



Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Renato Durrer

Automated Tuning of Double Quantum Dot Charge States

Master Thesis

Swiss Federal Institute of Technology (ETH) Zurich

Supervision

Benedikt Kratochwil, Dr. Jonne Koski, Dr. Eliska Greplova
Prof. Dr. Thomas Ihn

September 2019

Contents

Abstract	ii
1 Introduction	2
2 Quantum Dot Physics	4
2.1 Quantum Dots	4
2.1.1 Capacitance Model	7
2.1.2 Charge Stability Diagrams	9
3 Experimental Setup and Sample Characterization	12
3.1 Sample	12
3.2 Measurement Setup	12
3.3 Temperature Measurement	13
3.4 QPC as a Charge Detector	14
4 Machine Learning	18
4.1 Machine Learning Fundamentals	18
4.2 Machine Learning Tuning Algorithms	20
4.2.1 Finding a Reference Point	21
4.2.2 Recognizing Charge Transitions	22
5 Results	25
5.1 Machine Learning Models	25
5.2 Performance of the Auto-Tuner	28
5.3 Alternative Approach	35
5.3.1 Unsupervised Clustering	35
5.3.2 Clustering Charge Induced Current Offset	36
6 Conclusion and Outlook	40
A Data and Learning	42
A.1 Data Acquisition	42
A.2 Data Augmentation and Data Processing	42
A.3 Neural Network Architectures	44
A.4 Classification Results	47
B Automated QPC Calibration	50

Abstract

Quantum dots are extensively studied systems in semiconductor physics and are a promising candidate for scalable qubits. However, gate defined quantum dots - and thus the qubits - require careful tuning of the voltages applied to the individual gates in order to set physical parameters such as the charge occupation of the quantum dots. Tuning is intricate and lengthy and will require automation for scaled systems. In this work, we present a machine learning assisted auto-tuner capable of tuning to any charge occupation given a coarsely tuned double quantum dot. We employ two neural networks for this task, the first determines whether the double quantum dot is empty for a set of gate voltages, the second recognizes charge transitions within two points of the charge stability diagram. For training and validation of the networks we developed a ready-to-use data acquisition, data augmentation and data processing pipeline. The evaluation of the tuner on two GaAs/AlGaAs double quantum dot implementations yields a success rate of 57%. Given the limitations of the supervised learning approach we provide an unsupervised, clustering-based alternative, which does not require pre-training and can perform classifications given single measurement points. The clustering approach is scalable and can be further generalized and employed in practice in potential follow-up work.

Acronyms and Abbreviations

QD	Quantum Dot
DQD	Double Quantum Dot
QPC	Quantum Point Contact
MBE	Molecular Beam Epitaxy
ML	Machine Learning
CNN	Convolutional Neural Network
GMM	Gaussian Mixture Model
CICO	Charge Induced Current Offset
ETH	Eidgenössische Technische Hochschule

Chapter 1

Introduction

Recent developments in quantum dot (QD) physics has allowed for various implementations of semiconductor qubits. [1, 2, 3]. Proposals for scalable architectures have been made [4] and eventually, practical quantum computers should reach thousands to millions of qubits. Semiconductor qubits based on QDs are a promising candidate for quantum computers since, in contrast to many other qubit implementations [5, 6, 7], chips containing many gate defined quantum dots, and thus many qubits, can be fabricated using already existing fabrication techniques [8, 9]. Each of the gate defined QDs (and thus qubits) have several gates whose potentials must be carefully set in order to form the qubits in their operation regime. Examples of such QD qubits are the singlet-triplet qubit [10, 11, 12], the resonant exchange qubit [2, 13], the quadrupolar exchange-qubit [14] and the hybrid qubit [15, 16]. All of them rely on having the right number of electrons in each QD.

On the other hand, semiconductor QD systems are also extensively used to study the physics of solids such as the quantum Hall effect [17] or spin effects of electrons and holes [18]. In order to initialize these experiments, the QDs have to be tuned requiring a suitable set of gate voltages. However, a physicist would much prefer to specify a set of physical parameters instead, such as for example the interdot coupling or the number of electrons on the QDs. An auto-tuner capable of translating a set of physical parameters into a corresponding set of gate voltages would save a lot of time and largely simplify the use of QDs.

Tuning QDs is challenging because of the large parameter space, relatively slow measurements, and characterization of the device quality for given parameters. Up to now, the intricate task of tuning QDs is performed by human operators, using their knowledge and experience. However, due to time requirement and the complexity of the task, human operators will not be able to tune large scale semiconductor QD architectures and automated tuning is required.

Conventionally, the tuning is split into two stages. The first stage, called coarse tuning, is the forming of QD systems. The second step, referred to as fine tuning, is the adjustment of physical parameters such as tunnel couplings between QDs or electron occupation of the QDs [19]. Recently, attempts to automate each of the two steps have been made. Most of these attempts address the interdot tunnel couplings of a double quantum dot (DQD) and the tunnel coupling between QD and leads [20, 21, 22]. A computer automated tuner forming a DQD in the single electron regime for specific architectures has been presented [23]. Machine learning models are a promising tool for automated tuning as they can take over the task of the human operators consisting of interpreting measurements. Such models have been implemented to tune the interdot tunnel couplings of a DQD as well as the tunnel couplings between QDs and leads [19]. Also, coarse tuner capable of differentiating between single and double QDs using convolutional neural networks (CNNs) were proposed but not yet implemented [24]. However, the tuning into an arbitrary charge regime has not been addressed yet.

Within this work we present an auto-tuner capable of tuning to an arbitrary charge occupation of a previously coarse tuned DQD using convolutional neural networks. We further explore tuning of DQD charge states using unsupervised clustering techniques.

Chapter 2

Quantum Dot Physics

In this chapter we present the theory needed to understand the task of auto-tuning QDs. We first cover the formation of two dimensional electron gases followed by a discussion of QDs. We then use a capacitance model for describing the energetics of QD systems for single and DQDs. Finally, we introduce Coulomb blockade and discuss charge stability diagrams. The derivations in this chapter are based on “Semiconductor Nanostructures” by Dr. Prof. Ihn [25].

2.1 Quantum Dots

Quantum dots (QDs) are nanostructures that confine electrons or holes in small areas of space such that quantization of the energy levels occurs. These structures are typically weakly coupled to their surroundings as source and drain leads as well as lattice vibrations and thermal electromagnetic radiation [25, chap. 18] [26]. Examples of QDs are gate defined QDs [27], self-assembled QDs [28] or etched QDs [27] of which we will focus on gate defined QDs.

Confinement of electrons can be achieved by manipulating the local potential of a two dimensional electron gas (2DEG) by depleting the electrons locally with nearby electric gates resulting in gate defined QDs. A 2DEG is a region in a solid where electrons are confined in a plane such that the electron wave functions localize in z -direction and extend in x - and y - direction. By stacking

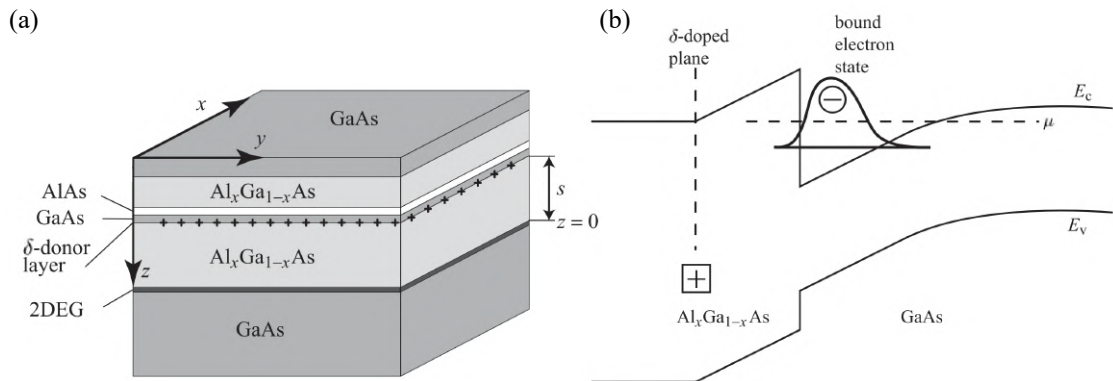


Figure 2.1: (a) A typical GaAs/AlGaAs heterostructure. Several layers of GaAs and $\text{Al}_x\text{Ga}_{1-x}\text{As}$ are stacked on top of each other where x denotes the fraction of Al to Ga. At the intersection $z = 0$, the energy of the conduction band E_c dips below the chemical potential μ allowing for bound electron states as indicated in (b) and a 2DEG forms. At $z = s$ a δ -donor layer provides electrons which occupy the free states in the conduction band at $z = 0$. Both figures are adapted from [25].

layers of semiconductors with different band gaps on each other, the conduction band may locally dip below the chemical potential allowing electrons to occupy conducting states. [29, chap. 3] [25, chap. 9] The electrons occupying the conducting states origin from a close by Si δ -donor layer. [30] A widely used technique for the growth of semiconductor heterostructures is molecular beam epitaxy (MBE) with which crystals are grown with atomic precision. [31] An example of an AlGa/AlGaAs heterostructure is visualized in Fig. 2.1 (a). In Fig. 2.1 (b) the energies of the conduction and valence band across the heterostructure are shown.

Electronic gates deposited on top of the heterostructure can deplete the 2DEG beneath induced by charge carriers accumulated in the gates. In practice, the charge accumulation is reached by applying voltages to the gates. The charge carriers on the gates enforce a positive local potential in the 2DEG close to the gates, illustrated in Fig. 2.2 (a). A gate arrangement as e.g. in Fig. 2.2 (b) on top of the heterostructure together with a suitable charge distribution on the gates can create areas in the 2DEG where electrons are confined. If the confinement extends to all three dimensions, quasi zero-dimensional objects - so-called quantum dots - are formed. The electron reservoirs next to the QD are called source (S) and drain (D). The gates indicated as LRG and RRG in Fig. 2.2 are used to control the potential walls, and thus the tunnel couplings between the QD and the reservoirs.

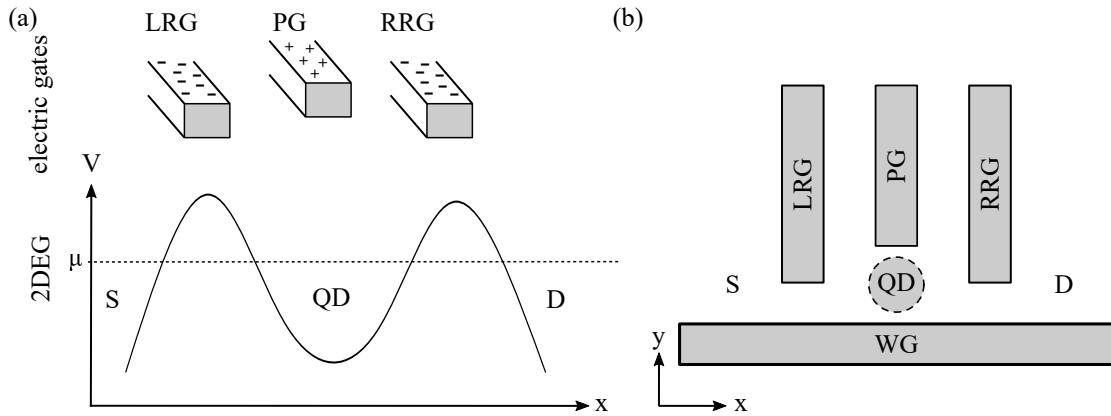


Figure 2.2: (a) The local electrostatic potential in the 2DEG is induced by charges on the gates on top of the 2DEG. The gates indicated with LRG and RRG are responsible for the potential walls on the left and right side respectively and enforce confinement in x-direction. Thus, the middle area (QD) is separated with tunnel barriers from the reservoirs source (S) and drain (D), respectively. With the plunger gate, PG, the depth of the potential well in the middle is controlled. (b) A top view of the gates stacked on top of the heterostructure is shown. The wall gate indicated as WG enables confinement in y-direction such that electrons in the region of the 2DEG indicated with QD are confined in all three dimensions and a quantum dot forms.

The two relevant energy scales for electrons in QDs are the confinement and the Coulomb energy. The confinement energy arises due to the confinement potential induced by the gates and obeys a particle-in-a-box-like behavior and as such, scales as r^{-2} where r is the radius of the QD. The Coulomb energy is due to the Coulomb interactions between the electrons and scales as r^{-1} . For a given material, the ratio between Coulomb and confinement energy solely depends on the size of the QD r . [25, chap. 18]. QDs larger than the Bohr radius a_B^* are dominated by the Coulomb energy [25, chap. 18], which is the case for the QDs used within this thesis¹. In GaAs typical Coulomb energies are of the order of several meV and the confinement energies are of the order of one meV.

¹ For GaAs the Bohr radius is approximately $a_B^* = 10\text{nm}$ [25, chap. 18] whereas the QDs formed by us obey a size of the order of 100nm.

Quantization of electrons on a QD only occurs if the QD is in the tunneling regime. This is the case when the relation between resistance quantum $R_K = \hbar/e^2$ and the tunnel-resistances R_t between QD and structures next to it is $R_t \gg R_K$ [25, chap. 18]. In the example of Fig. 2.2 the gates LRG and RRG are used to tune the tunneling rates and hence, the resistance R_t .

The energy required to add an electron to a QD is defined by its chemical potential μ . Electrons entering a QD origin from the tunnel coupled reservoirs or eventual other QDs next to the considered QD. The chemical energy for a QD with N electrons is

$$\mu_N = E(N) - E(N - 1) \quad (2.1)$$

where $E(N)$ corresponds to the energy of a QD containing N electrons. In Fig. 2.3 (a) an idealized energy level diagram of a QD coupled to two reservoirs including chemical potentials is shown. The chemical potential of a QD can be tuned using its plunger gate. Phenomenologically, it takes the form

$$\mu_N (V_{PG}^0 + \Delta V_{PG}) = \mu_N (V_{PG}^0) - |e|\alpha_{PG}\Delta V_{PG}, \quad (2.2)$$

where α_{PG} is the lever arm of the plunger gate, V_{PG} the voltage on the plunger gate relative to ground and e the electron charge. In QD physics, the energy difference between two consecutive levels μ_N and μ_{N-1} due to Coulomb repulsion is called charging energy E_C . Let us briefly cover the transport phenomenon called Coulomb blockade before we go on and express the energy of a QD, its chemical potential and the charging energy using the capacitance model.

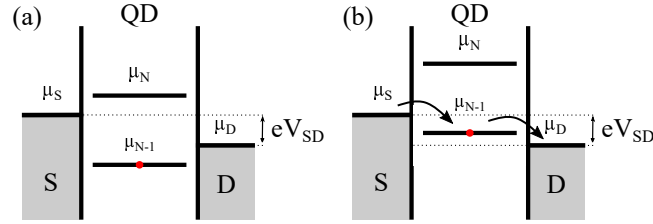


Figure 2.3: Energy level diagram of a QD coupled to the electron reservoirs source (S) and drain (D). The chemical potentials of the reservoirs and the QD are indicated with $\mu_{S/D}$ and μ_N , respectively. Occupied energy levels are marked with a red dot. (a) QD in the coulomb blockade regime. Source and drain are on different potentials, we say a bias window V_{SD} is open. The bias window is defined as $eV_{SD} = \mu_S - \mu_D$ with e the electron charge. (b) An energy state of the QD lies in the energy window opened by the applied bias and transport is possible as indicated by the arrows.

Coulomb blockade is a transport phenomenon at low temperatures exhibited through strong electron-electron correlations. [32, 33] At low temperatures the Fermi-function converges to a step function where all states below the chemical potentials of both, source and drain, are occupied, and all states above the chemical potentials of the reservoirs are unoccupied. In a situation as depicted in Fig. 2.3 (b), Coulomb repulsion prevents additional electrons to enter the QD system while the electrons on the QD cannot leave the dot due to missing free states in source and drain. Thus, Coulomb repulsion blocks transport through the dot. [25, chap. 18][34] Transport through a QD is only possible when the chemical potential μ_N of one of its states lies in between the chemical potential of source μ_S and drain μ_D

$$\mu_S \leq \mu_N \leq \mu_D \text{ or } \mu_S \geq \mu_N \geq \mu_D. \quad (2.3)$$

In such a case, illustrated in Fig. 2.3 (b), electrons tunnel from the reservoir with the higher chemical potential to the QD and from there to the other reservoir. The difference between the chemical potentials of source and drain is set externally by applying a source-drain voltage bias $eV_{SD} = \mu_S - \mu_D$. Coulomb blockade can only take place if the bias window is smaller than the

charging energy $|V_{SD}| < E_C$, since otherwise, there is always a QD state for which Eq. (2.3) is fulfilled. The phenomenon of Coulomb blockade is naturally extended to n QDs in series. An illustration of $n = 2$ dots in series is found in Fig. 2.4. For arbitrary n , transport from source to drain only occurs if each QD i has a state with chemical potential μ_N^i such that²

$$\mu_S \leq \mu_{N_0}^0 \leq \mu_{N_1}^1 \leq \dots \leq \mu_{N_n}^n \leq \mu_D \text{ or } \mu_S \geq \mu_{N_0}^0 \geq \mu_{N_1}^1 \geq \dots \geq \mu_{N_n}^n \geq \mu_D, \quad (2.4)$$

where the QDs are in series and indexed in order, i.e. QD 0 is next to source, QD n is next to drain and QD $i + 1$ is next to QD i . The electron tunnel from the reservoir with higher chemical potential through the QDs to the other reservoir.

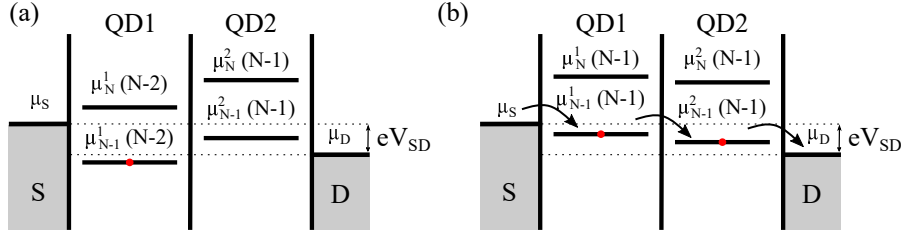


Figure 2.4: Scheme of the energy levels of a DQD coupled to reservoirs source and drain. The red dots indicate occupied energy levels. The chemical potential of source and drain differ and a finite bias window $eV_{SD} = \mu_S - \mu_D$ is opened. The chemical potentials of the energy levels are indicated with $\mu_N^1(N-1)$ which corresponds to the chemical potential of QD1 of the N -th electron with $(N-1)$ electrons present on QD2. In (a) Coulomb blockade prevents transport whereas in (b) transport is permitted and the direction is indicated with arrows.

2.1.1 Capacitance Model

In QD physics it is usually sufficient to consider the energy due to confinement and energy due to Coulomb interaction separately. We will assume that the confinement problem is solved quantum mechanically and that its energy levels correspond to the solutions of the particle-in-a-box problem. We will further assume, that all electrons occupy their lowest available energy state due to low temperature. The energy of the j -th electron due to confinement is then denoted as $\epsilon^{(j)}$. We consider the energy due to Coulomb interactions between the electrons semiclassically using the capacitance model³ [25]. Within this model, the QDs are described as metallic islands - and thus as capacitors - with discrete energy spectra. The interaction effects between the electrons and the gates are represented by a capacitance matrix C defined via

$$Q_i = \sum_{j=1}^n C_{ij} (V_j + V_j^{(0)}) = \sum_{j=1}^n C_{ij} V_j + Q_i^{(0)} \quad (2.5)$$

where C_{ij} are the entries of the matrix C and Q_i is the charge on island i which is at a potential $V_i + V_i^{(0)}$ relative to ground. Note that the capacitance between island i and j ($i \neq j$) is $C_{i \leftrightarrow j} = -C_{ij} > 0$. We further denote the self capacitance of island i as $C_i^{(0)}$. Then, the total capacitance of island i is

$$C_{ii} = C_i^{(0)} + \sum_{j=1, j \neq i}^n C_{i \leftrightarrow j}. \quad (2.6)$$

²In the case of co-tunneling transport can occur without fulfilling this equation. However, for this work co-tunneling is not relevant and we do not cover it here. We refer the interested reader to [26, 35]

³A quantum mechanical extension of the capacitance model using the Hubbard model can be found in [36, 37, 38]

Single Quantum Dots

We assume a setup with n metallic islands of which one is a QD (index $i = 1$) and $n - 1$ are gates (indices $i > 1$). We directly discuss the solution for the energy states for single QDs. For the derivation we refer the interested reader to [25, chap. 18]. Within the capacitance model the energy of a QD system containing N electrons can be written as

$$E(N) = \sum_{j=1}^N \epsilon^{(j)} + \frac{e^2 N^2}{2C_\Sigma} + |e|N \sum_{i=2}^n \frac{C_{1i}}{C_\Sigma} (V_i - V_i^{(0)}), \quad (2.7)$$

where $C_\Sigma \equiv C_{11}$ is the total capacitance of the QD and N is the number of electrons on the QD. The gates are at static potentials $V_i + V_i^{(0)}$ relative to ground. Following the definition of the chemical potential $\mu_N \equiv E(N) - E(N - 1)$ we find

$$\mu_N = \epsilon^{(N)} + \frac{e^2}{C_\Sigma} \left(N - \frac{1}{2} \right) + |e| \sum_{i=2}^n \frac{C_{1i}}{C_\Sigma} (V_i - V_i^{(0)}). \quad (2.8)$$

Comparing this expression with Eq. (2.2) yields the lever arms

$$\alpha_i = -\frac{C_{1i}}{C_\Sigma}. \quad (2.9)$$

Adding one electron to the QD without changing any gate voltages increases the chemical potential by $E_C + \epsilon^{(N)}$ with $E_C \equiv \frac{e^2}{C_\Sigma}$ the charging energy. By inserting the charging energy and the lever arms into Eq. (2.8) we obtain

$$\mu_N = \epsilon^{(N)} + E_C \left(N - \frac{1}{2} \right) - |e| \sum_{i=2}^n \alpha_i (V_i - V_i^{(0)}). \quad (2.10)$$

The first term is due to confinement and the remaining terms are due to Coulomb interactions.

Coupled Double Quantum Dots

We now consider a double quantum dot (DQD) system where two islands (indexed as one and two) are QDs and the remaining $n - 2$ islands are metallic gates. All islands are assumed to be capacitively coupled with each other where we denote the entries of the capacitance matrix C as C_{ij} according to (2.5). An illustration of such a system of a coupled DQD can be seen in Fig. 2.5. Since the two QDs are coupled via the interdot capacitance $C_{1 \leftrightarrow 2}$, the energies will change with respect to the single QD case. Following the derivation of the single QD energy, the total energy of the DQD system can phenomenologically be written as

$$\begin{aligned} E(N_1, N_2) = & \sum_{j=1}^{N_1} \epsilon_1^{(j)} + \frac{e^2 N_1^2}{2C_{\Sigma_1}} + |e|N_1 \sum_{i=3}^n \alpha_{1i} (V_i - V_i^{(0)}) \\ & + \sum_{j=1}^{N_2} \epsilon_2^{(j)} + \frac{e^2 N_2^2}{2C_{\Sigma_2}} + |e|N_2 \sum_{i=3}^n \alpha_{2i} (V_i - V_i^{(0)}) \\ & + \frac{e^2 N_1 N_2}{C_{12}} \end{aligned} \quad (2.11)$$

with $\epsilon_i^{(j)}$ the energy of the j -th electron in QDi due to quantization. In the above equation, the relations

$$\begin{aligned} C_{\Sigma_1} &= C_{11} \left(1 - \frac{C_{21}C_{12}}{C_{11}C_{22}} \right) \\ C_{\Sigma_2} &= C_{22} \left(1 - \frac{C_{21}C_{12}}{C_{11}C_{22}} \right) \\ \tilde{C}_{12} &= \frac{C_{11}C_{22} - C_{12}C_{21}}{-C_{12}} \\ \alpha_{1j} &= \frac{C_{12}C_{2j} - C_{22}C_{1j}}{C_{11}C_{22} - C_{12}C_{21}} \\ \alpha_{2j} &= \frac{C_{21}C_{1j} - C_{11}C_{2j}}{C_{11}C_{22} - C_{12}C_{21}} \end{aligned} \quad (2.12)$$

are used. The capacitance C_{11} is called the total capacitance of QD1 and C_{22} corresponds to the total capacitance of QD2, respectively. Coupling two QDs changes their charging energy $E_{C_i} = \frac{e^2}{C_{\Sigma_i}}$. Similarly, the lever arms α_{1j} and α_{2j} of the gates change as well. The first two lines of Eq. (2.11) formally correspond to the single dot case, whereas the third line describes the interaction between the QDs. The chemical potentials are

$$\mu_{N_1}^{(1)}(N_2) = \epsilon_1^{(N)} + E_{C_1} \left(N_1 - \frac{1}{2} \right) + |e| \sum_{i=3}^n \alpha_{1i} (V_i - V_i^{(0)}) + \frac{e^2}{\tilde{C}_{12}} N_2 \quad (2.13)$$

$$\mu_{N_2}^{(2)}(N_1) = \epsilon_2^{(N)} + E_{C_2} \left(N_2 - \frac{1}{2} \right) + |e| \sum_{i=3}^n \alpha_{2i} (V_i - V_i^{(0)}) + \frac{e^2}{\tilde{C}_{12}} N_1, \quad (2.14)$$

where $\mu_{N_1}^{(1)}(N_2)$ is the chemical potential for the N_1 -th electron entering QD 1 and N_2 electrons present on QD2. The chemical potential is formally expressed analogously to the single QD case given in Eq. (2.10) up to an additional term due to the interaction between the electrons on QD1 with the electrons on QD2. Lever arms α_{ij} as well as the charging energies are still solely expressed by the capacitances and the unit charge. For the limit of $C_{12} \rightarrow 0$ the case of uncoupled QDs in Eq. (2.10) is obtained.

2.1.2 Charge Stability Diagrams

For any given voltage configuration, there is one or several degenerate charge states that have the highest chemical potential below the reservoir potentials. When there are several, or there is one that equals to source or drain potential, the charge state is unstable. Otherwise, the charge is stable. The regions of stable charge can be illustrated with a charge stability diagram as a function of two gate voltages. In this diagram (see e.g. Fig. 2.6) the so-called charge transition lines indicate voltage configurations of unstable charge states. In the case of a DQD with N_1 electrons on QD1 and N_2 electrons on QD2 we indicate the charge state as (N_1, N_2) .

The formulation of the chemical potential of a DQD using the capacitance model in Eq. (2.13) and Eq. (2.14) enables to express the charge transition lines as a function of the gate voltages⁴. Hence, the charge transition lines in the charge stability diagram can directly be determined by the capacitances between the dots and the electronic gates. The six defining relations are:

⁴For a quantum mechanical description of charge stability diagrams of DQDs we refer to [37]

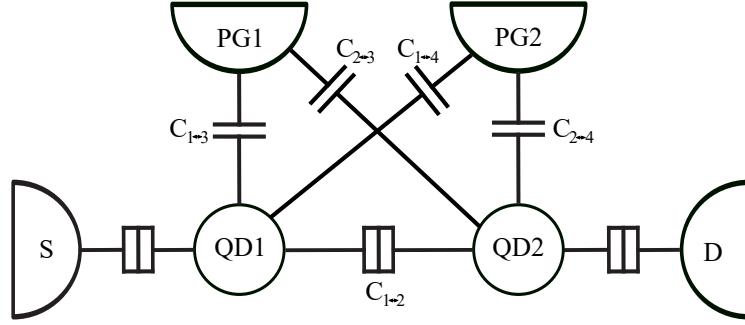


Figure 2.5: A scheme of a DQD connected to the reservoirs source (S) and drain (D). The relevant capacitances of the system are indicated in the scheme. Electron transport is only possible between S, QD1, QD2 and D, indicated with connected lines.

$$\begin{aligned}
 (N_1, N_2) \rightarrow (N_1 + 1, N_2) &\leftrightarrow \mu_{S/D} = \mu_{N_1+1}^{(1)}(N_2) \quad (1) \\
 (N_1, N_2) \rightarrow (N_1, N_2 + 1) &\leftrightarrow \mu_{S/D} = \mu_{N_2+1}^{(2)}(N_1) \quad (2) \\
 (N_1, N_2) \rightarrow (N_1 - 1, N_2) &\leftrightarrow \mu_{S/D} = \mu_{N_1}^{(1)}(N_2) \quad (3) \\
 (N_1, N_2) \rightarrow (N_1, N_2 - 1) &\leftrightarrow \mu_{S/D} = \mu_{N_2}^{(2)}(N_1) \quad (4)
 \end{aligned} \tag{2.15}$$

$$\begin{aligned}
 (N_1, N_2) \rightarrow (N_1 + 1, N_2 - 1) &\leftrightarrow \mu_{N_1+1}^{(1)}(N_2 - 1) = \mu_{N_2}^{(2)}(N_1) \quad (5) \\
 (N_1, N_2) \rightarrow (N_1 - 1, N_2 + 1) &\leftrightarrow \mu_{N_1}^{(1)}(N_2) = \mu_{N_2+1}^{(2)}(N_1 - 1) \quad (6)
 \end{aligned} \tag{2.16}$$

The relations in Eq. (2.15) describe charge transitions between reservoir and one of the QDs and the relations in Eq. (2.16) describe electrons transitioning from one dot into the other. All six transitions are visualized in the schematic charge stability diagram in Fig. 2.6 indexed as (1) - (6). Furthermore, the charging energies E_{C_i} defining the level spacing and the effective capacitance \tilde{C}_{12} between the two QDs are visualized. The distance between two resonances in terms of plunger gates voltages are $\Delta V_{PG1} = \frac{E_{C1}}{\alpha_{13}|e|}$ and $\Delta V_{PG2} = \frac{E_{C2}}{\alpha_{24}|e|}$. The shifts of the resonances due to the inderdot coupling are $\delta V_{PG1} = \frac{|e|}{\tilde{C}_{12}\alpha_{13}}$ and $\delta V_{PG2} = \frac{|e|}{\tilde{C}_{12}\alpha_{24}}$.

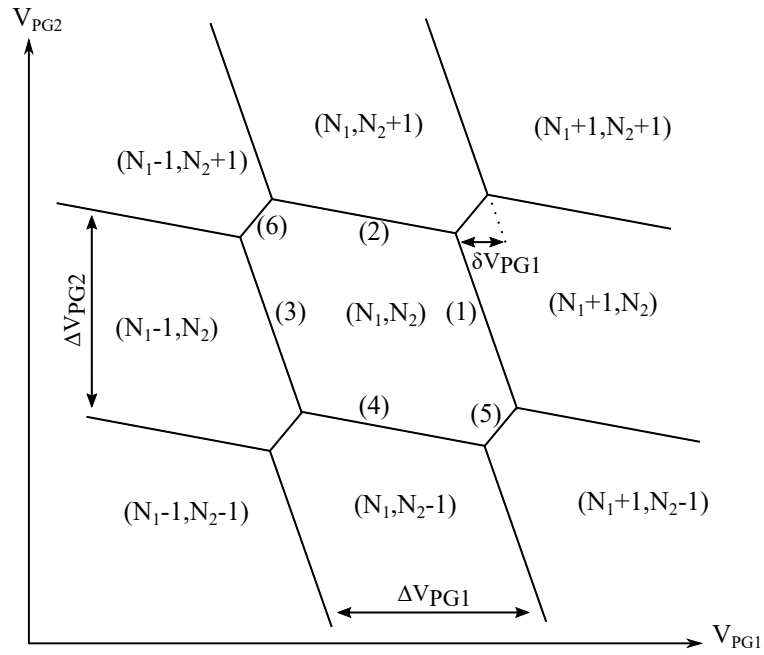


Figure 2.6: Charge stability diagram for a DQD system as modeled in Fig. 2.5. The solid lines indicate gate voltage configurations for which charge transitions occur. The transitions next to the numbers (1) - (6) are defined by the respective equation in Eq. (2.15) and Eq. (2.16). The indicated voltage differences relate the charging energies of the QDs and the effective capacitance between the QDs with the diagram.

Chapter 3

Experimental Setup and Sample Characterization

In this chapter we introduce the experimental setup and techniques used in this work. We present the sample used and discuss its properties in Sec. 3.1. Then, we introduce the setup with which measurements are taken in Sec. 3.2. In Sec. 3.3 we present the measurement of the electronic temperature and lastly, we introduce the charge detector used to obtain charge stability diagrams in Sec. 3.4.

3.1 Sample

We form electrostatically defined QDs using a GaAs/AlGaAs heterostructure grown with MBE [8, 31] by the Wegscheider Group at ETH Zürich. Our chip contains a 2DEG (see Sec. 2.1 or [25, chap. 9][29, chap. 3]) 90 nm below the surface with a mobility of $\mu = 3.2 \times 10^6 \text{cm}^2 \text{V}^{-1} \text{s}^{-1}$ and electron density of $n_e = 2.2 \times 10^{11} \text{cm}^{-2}$ measured at 4.2K. In order to contact and electrostatically control the electrons in the 2DEG ohmic contacts and gates are fabricated with standard photolithography, metal evaporation, and annealing techniques. For a detailed discussion of wafer growth, the creation of high mobility 2DEGs and the lithography processes we refer the interested reader to [8].

A scanning electron microscope image of the sample showing the gate structure is presented in Fig. 3.1. The shown architecture allows to form up to three QDs in series labeled with QD1, QD2 and QD3. The plunger gates marked as PG1, PG2 and PG3 are used to tune the chemical potential μ_i of the respective quantum dot QD1, QD2 and QD3. The gates labeled as QPC_M and QPC_R are used to form a quantum point contact (QPC) used as a charge detector, see Sec. 3.4 for a more detailed discussion. Two QPC gates allow to realize two different charge detectors that couple differently to the individual QDs. The source (S) and drain (D) contacts for the triple quantum dot and the QPC are labeled with S , D , S_{QPC} and D_{QPC} respectively.

3.2 Measurement Setup

The measurements are performed in a commercially available $^3\text{He}/^4\text{He}$ dilution refrigerator from Oxford Instruments with a base temperature of 12mK¹. A custom made set of ten voltage sources called 'compact 2012' is directly mounted on top of the refrigerator at room temperature. Placing the voltage source on the fridge allows using short cables and thus reduces noise from the cables. Additionally, two low noise voltage sources 'Yokogawa 7651 DC Source' and two 'Agilent 34401 Multimeter' are used to apply voltages and measure currents via two I-V converters. All electrical

¹The electronic temperature is measured in Sec. 3.3 and found to be 132mK.

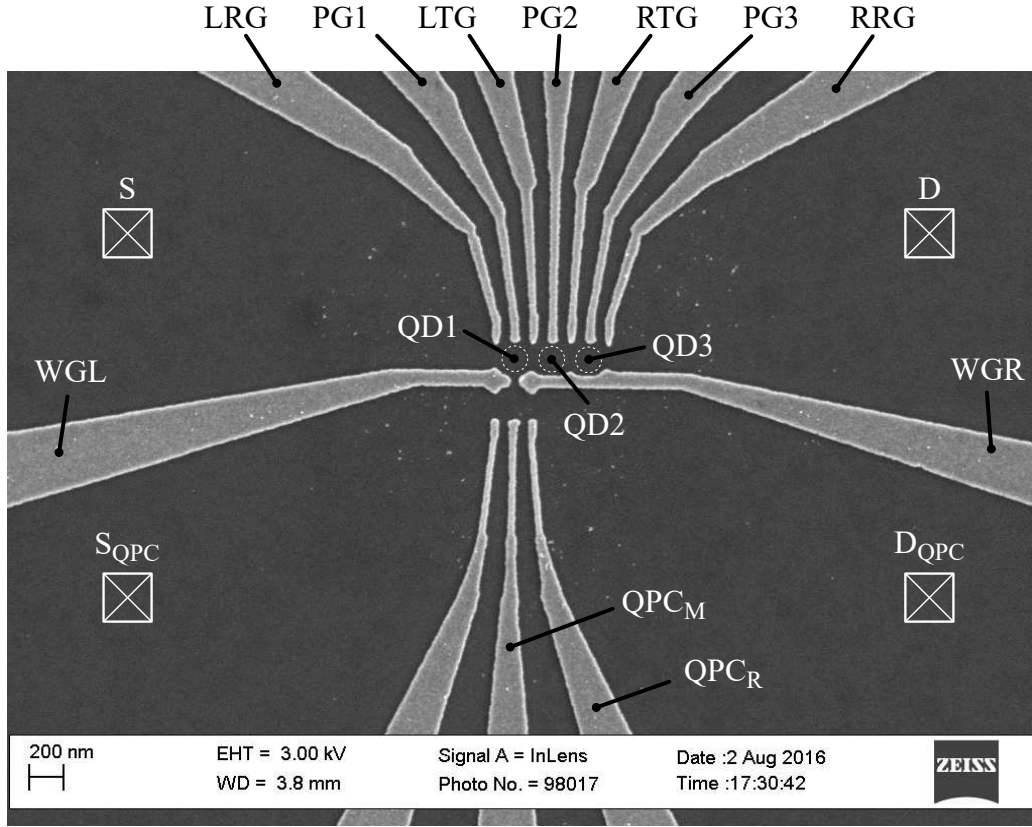


Figure 3.1: A scanning electron microscope image of the sample used with labeled gates. The dashed circles indicate the positions of the QDs. The gates to form the QDs are on the top (LRG, PG1, LTG, PG2, RTG, PG3, and RRG) while the gates to form quantum point contacts are on the bottom (QPC_M , QPC_R). The squares with a cross mark ohmic contacts to the 2DEG.

devices are connected to a measurement computer where measurements are performed using the software “Labber”.

The sample is mounted in the dilution refrigerator on a 12mK plate via a sample holder, see Fig. 3.2. The voltage sources and the I-V converters are directly connected to the sample via thermally anchored cables. It is common practice to mount an RC filter between voltage sources and sample holder in order to prevent high frequency current to heat up the system. However, no space is available on the 12mK plate and no filter was used.

3.3 Temperature Measurement

At low temperature, the electrons in the 2DEG are warmer than the base temperature of the refrigerator due to heat from the leads connecting the sample with the measurement devices at room temperature. [39, 40] We determine the electronic temperature by extracting the width of a non-tunnel broadened conductance peak of a Coulomb diamond at zero bias [39]. A measurement of the Coulomb diamond with which temperature is determined is shown in Fig. 3.3 (a) and (b). The current near the conductance peak is [25, chap. 18]

$$I = \frac{I_0}{\cosh^2 [\alpha_{PG} (V_{PG} - V_{res}) / 2k_B T]}, \quad (3.1)$$

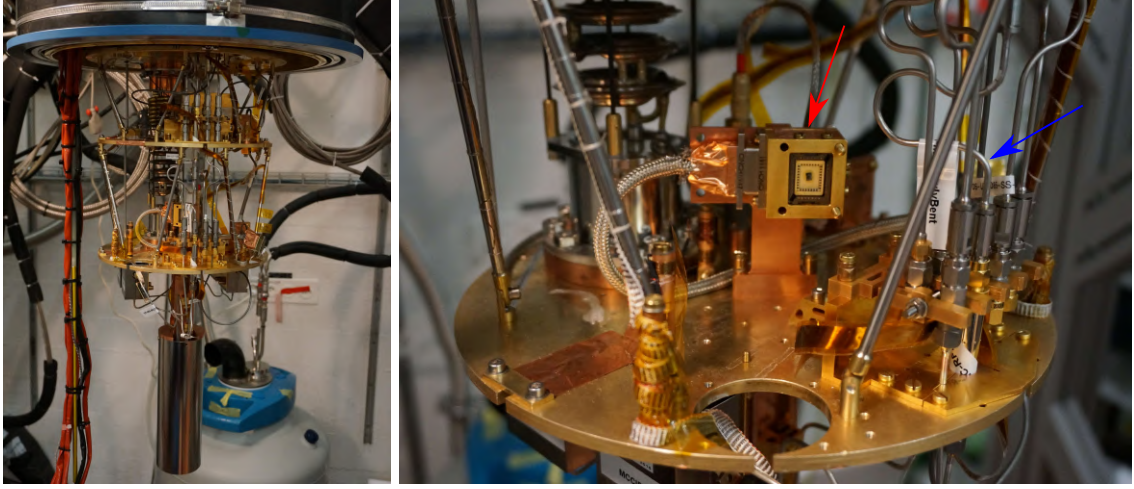


Figure 3.2: Pictures of the $^3\text{He}/^4\text{He}$ dilution refrigerator from Oxford Instruments. Left panel: inner part of the fridge. Right panel: The sample holder (red arrow) is placed on the 12mK plate and thermally anchored to it. The microwave waveguides (blue arrow) are not used.

with I_0 the height of the current peak, α_{PG} the lever arm of the plunger gate, V_{res} the mean of the resonance, V_{PG} the voltage applied to the plunger gate, k_{B} the Boltzmann constant and T the temperature. The lever arm of the plunger gate is the slope of the Coulomb diamond, see Fig. 3.3 (a). We fitted Eq. (3.1) with all measured traces of the current shown in Fig. 3.3 (b), an example is shown in Fig. 3.3 (c). However, the source drain bias must not broaden the current peak. Thus, only those traces for which the temperature does not show a bias dependence are taken into account and are shown in Fig. 3.3 (d). We found the electronic temperature $132\text{mK} \pm 9\text{mK}$. This is rather high, as for comparable experiments electronic temperatures of 20mK have been found. We believe that the cause of the high temperature is the missing RC filter which would shield from high-frequency noise in the cable connecting the measurement devices and voltage sources with the sample.

3.4 QPC as a Charge Detector

One example of a charge detectors are QPCs, which are one dimensional channels with a quantized number of transport modes [41, 42]. The conductance of a QPC depends on the applied electrostatic potential as for example induced by electrons on the QDs close by. Thus, by measuring the QPC conductance the charge occupation of a QD can be inferred.

We form a QPC by depleting the 2DEG beneath of the QPC gates (QPC_{M} or QPC_{R}) and the wall gates (WGL, WGR) indicated in Fig. 3.1. The depletion leads to an one dimensional conducting channel showing characteristic conductance plateaus, shown in Fig. 3.4. The derivative of the current with respect to the applied voltage on the QPC gate (dI/dV) shows the sensitivity of the QPC to a change of the local potential. Such a change of the local electrostatic potential can be caused by a change of the number of electrons confined in a close by QD. In order to maximize the sensitivity of the QPC, it is operated at the so called "sweet spot", or "operation point" given as the maximum of dI/dV . The operation point is indicated in the figure with an arrow.

A QPC allows to measure charge stability diagrams. For that purpose, one measures the current of the QPC next to the DQD^2 and sweep both plunger gates within a certain range. Such a

²In principle charge stability diagrams can be obtained for more than two QDs. However, within this thesis this is never the case.

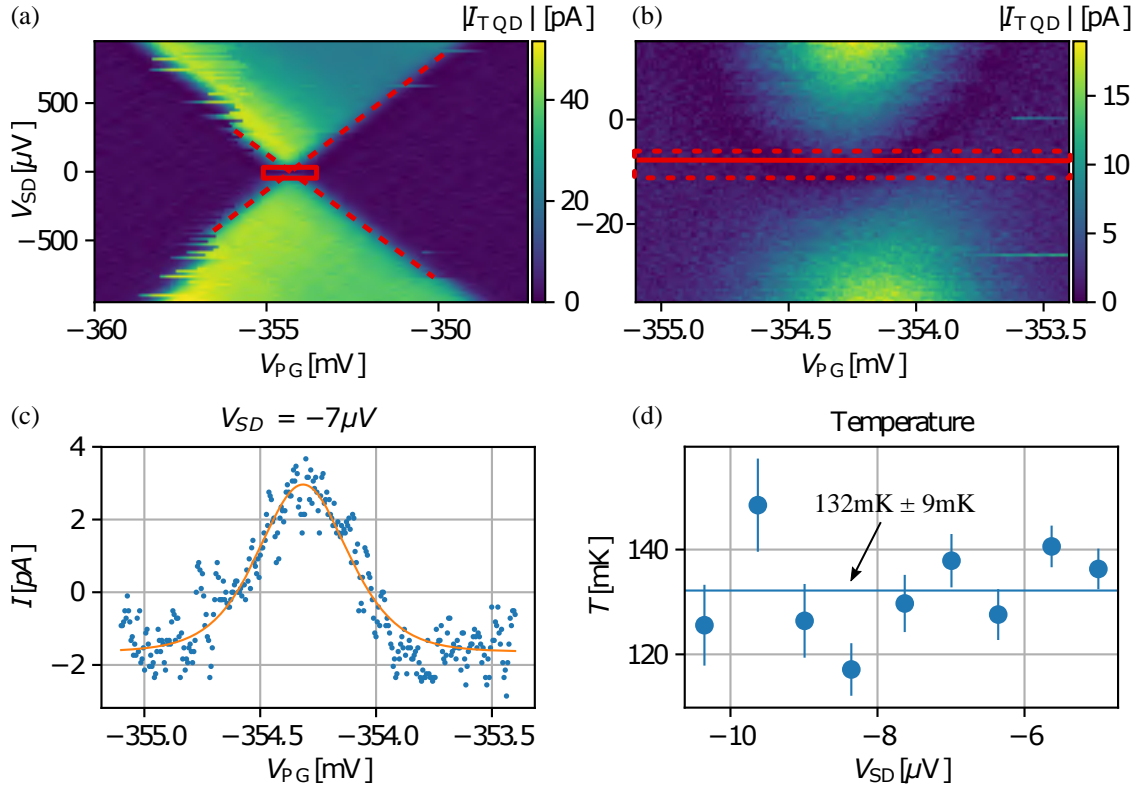


Figure 3.3: (a) - (b) Absolute value of the current through the QD showing the tips of the Coulomb diamond of the last electron. (a) The red dotted lines mark the boundaries of the Coulomb diamond, and their slope determines the leverarm α_{PG} . (b) Measurements taken in the range of the red solid box in (a). A non-zero background current is seen. (c) A fit of Eq. (3.1) for the trace indicated with a red solid line in (b). (d) The temperatures obtained by a fit for the traces in the red dotted box in (b).

measurement of the QPC is shown in Fig. 3.5. Discontinuities in the QPC signal indicate charge transitions in one of the QDs. Plotting the following superposition of derivatives (Fig. 3.6)

$$\frac{\partial I_{QPC}}{\partial PG} = \frac{1}{2} \frac{\partial I_{QPC}}{\partial PG1} + \frac{1}{2} \frac{\partial I_{QPC}}{\partial PG2} \quad (3.2)$$

visualizes these discontinuities and hence, charge transitions on the QDs manifest as solid lines, called charge transition lines. The slightly tilted vertical lines correspond to charge transitions on QD1, whereas the slightly tilted horizontal lines correspond to charge transitions on QD2, respectively. Starting from the lower left corner, it is possible to count the number of electrons present on each QD for a certain plunger gate voltage configuration. We indicate the electron occupation of a DQD as (N_1, N_2) with N_{QDk} the number of electrons on QDk.

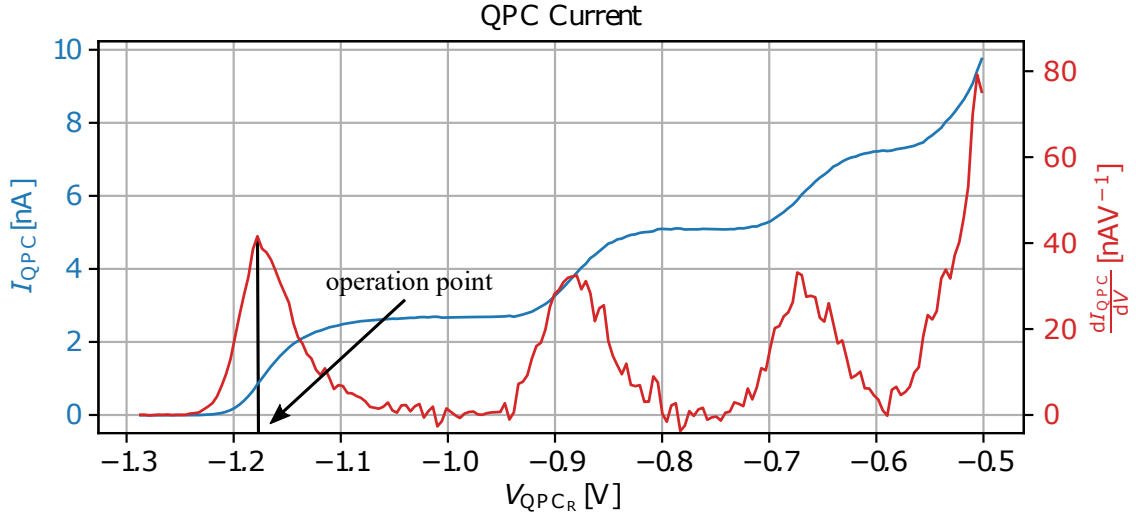


Figure 3.4: QPC current as a function of the QPC_R gate is shown in blue. Three conductance plateaus are seen. The red curve is the derivative of the blue curve, and is the differential QPC current. The QPC operation point is the one with maximal derivative indicated with an arrow in the plot.

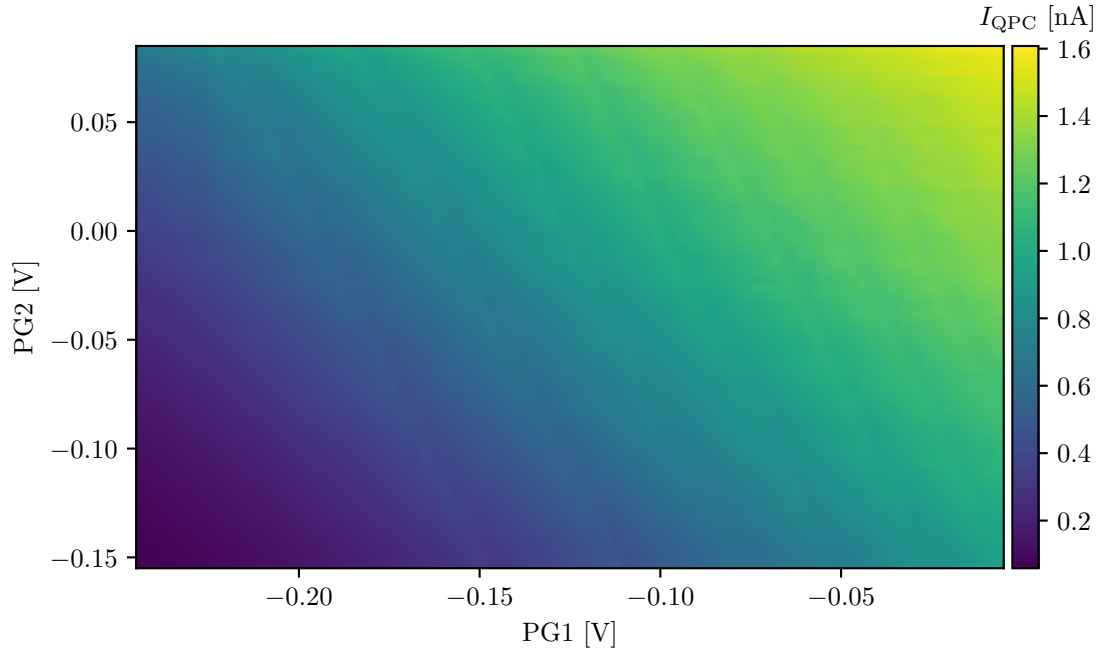


Figure 3.5: Shown is the QPC signal for a gate voltage configuration which we call DQD regime.

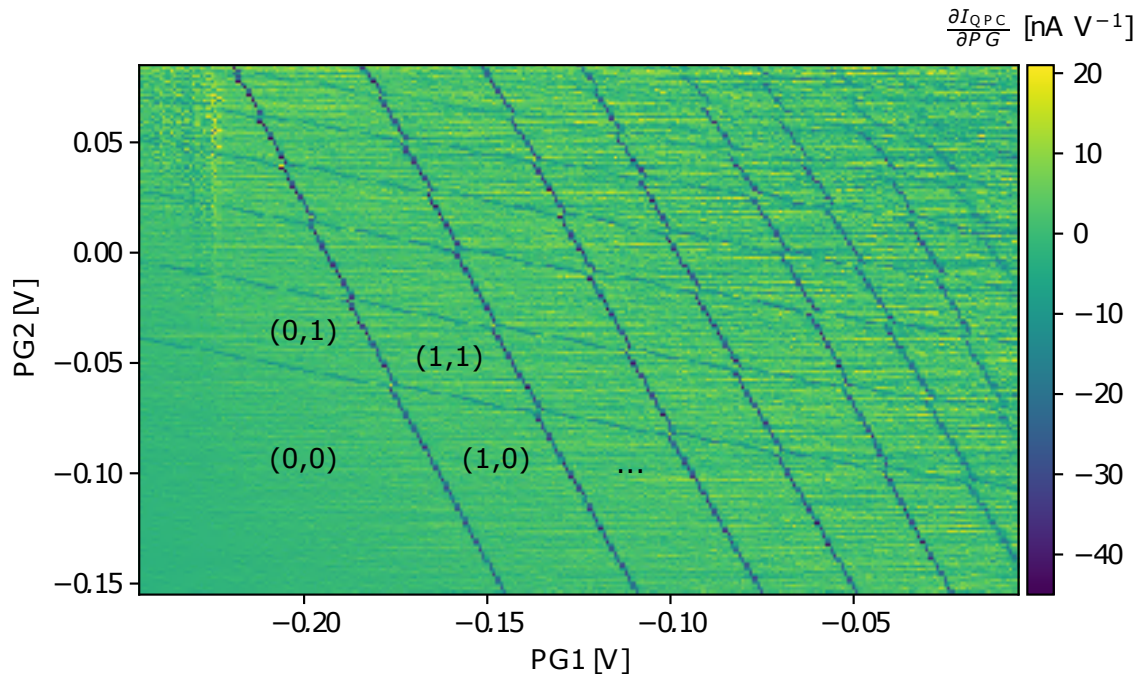


Figure 3.6: Charge stability diagram for the same regime as shown in Fig. 3.5. The plot shows an equal superposition of the derivative of the QPC current in $PG1$ and $PG2$ direction as defined in Eq. 3.2. The charge occupation for the stable charge regimes is indicated with (n, m) with n (m) the number of electrons on QD1 (QD2).

Chapter 4

Machine Learning

4.1 Machine Learning Fundamentals

Machine learning is the study of algorithms and statistical models that perform a class of tasks without explicit instructions. All machine learning models have in common that they learn general features from data. The procedure of learning can be very similar to the procedure of fitting, but in contrast to e.g. curve fitting, no explicit (i.e. task specific) underlying model is implied in machine learning problems. Machine learning tasks can be divided into three groups, supervised learning, unsupervised learning and reinforcement learning. In this thesis we mainly cover supervised learning.

In a machine learning task one is usually confronted with two sets, a set of data-points \mathcal{X} and a set of labels \mathcal{Y} . Data-points $x \in \mathcal{X}$ are represented as vectors, whose dimensions are called features. Every label $y_i \in \mathcal{Y}$ is allocated to a specific data-point $x_i \in \mathcal{X}$. Note that \mathcal{Y} and \mathcal{X} do not necessarily have the same cardinality. However, the underlying assumption here is that every data-point x possesses an associated label y , but the label might be unknown to us. The extreme case where \mathcal{Y} is empty corresponds unsupervised learning. The other extreme case, where \mathcal{Y} and \mathcal{X} have the same cardinality and all data is labeled is called supervised learning. All remaining cases are called semi-supervised learning. The Machine Learning task is then formulated as follows:

Problem Statement. Machine Learning Task

Given a set of data-points \mathcal{X} and labels \mathcal{Y} , find a model M with a corresponding set of hyper-parameters $\{p_i\}_i$ such that for any data point $x \notin \mathcal{X}$ the correct label y is predicted. Formally, $M(x, \{p_i\}_i) = y$.

Two typical examples of machine learning problems are classification and regression of which we concentrate on classification problems throughout this thesis. In classification problems, the set of possible labels \mathcal{Y} is discrete and finite. Every possible value of $y \in \mathcal{Y}$ is called a class. Within classification, the machine learning task is predicting the class y of a given data-point x .

Even though machine learning models do not need explicit, task specific, instructions one still needs to choose a model M and a set of corresponding hyper-parameters $\{p_i\}_i$. The challenging aspect in machine learning is to choose the right model with corresponding hyper-parameters. But how can we choose one model over another? In other words, how do we score the performance of a chosen model? Learning a model from data and looking at the quality of its predictions does not generalize as seen in the following thought-experiment. Imagine we wanted to fit data with a polynomial with more independent parameters than data-points (which is a common situation in machine learning). We would likely find a situation where our fit predicts all the training points correctly. Still, this does not mean that we have found the underlying function from which data points are drawn.

The most common approach for scoring a machine learning model is to split the data set into a training \mathcal{T} and an evaluation set \mathcal{E} . The model is trained on the training data set, and evaluated on the previously unseen data in the evaluation set. This allows to get an unbiased estimate about the performance of a model.

How can we quantify the performance of a classification model? A very instructive method for evaluation is the so-called confusion matrix as illustrated in Fig. 4.1. A confusion matrix is a matrix with as many rows and columns as there are classes. For illustration let us assume we are given a data set \mathcal{X} and a set of labels \mathcal{Y} consisting of three classes called "cats", "dogs" and elephants". After splitting \mathcal{X} into train \mathcal{T} and evaluation set \mathcal{E} and training our chosen model M on \mathcal{T} we can predict the classes for all remaining data points in \mathcal{E} . The entry ij of the confusion matrix for $i, j \in \{\text{elephants}, \text{cats}, \text{dogs}\}$ represents the number of data points in \mathcal{E} that have the label i and are classified as j . A perfect classifier thus has a diagonal confusion matrix.

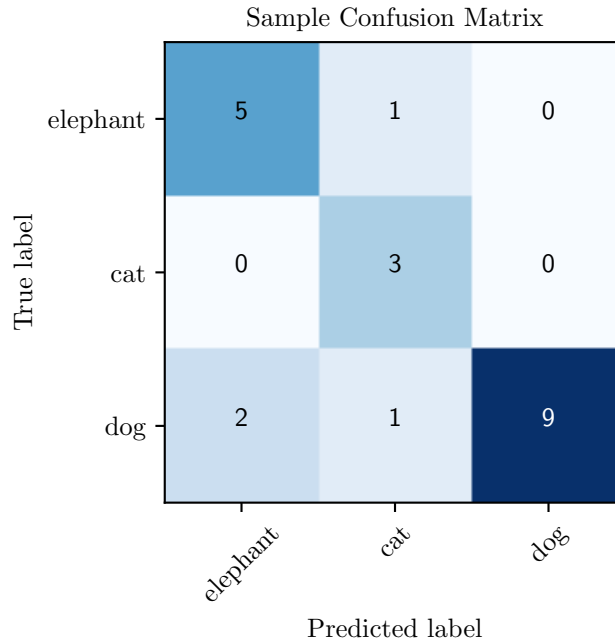


Figure 4.1: A sample confusion matrix of an invented classifier evaluated on an evaluation data set with a total of 21 data-points. The columns correspond to the predicted label by the classifier, whereas the rows correspond to the true, given, label of the data-points. In this example, 17 classifications are correct (sum of the diagonal elements) and 4 wrong (sum of the off diagonal elements).

Given a classification task and a classifier - and therefore its confusion matrix - it is still not clear on how to score the classifier's performance. Looking at an exemplary confusion matrix as e.g. in Fig. 4.1 there are several things that one can conclude. Column "dog" tells us that the probability that a data-point belongs to class "dog", given that the classifier classified it as "dog" is 100%. This does not mean, that all data points possessing the label "dog" are actually classified as "dog" which can be seen in row "dog". In particular, the probability to classify data points from class "dog" correctly is $\frac{9}{12}$. These two scores, the probability of classifying dogs as dogs and the probability of classified dogs being dogs, are called precision and recall, respectively. Let us formalize the classifications. Be Y the probability distribution of the true, given labels in the evaluation set. Similarly, let \hat{Y} be the probability distribution of the classified labels. We can then define precision and recall as the following.

Class	precision	recall	f1	support
elephant	0.71	0.83	0.77	6
cat	0.60	1.00	0.75	3
dog	1.00	0.75	0.86	12

Table 4.1: Sample scores for the predictions visualized in the confusion matrix of Fig. 4.1.

Definition. Precision

Precision p for class j is the probability that a data-point classified as j also belongs to class j .

$$p = \mathbb{P}[Y = j | \hat{Y} = j] \quad (4.1)$$

Precision is approximated as:

$$p \approx \frac{\text{count}(Y = j \wedge \hat{Y} = j)}{\text{count}(\hat{Y} = j)}. \quad (4.2)$$

Definition. Recall

Recall r for class j is the probability that a data-point from class j is classified as j .

$$r = \mathbb{P}[\hat{Y} = j | Y = j] \quad (4.3)$$

The recall score for class j is approximated as:

$$r \approx \frac{\text{count}(Y = j \wedge \hat{Y} = j)}{\text{count}(Y = j)}. \quad (4.4)$$

Definition. f1

The f1 score $f1$ is the harmonic mean between precision and recall.

$$f1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (4.5)$$

The f1 score is approximated as:

$$f1 \approx \frac{2 \cdot \text{count}(Y = j \wedge \hat{Y} = j)}{\text{count}(Y = j) + \text{count}(\hat{Y} = j)}. \quad (4.6)$$

The f1 score combines recall and precision, and it can only reach one if both, recall and precision equal to one as well.

For our example with the confusion matrix, we can calculate the precision, recall and f1 scores for each of the classes, given in Tab. 4.1. Confronted with many different scoring numbers, the problem of assessing a model's performance is still unsolved. Choosing the right score depends on the problem definition. Maybe it is crucial to always recognize elephants correctly, but it is okay to classify cats as dogs or vice versa. In that case, one would try to optimize the f1 score for elephants.

4.2 Machine Learning Tuning Algorithms

Within this section, we describe algorithms that we use for the auto-tuning of DQD charge states. We first introduce necessary expressions and define the problem statement. Then, we present an algorithm on how to auto-tune charge states by first, find what we call a reference point, and then, recognize charge transitions among a certain path.

Definition. DQD Regime

Within this thesis, we say that the gate voltage configuration correspond to a DQD regime if:

- it is possible to form a DQD with the sample under investigation and it also possesses a plunger gate for each of the individual QDs
- the gate voltage configuration is such that the charge stability diagram looks as depicted in Fig. 2.6. This means that both QDs and its charge occupation numbers are well defined over a range of PG1 and PG2 voltages. Furthermore, PG1 primarily couples to QD1 and PG2 primarily couples to QD2.

Definition. Charge Reference Point

A charge reference point is a gate voltage configuration within a DQD regime where the charge occupation number for both QDs is known. We indicate the charge occupation number for a DQD as (n, m) , where n is the number of electrons on QD1 and m the number of electrons on QD2 respectively.

Problem Statement. Auto Tuning of Quantum Dot Charge States

Given a sample tuned into the DQD regime and given a charge occupation (n, m) , find the plunger gate voltages such that the charge occupation of the DQD is (n, m) .

The auto-tuning task is split in two stages. First, we find a point for which the charge occupation for both QDs is $(0, 0)$. In the second stage we find a path in the plunger gate voltage space that starts in the $(0, 0)$ charge regime and ends up in the (n, m) configuration. We monitor transitions among path segments and depending on the charge transition we extend the path in different directions until we end up in the desired charge configuration. The monitoring of charge transitions is described in 4.2.2 and the procedure of finding a reference point $(0, 0)$ is explained in 4.2.1.

4.2.1 Finding a Reference Point

The goal of this section is to present an algorithm that finds a charge reference point. Within this thesis our reference point will always be a gate voltage configuration such that the charge occupation for both QDs is zero indicated as $(0, 0)$. Such an empty region can be seen in the lower left corner of Fig. 3.6. Knowledge of a reference point allows to later tune to any charge configuration.

One could argue that it is easy to find plunger gate voltages for which both QD have zero charge occupation $(0, 0)$. This case simply is achieved by applying very negative voltages to both plunger gates. While this procedure may work, the found reference point would likely be far away from the zero to one electron transition. Thus, the path built up in the second part of the auto-tuning algorithm would be longer. The longer this path, the more likely errors are to happen¹, and, the more time the tuning algorithm will need. Therefore, we focus on a different approach using a deep neural network differentiating between what we call empty and non empty areas.

Emptying the DQD using Machine Learning

Whether a DQD is empty for a certain gate voltage configuration can be determined given a frame that correspond to a part of the charge stability diagram. A schematic of such a frame is shown in Fig. 4.2 (b). The frame size is $(f + p) \times (f + p)$ where f denotes the inner-frame size and p is called padding. Each frame correspond to a certain point in the charge stability diagram, illustrated with the red dot in Fig. 4.2 (a). Thus, a charge occupation can be assigned to the frames and each frame belongs to one of four classes, it can be that for this specific point both QDs are empty, it can be that only QD2 is empty, there can be only QD1 empty or both QDs are occupied. We denote these four possibilities as $(0, 0)$, $(1, 0)$, $(0, 1)$ and $(1, 1)$, respectively. In Fig. 4.3 (a) examples for all four classes are given, in (b) the frames drawn by the algorithm "Finding a reference point" from below are shown.

¹Error sources are discussed in Sec. 5.2.

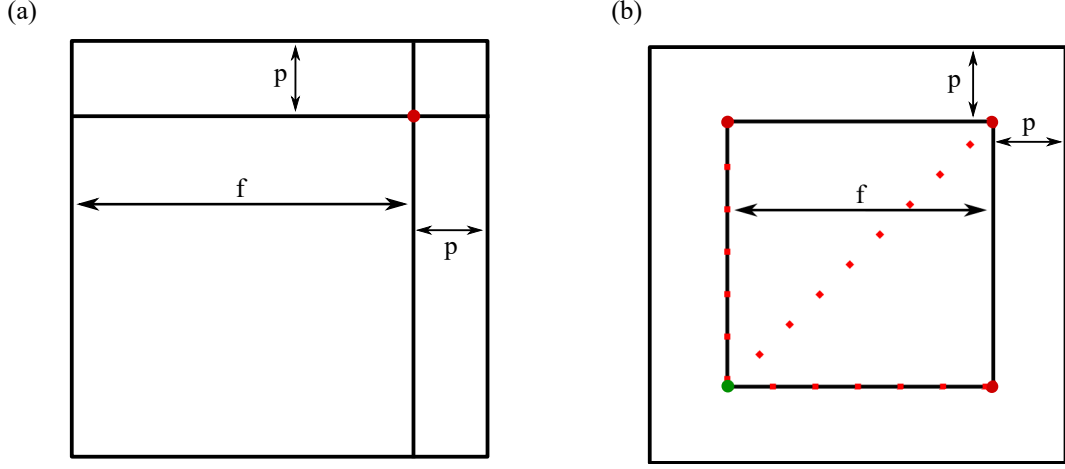


Figure 4.2: Scheme of the frames used for finding a reference point (a) and recognizing charge transitions (b). The distance p between inner and outer square is called padding. The inner part of a frame of size $f \times f$ is called inner-frame. (a) The frame used for determining whether a DQD is empty at the point indicated with the red dot. The frame has size $(f+p) \times (f+p)$. (b) The frame of size $(f+2p) \times (f+2p)$ used for recognizing charge transitions among path segments. Each frame comprises three path segments, connecting reference point (green dot) with the remaining corners of the inner-frames (red dots). The path segments are indicated with a red dashed line.

Algorithm. Finding a reference point

Let $PG1$ and $PG2$ denote the plunger gate voltages. Let $\delta > 0$.

1. Pick $PG1$ and $PG2$ randomly within some range.
2. Measure the frame corresponding to $(PG1, PG2)$.
3. Classify the frame. If both QDs are empty, return $PG1$ and $PG2$.
4. Otherwise, set $PG1 = PG1 - \delta$; $PG2 = PG2 - \delta$, and go back to step 2.

4.2.2 Recognizing Charge Transitions

The second part of the tuning algorithm finds a path connecting the previously found reference point with any point in the charge stability diagram possessing the desired charge occupation. This path is built up by individual path segments that are linked together. A machine learning classifier determines whether a charge transition occurred among each of the path segments. Based on the outcome, a new segment is added to the existing path.

Charge transitions are manifested as solid lines in the charge stability diagram called charge transition lines. In order to recognize the charge transitions among a given path segment starting from a certain point A and ending at a point B (A and B are points in the two-dimensional plunger gate voltage space), one needs to check how many, and what charge transition lines cross the chosen segment. We restrict ourselves to segments that have at most one transition per QD². There could either be no charge transition for any dot, there could be a transition in QD1, a transition in QD2 or a transition in both QDs. We indicate those four cases by $(0, 0)$, $(1, 0)$, $(0, 1)$ and $(1, 1)$, respectively. In machine learning terminology we call those cases classes. Every path segment obtains a label that belongs to one of the four classes. The machine learning task then consist of predicting the label given a path segment by recognizing which transition lines crossed a given path segment.

²Restricting to path segments with at most one transition per QD is achieved by choosing the individual path segments short enough. The step size between two measurement points for the frames needs to be chosen accordingly.

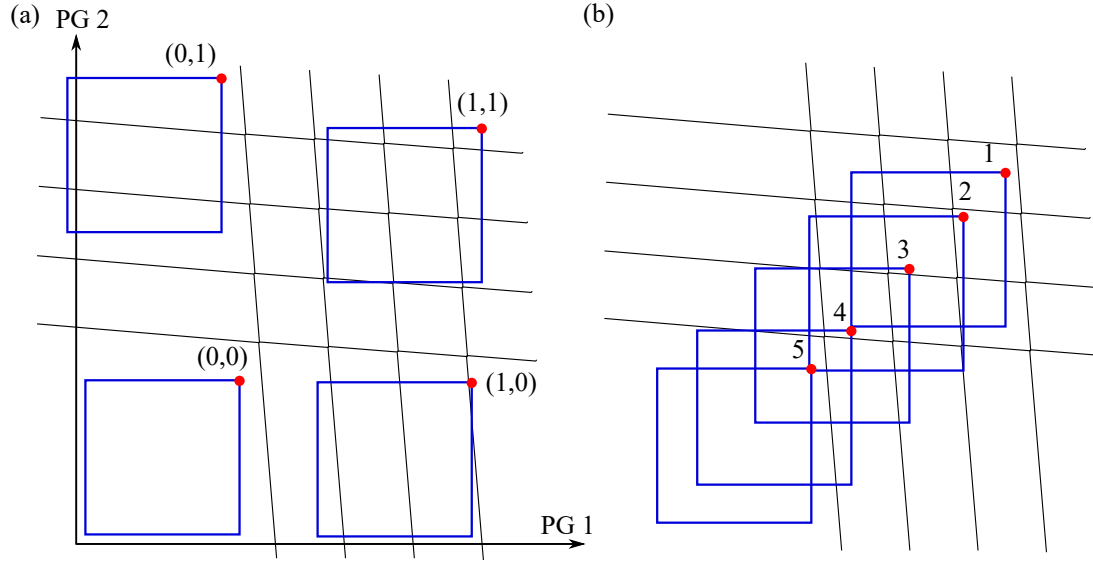


Figure 4.3: Idealized charge stability diagrams. The blue rectangles correspond to the inner frames used to determine whether or not the QDs have zero charge occupation at the upper right corner (red dot). (a) The four frames correspond to the four classification outcomes: both QDs empty (0,0), both QDs occupied (1,1), or one QD empty (1,0) and (0,1). (b) A series of inner-frames used to find a reference point. Both QDs are empty for frame nr. 5 and the algorithm "Finding a reference point" terminates.

The path segments are defined by frames that correspond to a small area of the charge stability diagram. A schematic of such a frame is shown in Fig. 4.2 (b). The frame size is $(f + 2p) \times (f + 2p)$ where f denotes the inner frame size and p is called padding. Each frame comprises three path segments (red dashed lines) each of which starts at the lower left corner of the inner frame and ends at one of its three remaining corners. In Fig. 4.4 (a) - (d) example inner frames with their different classification outcomes are shown. The corners of the inner frames then serve as potential starting point for the next path segment. The purpose of the padding of size p is to introduce additional information helpful to identify charge transition lines that lie close to any corner of the inner frame. By linking together path segment defined by frames we can build up a path that connects the found reference point with charge occupation (0,0) with a point with the desired charge occupation. An example of such a path created by linking frames can be found in Fig. 4.4 (e).

Algorithm. Auto Tuning of Double Quantum Dot Charge States

Objective: Find the voltages for plunger gate 1 (PG1) and plunger gate 2 (PG2) such that the corresponding state has charge occupation (n, m) for n, m freely chosen.

Let n_1^i and n_2^i be the charge occupation of QD1 and QD2 for each step i .

1. Find a reference point $x_0, i = 0$
2. While $(n_1^i, n_2^i) \neq (n, m)$, do:
 - Determine direction of next path $x_i \rightarrow x_{i+1}$
 - Measure the next frame and perform the classification
 - Update the charge configuration (n_1^i, n_2^i)
 - $i = i + 1$

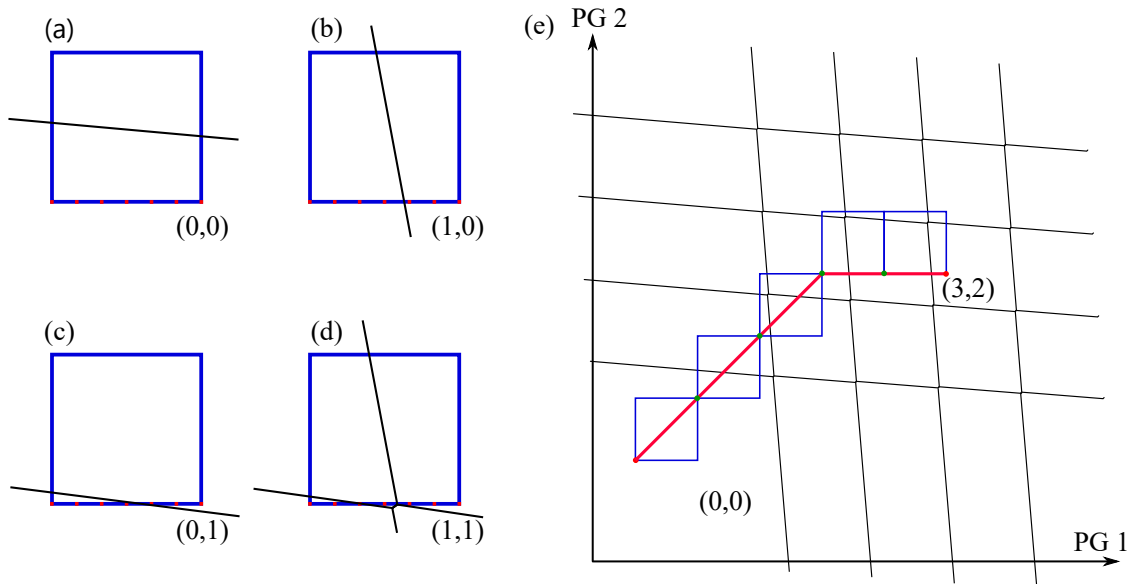


Figure 4.4: (a) - (d) Four frames including charge transition lines illustrating four different charge transitions among the path segment indicated with the red dashed line. The labels indicate whether a charge transition occurred on QD1 (1,0) (b), on QD2 (0,1) (c), both QDs (1,1) (d) or no transition occurred (0,0)(a). (e) A possible path taken by the auto-tuner. Starting from a reference point (lower left red dot) a frame is measured and a path segment is defined. Charge transitions are triggered among this segment. The occurred charge transitions determine the direction of the next path segment. Linked together, the path segments form a path connecting reference point with a point of the desired charge occupation. For visualization reasons, only the inner-frames are shown and paddings are omitted.

Chapter 5

Results

We present results obtained by applying the techniques described in Chap. 4 onto experimental data collected by procedure reviewed in Chap 4.2. Furthermore, we present an alternative approach of recognizing the charge occupation of QDs using unsupervised clustering techniques. The code of the complete auto-tuning routine is available on <https://github.com/redur/auto-tuner>.

5.1 Machine Learning Models

We developed two machine learning models including a data augmentation and processing pipeline. The model we call “reference point identifier” determines whether a DQD is empty for a certain gate voltage configuration. The second model, that we call “charge transition recognizer”, recognizes charge transitions occurring between two points in the charge stability diagram. Both models are convolutional neural networks (CNNs) built using the high level Tensorflow [43] API Keras[44].

Similarities of charge stability diagram frames and RGB images¹ allow to adopt well-developed image recognition and classification techniques [45, 46]. Such techniques include performing data augmentation [47] and using CNNs [48, 49] for the classification. The purpose of data augmentation is twofold. Firstly, the amount of training data is multiplied and secondly, the variety of the data is increased. The second aspect needs to be stressed at this point, as data augmentation enables to imitate measurements taken with other samples and thus, prevents overfitting to the sample at hand. Augmenting data consists of applying transformations that preserve the key features of charge stability diagrams that determine the classification outcomes. Both of our classifiers aim at recognizing charge transition lines. Transformations that conserve those lines are e.g. affine transformation, adding noise, introducing an offset and many more.

We augmented each of the measured charge stability diagrams by a factor of 18 using a random combination of rotations, scaling and axes flipping. Examples of augmented charge stability diagrams are found in Fig. 5.1. The augmentation transformations aim at simulating charge stability diagrams for different capacitive couplings. Thus, by applying these transformations different realizations of a DQD are imitated. All data augmentation has been performed using the python library imgaug [50].

Frames are randomly drawn from all augmented maps resulting in a training data-set of 650,000 frames for the charge transition recognizer and 530,000 frames for the reference finder. All those frames are then processed individually, since during the runtime of the auto-tuner, frames will be

¹Charge stability diagrams can be seen as images. However, computationally charge stability diagrams consist of two dimensional arrays whose entries are floats. RGB images are represented using three dimensional arrays consisting of 8-bit integers.

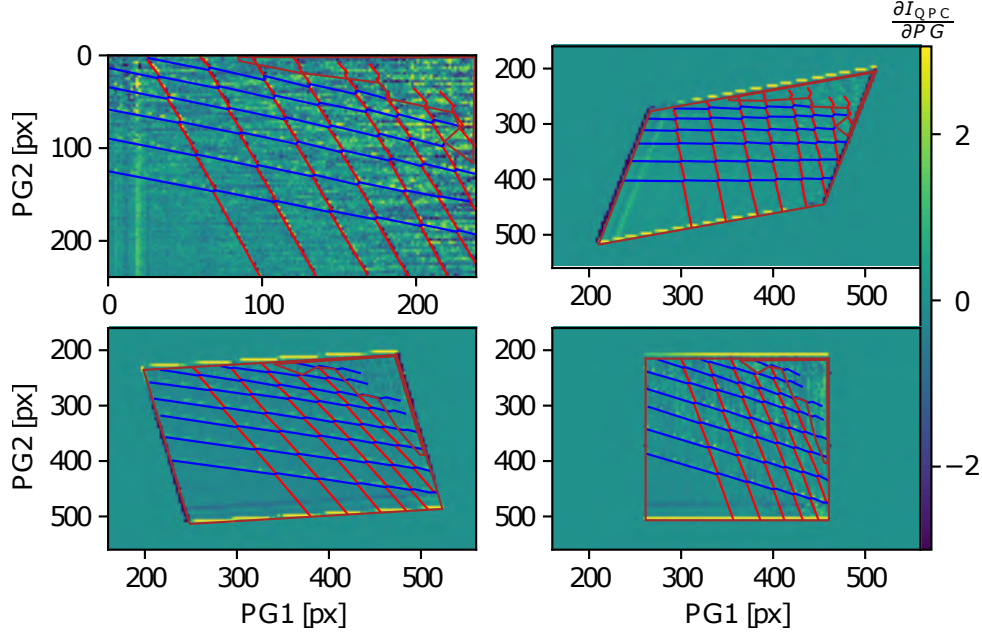


Figure 5.1: A charge stability diagram with three different augmentation transformations. The blue lines indicate charge transitions in PG2, the red line indicate charge transitions in PG1 and the brown lines encloses an area in which a charge allocation can not be made. The top left plot shows the non-augmented charge stability diagram. All other diagrams origin from a combination of scaling (in PG1 and PG2 axis), rotations and axis flipping applied on the top left one. All units are in pixels corresponding to individual measurement points.

the only available information for the classifications. Every frame is rescaled such that the variance equals to one². The derivative according to Eq. (3.2) is calculated, outliers³ are removed and the median is subtracted. For a detailed discussion of the measured charge stability diagrams, the applied transformations as well as the data processing we refer to the appendix A.2.

CNNs are a class of multi layer (or deep) neural networks that are commonly used for classifying images since they manage to both, efficiently handle large feature spaces and capturing correlation between features (pixels) [48, 49]. The reference identifier CNN consists of two convolutional and one fully connected layer with a total of 141,164 trainable parameters. The detailed architecture is shown in Fig. 5.2. We used a similar architecture for the charge transition recognizer shown in Fig. 5.3. This architecture consists of three convolutional and one fully connected layer resulting in 248,642 trainable parameters. This CNN further has three outputs, whose outcomes correspond to the predicted charge transitions of the three path segments defined within each frame. Both CNNs were created with the python library Keras [44], a high level API of Tensorflow [43] and trained using the ETH supercluster "Leonhard". For more information on the CNN architectures and training of the models we refer to the appendix A.3 and A.4.

We evaluate both CNNs using their corresponding evaluation data-set. No augmentation is performed on the evaluation sets, in order to get a performance score on real measurements. The evaluation data-set for the reference identifier consists of 3152 frames and the evaluation set for the charge transition recognizer contains 2054 frames. The classification outcomes for the evaluation data-sets for both CNNs are visualized in the confusion matrices shown in Fig. 5.4. For the reference finder, it is important not to classify occupied regions as empty. Thus, the score for

²For computational reasons, the complete maps are already rescaled before augmentation transformations are applied. However, we rescale the frames again at this point.

³For a definition of what we consider as outliers we refer to the Appendix A.2

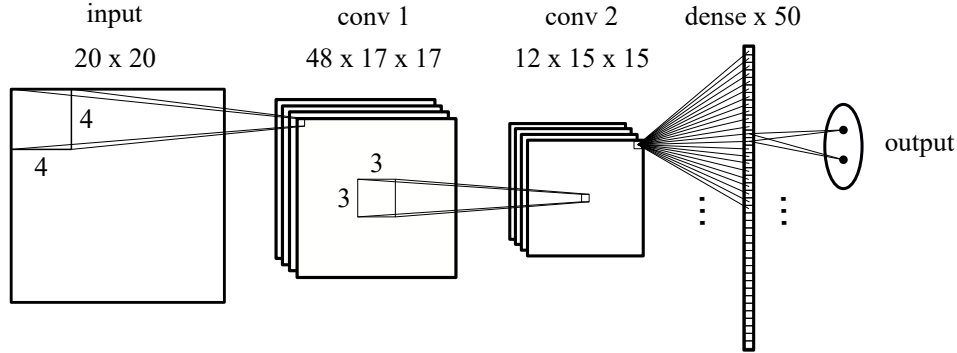


Figure 5.2: A scheme of the architecture of the neural network used for whether the DQD is empty for a certain gate voltage configuration. We call the model "reference point identifier". The input image consists of 20x20 floating point numbers. In a first step, 48 filters of size 4x4 are used, resulting to 48 maps of dimension 17x17. This layer is called a convolutional layer. In a next step, another 12 filter of size 48x3x3 are applied, reducing the dimension to 12x15x15. Then a fully connected (or dense) layer is applied with 50 nodes. Finally, there are two output nodes corresponding to the classification results "empty" and "occupied". In total, this architecture has 141,164 trainable parameters.

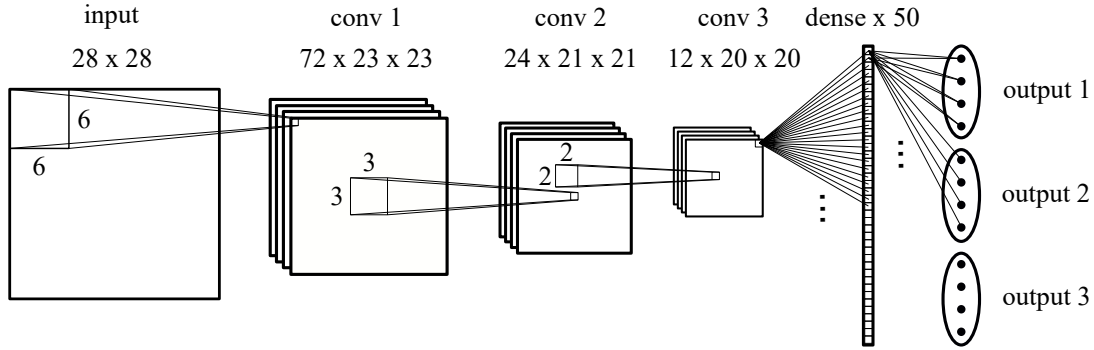


Figure 5.3: A scheme of the architecture of the neural network used for recognizing charge transitions, by us called charge transition recognizer. The input image consists of 28x28 floating point numbers. In a first step, 72 filters of size 6x6 are used, resulting to 72 maps of dimension 23x23. This layer is called a convolutional layer. Two more convolutional layers are applied, the first consisting of 24 filters of size 72x3x3 are and the second consisting of 12 filters of size 24x2x2 reducing the dimension to 12x20x20. Then a fully connected (or dense) layer is applied with 50 nodes. Finally, there are three outputs corresponding to the three corners of the frame that are classified. Each of the outputs has four nodes, corresponding to the four possible classes a corner of an inner-frame can belong to. In total, this architecture has 248,642 trainable parameters.

which we optimized our model is precision on the class 'empty' given as 0.975. The overall accuracy remains high with 0.989. There are three confusion matrices for the charge transition recognizer corresponding to the three classifications made per frame. Here, we optimized the model for the average overall accuracy which is 0.963. The performance of the classifiers on real-time measured data during runs of the auto-tuner is discussed in Sec. 5.2.

5.2 Performance of the Auto-Tuner

To assess the performance of the auto-tuner we performed 160 evaluation runs. The auto tuner successfully found the correct plunger gate voltages in 91 runs or in 57%. In 142 runs or 89% the tuner found the correct charge occupation allowing for a discrepancy of one electron. Within the 160 evaluation runs, the task of the tuner was to find the charge regimes (1, 1), (1, 2), (2, 1) and (2, 2). For each of these charge occupations we used four different gate voltage configurations in different DQD realizations with e.g. different tunnel couplings and charging energies. The confusion matrix (see Fig. 5.5) illustrates the outcomes of the evaluation runs.

The DQD formed with QD1 and QD2, called “training DQD”, is the one with which the training data was acquired. The other DQD formed by QD2 and QD3 we call “testing DQD”. We expect the auto-tuner to perform better on the training DQD, since the Machine Learning models were trained on its data and thus, potentially overfitted to specific features⁴. In order to assess to what extent the performance generalizes to other samples we performed half of the evaluation runs on the testing DQD. Furthermore, for the evaluation runs on the testing DQD we used the right QPC formed with the QPC_R gate which further differentiates its measurement signal from the initial, training DQD. For both DQDs we used the same parameters for the tuning algorithm as e.g. measurement resolution or the number of frames that are drawn before the QPC is recalibrated. On the training DQD 56% of the runs were successful whereas on the testing DQD 58% were successful. The comparable performance on the testing DQD indicates a good generalization to other samples. We explain the discrepancy of 2% (corresponding to one run) by statistical fluctuations. In the following we investigate both parts of the algorithm, the reference finding and the monitoring of charge transitions among path-segments in more details.

The reference finding succeeded in 144 out of 180 runs or 90% of the cases. An example of such a run where a reference point is successfully found is shown in Fig. 5.7. The classification outcomes of all frames of the evaluation runs are represented in a confusion matrix in Fig. 5.6 (a). The relevant score, precision on class empty, is 0.9 and the overall accuracy is 0.92.

The prediction of the charge transitions of the individual frames of the evaluation runs are 92.3% accurate, see the confusion matrix in Fig. 5.6 (b). Linked together, the frames extend to a path connecting plunger gate configurations of the desired charge occupation with the previously found reference point. Given a correct reference point, the total path is correctly found with 63% accuracy. Successful examples of such a path are shown in Fig. 5.9.

We identified two sources of errors, weak signal-to-noise ratio and what we call misalignment of frames. For weak signal-to-noise ratios, the classifiers are not able to recognize charge transition lines and misclassifications become frequent. Misclassifications may also occur, when the point that determines the class allotment of a frame exactly lies on a charge transition line. For example, the charge transition recognizer may miss a charge transition line crossing one of the frames corner. We call this case misalignment of frames.

⁴Such a feature could for example be the capacitances between plunger gates and QDs.

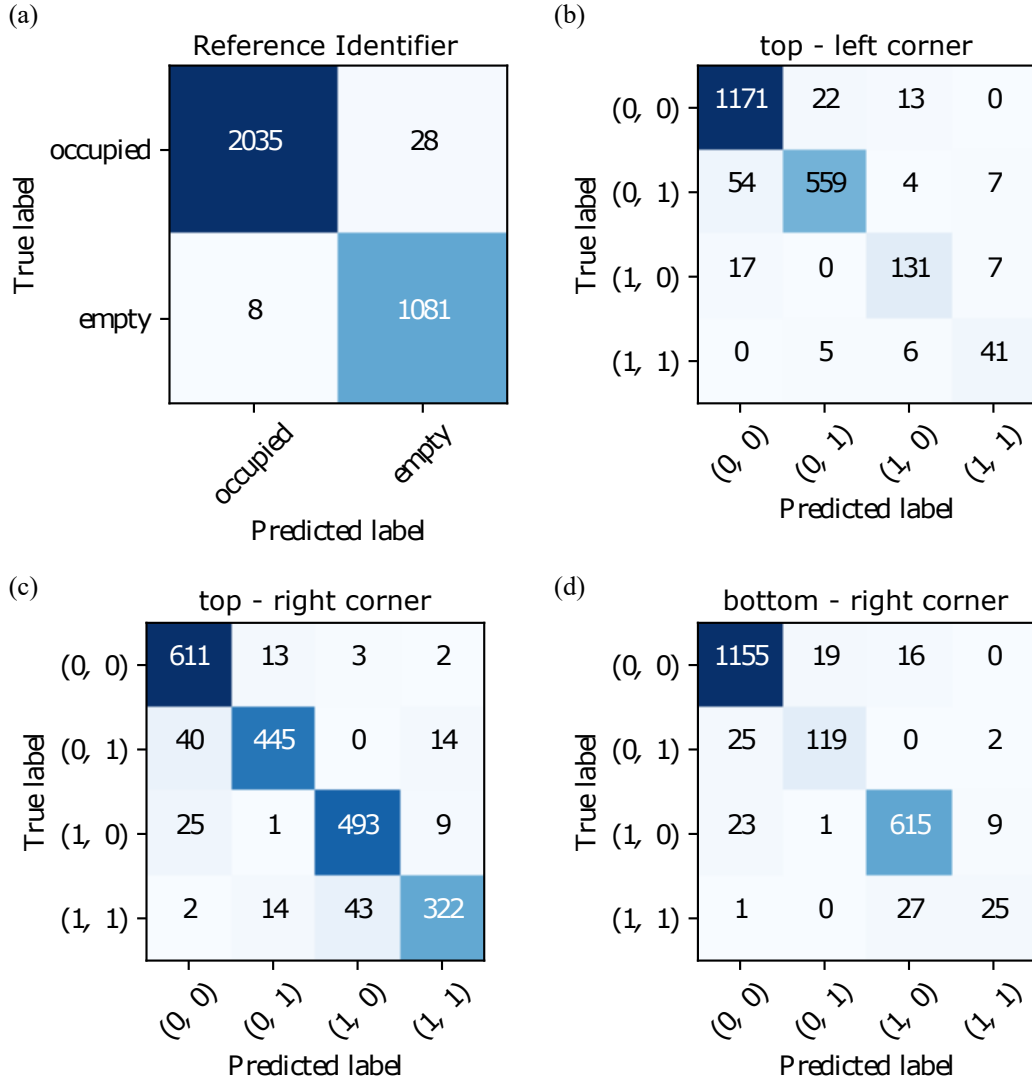


Figure 5.4: Confusion matrices for the machine learning models used evaluated on their corresponding evaluation data-set. (a) Confusion matrix for the reference identifier. Its evaluation set consists of 3152 data-points. (b) - (d) The confusion matrices for the charge transition recognizer. Classification outcomes of the top-left corner (b), top-right corner (c) and bottom-right corner (d) of the inner-frames. For these three confusion matrices the same evaluation data-set containing 2054 data-points is used. The label (n, m) indicates the charge transitions. For $n = 1$ a transition on QD1 occurred, similarly for $m = 1$ there is a charge transition on QD2. Where $m = 0$ or $n = 0$ no charge transition happened in the respective QD.

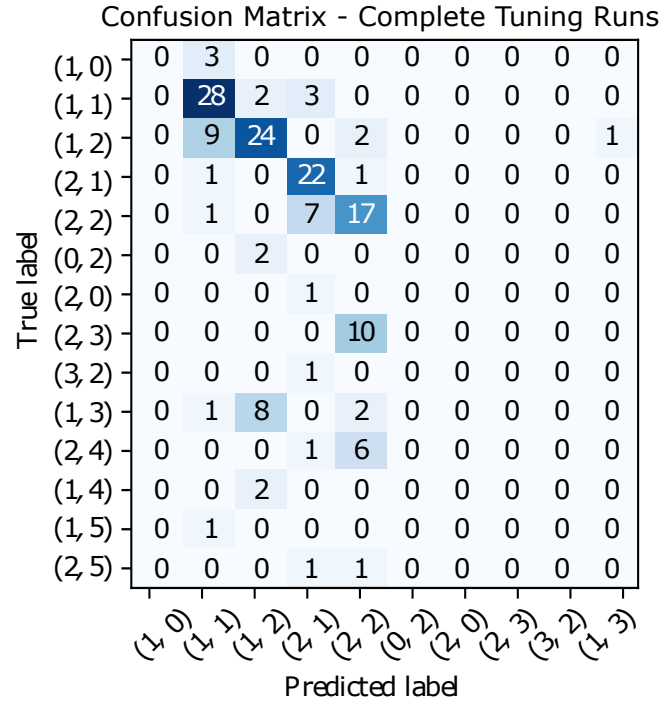


Figure 5.5: Confusion matrix of the evaluation runs of the the auto tuning of a DQD's charge state. The rows correspond to the true charge occupation corresponding to the plunger gate voltage configuration found by the auto tuner. Columns represent the charge state predicted by the auto tuner. For all diagonal entries, true and predicted label match and thus, the tuner found suitable voltages for the plunger gates. Two of the 160 evaluation runs are not listed here, since a label could not be allocated to them.

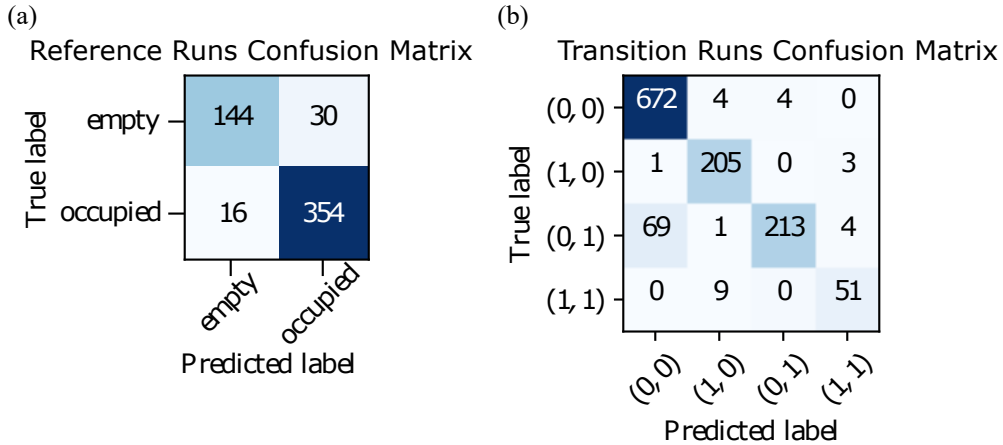


Figure 5.6: Confusion matrices for the classification outcomes of all frames of the evaluation runs for the reference identifier (a) and the charge transition recognizer (b). For the task of finding a charge reference point, the relevant score is precision for the class 'empty' which is 90%. For the classification outcomes of the charge transition recognizer 'accuracy' is the relevant score and is 92.3%. The misclassification of (0, 1) as (0, 0) account for 73% of all misclassifications.

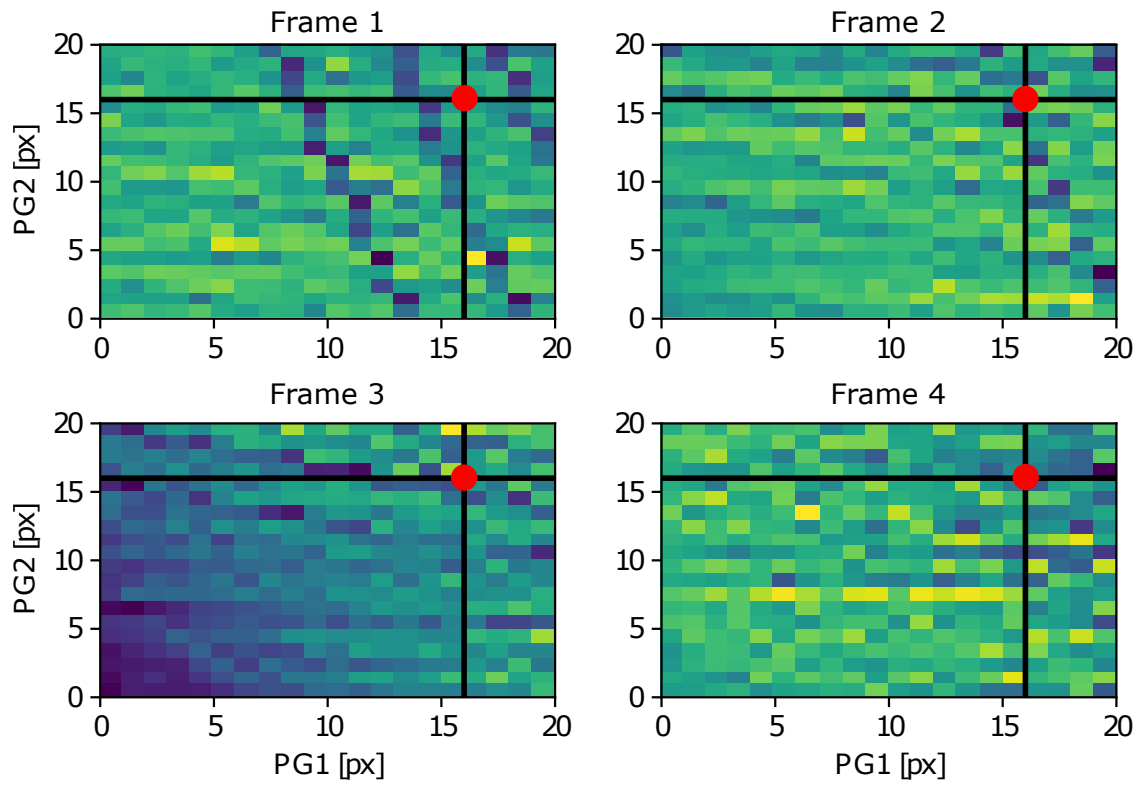


Figure 5.7: A successful run of the reference charge finding algorithm. The red dot indicates the voltage configuration to which each of the frames belong to. All four frames are correctly classified. Frame one to three correspond to an 'occupied' state and frame number four is 'empty'. Thus, the red dot in frame number four is a charge reference point.

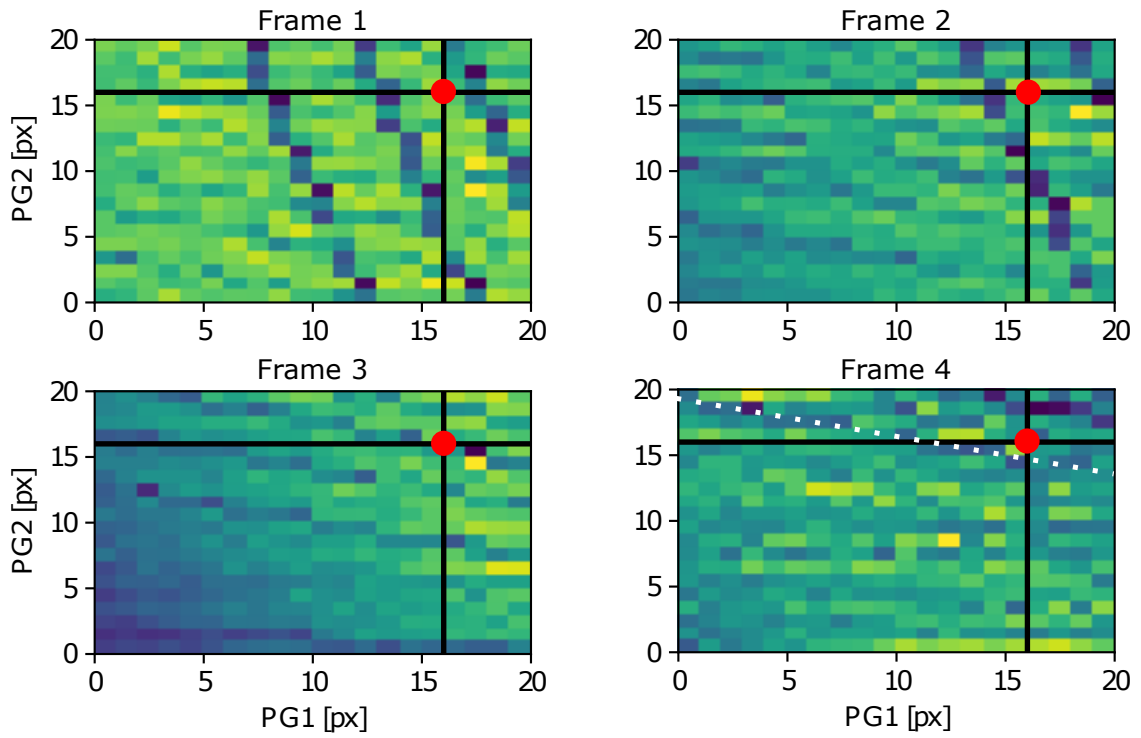


Figure 5.8: An unsuccessful run of the reference charge finding algorithm. The red dot indicates the voltage configuration to which each of the frames belong to. Frames one to three are correctly classified as 'occupied'. However, the 'occupied' frame four is classified as 'empty'. The charge transition line of the last electron in frame 4 is indicated with a white dashed line.

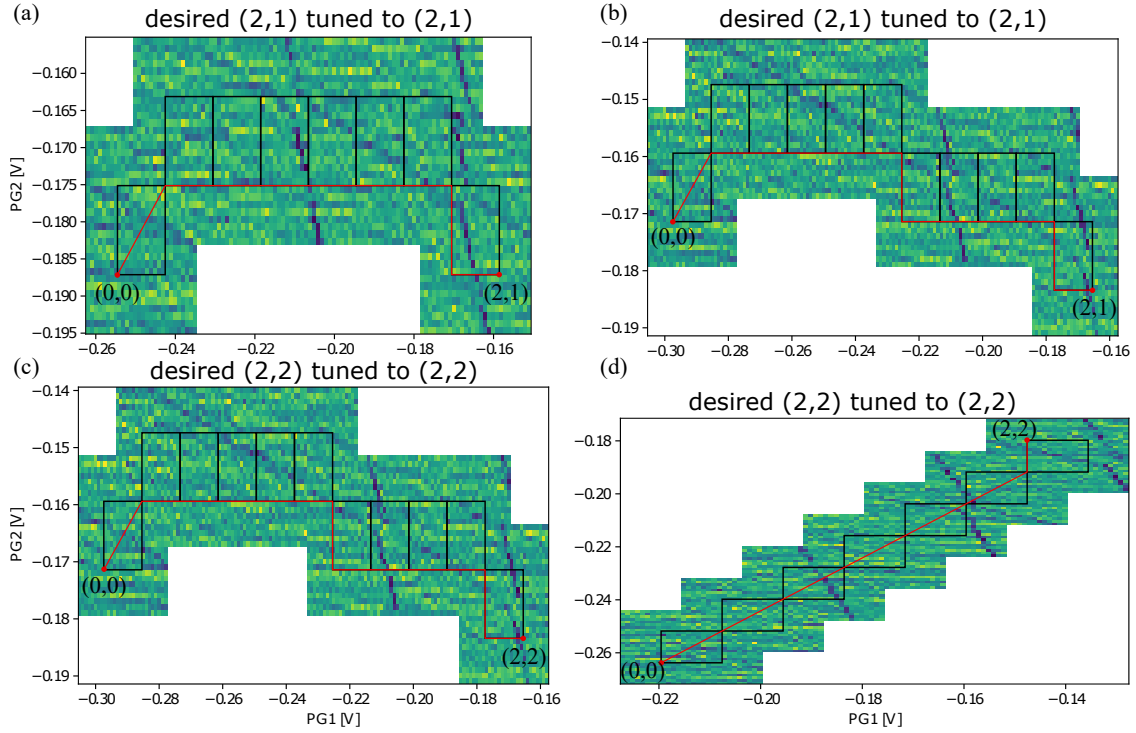


Figure 5.9: Four successful tuning runs. Each of them started in the (0,0) charge configuration. The black boxes are the inner-frames defining path segments among which charge transitions are recognized. Together, the path segments form a path starting from (0,0) to the desired charge occupation.

An example of a case where a false reference point found - i.e. a non-empty area is classified as empty - is shown in Fig. 5.8. The white dashed line in the figure indicates the charge transition of the last electron for QD3 which was not recognized due to a weak signal-to-noise ratio. Since QD3 is placed further away from the QPC, its signal-to-noise ratio for QD3 is smaller than for the other QDs. Examples for false classifications due to weak signal-to-noise ratios for the charge transition recognizer are shown in Fig. 5.10 (c) and (d). Examples for false classifications due to misalignment of frames are shown in 5.10 (a) and (b). Areas where transitions were not recognized are indicated with a white dashed circle.

Weak signal-to-noise ratios account for 72% of all errors made and misalignment of frames account for 24% of the errors. The remaining 4% could not be assigned to either of the error sources. A tree representation dividing the unsuccessful runs into its error categories is shown in Fig. 5.11. The confusion matrix of the transition runs shown in Fig. 5.6 (b) illustrates the domination of signal-to-noise induced errors. For these errors, typically transitions are not recognized as such, resulting to entries in the confusion matrix like classifying (0,1) as (0,0) or classifying (1,1) as (1,0). The charge transition lines of the QD whose charge is indicated with the second number (\cdot) are weaker, since this QD is further away from the QPC. Thus, the signal-to-noise ratio decreases and errors get more frequent. These mistakes of missing transitions of the QD placed further away from the QPC account for 82% of all misclassifications of the charge transition recognizer.

Errors due to misalignment of frames could be corrected by repeated classification of slightly shifted frames. However, the majority of the errors are due to weak signal-to-noise ratios. There are several - but limited - ways to overcome this issue. One way is to increase the measurement time by increasing the integration time of the measurement. Alternatively, one could also measure more points, i.e. increase the number of points within one frame. Another possibility is to increase

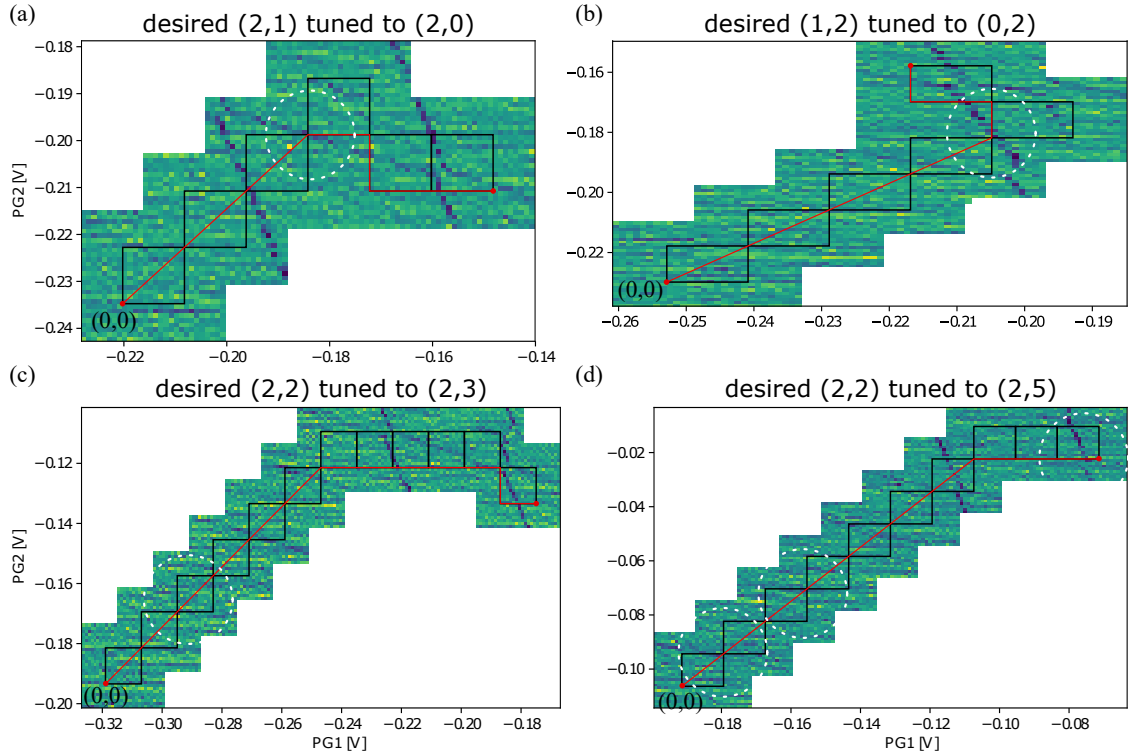


Figure 5.10: Four unsuccessful tuning runs. Each of them started in the $(0,0)$ charge configuration. The black boxes are the inner-frames defining path segments among which charge transitions are recognized. The dashed white circles indicate misclassifications. In (a) and (b) an edge of an inner-frame lies exactly on a charge transition line, and thus the transition line is recognized as such in two consecutive path segments. The misclassifications in (c) and (d) are due to a weak signal-to-noise ratio for the transitions in QD3 (horizontal charge transition lines).

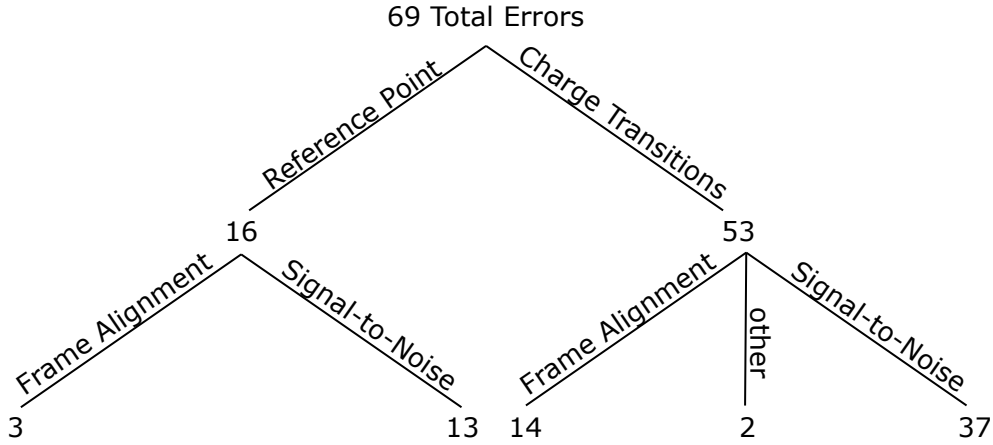


Figure 5.11: Tree representation of the unsuccessful evaluation runs and their error source. An error either occurred in the reference point finding, or in the recognition of charge transitions. The identified error sources are misalignment of frames and weak signal-to-noise ratios.

noise level in the training data via data augmentation transformations such that a classifier learns to handle weak signal-to-noise ratios.

5.3 Alternative Approach

In this section, we present a method that does not require training on manually labeled data and additionally performs a classification given a single measurement point in contrast to a complete frame. We call the method Clustering of Charge Induced Current Offset. The idea of this approach is to find a quantity that is constant for regions of equal charge occupation. Being able to compute this quantity efficiently and reliably will allow to cluster different areas of the charge stability diagram that correspond to states with different charge occupations. We call the quantity we define within this chapter Charge Induced Current Offset. For the clustering we use unsupervised machine learning techniques, particularly Gaussian Mixture Models. First, we introduce to Gaussian Mixture Models, a commonly used method in unsupervised clustering, then we introduce the charge induced current offset and discuss its clustering for determining the charge occupation of a DQD.

5.3.1 Unsupervised Clustering

Unsupervised learning is a subfield of Machine Learning where the given data-points' labels are unknown. A common unsupervised problem is clustering, where one aggregates data-points that share common characteristics together. These bunches of data-points are called clusters. One then assigns different labels to different clusters. One widely used clustering approach is called Gaussian Mixture Model (GMM) which we use within this thesis. For a general introduction to unsupervised learning methods and Gaussian Mixture Models we refer to [45, chap. 2, 9].

The goal of a Gaussian Mixture Model is to approximate the probability distribution from which data-points are drawn. Each cluster is assumed to be of Gaussian form, i.e. we assume that a data-point belonging to cluster i is drawn from a Gaussian distribution $\mathcal{N}_i(\theta_i)$ with parameter θ_i . In order to get the total probability function we assume that all data-points are allocated to a cluster. Further, data-points associated with different clusters are drawn from different probability

distributions.⁵ Formally, a Gaussian Mixture is then a probability distribution p formed by a convex combination of Gaussian distributions

$$p(x) = \sum_{i=1}^k p_i \mathcal{N}_i(x|\theta_i) \quad (5.1)$$

with k the number of clusters and $\sum_{i=1}^k p_i = 1$. The θ_i are the parameters of the individual Gaussian distributions \mathcal{N}_i . [51, 52, 53] Given the number of clusters k one can maximize the likelihood of the observed data-points as a function of the parameter θ_i . The optimization can be performed by using the Expectation Maximization algorithm. [54] The number of clusters k can be determined by comparing the obtained likelihood on previously unseen data.

In clustering, the optimization of the likelihood is called training. Once trained, a GMM is able to perform classifications by comparing the likelihood of a data-point belonging to one of the clusters with which the mixture is build. Formally, given a data-point x the label j is determined as

$$j = \arg \max_i \mathbb{P}[Y = i|x] \quad (5.2)$$

where Y denotes the cluster or label. This translates to

$$j = \arg \max_i (p_i \mathcal{N}_i(x|\theta_i)). \quad (5.3)$$

5.3.2 Clustering Charge Induced Current Offset

In this section we define the Charge Induced Current Offset and describe how its clustering can be used for determining the charge occupation of a DQD.

Assume we knew the QPC current as a function of the electrostatic potential induced by the gates. In other words, assume that we are given a function of the gate voltages $\hat{I}_{\text{QPC}}(V_{\text{PG1}}, V_{\text{PG2}}, \dots)$, which we call predicted current, that predicts the current through the QPC due to a set of voltages applied on the gates. A charge located on any of the QDs is inducing an additional potential at the QPC affecting its current as well. Thus, the true, measured QPC current I_{QPC} does not equal to the predicted current, i.e. $\hat{I}_{\text{QPC}} \neq I_{\text{QPC}}$, unless the charge occupation is $(0, 0)$.

Definition. Charge Induced Current Offset (CICO)

We call the difference between predicted \hat{I}_{QPC} and measured current I_{QPC} *Charge Induced Current Offset* $\Delta_{I_{\text{QPC}}}$.

$$\Delta_{I_{\text{QPC}}} = \hat{I}_{\text{QPC}}(V_{\text{PG1}}, V_{\text{PG2}}, \dots) - I_{\text{QPC}}(V_{\text{PG1}}, V_{\text{PG2}}, \dots). \quad (5.4)$$

In order to get a prediction for QPC current \hat{I}_{QPC} , we first empty the DQD by tuning the plunger gates appropriately. We then measure I_{QPC} sweeping the QPC gate voltage. Interpolating these measurement results using a spline leads to a function⁶ f . This functions describes the effect of a local potential on the QPC current induced by the QPC gate and we can write $f(V_{\text{QPC}}) = \hat{I}_{\text{QPC}}$. By assuming constant couplings between QPC and all gates we can write

$$\hat{I}_{\text{QPC}}(V_{\text{PG1}}, V_{\text{PG2}}, V_{\text{QPC}}) = f(\alpha_1 V_{\text{PG1}} + \alpha_2 V_{\text{PG2}} + \beta V_{\text{QPC}} + \epsilon). \quad (5.5)$$

The real parameters α_1 , α_2 , β and ϵ are found by fitting f to measured data for which the DQD's charge occupation is $(0, 0)$.

⁵ A detailed list of assumptions made using GMMs is beyond the scope of this discussion and we refer to [51].

⁶ $f : \mathbb{R} \rightarrow \mathbb{R}; V \mapsto I_{\text{QPC}}$

Two plots comparing the predicted QPC current \hat{I}_{QPC} with the measured values I_{QPC} are shown in Fig. 5.12 (a) and (b). Knowing both, the predicted and measured current for a range of PG1 and PG2 voltages, it is possible to calculate the CICO $\Delta_{I_{\text{QPC}}}$ which is shown in Fig. 5.12 (c). It is seen that areas with equal charge occupation have a similar CICO.

Since the CICO $\Delta_{I_{\text{QPC}}}$ solely depends on the charge occupation of the DQD⁷, the clusters obtained by a GMM directly correspond to different charge occupations. Thus, once trained, a GMM can differentiate between certain charge occupations of the DQD for a range of gate voltages. The GMM allocates a gate voltage configuration to a cluster given a single measurement point only. In Fig. 5.13 (a) and (b) we illustrate the clustering of the CICO. The in (a) encircled points all correspond to the charge occupation (0, 0) which matches with the lowest cluster of the CICO seen in (b). We trained a GMM on a fraction of $\frac{1}{200}$ of the data-points shown in Fig. 5.12 (c) distributed on a grid. We clustered all points and visualized the allotment to the clusters in Fig. 5.14. The fitted GMM has 13 classes. Each cluster contains several charge occupations. In particular, points with the same total charge occupation of the DQD are mostly assigned to the same cluster. This allocation manifests in the diagonal distribution of the classes in Fig. 5.14.

A GMM that is able to differentiate between different charge states can be used for auto-tuning of quantum dot charge states in a similar manner as presented Sec. 4.2. Starting at (0,0) one can decide on where to measure next, make the classification and go on. Since the CICO is a monotonic function for each of the gates individually, binary search could be used.

Clustering of CICO has a few advantages over the method of classifying frames that build up a path connecting reference point with the final charge regime. The classified frames are composed of $n \times n$ measurement points⁸. These frames are linked together building up a path. Therefore, a classification error in any of the frames leads to the wrong charge regime found by the tuner. Not all points in the charge stability diagram can be reached, but only those points that can be connected with the reference point by linking frames starting from the reference point. In contrast to the frame classification approach, clustering of CICO only needs one measurement point. The clustering of each point is independent of the outcome of any other clustering. Furthermore, any point in the charge stability diagram can be reached. Since the CICO is an monotonous function for each gate individually, binary search could be used to find the desired charge state which would further speed up the tuning procedure. Also, clustering of CICO naturally extends to tuning more gates and also larger architectures containing more than two QDs.

⁷ This is the assumption made here. Within a QD system, this is true as long as the couplings between gates, quantum dots and QPC stay constant.

⁸ In our case either $n = 29$ or $n = 21$.

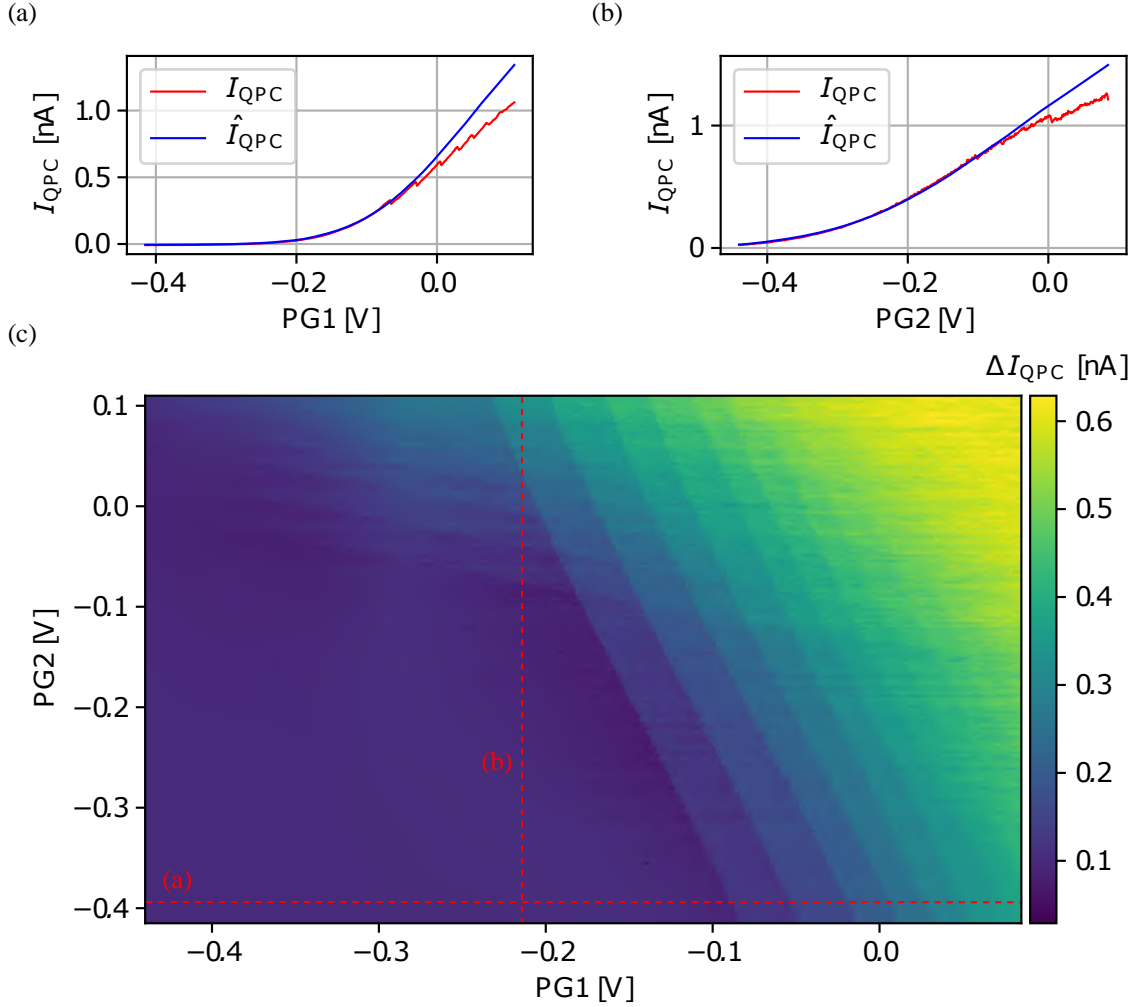


Figure 5.12: (a) - (b) Measured I_{QPC} and predicted \hat{I}_{QPC} QPC current for the data indicated with the dashed lines in (c). (c) Difference between measured and predicted QPC current, called charge induced current offset ΔI_{QPC} . Points corresponding to different charge occupations of the DQD differ in ΔI_{QPC} .

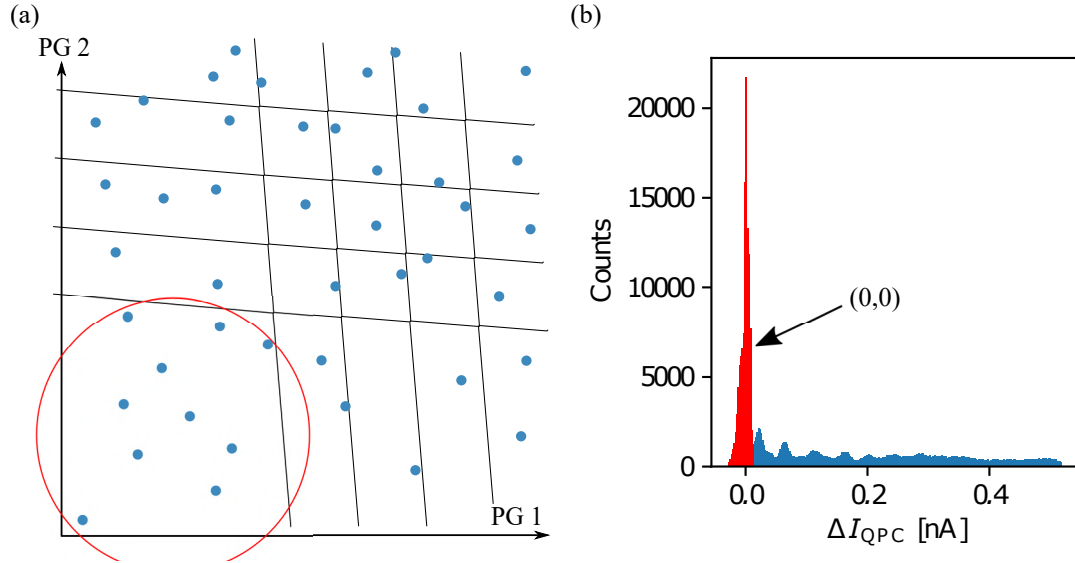


Figure 5.13: (a) An idealized charge stability diagram. Randomly taken measurements are indicated with blue dots. (b) Histogram of the charge induced current offset for randomly distributed plunger gate voltage configurations as indicated in (a). The lowest cluster indicated in red corresponds to the (0,0) charge area encircled in (a). Note that (a) is an idealized scheme whereas the data in (b) is based on measurements.

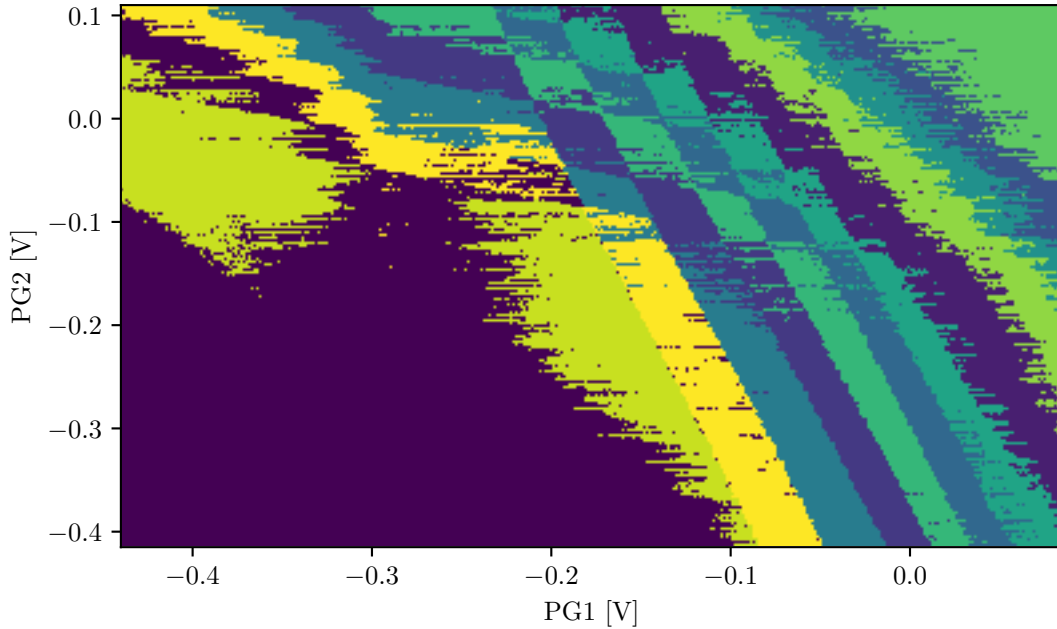


Figure 5.14: Visualization of a clustering of the charge induced current offset. It is the result of a Gaussian Mixture Model with 13 classes fitted on the charge induced current offset shown in Fig. 5.12 (c). For the fitting, only every 200st data point is used, but all points are classified for visualization purposes. Points with equal total charge occupation (i.e. the sum of the number of electrons on each dot is equal) are mostly assigned to the same cluster.

Chapter 6

Conclusion and Outlook

In this thesis we successfully implemented a machine learning assisted auto-tuner that determines the plunger gate voltages for any DQD charge configuration with 57% accuracy. Allowing for a deviation of one electron, this score extends to 89%. The auto-tuner is realized using two convolutional neural networks. The task of the first network is to determine whether the DQD is empty for a certain voltage configuration. The second neural network triggers charge transitions between two points in the charge stability diagram. We furthermore developed a data acquisition, augmentation and processing pipeline. Lastly, we presented a first implementation of an alternative approach for identifying charge states of DQDs using unsupervised clustering techniques.

The performance of the auto-tuner was found to be independent of the used DQD. However, both DQDs on which evaluation took place have a very similar architecture, the measurements have been recorded within the same cooldown¹ and one QD is shared between the realizations of the DQDs. For a more stringent check on the auto-tuner, we propose to test on completely different samples in order to assess its generalization.

We identified weak signal-to-noise ratios and what we call misalignment of frames being the major sources of errors. Weak signal-to-noise ratio accounts for 72% of all errors. Potential improvements involve optimizing integration time of the measurements and optimizing the measurement resolution as well as the frame dimensions. One could further introduce noise to the training set as an augmentation transformation and improve the performance of the CNN. On the other hand, errors due to misalignment of frames can be overcome by adjustments to the algorithm. For example, performing two classifications with slightly shifted frames should overcome misalignment problems.

We furthermore presented an approach which we call clustering of charge induced current offset (CICO). This approach does not require training on manually labeled data-points and performs a classification given a single measurement point of the QPC, in contrast to the auto-tuner where several frames of the charge stability diagram need to be measured. The idea here is to cluster the CICO, a quantity that remains constant for regions of equal charge occupation. Thus, different clusters of the CICO correspond to different charge occupations. We trained a Gaussian Mixture Model that is able to differentiate between different total charge occupations. The clustering of CICO naturally expands to larger architectures containing more gates as well as more QDs. Furthermore, it is possible to use it without the need of training machine learning models on other architectures first. Thus, no overfitting to specific samples or architectures may occur. Also, the charge occupation of any gate voltage configuration may be obtained contrary to the auto-tuner that only reaches points lying on a corner of an inner-frame.

¹Heating up and cooling down again would amongst other things lead to a change of the capacitances between the QDs and the gates.

Despite the numerous advantages of clustering CICO over the tuning approach using frames, challenges still remain. It is not clear yet on how to identify the charge occupation of a cluster. Also, by now several charge occupations are assigned to the same cluster. Particularly, areas with the same total charge occupation (sum of both QDs) are mainly allocated to the same cluster. And thus, no auto-tuner using clustering of CICO has been worked out yet.

There are several possibilities that may lead to an improved clustering of charge occupations. As of now, we only explored the CICO. Including additional information as for example the gate voltages applied may lead to better clustering as points with similar gate voltages are likely to have the same charge occupation. One could use a sample where QD1 and QD2 couples differently to the QPC such that the clustering is sensitive to charge transport between the dots as well. Furthermore, any procedure increasing the signal-to-noise ratio of the QPC will lead to a better result, one could for example optimize the integration time of the measurements.

There are two further research direction that may successfully extend upon results presented here: generalization of the supervised approach to larger training sets consisting of data from other samples and wide variety of noise. Exploring this direction would entail collecting and label more data for larger amount of experiments and adapting the neural network architecture presented. Another direction is to explore further the GMM clustering technique that offered initial promising results in our case. While the supervised approach should show more precise and resilience to noise, the unsupervised clustering approaches can offer scalability that will be important as devices with larger number of quantum dots are built.

Appendix A

Data and Learning

In this section we present the details on creating a machine learning model the acquiring data, processing it, defining an architecture and finally training it. How we acquired the data is explained in Sec. A.1. Data augmentation is outlined in Sec. A.2 and the neural network architecture and its parameters are covered in Sec. A.3 and finally, evaluation scores of the classifiers are discussed in Sec. A.4.

A.1 Data Acquisition

We use labeled frames as illustrated in Fig.4.2 as an input for the classifiers. In order to retrieve those frames, we measure what we call maps. By map we mean a plot (or measurement) of the QPC current as a function of both PG voltages over a range much larger than the frame size. Fig. 3.5 shows a measured map and Fig 3.6 shows its differential QPC signal. We call the maps used to learn the charge transition recognizer fine maps and the maps for the reference finder, that determines whether the DQD is empty coarse maps due to its different resolutions¹. In both cases we varied the gates indicated in Fig. 3.1 as LRG, LTG, RTG as well as WGL and WGR. Additionally, for the coarse maps we also varied the step size for the plunger gates which we set to $(6 - 8.5mV)$. For the fine maps the step size is constant and equals to $1mV$. We measured 128 fine maps and 472 coarse maps. Fine maps are augmented² by a factor of 18 and coarse maps are augmented by a factor of 36. For the charge transition recognizer we drew about 300 frames within each of the obtained maps resulting to approximately 650,000 frames. For the reference identifier we drew about 45 frames per map which results to approximately 530,000 frames.

For each of the fine maps, we manually indicated all charge transition lines using the Python library `pyplot.matplotlib` and saved the results as a pickle file. We differentiated between QD1 and QD2 transitions. Furthermore, we marked areas of the charge stability diagram in which we could not recognize eventual charge transition lines and ignored those areas in the subsequent steps. For the coarse maps, we marked the transitions between regions where the DQD is empty and where it is occupied in the charge stability diagram. A fine map including marked charge transition lines can be seen in Fig. 5.1 (a). The measured maps together with the manually indicated charge transition lines serve as raw data.

A.2 Data Augmentation and Data Processing

This section gives a detailed overview of the data augmentation and data processing pipeline. Data augmentation aims at increasing the variety and the amount of training data. Data processing

¹The transition recognizer is used in Sec. 4.2.2 and the reference finder in Sec. 4.2.1.

²For a detailed discussion about data augmentation we refer to Sec. A.2.

methods are transformation applied on the data, that facilitate the information extraction by the machine learning models.

Data augmentation is a widely used technique in the field of image recognition [47]. Its goal is twofold. Firstly, it is a method to multiply the available training-data which improves the classifiers performance. Secondly, it also enlarges the variety of the data. We would like to emphasize the second aspect. By applying the right transformations on our measured maps it is possible to simulate maps perceived from other samples possessing different intrinsic properties. Hence, data augmentation enables to learn a classifier that is more generally applicable to DQD systems and not overfitted to the specific sample used to acquire the data. The augmentation consists of applying transformations on a map. In principle, these transformations can be anything preserving the key features needed to make the classification. To classify charge transitions or determining whether a DQD is empty, we need to be able to recognize charge transition lines. Affine transformation, adding noise, suppressing single pixes and cropping all preserve the key features of the data - i.e. charge transition lines and are thus candidates for data augmentation transformations.

The DQD system is described using the capacitance model discussed in Sec. 2.1.1. The capacitance matrix C fully determines the system. Thus, in order to simulate other DQD by applying transformations to the charge stability diagram, this matrix C needs to be varied accordingly. The influence of the individual capacitances $C_{i \leftrightarrow j}$ to the charge stability diagram is illustrated in Fig. 2.6. We vary the parameter ΔV_{PG1} , ΔV_{PG2} , δV_{PG1} and δV_{PG2} and thus, implicitly the lever arms α_{ij} , the charging energies E_{C_i} and ultimately also the capacitances $C_{i \leftrightarrow j}$.

The individual augmentation transformations are:

- Rotation of the whole charge stability diagram: corresponds to increasing the lever arm α_{13} and decreasing α_{24} or vice versa (depending on the sign of the rotation angle)
- Rotation of 45 degrees followed by scaling the diagram in x and/or y direction followed by a rotation of -45 degrees: Increases or decreases α_{13} and α_{24} simultaneously. It could as well be seen as a joint increase or decrease of δV_{PG1} , ΔV_{PG1} , δV_{PG2} and ΔV_{PG2} .
- Scaling among PG1 axis: increases or decreases ΔV_{PG1} as well as δV_{PG1}
- Scaling among PG2 axis: increases or decreases ΔV_{PG2} as well as δV_{PG2}
- Flip PG1 and PG2 axis: exchange index 1 with 2.

Flipping the axes simulates measurements for which the plunger gate voltages are swept in reverse order. For both, coarse and fine maps, the final augmentation sequence consists of randomly picking 1 to all transformations with random transformation parameters as e.g. rotation angle or scaling factor. Three examples of augmented fine maps are shown in Fig. 5.1.

After having augmented all maps frames are drawn. In order to distribute the frames equally over the map, but still remain randomness we decided to draw the frames on a grid where the grid points are randomly shifted by up to five pixels in each direction. We chose to take a grid constant of $f = 12\text{px}$, the inner-frame size. Fig. A.1 shows a complete map including the drawn frames without padding.

Each of the frames is processed separately, since at a later point during the runtime of the auto-tuner single frames will be the only information available. In a first processing step, the data is rescaled such that the variance equals to one. Then, an equal superposition between the PG1 and the PG2 derivative³ of the frame is created according to Eq. (3.2). Subsequently, outliers are removed. We define outliers as all points that lie further away from the median than 4.5 standard

³The derivative is calculated using finite differences.

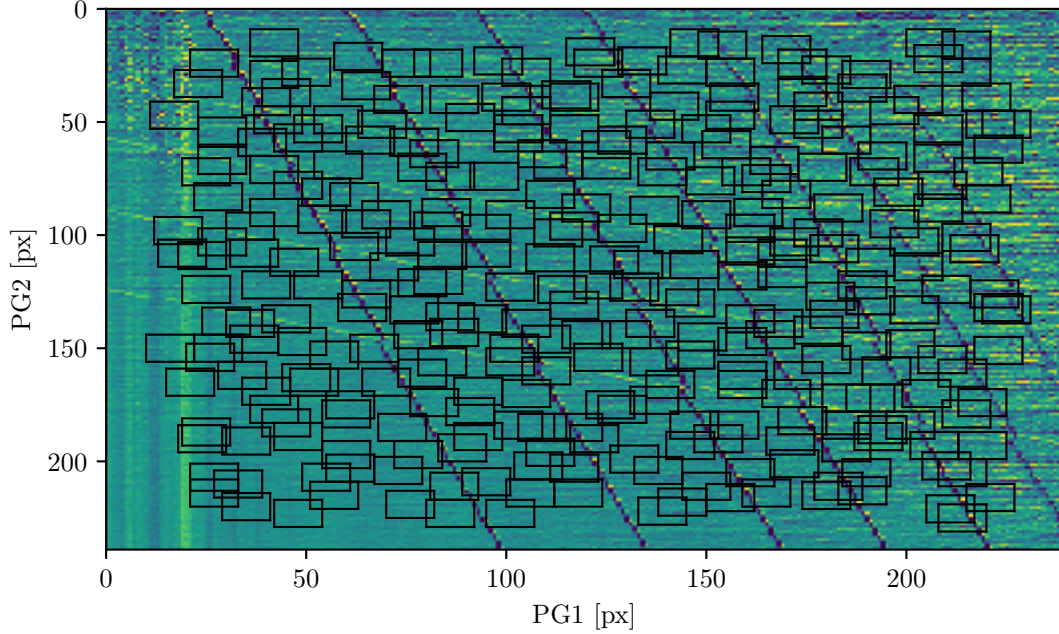


Figure A.1: A complete fine map with drawn frames. The frames are drawn on a grid with grid constant equal to the inner-frame size ($f = 12px$). Subsequently, the frames are randomly moved 0 to 5px in each direction. For better visualization, the paddings of the frames, the labels as well as the charge transition lines are not indicated. In total, on this map 310 frames are drawn.

deviations of 98.4% of the values closest to the mean of the frame. For the frames of the charge transition recognizer an additional rescaling setting the variance to one is performed. Finally, for both cases the median is subtracted.

A.3 Neural Network Architectures

In this section we present both neural network architectures in details. Furthermore, we state all parameters used for the training which was conducted on the supercluster "Leonhard" at ETH Zurich. Finally, the classification results on the evaluation data-sets are discussed.

The architecture of the convolutional neural network used to recognize charge transitions, called charge recognizer, is shown in Fig. 5.3. It consists of three convolutional layers, one fully connected layer and three outputs for each of which a softmax activation is applied. Each output has four neurons and corresponds to one of the three to be classified edges of the input frame. In total the charge recognizer neural network has 248,642 trainable parameters. The neural network used for predicting whether the DQD is occupied by electrons, called reference finder, consists of only two convolutional layers, one fully connected layer and it has one output with two neurons corresponding to the classification outcome "empty" and "occupied". In total the reference finder neural network has 141,164 trainable parameters. The architecture of this neural network can be found in Fig. 5.2. Both neural networks were created using the Python library Keras [44]. A detailed list of all used parameters for architecture and training can be found in Tab. A.1 for the transition recognizer and in Tab. A.2 for the reference identifier.

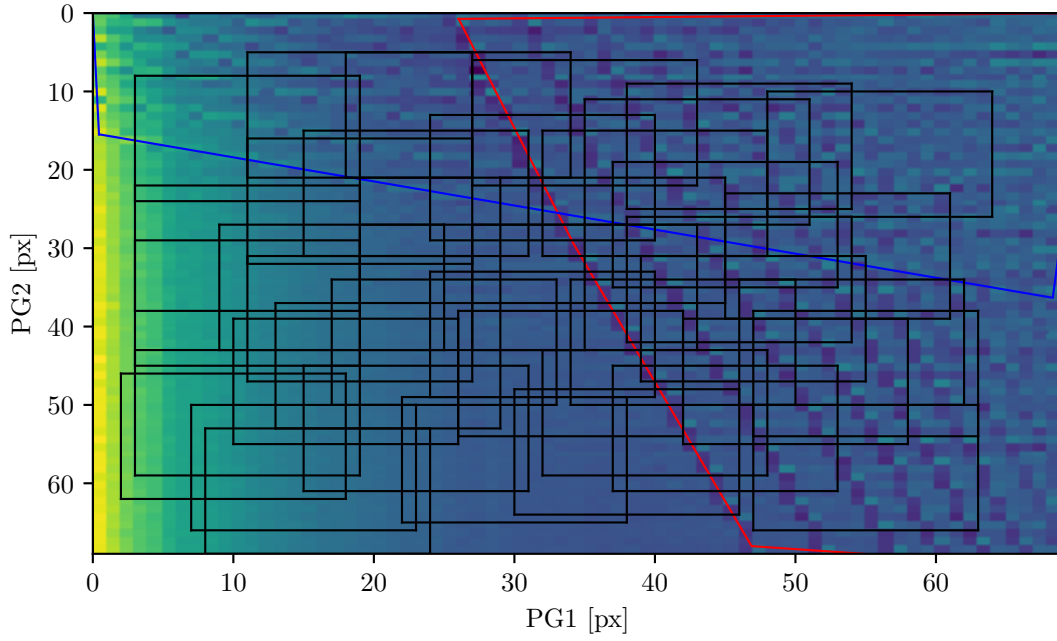


Figure A.2: A complete coarse map including marked charge occupation boundaries and drawn inner-frames. For better visibility, the paddings of the frames are omitted. The inner-frames are drawn on a grid with grid constant equal to the inner-frame size ($f = 16px$). Subsequently, the frames are randomly moved 0 to 3px in both directions. For better visualization, the labels of the frames are not indicated. The top right corner of every inner-frame determines its label depending on which area it belongs to. The areas are bounded by the blue and red lines. Hence, a frame either belongs to the blue region (QD2 occupied), the red (QD1 occupied), both areas (both QD occupied) or none of the indicated regions (both QD empty). In total, there are 35 frames drawn on this map.

Transition Recognizer Parameters	
Parameter	Value
Dropout 1	0.2
Conv. Layer 1	$(6, 6) \times 72$
Activation 1	ReLu
Dropout 2	0.1
Conv. Layer 2	$(3, 3) \times 24$
Activation 2	ReLu
Dropout 3	0.2
Conv. Layer 3	$(2, 3) \times 12$
Activation 3	ReLu
Dropout 4	0
Dense Layer	50
Activation	sigmoid
Outputs	3×4
Activation	softmax
Loss	categorical crossentropy
Optimizer	adam
Batch size	128
Learning rate	0.001
Epochs	8
Class weights	False

Table A.1: In this table all parameters used to create the neural network for the transition recognizer using the python library Keras are stated in the upper half. The parameter in the lower half define the training. For parameter that are not stated here, the default values of the Keras are used.

Reference Identifier Parameters	
Parameter	Value
Dropout 1	0.05
Conv. Layer 1	$(4, 4) \times 48$
Activation 1	ReLu
Dropout 2	0.05
Conv. Layer 2	$(3, 3) \times 12$
Activation 2	ReLu
Dropout 3	0.4
Dense Layer	50
Activation	sigmoid
Outputs	2
Activation	softmax
Loss	categorical crossentropy
Optimizer	adam
Batch size	128
Learning rate	0.001
Epochs	8
Class weights	True

Table A.2: In this table all parameters used to create the neural network for the reference identifier using the python library Keras are stated in the upper half. The parameter in the lower half define the training. For parameter that are not stated here, the default values of the Keras functions are used.

A.4 Classification Results

For assessing the performance of the classifiers we used evaluation data-sets consisting of measured frames only. No data augmentation that may lead to biased evaluation scores were performed. Of course, all processing steps are kept the same for training and evaluation. The evaluation-set for the reference identifier contains 3152 frames drawn from 85 coarse maps and the evaluation-set for the transition recognizer 2054 frames drawn from 10 fine maps.

The confusion matrix for the reference identifier is shown in Fig. 5.4 (a). For the sake of finding any reference point it is important not to classify occupied states as empty. The reverse, classifying empty states as occupied, would not lead to false reference points. In other words, what we want to have a look at is the precision score for the class empty which is estimated as $p = 0.975$. The slight class imbalance lets the model classify uncertain points as occupied rather than empty which is to our favor. Nonetheless, all scores obtained with the evaluation set are 97% and higher and shown in Tab. A.3. What makes us confident about the classifiers performance is that data points for which the classifier is confident are very likely classified correctly. If we for example only include data-points that are classified with 70% and higher, 98% of the data is still classified and the accuracy increases from 98.9% to 99.5%. This behavior indicates that the classifier learned actual features and did not overfit to noise. However, the training and the evaluation data-set are obtained from the same sample within the same cooldown of the fridge. Heating up the sample would have led to changes of the capacitances. Thus, we cannot eliminate the possibility of having overfitted to the particular sample at hand.

The transition recognizer’s confusion matrices are shown in Fig. 5.4 (b) - (d). Since every frame has three classification outcomes indicating the charge transition between lower left and the remaining corners of the frame there are three confusion matrices. Different corners suffer under different class imbalance. For example for the top-left corner the class (1,1) only has 52 data-points. Whereas for the top-right corner there are 405 data-points for the same class. The unequal class balance for the different corners makes it difficult to correct for them, as for example resampling would not solve the issue. However, despite the class imbalance all three corners have overall accuracy of 96% - 97%. Recall, precision, f1 and accuracy are stated in Tab. A.4, Tab. A.5 and Tab. A.6

The question whether more training-data would lead to a better classification result can be answered. We trained both CNNs for a random subsample of the complete data several times, and plotted the obtained evaluation scores and losses⁴ in Fig. A.3. The number of weight updates were kept fix by learning longer for smaller data-sets. In (a) and (b), the scores for the reference finder are shown. They quickly saturate and one can see, that more data would not lead to large improvements. It is expected, that accuracy is highest for the complete data, as optimization was conducted on the complete data set, and thus, potential overfitting might have occurred. In Fig. A.3. In (c) and (d) the accuracy and loss of the transition recognizer is shown. Here, the scores differ significantly for different training runs. However, saturation seems to take place for a small fraction of the data already. We believe, that more data would not lead to significant improvements here as well. However, this only means that more of the same kind of data does not improve the classification results. In particular, it would be favorable to increase the variety of the data by measuring other samples.

⁴We employed a categorical crossentropy loss. For further informatio we refer to [45].

Class	precision	recall	f1	support
occupied	0.996	0.986	0.991	2063
empty	0.975	0.993	0.984	1089
Micro Avg.	0.989	0.989	0.989	3152

Table A.3: Classification scores for the classifier determining whether or not the DQD is empty. The numbers are obtained from the evaluation data set containing 3152 data-points. The micro average corresponds to the overall accuracy, i.e. the fraction of data points that are correctly classified.

Class	precision	recall	f1	support
(0, 0)	0.97	0.97	0.98	1202
(0, 1)	0.96	0.95	0.95	643
(1, 0)	0.97	0.93	0.95	157
(1, 1)	0.83	0.83	0.83	52
Micro Avg.	0.96	0.96	0.96	2054

Table A.4: Classification scores for the top-left corner for the classifier determining whether or not a charge transition occurred. The numbers are obtained from the evaluation data set containing 2054 data-points. The micro average corresponds to the overall accuracy, i.e. the fraction of data points that are correctly classified.

Class	precision	recall	f1	support
(0, 0)	0.97	0.98	0.97	632
(0, 1)	0.95	0.96	0.95	517
(1, 0)	0.96	0.96	0.96	500
(1, 1)	0.95	0.93	0.94	405
Micro Avg.	0.96	0.96	0.96	2054

Table A.5: Classification scores for the top-right corner for the classifier determining whether or not a charge transition occurred. The numbers are obtained from the evaluation data set containing 2054 data-points. The micro average corresponds to the overall accuracy, i.e. the fraction of data points that are correctly classified.

Class	precision	recall	f1	support
(0, 0)	0.98	0.99	0.98	1193
(0, 1)	0.90	0.87	0.89	151
(1, 0)	0.98	0.97	0.97	643
(1, 1)	0.84	0.78	0.81	63
Micro Avg.	0.97	0.97	0.97	2054

Table A.6: Classification scores for the bottom-right corner for the classifier determining whether or not a charge transition occurred. The numbers are obtained from the evaluation data set containing 2054 data-points. The micro average corresponds to the overall accuracy, i.e. the fraction of data points that are correctly classified.

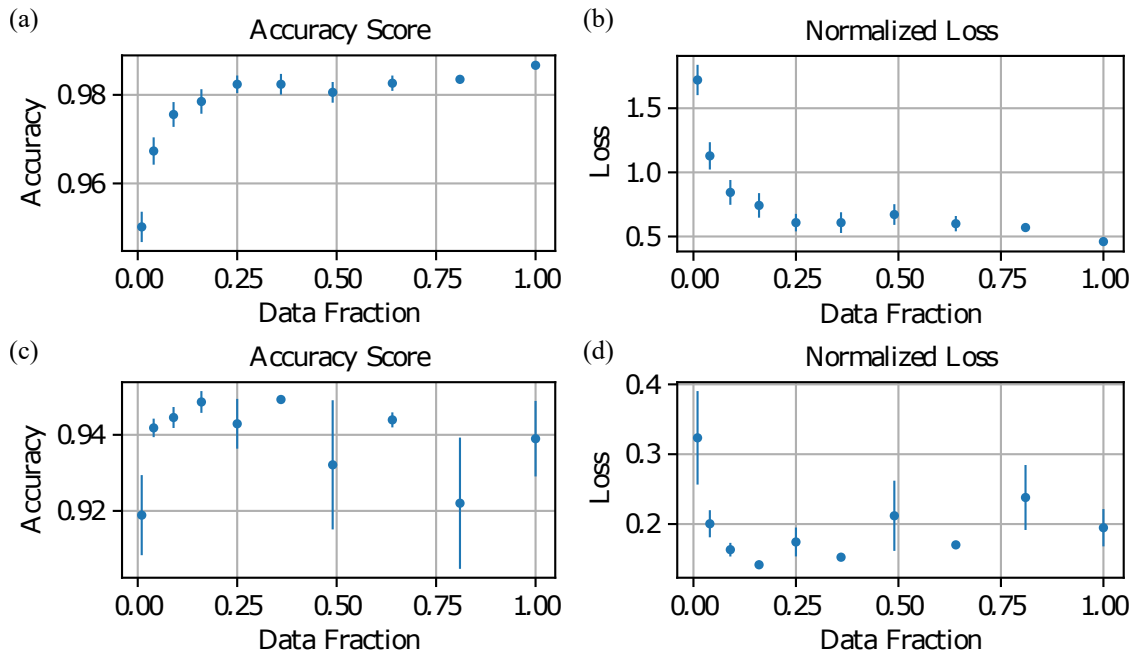


Figure A.3: Accuracy and normalized loss of both CNNs versus the amount of training data used. (a) - (b) Scores of the reference finder. (c) - (d) Scores of the transition recognizer. The loss function is the categorical crossentropy.

Appendix B

Automated QPC Calibration

In order to perform measurements with a QPC, a calibration needs to be performed first. For automated measurements for arbitrary gate voltages the calibration of the QPC needs to be automated as well. The below algorithm reliably finds the sweet spot, or operation point of the QPC. It requires prior knowledge of the pinch off voltage for the QPC current as well as an upper bound for the QPC voltage which is more positive than the operation point.

Algorithm. Automated QPC Calibration

Requirements: V_{start} that defines an upper bound on the QPC gate voltage and $V_{\text{pinch off}}$ a QPC gate voltage for which the QPC is pinched off.

1. Get the mean pinched off QPC current denoted as *mean* and its standard deviation *std* by 51 identical measurements of I_{QPC} at $V_{\text{pinch off}}$
2. Measure $I_{\text{QPC}}(V_{\text{QPCG}})$ from V_{start} to $V_{\text{pinch off}}$ with V_{QPCG} the QPC gate voltage
3. Calculate the derivative of $I_{\text{QPC}}(V_{\text{QPCG}})$
4. Return the first maximum of the above derivative for which $|I_{\text{QPC}}| > |\text{mean}| + 15 \cdot \text{std}$

Bibliography

- [1] T. Frey, P. J. Leek, M. Beck, A. Blais, T. Ihn, K. Ensslin, and A. Wallraff. Dipole coupling of a double quantum dot to a microwave resonator. *Physical Review Letters*, 108(4), jan 2012.
- [2] A. J. Landig, J. V. Koski, P. Scarlino, U. C. Mendes, A. Blais, C. Reichl, W. Wegscheider, A. Wallraff, K. Ensslin, and T. Ihn. Coherent spin–photon coupling using a resonant exchange qubit. *Nature*, 560(7717):179–184, jul 2018.
- [3] V. Srinivasa, J. M. Taylor, and Charles Tahan. Entangling distant resonant exchange qubits via circuit quantum electrodynamics. *Physical Review B*, 94(20), nov 2016.
- [4] L. M. K. Vandersypen, H. Bluhm, J. S. Clarke, A. S. Dzurak, R. Ishihara, A. Morello, D. J. Reilly, L. R. Schreiber, and M. Veldhorst. Interfacing spin qubits in quantum dots and donors—hot, dense, and coherent. *npj Quantum Information*, 3(1), sep 2017.
- [5] L. DiCarlo, M. D. Reed, L. Sun, B. R. Johnson, J. M. Chow, J. M. Gambetta, L. Frunzio, S. M. Girvin, M. H. Devoret, and R. J. Schoelkopf. Preparation and measurement of three-qubit entanglement in a superconducting circuit. *Nature*, 467(7315):574–578, sep 2010.
- [6] C. Monroe and J. Kim. Scaling the ion trap quantum processor. *Science*, 339(6124):1164–1169, mar 2013.
- [7] C A Ryan, M Laforest, and R Laflamme. Randomized benchmarking of single- and multi-qubit control in liquid-state NMR quantum information processing. *New Journal of Physics*, 11(1):013034, jan 2009.
- [8] Christian Reichl. *MBE growth of ultrahigh-mobility 2DEGs in GaAs/AlGaAs*. PhD thesis, ETH Zurich, 2014.
- [9] 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. *Physical Review Applied*, 6(5), nov 2016.
- [10] C. Barthel, J. Medford, C. M. Marcus, M. P. Hanson, and A. C. Gossard. Interlaced dynamical decoupling and coherent operation of a singlet-triplet qubit. *Physical Review Letters*, 105(26), dec 2010.
- [11] J. R. Petta. Coherent manipulation of coupled electron spins in semiconductor quantum dots. *Science*, 309(5744):2180–2184, sep 2005.
- [12] M. D. Shulman, O. E. Dial, S. P. Harvey, H. Bluhm, V. Umansky, and A. Yacoby. Demonstration of entanglement of electrostatically coupled singlet-triplet qubits. *Science*, 336(6078):202–205, apr 2012.
- [13] J. Medford, J. Beil, J. M. Taylor, E. I. Rashba, H. Lu, A. C. Gossard, and C. M. Marcus. Quantum-dot-based resonant exchange qubit. *Physical Review Letters*, 111(5), jul 2013.
- [14] Mark Friesen, Joydip Ghosh, M. A. Eriksson, and S. N. Coppersmith. A decoherence-free subspace in a charge quadrupole qubit. *Nature Communications*, 8(1), jun 2017.

- [15] Gang Cao, Hai-Ou Li, Guo-Dong Yu, Bao-Chuan Wang, Bao-Bao Chen, Xiang-Xiang Song, Ming Xiao, Guang-Can Guo, Hong-Wen Jiang, Xuedong Hu, and Guo-Ping Guo. Tunable hybrid qubit in a GaAs double quantum dot. *Physical Review Letters*, 116(8), feb 2016.
- [16] Zhan Shi, C. B. Simmons, J. R. Prance, John King Gamble, Teck Seng Koh, Yun-Pil Shim, Xuedong Hu, D. E. Savage, M. G. Lagally, M. A. Eriksson, Mark Friesen, and S. N. Copper-smith. Fast hybrid silicon double-quantum-dot qubit. *Physical Review Letters*, 108(14), apr 2012.
- [17] Yuanbo Zhang, Yan-Wen Tan, Horst L. Stormer, and Philip Kim. Experimental observation of the quantum hall effect and berry’s phase in graphene. *Nature*, 438(7065):201–204, nov 2005.
- [18] Daisy Q. Wang, Oleh Klochan, Jo-Tzu Hung, Dimitrie Culcer, Ian Farrer, David A. Ritchie, and Alex R. Hamilton. Anisotropic pauli spin blockade of holes in a GaAs double quantum dot. *Nano Letters*, 16(12):7685–7689, dec 2016.
- [19] Julian D. Teske, Simon Sebastian Humpohl, René Otten, Patrick Bethke, Pascal Cerfontaine, Jonas Dedden, Arne Ludwig, Andreas D. Wieck, and Hendrik Bluhm. A machine learning approach for automated fine-tuning of semiconductor spin qubits. *Applied Physics Letters*, 114(13):133102, apr 2019.
- [20] Tim Botzem, Michael D. Shulman, Sandra Foletti, Shannon P. Harvey, Oliver E. Dial, Patrick Bethke, Pascal Cerfontaine, Robert P. G. McNeil, Diana Mahalu, Vladimir Umansky, Arne Ludwig, Andreas Wieck, Dieter Schuh, Dominique Bougeard, Amir Yacoby, and Hendrik Bluhm. Tuning methods for semiconductor spin–qubits.
- [21] A. R. Mills, M. M. Feldman, C. Monical, P. J. Lewis, K. W. Larson, A. M. Mounce, and J. R. Petta. Computer-automated tuning procedures for semiconductor quantum dot arrays. 2019.
- [22] C. J. van Diepen, P. T. Eendebak, B. T. Buijtendorp, U. Mukhopadhyay, T. Fujita, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen. Automated tuning of inter-dot tunnel coupling in double quantum dots. *Applied Physics Letters*, 113(3):033101, jul 2018.
- [23] T. A. Baart, P. T. Eendebak, C. Reichl, W. Wegscheider, and L. M. K. Vandersypen. Computer-automated tuning of semiconductor double quantum dots into the single-electron regime. *Applied Physics Letters*, 108(21):213104, may 2016.
- [24] Sandesh S. Kalantre, Justyna P. Zwolak, Stephen Ragole, Xingyao Wu, Neil M. Zimmerman, M. D. Stewart, and Jacob M. Taylor. Machine learning techniques for state recognition and auto-tuning in quantum dots.
- [25] Thomas Ihn. *Semiconductor Nanostructures*. Oxford University Press, nov 2009.
- [26] 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.
- [27] Lucjan Jacak, Arkadiusz Wójcik, and Paweł Hawrylak. Creation and structure of quantum dots. In *Quantum Dots*, pages 5–14. Springer Berlin Heidelberg, 1998.
- [28] Zhiming M. Wang. *Self-Assembled Quantum Dots*. Springer New York, 2008.
- [29] John H. Davies. *The Physics of Low-Dimensional Semiconductors*. Cambridge University Press, 1997.
- [30] J.J. Harris. Delta-doping of semiconductors. *Journal of Materials Science: Materials in Electronics*, 4(2), jun 1993.
- [31] Mohamed Heini. *Molecular Beam Epitaxy: From Research to Mass Production*. ELSEVIER SCIENCE & TECHNOLOGY, 2012.

- [32] M. A. Kastner. The single-electron transistor. *Reviews of Modern Physics*, 64(3):849–858, jul 1992.
- [33] C. A. Stafford and S. Das Sarma. Collective coulomb blockade in an array of quantum dots: A mott-hubbard approach. *Physical Review Letters*, 72(22):3590–3593, may 1994.
- [34] D. A. Wharam and T. Heinzel. Coulomb blockade in quantum dots. In *Quantum Dynamics of Submicron Structures*, pages 311–325. Springer Netherlands, 1995.
- [35] S. De Franceschi, S. Sasaki, J. M. Elzerman, W. G. van der Wiel, S. Tarucha, and L. P. Kouwenhoven. Electron cotunneling in a semiconductor quantum dot. *Physical Review Letters*, 86(5):878–881, jan 2001.
- [36] J. Hubbard. Electron correlations in narrow energy bands. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 276(1365):238–257, nov 1963.
- [37] Xin Wang, Shuo Yang, and S. Das Sarma. Quantum theory of the charge stability diagram of semiconductor double quantum dot systems.
- [38] Shuo Yang, Xin Wang, and S. Das Sarma. Generic hubbard model description of semiconductor quantum-dot spin qubits. *Physical Review B*, 83(16), apr 2011.
- [39] Beat A. Braem. *Imaging Charge Transport through Nanostructures on High-Mobility Electron Gases*. PhD thesis, ETH Zurich, Zurich, 2018.
- [40] Francesco Giazotto, Tero T. Heikkilä, Arttu Luukanen, Alexander M. Savin, and Jukka P. Pekola. Opportunities for mesoscopes in thermometry and refrigeration: Physics and applications. *Reviews of Modern Physics*, 78(1):217–274, mar 2006.
- [41] Henk van Houten and Carlo Beenakker. Quantum point contacts. *Physics Today*, 49(7):22–27, jul 1996.
- [42] Thomas Ihn, Simon Gustavsson, Urszula Gasser, Bruno Küng, Thomas Müller, Roland Schleser, Martin Sigrist, Ivan Shorubalko, Renaud Leturcq, and Klaus Ensslin. Quantum dots investigated with charge detection techniques. *Solid State Communications*, 149(35-36):1419–1426, sep 2009.
- [43] Tensorflow. <https://github.com/tensorflow/tensorflow>, 2019.
- [44] Keras. <https://github.com/keras-team/keras>, 2019.
- [45] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York Inc., 2006.
- [46] M. Egmont-Petersen, D. de Ridder, and H. Handels. Image processing with neural networks—a review. *Pattern Recognition*, 35(10):2279–2301, oct 2002.
- [47] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. 2017.
- [48] Dan Claudiu Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.
- [49] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [50] Aleju. imgaug. GitHub, August 2018.

- [51] Jean-Michel Marin, Kerrie Mengersen, and Christian P. Robert. Bayesian modelling and inference on mixtures of distributions. In *Handbook of Statistics*, pages 459–507. Elsevier, 2005.
- [52] B. S. Everitt and D. J. Hand. Mixtures of normal distributions. In *Finite Mixture Distributions*, pages 25–57. Springer Netherlands, 1981.
- [53] G. J. McLachlan, S. K. Ng, and D. Peel. On clustering by mixture models. In *Studies in Classification, Data Analysis, and Knowledge Organization*, pages 141–148. Springer Berlin Heidelberg, 2003.
- [54] T.K. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996.