences, Sept. 2021.
- [353] S Kafanov, H Brenning, T Duty, and P Delsing. “Charge Noise in Single-Electron Transistors and Charge Qubits May Be Caused by Metallic Grains”. In: *Phys. Rev. B* 78.12 (Sept. 2008). DOI: 10.1103/PhysRevB.78.125411.
- [354] Christian Grewing, Jonas Bühler, Arun Ashok, and Stefan van Waasen. “Auslesevorrichtung und Verfahren zum zeitversetzten Auslesen mehrerer Quanteneinrichtungen, insbesondere Quantenpunkte oder Quantenbits, sowie Quantenvorrichtung”. Deutsche Patentanmeldung Nr. 10 2022 124 861.5 (Germany). Sept. 2022.
- [355] Shahanur Alam, Chris Yakopcic, Qing Wu, Mark Barnell, Simon Khan, and Tarek M. Taha. “Survey of Deep Learning Accelerators for Edge and Emerging Computing”. In: *Electronics* 13.15 (Jan. 2024), p. 2988. ISSN: 2079-9292. DOI: 10.3390/electronics13152988.
- [356] T. Hui Teo, Huan-Ke Hsu, Yu-Chiau Chen, and I-Chyn Wei. “U-Net Hardware Accelerator”. In: *2024 IEEE 17th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoc)*. 2024, pp. 345–348. DOI: 10.1109/MCSoc64144.2024.00063.
- [357] Bert Moons, Roel Uytterhoeven, Wim Dehaene, and Marian Verhelst. “14.5 Envision: A 0.26-to-10TOPS/W subword-parallel dynamic-voltage-accuracy-frequency-scalable Convolutional Neural Network processor in 28nm FDSOI”. In: *2017 IEEE International Solid-State Circuits Conference (ISSCC)*. 2017, pp. 246–247. DOI: 10.1109/ISSCC.2017.7870353.
- [358] Markus Nagel, Marios Fournarakis, Rana Ali Amjad, Yelysei Bondarenko, Mart van Baalen, and Tijmen Blankevoort. “A White Paper on Neural Network Quantization”. In: (June 2021). DOI: 10.48550/arXiv.2106.08295.
- [359] Olivia Weng. “Neural Network Quantization for Efficient Inference: A Survey”. In: (Jan. 2023). DOI: 10.48550/arXiv.2112.06126.
- [360] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. “Incremental Network Quantization: Towards Lossless CNNs with Low-Precision Weights”. In: (Aug. 2017). DOI: 10.48550/arXiv.1702.03044.

- [361] Bluefors. *XLDsl Dilution Refrigerator Measurement System*. Mar. 2025. URL: <https://bluefors.com/products/dilution-refrigerator-measurement-systems/xldsl-dilution-refrigerator-measurement-system/> (visited on 03/04/2025).
- [362] Benjamin Papajewski. "Sensor Scan Simulation for Automated Tuning of Gate-Defined Semiconductor Quantum Dots". MA thesis. Jülich, Germany: FH Aachen, University of Applied Sciences, Nov. 2024.
- [363] F. J. Richards. "A Flexible Growth Function for Empirical Use". In: *Journal of Experimental Botany* 10.2 (June 1959), pp. 290–301. ISSN: 0022-0957. DOI: 10.1093/jxb/10.2.290.
- [364] Stefanie Czischek et al. "Miniaturizing neural networks for charge state autotuning in quantum dots". In: *Machine Learning: Science and Technology* 3.1 (Mar. 2022). ISSN: 2632-2153. DOI: 10.1088/2632-2153/ac34db.
- [365] Pengzhen Ren et al. "A Comprehensive Survey of Neural Architecture Search: Challenges and Solutions". In: *ACM Comput. Surv.* 54.4 (May 2021). ISSN: 0360-0300. DOI: 10.1145/3447582.
- [366] Hadjer Benmeziane, Kaoutar El Maghraoui, Hamza Ouarnoughi, Smail Niar, Martin Wistuba, and Naigang Wang. "A Comprehensive Survey on Hardware-Aware Neural Architecture Search". In: (Jan. 2021). DOI: 10.48550/arXiv.2101.09336.
- [367] Shashank Shekhar, Adesh Bansode, and Asif Salim. "A Comparative study of Hyper-Parameter Optimization Tools". In: *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*. Los Alamitos, CA, USA, Dec. 2021, pp. 1–6. DOI: 10.1109/CSDE53843.2021.9718485.

Publications Related to This Work

Parts of this doctoral thesis have been published. The following list gives an overview of current publications in which parts of this doctoral thesis have been published.

Journal Publications

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

Fabian Hader, Sarah Fleitmann, Jan Vogelbruch, Lotte Geck, and Stefan van Waasen. “Simulation of Charge Stability Diagrams for Automated Tuning Solutions (SimCATS)”. in: *IEEE Transactions on Quantum Engineering* 5 (2024), pp. 1–14. DOI: [10.1109/TQE.2024.3445967](https://doi.org/10.1109/TQE.2024.3445967)

Justyna P. Zwolak, Jacob M. Taylor, Reed W. Andrews, Jared Benson, Garnett W. Bryant, Donovan Buterakos, Anasua Chatterjee, Sankar Das Sarma, Mark A. Eriksson, Erika Greplová, Michael J. Gullans, Fabian Hader, Tyler J. Kovach, Pranav S. Mundada, Mick Ramsey, Torbjørn Rasmussen, Brandon Severin, Anthony Sigillito, Brennan Undseth, and Brian Weber. “Data needs and challenges for quantum dot devices automation”. In: *npj Quantum Information* 10.1 (Oct. 2024), pp. 1–8. ISSN: 2056-6387. DOI: [10.1038/s41534-024-00878-x](https://doi.org/10.1038/s41534-024-00878-x)

Fabian Hader, Fabian Fuchs, Sarah Fleitmann, Karin Havemann, Benedikt Scherer, Jan Vogelbruch, Lotte Geck, and Stefan van Waasen. “Automated Charge Transition Detection in Quantum Dot Charge Stability Diagrams”. In: *IEEE Transactions on Quantum Engineering* 6 (2025), pp. 1–14. DOI: [10.1109/TQE.2025.3596392](https://doi.org/10.1109/TQE.2025.3596392)

Conferences

Fabian Hader, Sarah Fleitmann, Jan Vogelbruch, Lotte Geck, and Stefan van Waasen. “A Geometric Approach for the Modelling of Charge Stability Diagrams”. In: NIST Workshop on Advances in Automation of Quantum Dot Devices Control, Rockville (USA), July 19-20, 2023. URL: <https://juser.fz-juelich.de/record/1009677>

Fabian Hader, Fabian Fuchs, Sarah Fleitmann, Karin Havemann, Benedikt Scherer, Jan Vogelbruch, Lotte Geck, and Stefan van Waasen. “Quantum Dot CSD Simulation and Automated Charge Transition Detection”. In: SpinQubit 6, Sydney (Australia), Nov. 4-8, 2024. DOI: [10.34734/FZJ-2024-06298](https://doi.org/10.34734/FZJ-2024-06298)

Fabian Hader, Fabian Fuchs, Sarah Fleitmann, Karin Havemann, Benedikt Scherer, Jan Vogelbruch, Simon Humpohl, Tobias Hangleiter, Till Huckemann, Lotte Geck, and Stefan van Waasen. "Towards Scalable Robust Charge Transition Detection for Quantum Dot Devices". In: *Advances in Automation of Quantum Dot Devices Control*, Los Angeles (USA). Oct. 5, 2025. DOI: [10.34734/FZJ-2025-04195](https://doi.org/10.34734/FZJ-2025-04195)

Fabian Hader, Fabian Fuchs, Sarah Fleitmann, Karin Havemann, Benedikt Scherer, Jan Vogelbruch, Lotte Geck, and Stefan van Waasen. "Towards Scalable Cryogenic Charge Transition Detection for Automated Quantum Dot Tuning". In: *Silicon Quantum Electronics Workshop*, Los Angeles (USA), Oct. 6-8, 2025. DOI: [10.34734/FZJ-2025-04196](https://doi.org/10.34734/FZJ-2025-04196)

Patent Application

Fabian Hader and Jan Vogelbruch. "Verfahren zur Ermittlung wenigstens eines Gesamtladungsüberganges eines Ladungsstabilitätsdiagramms, Computerprogramm und computerlesbares Medium". Deutsche Patentanmeldung Nr. 10 2023 118 751.1 (Germany). July 2023

Curriculum Vitae

Fabian António Hader

Personal data

Date of birth September 10, 1997
Place of birth Neuss, Germany
Nationality German

Education

University of Duisburg-Essen, Duisburg, Germany

PhD Student, Electrical Engineering

December 2021 – May 2025

FH Aachen – University of Applied Sciences, Jülich, Germany

M.Sc., Energy Economics & Informatics

September 2019 – November 2021

B.Sc., Scientific Programming

September 2016 – August 2019

Professional Experience

Central Institute of Engineering, Electronics and Analytics – Electronic Systems (ZEA-2) / Peter Grünberg Institute – Integrated Computing Architectures - ICA (PGI-4), Forschungszentrum Jülich GmbH, Jülich, Germany

Doctoral Researcher

December 2021 – May 2025

Scientific Assistant

September 2019 – November 2021

Mathematical-Technical Software Engineer

September 2018 – August 2019

Projekträger Jülich, Forschungszentrum Jülich GmbH, Jülich, Germany

Mathematical-Technical Software Engineer

September 2016 – August 2018

Band / Volume 108

Spin selectivity of chiral molecules on surfaces

M.R. Safari (2025), xiv, 165 pp

ISBN: 978-3-95806-810-0

Band / Volume 109

Redox-based Random Access Memory Arrays for Computing-In-Memory and Neuromorphic Computing

H. Chen (2025), x, 154 pp

ISBN: 978-3-95806-814-8

Band / Volume 110

Mechanics of deep neural networks beyond the Gaussian limit

K. Fischer (2025), xvi, 138 pp

ISBN: 978-3-95806-815-5

Band / Volume 111

Characteristics of plastically deformed *fcc* and *bcc* High-Entropy Alloys

T. Meenen (2025), x, 115 pp

ISBN: 978-3-95806-820-9

Band / Volume 112

Software-Configurable Analog-To-Digital Converters for Configurable Pulse Detection

L. Krystofiak (2025), xvii, 113 pp, xxix

ISBN: 978-3-95806-826-1

Band / Volume 113

Development of Superparamagnetic Based Biological Sensor for the Detection of Brucella DNA Using Frequency Mixing Magnetic Detection

A. Abuawad (2025), X, 129 pp

ISBN: 978-3-95806-836-0

Band / Volume 114

A System for the Cryogenic Power Management of Quantum Computing Electronics: Development, Integration, and Test

A. R. Cabrera Galicia (2025), xxv, 110, lviii pp

ISBN: 978-3-95806-844-5

Band / Volume 115

Investigation of 2D Materials using Low Energy Electron Microscopy (LEEM)

H. Yin (2025) viii, 137 pp

ISBN: 978-3-95806-848-3

Band / Volume 116

Topotactic phase transition in $\text{La}_{0.6}\text{Sr}_{0.4}\text{CoO}_{3-\delta}$ thin films: oxygen content, dynamics and reversibility

S. He (2025) ix, 137 pp

ISBN: 978-3-95806-868-1

Band / Volume 117

Electrical anisotropy and shear-resistant topology in the quasi one-dimensional van-der-Waals material $\alpha\text{-Bi}_4\text{Br}_4$

J.K. Hofmann (2025) xv, 129 pp

ISBN: 978-3-95806-869-8

Band / Volume 118

Spin-orbital mixing in the topological ladder of the two-dimensional metal PtTe_2

M. Qahosh (2025), ix, 170 pp

ISBN: 978-3-95806-872-8

Band / Volume 119

Functions of SNNs constrained by biology

A. Korcsak-Gorzo (2025), xvi, 145 pp

ISBN: 978-3-95806-876-6

Band / Volume 120

On Scalable Integrated Charge State Tuning for Semiconductor Quantum Dot Devices

F. A. Hader (2026), xx, 145 pp

ISBN: 978-3-95806-884-1

Information

Band / Volume 120

ISBN 978-3-95806-884-1

Mitglied der Helmholtz-Gemeinschaft

