



**HAL**  
open science

# Quantum transport in two-dimensional systems : artificial intelligence applied to material science

Gaëtan Percebois

► **To cite this version:**

Gaëtan Percebois. Quantum transport in two-dimensional systems : artificial intelligence applied to material science. Condensed Matter [cond-mat]. Université de Strasbourg, 2023. English. NNT : 2023STRAE033 . tel-04587155

**HAL Id: tel-04587155**

**<https://theses.hal.science/tel-04587155v1>**

Submitted on 24 May 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

*ÉCOLE DOCTORALE DE PHYSIQUE ET CHIMIE*

INSTITUT DE PHYSIQUE ET CHIMIE DES MATÉRIAUX DE STRASBOURG

**THÈSE** présentée par :

**Gaëtan PERCEBOIS**

soutenue le : 29 septembre 2023

pour obtenir le grade de : **Docteur de l'Université de Strasbourg**

Discipline/ Spécialité : Physique de la matière condensée

**Quantum transport in two-dimensional systems : Artificial intelligence applied to material science**

**THÈSE dirigée par :**

**Dr. Dietmar WEINMANN**

Université de Strasbourg, IPCMS

**RAPPORTEURS :**

**Prof. Natalia ARES**

University of Oxford

**Prof. Florian MARQUARDT**

University of Erlangen-Nuremberg, Max-Planck-Institut

---

**AUTRES MEMBRES DU JURY :**

**Prof. Fabrice THALMANN**

Université de Strasbourg, ICS

**Dr. Riccardo HERTEL**

Université de Strasbourg, IPCMS

**Dr. Hermann SELLIER**

Université Grenoble Alpes, Institut Néel

**Prof. Rodolfo JALABERT**

Université de Strasbourg, IPCMS (membre invité)







## *Acknowledgements*

Je souhaite tout d'abord remercier Natalia Ares, Florian Marquardt, Hermann Sellier, Riccardo Hertel et Fabrice Thalmann pour avoir accepté de faire partie du jury ainsi que pour le temps passé à lire et juger les travaux présents de cette thèse.

Bien sûr, je tiens à remercier chaleureusement mon directeur de thèse Dietmar Weinmann pour les trois années qui viennent de s'écouler. Merci à lui de m'avoir encadré avec bienveillance et permis de travailler dans les meilleures conditions. Je suis très fier du travail que nous avons fait ensemble. Je tiens aussi à souligner ses qualités humaines qui font de lui un superviseur hors pair. Au-delà de la recherche, j'ai passé d'agréables moments en sa compagnie et j'espère que nous aurons l'occasion d'en partager d'autres, notamment à travers des sorties vélos.

Les résultats présentés dans ce manuscrit doivent beaucoup à certains Grenoblois. Merci à Xavier Waintal et Antonio Lacerda-Santos de m'avoir appris à me servir de leur solver Poisson, de me l'avoir mis à disposition et d'avoir assuré le service après-vente via nos réunions zoom. Je tiens aussi à remercier Boris Brun qui a pu m'apporter son savoir sur la partie expérimentale de l'expérience SGM.

Je veux aussi remercier mes collègues de travail et amis, Guillaume Weick, Rodolfo Jalabert, Thomas Allard, Nico Leumer et Mauricio Gomez pour tous les moments passés en leur compagnie et toutes les discussions que nous avons pu avoir. Grâce à eux j'ai passé trois années dans un cadre de travail chaleureux et épanouissant.

Je tiens aussi à remercier tout particulièrement Clémentine, grâce à qui ces trois années furent particulièrement agréables. Merci d'avoir su me changer les idées lorsque j'en avais besoin et de m'avoir permis de travailler dans les meilleures conditions.

Pour finir, je voudrais aussi remercier mes amis et ma famille pour leur soutien durant ces trois années.



# Contents

<b>Acknowledgements</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Artificial intelligence and deep learning</b>	<b>5</b>
2.1 Basics on neural networks . . . . .	7
2.1.1 Neural network as a function . . . . .	7
2.1.2 Activation function . . . . .	9
2.2 Training procedure . . . . .	9
2.2.1 Organizing the data . . . . .	10
2.2.2 Loss function . . . . .	10
2.2.3 Backpropagation . . . . .	11
2.2.4 Gradient descent and optimizers . . . . .	12
2.2.5 Overview over the whole training algorithm . . . . .	13
2.2.6 Finding the best model with $k$ -fold cross-validation and grid search . . . . .	15
2.3 Convolutional neural networks . . . . .	16
2.3.1 Convolution layer . . . . .	17
2.3.2 Pooling layer . . . . .	18
2.3.3 Encoder-decoder . . . . .	18
2.3.4 Training acceleration using a GPU . . . . .	19
2.4 Adversarial training . . . . .	20
<b>3 Quantum transport in two-dimensional electron gases</b>	<b>23</b>
3.1 Landauer - Büttiker formalism . . . . .	23
3.1.1 Two-terminal systems . . . . .	23
3.1.2 Multi-terminal systems . . . . .	27
3.2 Quantum transport simulations . . . . .	28
3.3 Description of the heterostructure model . . . . .	29
3.4 Potential simulation . . . . .	30
3.4.1 Background disorder potential . . . . .	30
3.4.2 Simulation of the top gates . . . . .	34
3.4.3 Simulation of the tip . . . . .	35
3.5 Conductance through a quantum point contact . . . . .	35
3.6 Scanning gate microscopy . . . . .	37
<b>4 Global characterization of the disorder</b>	<b>41</b>
4.1 Model of the sample . . . . .	41
4.2 Presentation of the dataset . . . . .	42
4.3 Architecture of the neural network . . . . .	43



4.4	Prediction of the global parameters . . . . .	45
4.5	Beyond the black box . . . . .	50
4.5.1	Where is the information? . . . . .	50
4.5.2	Integrated gradient method . . . . .	52
<b>5</b>	<b>Full potential landscape from PLDOS</b>	<b>55</b>
5.1	Regression on the Fourier coefficients . . . . .	55
5.1.1	Architecture of the neural network . . . . .	55
5.1.2	Accuracy of the predicted Fourier coefficients . . . . .	57
5.2	Encoder-decoder neural network . . . . .	62
5.2.1	Architecture of the neural network . . . . .	62
5.2.2	Average error in real space . . . . .	63
5.2.3	Variation of the prediction accuracy when varying the disorder characteristics . . . . .	66
5.2.4	Image translation through the network . . . . .	68
5.3	Magnetic field . . . . .	70
5.3.1	Three images input neural network architecture . . . . .	71
5.3.2	Advantage of the multi-input neural network . . . . .	72
5.4	Potential prediction without branch pattern in the PLDOS . . . . .	73
<b>6</b>	<b>Realistic simulation of an experimental sample</b>	<b>77</b>
6.1	Presentation of the sample . . . . .	77
6.1.1	Properties of the experimental sample . . . . .	77
6.1.2	Realistic model . . . . .	79
6.2	Electrostatic simulations . . . . .	81
6.2.1	Presentation of the PESCADO python package . . . . .	81
6.2.2	Electrostatics of the model . . . . .	82
6.2.3	Calibration of the gate voltage . . . . .	86
6.2.4	Approximation on the electrostatic potential for quantum transport simulations . . . . .	87
6.2.5	Advantage of the PESCADO computation . . . . .	90
6.3	Description of the dataset . . . . .	92
<b>7</b>	<b>Disorder potential prediction for an experimental sample</b>	<b>95</b>
7.1	Neural network architecture . . . . .	95
7.2	Performance of the neural network . . . . .	97
7.2.1	Pre-training . . . . .	97
7.2.2	Performance on simulated SGM data . . . . .	97
7.3	Smoothing of simulated SGM response . . . . .	99
7.4	Indirect evaluation of the predicted potential . . . . .	100
7.4.1	Quantitative method . . . . .	100
7.4.2	Qualitative method . . . . .	105
7.5	Prediction from real images . . . . .	106
<b>8</b>	<b>Conclusions</b>	<b>111</b>
8.1	Proof of principle . . . . .	111
8.2	Disorder determination for an existing sample . . . . .	113

8.3	General method for unveiling the disorder potential in a heterostructure	113
8.4	Perspectives	114
<b>A</b>	<b>Examples of potential construction</b>	<b>117</b>
<b>B</b>	<b>Gaussian Filter examples</b>	<b>119</b>
<b>C</b>	<b>Résumé en français</b>	<b>121</b>
C.1	Introduction	121
C.2	Pré-requis	122
C.2.1	Description du système	122
C.2.2	Deep learning	124
C.3	Preuve de principe	126
C.3.1	Modèle	126
C.3.2	Caractérisation du potentiel par deux paramètres	128
C.3.3	Détermination de l'image du potentiel observé par les électrons	129
C.4	Application de la méthode à un vrai échantillon expérimental	130
C.4.1	Présentation de l'échantillon et du modèle	130
C.4.2	Résultats	132
C.5	Conclusion	133
	<b>Bibliography</b>	<b>137</b>



# List of Figures

2.1	Sketch of perceptron and deep dense neural network . . . . .	8
2.2	Backpropagation example . . . . .	12
2.3	Grid search and $k$ -fold cross-validation . . . . .	16
2.4	Example of the convolution operation . . . . .	17
2.5	Example of maxpooling and upsampling operation . . . . .	19
2.6	Sketch of generative adversarial training . . . . .	21
3.1	Two-terminal system . . . . .	24
3.2	Example of KWANT system . . . . .	29
3.3	Sketch of an heterostructure . . . . .	30
3.4	Global parameters for the potential characterization . . . . .	33
3.5	Examples of disorder potential . . . . .	34
3.6	Quantization of the conductance . . . . .	36
3.7	Thermal effect within the saddle point model . . . . .	37
3.8	Sketch and exemple of SGM experiment . . . . .	38
3.9	Example of potential disorder with the associated PLDOS and SGM images . . . . .	39
4.1	Sketch of the model used for the study on simulated samples . . . . .	42
4.2	Example of a dataset sample . . . . .	43
4.3	Results of the grid search for the neural network that determines the global parameters . . . . .	44
4.4	Precision of the neural network for the determination of the global parameters as a function of the dataset size . . . . .	46
4.5	Influence of the number of parameters on the precision of the determination of the global disorder properties . . . . .	47
4.6	Confusion matrix for the two sets of global parameters . . . . .	48
4.7	Relative error of prediction of the two global parameters . . . . .	49
4.8	Impact of the PLDOS features in the prediction . . . . .	51
4.9	Application of the integrated gradient method . . . . .	54
5.1	Results of the grid search for the neural network that determines the Fourier coefficients . . . . .	56
5.2	Precision of the neural network for the determination of the Fourier coefficients as a function of the dataset size . . . . .	57
5.3	Precision of the predicted Fourier coefficients in the reciprocal space . . . . .	59
5.4	Precision of the potential landscape reconstructed from the Fourier coefficients . . . . .	60
5.5	Impact of the Fourier coefficients on the reconstructed potential . . . . .	61

5.6	Results of the grid search for the neural network that determines the potential landscape . . . . .	63
5.7	Precision of the neural network for the determination of the potential landscape as a function of the dataset size . . . . .	64
5.8	Precision of the predicted potential landscape . . . . .	65
5.9	Average error in the real space . . . . .	66
5.10	Evolution of the precision with the variation of a global parameter . . . . .	67
5.11	Examples of image translation through a neural network . . . . .	69
5.12	Examples of PLDOS computed with different magnetic fields . . . . .	71
5.13	Architecture of the three inputs neural network . . . . .	72
5.14	Influence of the multiple inputs on the precision . . . . .	73
5.15	Average spatial error depending on the magnetic field present in the input(s) . . . . .	74
5.16	Wire system . . . . .	75
5.17	Comparison of the accuracy between wire and QPC systems . . . . .	75
6.1	Experimental sample . . . . .	78
6.2	Experimental SGM-responses . . . . .	79
6.3	Sketch of the model that represents the experimental sample . . . . .	80
6.4	Charge in cell as a function of the electro-chemical potential . . . . .	82
6.5	Meshing of the sample for the electrostatic computation . . . . .	83
6.6	Evolution of the state of the cell with the number of iterations . . . . .	85
6.7	Gate voltages calibration . . . . .	87
6.8	Comparison of two methods for the tip implementation . . . . .	88
6.9	Example of potential construction . . . . .	89
6.10	Comparison between the potential of PESCADO and the fast-potential implementation used for transfer learning . . . . .	91
6.11	Example of an SGM-potential pair that composes the dataset . . . . .	93
7.1	Architecture of the neural network that predicts the disorder from the SGM . . . . .	96
7.2	Precision of the predicted potential from the SGM-response . . . . .	98
7.3	Evolution of the precision with the width of the Gaussian filter . . . . .	100
7.4	Example of the quantitative indirect evaluation method . . . . .	101
7.5	Correlation between the estimated precision and the real one . . . . .	102
7.6	Correlation between non-measurable Pearson coefficient and indirect estimation of Pearson coefficient . . . . .	103
7.7	Evolution of the parameters for accuracy prediction as a function of the blurring width . . . . .	104
7.8	Qualitative indirect evaluation method . . . . .	105
7.9	Uncorrelation between the similarity of reconstructed and original SGM and expected and predicted potential . . . . .	106
7.10	Predicted potential of the experimental sample . . . . .	107
7.11	Accuracy of the model used for the potential prediction of the experimental sample . . . . .	108
7.12	Reconstructed SGMs from the predicted disorder potentials . . . . .	109

A.1	Examples of potentials used for the SGM simulations . . . . .	118
B.1	Examples of application of Gaussian filters . . . . .	120
C.1	Schéma de l'hétérostructure . . . . .	123
C.2	Quantification de la conductance . . . . .	124
C.3	Schéma de l'expérience de microscopie de grille locale et résultat de cette expérience . . . . .	124
C.4	Schéma du perceptron et d'un réseau neuronal dense profond . . . . .	125
C.5	Exemple d'un échantillon . . . . .	127
C.6	Schéma du modèle utilisé pour l'étude sur des échantillons simulés . . . . .	128
C.7	Erreur relative de prédiction des deux paramètres globaux . . . . .	129
C.8	Précision du paysage de potentiel prédit . . . . .	131
C.9	Réponse SGM expérimental . . . . .	132
C.10	Exemple de la méthode d'évaluation indirecte quantitative . . . . .	134
C.11	Potentiel prédit de l'échantillon expérimental . . . . .	135



# Chapter 1

## Introduction

The advancements in technology throughout the last century have made it possible to create nanometric systems in the late 1980s, enabling the investigation of a new physics regime called mesoscopic [1, 2]. This regime emerges at low temperature on the nanoscale and corresponds to the study of an ensemble of particles that exhibits quantum mechanical effects [3]. Mesoscopic physics is of great importance in addressing present and future challenges such as information transport, information storage or computation power [4, 5].

Due to the nanoscopic size of mesoscopic samples, a new issue arises concerning the microscopic fluctuations of sample properties with identical fabrication processes. These fluctuations, which are due to imperfections in the material, can considerably affect the properties of the samples, like the superconducting gap [6], the quantum transport in systems like graphene [7], disordered metals [8, 9] or semiconductor heterostructures [10].

In parallel to the miniaturization of electronic devices, the computational power, storage, and memory has been increasing exponentially. This advancement allowed the development of a new kind of algorithm based on what is called artificial intelligence (AI), which uses experience to learn how to solve highly complex problems. In the case of supervised learning, the problem corresponds to the prediction of an output based on a given input. The experience corresponds to an ensemble of examples composed of input/output pairs. Then, during the learning procedure, the parameters of the neural network (that can be seen as a complex function) are optimized to fit the known outputs [11].

In recent years, AI has become an important tool for researchers helping them to solve a multitude of problems. Most of them can be solved manually or with the help of a classical algorithm and aim to either accelerate (To save time or money) [12, 13] or automate (task only doable by a human being) [14, 15] tasks that can already be done nowadays. Furthermore, artificial intelligence made it possible to solve problems that could not have been solved otherwise like the inverse problem [16, 17]. An important category of problems that can be solved with the aid of machine learning is image processing. Leveraging machine learning, it is possible to automatically extract information from large image datasets solely utilizing the pixels contained within the images, making this method remarkably versatile. Thus, image processing can be applied from large length scales, like in astrophysics with



images from telescopes [18, 19], as well as small length scales with electronic microscope images [14, 20].

In the following study, we will apply image processing techniques to investigate the quantum transport of a high-mobility electron gas confined in a two-dimensional plane. Such systems are present in GaAs/AlGaAs heterostructures [21] and are broadly used for the research on nanostructures [4, 22, 23]. The high-mobility property is due to the presence of layers that separate the electron gas from the ionized dopants [24]. However, those ionized dopants are at the origin of small-angle scattering of the electrons. The weakness of the disorder effects comes from the relatively long correlation length and the amplitude variation that is weaker than the Fermi energy. The aim of the project is to characterize this disorder using the transport properties. A technique called scanning gate microscopy (SGM) allows us to observe the local transport properties using a scanning tip with negative bias voltage and measuring the conductance of the sample for all the positions of the tip [25, 26]. At the end, experimentalists obtain an image where a pixel corresponds to the conductance of the sample when the tip is located at the position of the pixel. It has been proposed that this image shows an approximation of the electron flow [25]. In the case where the electron flow goes through a constriction, one can observe a branch pattern in the image [27]. It has been demonstrated that this branching pattern is due to the disorder present in the sample. It is therefore theoretically possible to extract disorder characteristics from the SGM images.

The applications of machine learning techniques to the disorder characterization from electronic transport properties has been studied recently. Such methods have been used to adjust device parameters to compensate for uncontrolled disorder effects in the case of a double quantum dot nanostructure [28]. It has also been suggested that properties of the disorder between the fingers of a QPC can be extracted from SGM data using cellular neural networks [29] or a swarming algorithm [30]. A proof of principle consisting to determine the disorder from a Majorana nanowire system has also been performed recently [17].

In order to treat this interdisciplinary problem, we introduce separately concepts of artificial intelligence, emphasising on deep learning concepts in chapter 2, and the basics of quantum transport including a presentation of the scanning gate microscopy in chapter 3. Since we have faced considerable time constraints while simulating SGM images, in the first part of the study we used a less time-consuming signal to compute, which is the partial local density of states (PLDOS). In our first paper, we used this signal to obtain global characterization parameters of the potential as detailed in chapter 4 and to unveil the full potential landscape as presented in chapter 5. After completing the proof of principle concerning the ability of neural networks to give a precise characterization of the potential from the PLDOS, we decided to apply our method on a real sample, where the electronic transport information used to determine the disorder is the SGM. We describe the sample and the model used to create the dataset in the chapter 6, the analysis with artificial intelligence is presented in chapter 7. Finally, in chapter 8, we present

our conclusions and outlooks of the study.

## Related publications

- [G. J. Percebois](#) and [D. Weinmann](#) *Deep neural networks for inverse problems in mesoscopic physics: characterization of the disorder configuration from quantum transport properties*, Phys. Rev. B 104, 075422 (2021)
- [G. J. Percebois](#), [A. Lacerda-Santos](#), [B. Brun](#), [B. Hackens](#), [X. Waintal](#) and [D. Weinmann](#) *Reconstructing the potential configuration in a high-mobility semiconductor heterostructure with scanning gate microscopy*, SciPost Phys. 15, 242 (2023).



## Chapter 2

# Artificial intelligence and deep learning

*"A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $P$ , if its performance at tasks in  $T$ , as measured by  $P$ , improves with experience  $E$ ."*

T. Mitchell (1997) [31]

Thanks to the significant improvement in storage capacity and computational power, it has become possible to develop a new class of powerful and efficient algorithms known as machine learning. These algorithms are capable of solving a diverse range of complex tasks. Over the past few years, machine learning algorithms have revolutionized numerous domains, spanning from industry to scientific research. They find applications in various fields such as self-driving cars, speech recognition, medicine, finance, and language translation.

Recently, machine learning has emerged as a powerful tool in many scientific research areas. For example in biology, machine learning is used to sequence DNA [32] or predicting protein structures [33]. It can also be used as a tool to reduce greenhouse gas emissions by reducing the transport activity or forecasting electricity supply and demand [34].

The efficiency of machine learning algorithms has captured the attention of physicists, who have found numerous applications to physics problems [35–37]. Indeed, this tool allows scientists to tackle problems with new perspectives. The diversity of applications of machine learning resides in the diversity of learning methods that can be used. We can distinguish three important categories of training methods: (i) the supervised learning, (ii) the unsupervised learning and (iii) the reinforcement learning.

The process of **supervised learning** involves training a neural network, which can be viewed as a complex function with an output that depends on the input and a set of parameters known as weights. This is achieved by utilizing labeled examples, where the label represents the desired output. During the training process, the algorithm optimizes the weights of the neural network in order to make the best possible correspondence between the input and the output by minimizing the difference between the prediction and the expected output label. Supervised learning algorithms in machine learning are commonly used for tasks such as classification or regression.

In physics, a well-known classification task involves identifying different phases of matter, like gel and fluid in soft matter [38] or topological phases [39] and magnetic phases [40, 41], in condensed matter. In the example of magnetic phases, it can correspond to the paramagnetic and ferromagnetic phase of the Ising model. Applying machine learning to the Ising model allows to determine the critical temperature, which corresponds to the phase transition. This can be achieved by representing the lattice by a two dimensional array that will be the input of the neural network. Then, systems with very low temperatures are labeled as ferromagnetic, while those with very high temperatures are labeled as paramagnetic. After training the neural network, one can apply it to a range of temperatures that includes the critical temperature and observe in which category the neural network classifies the system.

Another application of classifier is the prediction of the topology of a material from its chemical composition and symmetry properties [42]. This task does not require utilization of machine learning, and could be done with a numerical density functional theory (DFT) calculation. In this specific case, machine learning is just used to get a prediction in a shorter amount of time compared to the time needed for a DFT calculation.

Neural networks performing regression tasks can be used to obtain information on the properties of quantum many-body systems like the potential or hopping amplitude from observables [43].

The **unsupervised learning** consists in creating an algorithm that learns from unlabeled data. One important application of this approach is to identify similarities among examples within the training data. The samples with common underlying properties will be part of the same cluster. The determination of these clusters is based on computing distances between samples, which correspond to their similarity rates.

An example of application of unsupervised learning is classifying states according to their topological properties without knowledge of the underlying topological invariant [44]. This example is applied to the XY model where the topological invariant is the winding number. The algorithm learns the topological phase transition, which cannot be identified by a local order parameter.

Another interesting application of unsupervised learning, is to train a neural network to reproduce the input. In this case, the neural network has a peculiar architecture called *autoencoder*. This architecture has the property to compress the input (i.e. in the middle of the neural network there is less information than at the input) thanks to a bottleneck shape of the neural network architecture. The layer in the middle of the bottleneck corresponds to the latent space and has the property to contain only the relevant features of the input. Such neural networks can be used to denoise data [45] or tuning semiconductor quantum devices using the property of the latent space [46].

Contrary to supervised and unsupervised learning, **reinforcement learning** is a distinct category of machine learning that does not require training data. Instead, in reinforcement learning, an *agent* interacts with an *environment* [47, 48]. At a time  $t$ , the agent has to choose to do one action among

a given list of possible actions. Each action is associated with a *reward*, and through learning, the agent aims to select the action that maximizes its cumulative reward. Thus, the agent is trained by performing actions in the environment. The reward associated for each possible action depends on the *policy* of the agent.

This machine learning approach can be useful in physics. For example, if one wants to stabilize a quantum state in a cavity, it is possible to use such an algorithm that tunes parameters like the drive amplitude or the frequency of the drive depending on the current quantum states [49, 50]. Reinforcement learning can also be applied to semiconductor quantum devices for tuning the gates voltages of samples [51] or finding specific transport features [52].

Recently, the utilization of physics informed neural networks [53] has been increasing. These neural networks have the properties to take into account the physics governing the dataset. This technique can be applied to systems submitted to nonlinear differential equations. They can be especially interesting when the training data set is relatively small. Indeed, the final result of the neural network has to be physically consistent which considerably decreases the number of output possibilities.

In recent years, a new field of research known as quantum machine learning has been emerging. In this domain, neural networks are no longer numerical calculations but applications of unitary operators on quantum states [54]. An example of application is the quantum autoencoder, which can be used to compress quantum information. The compression can be the decline of the number of qubits to encode the information [55] or encoding the information in a qudit of dimension  $d$  to a qudit of dimension  $n < d$  [56]. In the latter example the training is performed using an optical setup. The set of parameters, which were previously represented as matrix elements, now correspond to the angles of rotation of half-wave and quarter-wave plates. This kind of compression can have a significant impact in the field of quantum computation and simulation, where quantum resources are highly valuable.

The following study specifically focuses on supervised deep learning algorithms, which possess two key properties: a supervised learning procedure and the utilization of deep neural networks. This chapter is dedicated to providing an introduction to all the essential concepts required to understand the work performed throughout the thesis. In the first part, we describe the basics of neural networks and the training procedure. The second part is dedicated to exploring neural networks specifically optimized for image processing.

## 2.1 Basics on neural networks

### 2.1.1 Neural network as a function

Deep neural networks are defined as artificial neural networks (ANNs) composed of multiple layers [11, 57, 58]. These layers can be decomposed into three parts, the input and the output layer, where the number of neurons corresponds to the number of input and output values. Between those two

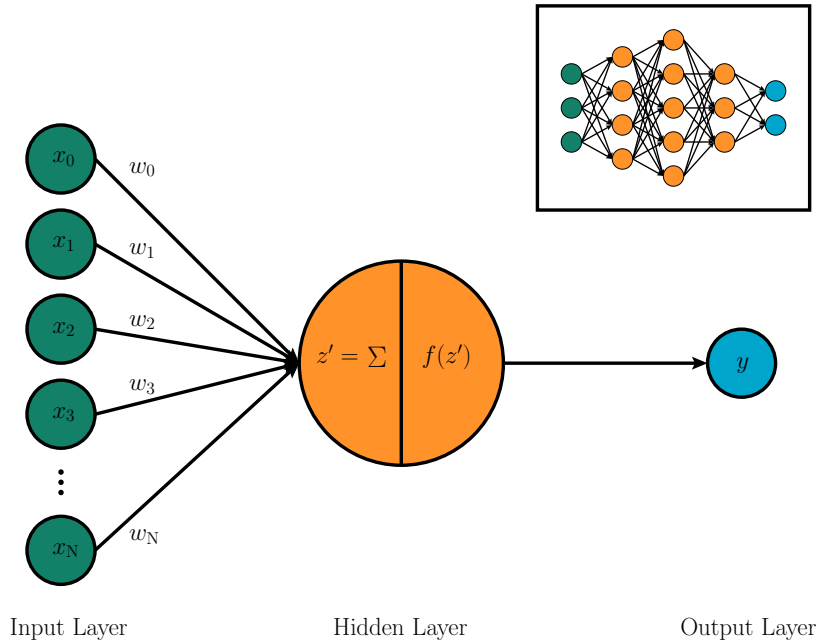


FIGURE 2.1: Representation of a perceptron. The input layer is composed of  $N$  neurons which are connected to the single neuron of the single hidden layer. The two operations performed in the neuron of the hidden layer are decomposed. The first operation corresponds to computing the weighted sum of the values of input neurons according to Eq. 2.1 and the second one is the application of an activation function. The inset represents a dense neural network with several hidden layers.

layers, we have the hidden layers which perform nonlinear transformations of the input information. Those operations are not visible by the user. This model is based on the neural network of the brain in which the signal is transmitted and analyzed through the neuronal connections. In general many concepts used in AI are inspired by brain science [59]. In an ANN, a neuron receives numbers from neurons of the previous layer, performs operations and then transmits the resulting number to the neurons of the next layer. The building block of an ANN, called perceptron, is a neural network with only one hidden layer containing one neuron. In Fig 2.1, the neuron of the hidden layer takes the set of inputs  $\mathbf{x}$  and performs two operations, the first one is

$$z' = \mathbf{w}^T \mathbf{x} + b \quad (2.1)$$

where  $\mathbf{w}$  is the set of weights and  $b$  the bias, both associated to this neuron. The second operation is the application of a so-called activation function  $z = f(z')$  that is present to add non-linearity. For the perceptron  $y = z$ . Examples of activation functions are given in Sec. 2.1.2.

It becomes then possible to extend this concept to a multiple-layer fully connected neural network as depicted in the inset of Fig. 2.1. The value of

the neurons on the  $l$ -th layer is given by the vector  $\mathbf{z}_l = f(\mathbf{z}'_l)$ , where

$$\mathbf{z}'_l = [\mathbf{w}_l^T] \mathbf{z}_{l-1} + \mathbf{b}_l. \quad (2.2)$$

Here  $[\mathbf{w}_l]$  is a matrix of weights where the element  $w_l^{ji}$  corresponds to the weight that connects the  $j$ -th neuron of the  $(l-1)$ -th layer to the  $i$ -th neuron of the  $l$ -th layer.  $\mathbf{b}_l$  corresponds to the vector of biases of the  $l$ -th layer. The activation function applied on  $\mathbf{z}'_l$  is the same for all neurons in the layer (and is often the same through all the computational pipeline except for the output layer).

### 2.1.2 Activation function

The activation functions are present to add non-linearity through the neural network. The latter are non-linear, differentiable and often monotonous. One of the most widely used activation functions is the rectified linear unit (ReLU) function [60, 61]

$$f(z') = \max(0, z') \quad (2.3)$$

that is efficient in numerous cases. The non-saturated property has the advantage to avoid the exploding and vanishing gradient issues and to accelerate the convergence speed. However, in some cases, one may encounter what is known as the "dying ReLU" problem, which refers to situations where the output of the neural network becomes independent of the input because the outputs after the application of the ReLU function is 0 for a majority of the neurons in the network. To avoid this problem, one can use the Leaky ReLU function [62]

$$f(z') = \begin{cases} z', & z' \geq 0 \\ \alpha z', & z' < 0 \end{cases} \quad (2.4)$$

where the value of  $\alpha$  is typically between 0 and 1. Other interesting activation functions are the sigmoid and the hyperbolic tangent

$$f(z') = \frac{1}{1 + e^{-z'}}, \quad f(z') = \tanh(z'), \quad (2.5)$$

that returns a number in the finite interval  $[0, 1]$  and  $[-1, 1]$ , respectively. It can be observed that such activation functions can be employed at the end of a neural network when it is desired to constrain the values of output neurons within specific intervals.

## 2.2 Training procedure

In this section, we restrict the discussion to the supervised learning method. The training process of such a supervised learning algorithm consists in finding the best set of parameters  $\{\mathbf{w}_l, \mathbf{b}_l\}$  that makes the model (i.e. the function provided by the ANN) fit, as much as possible, the desired output for



the data of the training set.

### 2.2.1 Organizing the data

When dealing with a machine learning problem, it is crucial to organize the data by applying two important procedures: data splitting and normalization.

In order to train and evaluate a neural network, one has to split the dataset in three parts. First, approximately 10% to 20% of the data needs to be set aside as the **test set**. The purpose of this test set is to assess the accuracy of the best model once it has been identified. To find this best model, we split again the remaining data in two categories. The **training set** (70% to 80%) is used only to train the model i.e. finding the best set of weights of a neural network with a given architecture. The **validation set** (10% to 20%) corresponds to a set of data that are used to evaluate the model during the training. The latter set is useful to find the best architecture for the neural network, as well as identifying hyperparameters (a specific kind of parameters that remain fixed during training and are defined when creating the neural network). Examples of hyperparameters include the number of layers and the choice of loss function.

The normalization of the data is crucial in machine learning [63]. It allows the neural network to deal with numbers which are all of the same order of magnitude and make the training more efficient. It can take the form of pre-treatment where we normalize the set of data  $\mathbf{X}$  all at once. It is also possible to use specific layers at the beginning of the neural network to normalize the data. The two main kinds of normalization are the min-max scaling

$$\hat{\mathbf{X}} = \frac{\mathbf{X} - \min(\mathbf{X})}{\max(\mathbf{X}) - \min(\mathbf{X})} \quad (2.6)$$

which is useful when the data distribution is unknown but is affected by outliers. The second one is the standardization

$$\hat{\mathbf{X}} = \frac{\mathbf{X} - \text{mean}(\mathbf{X})}{\text{std}(\mathbf{X})} \quad (2.7)$$

where  $\text{mean}(\mathbf{X})$  means the average of  $\mathbf{X}$  and  $\text{std}(\mathbf{X})$  concern the standard deviation of the data  $\mathbf{X}$ . This method is more efficient when dealing with data that have a normal distribution.

### 2.2.2 Loss function

The optimization of the weights and biases corresponds to the minimization of the difference between the expected output  $\mathbf{Y}^e$  and the prediction  $\mathbf{Y}^p$  done by the neural network. This difference is evaluated through the so-called loss function  $L(\mathbf{Y}^e, \mathbf{Y}^p)$  [11]. The loss function plays a crucial role in the training process. It has to be chosen depending on the kind of problem we want to solve (regression or classification). For instance, when performing a classification between two categories, the output can have a value between 0 and 1.

In this case, an example of loss function is the binary cross-entropy

$$L(\mathbf{Y}^e, \mathbf{Y}^p) = -\frac{1}{N} \sum_{i=1}^N [\mathbf{Y}_i^e \log(\mathbf{Y}_i^p) - (1 - \mathbf{Y}_i^e) \log(1 - \mathbf{Y}_i^p)]. \quad (2.8)$$

where  $N$  is the number of computed values. For regression, it can be any number<sup>1</sup> and we can use for example the mean squared error

$$L(\mathbf{Y}^e, \mathbf{Y}^p) = \frac{1}{N} \sum_{i=1}^N (\mathbf{Y}_i^e - \mathbf{Y}_i^p)^2. \quad (2.9)$$

We bring to light the dependency of the prediction  $\mathbf{Y}^p$  on the input  $\mathbf{X}$  and the current set of parameters of the neural network  $\{\mathbf{w}_l\}$ . This allows us to rewrite the loss function as  $L(\mathbf{Y}^e, \mathbf{Y}^p) = L(\mathbf{Y}^e, \mathbf{X}, \{w\})$ . It becomes then clear that the minimization of the loss function corresponds to finding the minimum in the hyperspace of the loss function where each dimension corresponds to one of the parameters (weight or bias). The search of these minimum is done by performing a gradient descent and the gradient of the loss function is determined through the backpropagation algorithm that is detailed in the next section.

### 2.2.3 Backpropagation

The procedure of the backpropagation algorithm [64] is as follows: If one wants to evaluate the gradient of the loss function with respect to the weight  $w_l^{ij}$ , one has first to evaluate the gradient of the output with respect to the neuron of the last hidden layer and then evaluate the gradient of the neuron of the last hidden layer with respect to the neurons of the second last hidden layer and so on until evaluating the gradient of the  $j$ -th neuron of the  $l$ -th layer with respect to  $w_l^{ij}$ . A simple example is depicted in Fig. 2.2, where we want to compute  $\partial L / \partial w_1^{21}$ . The latter value is given by

$$\frac{\partial L}{\partial w_1^{21}} = \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial w_1^{21}} \quad (2.10)$$

$$= \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial h_2^1} \frac{\partial h_2^1}{\partial w_1^{21}} + \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial h_2^2} \frac{\partial h_2^2}{\partial w_1^{21}} \quad (2.11)$$

$$= \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial h_2^1} \frac{\partial h_2^1}{\partial h_1^2} \frac{\partial h_1^2}{\partial w_1^{21}} + \frac{\partial L}{\partial Y} \frac{\partial Y}{\partial h_2^2} \frac{\partial h_2^2}{\partial h_1^2} \frac{\partial h_1^2}{\partial w_1^{21}}. \quad (2.12)$$

In this example, we observed that all the paths from the output to the given neuron have to be considered. In the general case, the dependency of

<sup>1</sup>We note that it is often convenient in machine learning to normalize the data which makes it possible and sometimes more efficient to use a classification loss function like the binary cross-entropy for regression tasks.

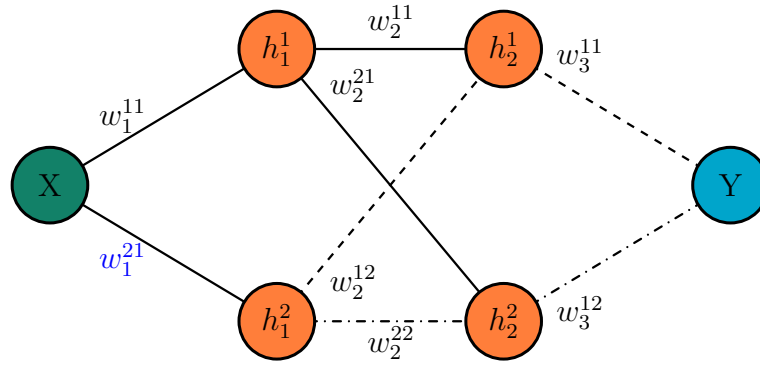


FIGURE 2.2: Schematic neural network. Backwards paths from output to  $w_1^{21}$  correspond to dot and dashdotted paths.

the loss function on a given parameter  $w_l^{ji}$  located on layer  $l$  writes

$$\frac{\partial L}{\partial w_l^{ji}} = \frac{\partial L}{\partial Y} \left[ \sum_{\delta \in \mathcal{P}} \frac{\partial Y}{\partial h_L^{\delta_{L-1}}} \prod_{k=L-1}^{l-1} \frac{\partial h_{k+1}^{\delta_{k+1}}}{\partial h_k^{\delta_k}} \right] \frac{\partial h_l^j}{\partial w_l^{ji}}, \quad (2.13)$$

where  $\mathcal{P}$  represents all the paths that go from the output back to  $h_l^j$ . The ensemble neurons corresponding to the path  $\mathcal{P}$  is  $\{h_l^{\delta_l}\}$ , where  $\delta_l$  is the neuron of layer  $l$  that corresponds to the path.

## 2.2.4 Gradient descent and optimizers

Once the gradient is determined, one has to update the weights in order to reduce the loss function as follows

$$w^{(t)} = w^{(t-1)} - \eta \nabla_{w^{(t)}} L \quad (2.14)$$

where  $t$  corresponds to the iteration variable and  $\eta$  is an hyperparameter called the learning rate<sup>2</sup>. We notice that a large value of the learning rate could lead to a faster convergence, but can also lead to the overshoot of an optimum minimum of the loss function. However, it is generally not advisable to set an excessively small learning rate as it can result in significantly increased training time. Fortunately, there are very efficient algorithms with adaptive learning rate. One of the most common optimizers is *Adam* [65] which "combine the advantages of [...]: *AdaGrad* [66], which works well with sparse

<sup>2</sup>Unlike other hyperparameters, the learning rate can be modified during training if we use a callback mechanism (This notion will be discussed in Sec. 2.2.5) during the training the allows to decrease the learning by a given factor if the precision gets stuck on a plateau. However, in this case, we have new hyperparameters like the initial learning rate or the decreasing factor.

gradients, and RMSProp [67], which works well in on-line and non-stationary settings"<sup>3</sup>. Using Adam [65], the gradient descent (2.14) becomes

$$w^{(t)} = w^{(t-1)} - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \epsilon}} \quad (2.15)$$

where  $\epsilon = 10^{-7}$  (by default on Keras<sup>4</sup> [69]) is an infinitely small number to avoid the division by zero. The terms  $\hat{m}_t$  and  $\hat{v}_t$  correspond to

$$\hat{m}_t = \frac{m_t}{1 - \beta_1} \quad \text{and} \quad \hat{v}_t = \frac{v_t}{1 - \beta_2}. \quad (2.16)$$

where  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$  (by default on Keras) are the first and the second decay rates, respectively. The term

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) (\nabla_{w^{(t)}} L). \quad (2.17)$$

takes into account the average of previous gradient values with an exponential weight. While the term

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_{w^{(t)}} L)^2 \quad (2.18)$$

takes into account the exponential moving average. The terms  $m_t$  and  $v_t$  have been redefined in Eq. 2.16 because of their initialization at 0, those parameters are often biased towards 0 since  $\beta_1$  and  $\beta_2 \approx 1$ .

The iteration stops when the set of weights has converged. It is observed that Adam outperforms the majority of the other optimizers in terms of convergence speed and training cost [65].

### 2.2.5 Overview over the whole training algorithm

In this section, we concatenate all the notions seen previously to have an overview of the easiest general procedure to apply in order to find and evaluate a reasonable model. In a second part, we also detail the training procedure.

At the very beginning, one is highly recommended to first to pre-process the data, which means, first normalize them<sup>5</sup> and then split them in three parts which corresponds to the training, validation and test set. Once the data are prepared, we train several different models and observe their performance on the validation set. The model with the highest performance is then tested on the test set. A more rigorous method concerning the choice of the model is described in the following section. This last step is important to ensure that the selected model has the expected performance on unseen data, and is not just the best one to fit only the data of the validation set. Of

<sup>3</sup>Citation from [65]

<sup>4</sup>Keras is a high-level application programming interface (API) of the machine learning TensorFlow [68] platform

<sup>5</sup>if we do not use a normalization layer after the input layer of the ANN

course, if there is no bias on the splitting part and if we have a sufficiently large validation set, such a problem is unlikely to occur.

Since the gradient descent cannot reach a minimum of the loss function in a single step, even with a large training step, we have to perform several updates applying (2.15) until reaching the convergence of the weights. One training iteration over all the training set is called an epoch. The number of epochs is also an hyperparameter<sup>6</sup>. Another hyperparameter is the mini-batch, it corresponds to the number of samples of the training set that is shown "in the same time" to the neural network. In other words, one iteration of weights update will take into account one mini-batch of samples. We note that this parameter has an important impact on the convergence of the neural network. The whole training procedure is detailed on Alg. 1.

In some cases, it can be interesting to perform actions during the training. Although we cannot perform action manually during the training, it is possible to define a set of automatic actions that have to be executed between the different epochs. Those automatic actions are the callbacks. For example, it is convenient to save the neural network after each epoch<sup>7</sup>. The method allows to keep the neural network that has the best performance on the validation set which can be interesting if the weight values have changed such that the loss function left an interesting local minimum. To avoid the resulting decrease in performance, it is also possible to stop the training when the model is no longer improving. This method can also help to avoid overfitting (recurrent issue in machine learning that is explained in the following paragraph). Another callbacks that we use in the study is the automatic reduction of the learning rate. Indeed, when the model is not improving significantly, we can reduce the learning rate by a chosen factor, which can help to improve the performance of a model.

---

#### Algorithm 1 Neural network training

---

```

Initialize the neural network  $\text{Net} \left( X, \{w^{(t)}\} \right)$ 
for the number of epochs do
  for the number of mini-batches in one epoch do
    Forward pass the mini-batch through the neural network and evaluate
    the loss function  $L$ 
    Perform backpropogation to find  $\nabla_{\{w^{(t)}\}} L$ 
    Use the optimizer to determine  $\{w^{(t+1)}\}$ 
  end for
  Evaluate the neural network on the validation set
  if callbacks then
    Perform callback statements
  end if
end for

```

---

<sup>6</sup>The number of epochs is indeed fixed before the training process. It is nonetheless possible to stop the training before reaching the fixed number of epochs using a callback.

<sup>7</sup>One can incorporate a condition to save the neural network only if it outperforms the previous best model on the validation set thus far.

During the training process, the neural network is evaluated on both the training and validation set. This evaluation corresponds to the difference between the expected and predicted output of the neural network. Quantitatively, it is computed through the metric function(s) that can be the same one as the loss function or other(s). One has to pay attention to the evolution of the performance of the neural network on the training set and the validation set. In the general case, the performance improves for both sets of data until reaching a convergence. However, in some cases the performance on the validation set increases until finding a maximum and then begins to decrease while the performance on the training set is still increasing or converging. This particular case is called overfitting. It means that the neural network overfits the training data but has bad performance on unseen data.

To prevent the overfitting, several methods can be used. One of the most common ones is the dropout layer [70]. This layer disconnects some neurons (each neuron concerned by the dropout layer has a probability  $p$  to be disconnected during one epoch). The choice of the disconnected neurons changes at each epoch. We note that, once the training is done, all the neurons are reconnected.

### 2.2.6 Finding the best model with $k$ -fold cross-validation and grid search

We notice the empirical nature of the procedure to find the best neural network. Indeed, it currently does not exist any precise recipe to tune the neural network to the optimal configuration. However, it exists tedious procedures to find a correct tuning of the neural network. One of them consists in defining a set of values for each of the parameters that we want to determine. Then, all the models that correspond to all the possible combinations are tested to find the one with the highest accuracy. This method requires a considerable amount of computation time. In order to use this method, one must have already an idea of parameters that work well. Otherwise, it is, for example, preferable to use a random search that picks random values in a bounded domain for each parameter. To test the model, we first have to split the entire dataset in two parts, the test set and the training set (see in Fig. 2.3). The test set is kept for the very end of the procedure in order to evaluate the chosen model on unseen data. The rest is used to find the best model and to train the selected one.

The grid search method is often used with a precise evaluation procedure for the model. Indeed, in some cases, the accuracy variation from one model to another can be relatively small and can be biased by the finite size of the set of data used to evaluate the model. As depicted in Fig. 2.3, the  $k$ -fold cross-validation [71] consists to split the training set in  $k$  folds of equal size and use  $k - 1$  fold to train the model while the remaining fold is used for the evaluation. We repeat this operation  $k$  times where each time the evaluation fold is never the same and we average the precision for the given model over

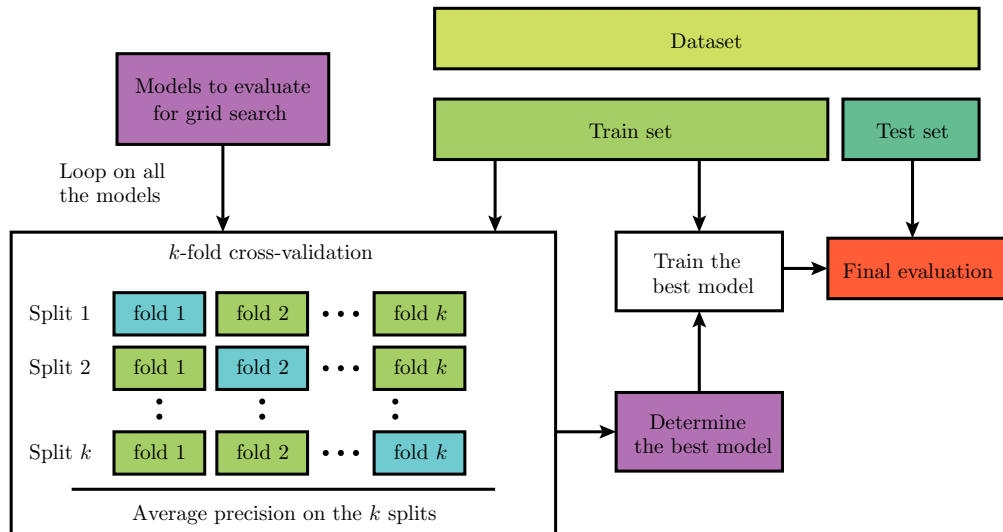


FIGURE 2.3: Schematic representation of the procedure to determine the best model among a list of models using the  $k$ -fold validation technique. In the  $k$ -fold cross-validation section, the green folds correspond to the training data while the blue folds are the validation data.

the  $k$  trainings<sup>8</sup>. At the end, we use the whole training set to train the best model and evaluate it on the test set.

## 2.3 Convolutional neural networks

In the previous section, we have presented the general learning procedure of neural networks. Now, we focus on neural network architectures that are more specific for image processing. This machine learning category has literally been the object of a famous challenge called *ImageNet* [72] which is the name of the database used for the challenge. One of most significant events of this challenge happened in 2012 with the utilization of convolutional neural networks (CNN) that allowed to shatter the previous performance of other kinds of algorithms [73]. The main feature of CNNs is to process the image while keeping the spatial coherence of it [11, 74, 75].

CNNs are generally composed of two parts: the convolution part at the beginning of the network which is followed by a part composed of dense layers that allows to perform a regression or a classification. The two parts are merged by flattening the last convolution layer which becomes a dense layer. In the following study, we also use an *encoder-decoder* network that is designed to take an image as input and produce another image as output. We note that this category of neural networks does not have dense layers.

<sup>8</sup>We note that we can use more than one fold for this procedure, but the set of evaluation folds has never to be the same

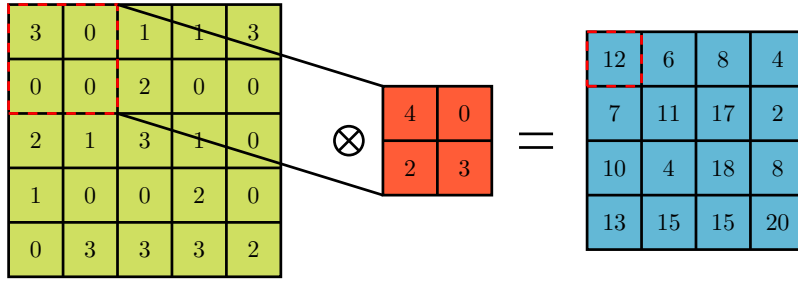


FIGURE 2.4: Example of a convolution operation. The green matrix represents the initial image, the red one corresponds to the filter and the blue one is the result of the convolution.

This section is dedicated to the presentation of those neural networks by mainly explaining the working procedure of convolutional and pooling layers. We also briefly discuss the architecture of an encoder-decoder and the role of the graphical processing unit (GPU) in the training process.

### 2.3.1 Convolution layer

The convolution layer is a function that returns *feature* maps. Each of these feature maps corresponds to the input images of the convolution layer, on which we apply filters (also called *kernels*). As in the dense layer, we also add a bias to the result and apply an activation function  $f$ . The convolution operation that corresponds to one filter  $w$  applied to one image  $x$  without activation function nor bias is depicted in Fig. 2.4 which shows that for the pixel  $(i, j)$ , the output  $y$  is given by

$$y_{ij} = \sum_{\alpha}^{k_1} \sum_{\beta}^{k_2} w_{\alpha\beta} x_{\alpha+i-1, \beta+j-1} \quad (2.19)$$

where  $k_1$  and  $k_2$  correspond to the size of the kernel, in the example  $k_1 = k_2 = 3$ . In practice, many filters are used in parallel such that the input and the output of a convolution layer are in general not single images but a set of images that all are feature maps. In that case, the formula that returns the  $n$ -th feature map of the  $l$ -th layer is

$$y_{n,ij}^{(l)} = f \left( \sum_s^{N_f^{(l-1)}} \sum_{\alpha}^{k_1^{(l)}} \sum_{\beta}^{k_2^{(l)}} w_{ns,\alpha\beta}^{(l)} y_{s,\alpha+i-1, \beta+j-1}^{(l-1)} + b_n \right) \quad (2.20)$$

where  $N_f^{(l-1)}$  is the number of feature map in the previous convolution layer and  $b_n$  is the bias corresponding to the  $n$ -th feature map.

For CNNs, the weights are the values of the filter and the bias. The training process for CNN is similar to the one of a classical neural network presented above.



The main difference between a dense layer and a convolution layer is the number of connected neurons. Indeed, in CNN all the pixels of the feature maps between one layer and the next one are not connected to each other which reduces the number of parameters and makes the training faster. This architecture of neural networks allows also to keep processing the information in two-dimensions which keeps a spatial information. This is not possible with a dense neural network.

When using convolution layers, one has to fix new hyperparameters that do not exist for dense layers:

- **Kernel size:** corresponds to the size (in pixel) of the filter.
- **Number of filters:** corresponds to the number of feature maps in output of the concerned convolution layer.
- **Padding:** number of pixel line added (with the value 0) at the edge of the image on which the filter is applied. It can be used to keep the same image dimension after the convolution layer.
- **Stride:** corresponds to the number of pixels that the filter moves between two consecutive convolution operations. (Impact the dimension of the output feature maps).

The aim of the succession of convolution layer is to detect the main features of an image which are necessary for the classification or the regression. However, in order to keep only the relevant information, one has to reduce the dimension of the images through the convolution part. This operation is performed using a pooling (or downsampling) layer, explained in the next section.

### 2.3.2 Pooling layer

The pooling layer plays a dual role in the network: it reduces the dimension of an image and contributes to the non-linearity, alongside the activation function. Additionally, by reducing the amount of information, it helps mitigate overfitting.

To decrease the image size, the pooling layer partitions the image in equal size rectangles (that are often squares) which correspond to an ensemble of pixels. Then, the pooling layer applies a function on each of those rectangles in order to obtain a single number. The function can be a simple average of all the numbers in the rectangle for instance. The most famous function is the maxpooling which corresponds to the  $\max()$  function. An example of the application of maxpooling corresponds to the first operation depicted in Fig. 2.5.

### 2.3.3 Encoder-decoder

In the majority of applications discussed in this thesis, we need to use a CNN that predicts an image from another image. The encoder-decoder architecture serves as a fundamental framework for addressing such problems. The

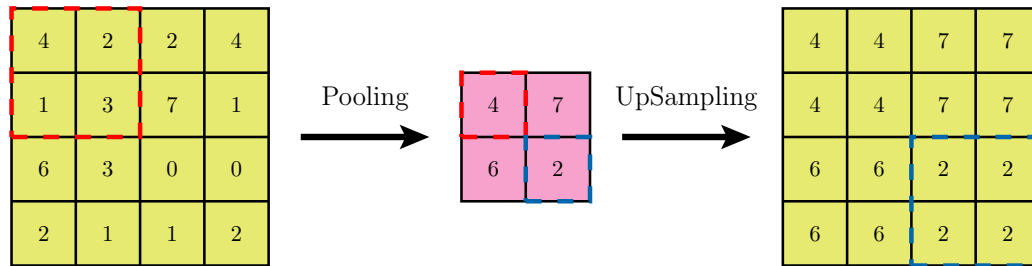


FIGURE 2.5: Example of a pooling and Upsampling operation. The left green matrix represents the initial image, the pink one corresponds to the results of the pooling operation on the left matrix. The ensemble of pixels in the red dashed square of the left matrix felt the  $\max()$  function to become the pixel in the pink matrix highlighted by the red dashed square. The right green matrix corresponds to the result of an upsampling operation applied to the pink center matrix. The pixel highlighted by the blue square in the pink matrix becomes the ensemble of pixels of the right matrix in the blue dashed square.

first part of the neural network is similar to the CNN described before with a succession of convolution and pooling layers, which decrease the size of the image but increase the number of feature maps. The difference appears in the second part, where we do no longer have dense layers but an architecture symmetric to the first part. Indeed, the second part is composed of a succession of convolution layers and upsampling layers. Contrary to a pooling layer, an upsampling layer increases the dimension of the image by creating a rectangle (that is often a square) composed of several pixels with the same value from the original image. An example of an application of upsampling layer is depicted by the second operation present in Fig. 2.5.

It is worth noting that such image translation using a neural network is a common task in AI. However, in the large majority of cases, the input and output images exhibit shared local features. For example, extracting a road map from a satellite image, or segmentation tasks. In these cases, it is frequent to use a neural network with a peculiar architecture called a U-net [76] introduced in 2015 for biomedical image processing. The specificity of this architecture is an additional connection between the convolution layers of the encoder part to the convolution layers (with the same image dimension) of the decoder part. We will not use U-net architecture because of the absence of shared local features between the input and output image in our case.

### 2.3.4 Training acceleration using a GPU

The training process of a neural network is generally time-consuming. However, one can accelerate the training using a GPU [77]. Indeed, GPUs have two interesting features when performing data science. The first one is the parallel architecture with an important number of threads. This property allows the GPU to be efficient for matrix and convolution operations. The

second interesting property is the consequent bandwidth of the GPU that allows to handle a large amount of data [78]. In the framework of this thesis, we used an Nvidia RTX 5000 GPU for the training process.

## 2.4 Adversarial training

When performing image translation, one can also use Generative Adversarial Neural networks (GANs) [79, 80]. This training procedure involves incorporating an additional loss function. This second loss function has the specificity to be a neural network (discriminator) itself that is continuously trained and improved during the training procedure. However, in our case, we do not use such an architecture for the generation of original data but just to perform image translation. The latter image translation task is performed by the neural network called generator. If we consider only the thick black lines of Fig. 2.6, we have the classical training procedure. Considering now all the solid lines, we see in addition the training of the generator by the discriminator. The training of the discriminator is depicted by the dashed lines. The discriminator is trained to give 1 when the two input images are the same (switch on red dashed line) and 0 when they are different (switch on green dashed line), even for tiny differences. Therefore, when training the generator, the value given by the discriminator when considering the expected and predicted image corresponds to a loss value. Due to the training of the discriminator, we expect to have a loss value that remains bad, even when the generator quality prediction is increasing, in order to obtain the best performance possible.

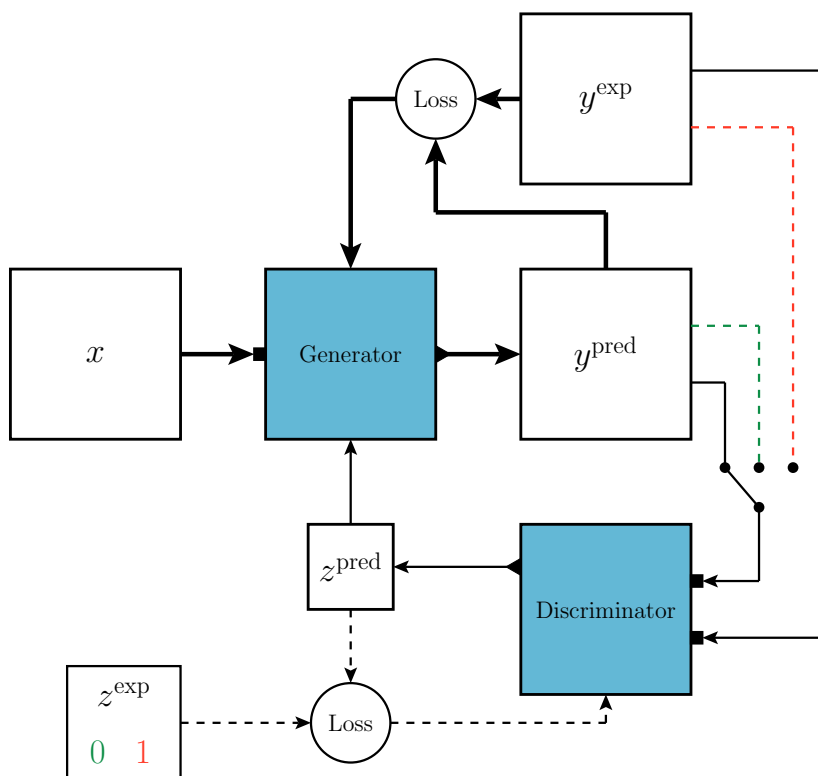


FIGURE 2.6: Sketch of the training procedure of a generative adversarial network. The two neural networks (generator and discriminator) are depicted by blue squared. The input(s) (output) of the neural networks is depicted by a small rectangle (triangle). Thick black dashed lines represent the training procedure of a classical neural network. All the solid black lines represent the training procedures of the generator. The dashed lines correspond to the training of the discriminator. A switch is depicted by the black dots. When the latter is connected to the green (red) dashed line, the expected value  $z^{\text{exp}}$  is 0 (1).



## Chapter 3

# Quantum transport in two-dimensional electron gases

The progress made in the domain of miniaturization of electronic devices at the end of the last century enabled the creation of systems that are sufficiently small to exhibit quantum effects. In particular, for two-dimensional electron gases (2DEGs), nanostructured samples based on high-mobility modulation-doped heterostructures are a staple of condensed matter experiments [81, 82]. Although those heterostructures exhibit a high mobility and a large transport mean free path up to tens of micrometers, the disorder created by the ionized dopants still affects the quantum transport properties [83]. In particular, the branching pattern in the electron flux [26], observed in SGM experiments, is a consequence of the weak disorder seen by the electrons.

In this chapter we describe the state of the art of electronic quantum transport [3, 84, 85] and explain the basics of the quantum transport simulations we use in this thesis. Then, we describe the model of the heterostructure and a model of the disorder present in it. Finally, we present the quantum point contact (QPC) nanostructure and the scanning gate microscopy experiment which produces the transport data that we will use in the following chapters to infer the disorder potential landscape.

### 3.1 Landauer - Büttiker formalism

#### 3.1.1 Two-terminal systems

One can consider a system composed of a scattering region ( $S$ ) connected to two reservoirs  $R_1$  and  $R_2$  with chemical potential  $\mu_1$  and  $\mu_2$ , respectively. The connection is made by leads  $L_1$  and  $L_2$  with a perfect contact (see Fig. 3.1). We suppose that we apply a voltage  $V$  between the two reservoirs, such that  $\mu_1 - \mu_2 = eV$ .

In this approach we consider asymptotically-free quantum states, which are present in the leads and the reservoirs. Thus, in the reservoir we consider a free electron gas with a dispersion relation

$$\epsilon = \frac{\hbar^2 k^2}{2m_e} \quad (3.1)$$

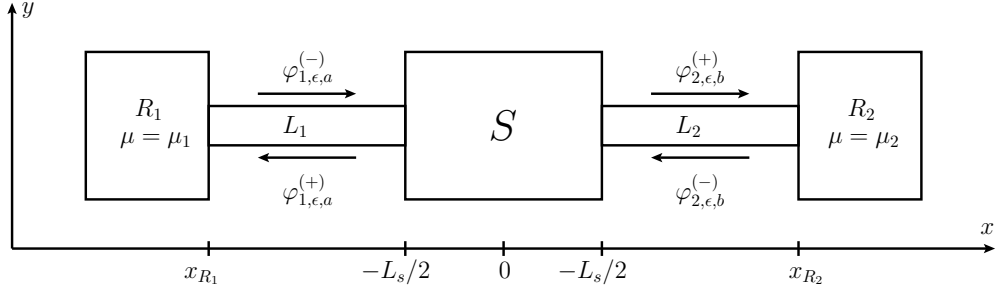


FIGURE 3.1: Two-terminal system composed of a scatterer  $S$  connected to two reservoirs  $R_1$  and  $R_2$  through leads  $L_1$  and  $L_2$ , respectively. The outgoing (incoming) states are labeled by an upper index (+) or (-).

where  $k$  is the magnitude of the wave vector,  $\hbar$  is the reduced Planck constant and  $m_e$  is the electron mass. The leads are also considered disorder-free and quasi one-dimensional. In case of hard-wall boundary conditions in the transverse direction at  $y = 0$  and  $y = W$ , the dispersion relation of the  $a^{\text{th}}$  channel in a lead  $\epsilon = \epsilon_a + \epsilon_{k_a}$  is the sum of the transverse  $\epsilon_a$  and longitudinal  $\epsilon_{k_a}$  energy

$$\epsilon_a = \frac{\hbar^2}{2m_e} \left( \frac{\pi a}{W} \right)^2 \quad \epsilon_{k_a} = \frac{\hbar^2 k_a^2}{2m_e} \quad (3.2)$$

where  $k_a$  is the longitudinal ( $x$  direction) wave-vector in the lead which satisfies  $k^2 = (\pi a/W)^2 + k_a^2$ . The corresponding transversal ( $y$  direction) wave function is given by

$$\phi_a(y) = \sqrt{\frac{2}{W}} \sin \left( \frac{\pi a}{W} (y - W) \right), \quad (3.3)$$

while the longitudinal wave functions are defined by

$$\psi_{k_a}(x) = A_a^{(+)} e^{ik_a x} + A_a^{(-)} e^{-ik_a x}, \quad x \in L_1 \quad (3.4)$$

$$\psi_{k_a}(x) = B_a^{(-)} e^{ik_a x} + B_a^{(+)} e^{-ik_a x}, \quad x \in L_2 \quad (3.5)$$

where the coefficients  $A_a^{(+,-)}$  and  $B_a^{(+,-)}$  correspond to the incoming (+) and outgoing (-) waves of channel  $a$  and are related through the scattering matrix  $S$  such that

$$\begin{pmatrix} A_a^{(-)} \\ B_a^{(-)} \end{pmatrix} = S \begin{pmatrix} A_a^{(+)} \\ B_a^{(+)} \end{pmatrix}. \quad (3.6)$$

Then, a complete basis of incoming (-) and outgoing (+) wave functions in leads 1 and 2 with an energy  $\epsilon$  is given by

$$\varphi_{1,\epsilon,a}^{(\mp)}(\mathbf{r}) = \frac{1}{\sqrt{2\pi\hbar v_a}} e^{\pm ik_a x} \phi_a(y), \quad x \in L_1 \quad (3.7)$$

$$\varphi_{2,\epsilon,a}^{(\mp)}(\mathbf{r}) = \frac{1}{\sqrt{2\pi\hbar v_a}} e^{\mp ik_a x} \phi_a(y), \quad x \in L_2. \quad (3.8)$$

with the prefactor  $1/\sqrt{2\pi\hbar v_a}$  that corresponds to the unit-flux normalization.  $v_a$  corresponds to the longitudinal velocity of mode  $a$ . In the presence of a scatterer (of length  $L_s$  at  $x = 0$ ), an incident flux from one reservoir only in a fixed channel leads to the scattering states

$$\Psi_{1,\epsilon,a}^{(+)}(\mathbf{r}) = \begin{cases} \varphi_{1,\epsilon,a}^{(-)}(\mathbf{r}) + \sum_{b=1}^N r_{ba}(\epsilon_{k_a}) \varphi_{1,\epsilon,b}^{(+)}(\mathbf{r}), & x \ll -L_s/2 \\ \sum_{b=1}^N t_{ba}(\epsilon_{k_a}) \varphi_{2,\epsilon,b}^{(+)}(\mathbf{r}) & x \gg L_s/2 \end{cases} \quad (3.9)$$

$$\Psi_{2,\epsilon,a}^{(+)}(\mathbf{r}) = \begin{cases} \varphi_{2,\epsilon,a}^{(-)}(\mathbf{r}) + \sum_{b=1}^N r'_{ba}(\epsilon_{k_a}) \varphi_{2,\epsilon,b}^{(+)}(\mathbf{r}), & x \gg L_s/2 \\ \sum_{b=1}^N t'_{ba}(\epsilon_{k_a}) \varphi_{1,\epsilon,b}^{(+)}(\mathbf{r}) & x \ll -L_s/2 \end{cases} \quad (3.10)$$

where  $N$  is the number of propagating channels and the  $k_a$  dependent  $N \times N$  matrices  $r$  ( $r'$ ) and  $t$  ( $t'$ ) correspond to the reflection and transmission matrices for particles entering the scatterer from lead  $l = 1$  ( $l = 2$ ) and define the scattering matrix

$$S = \begin{pmatrix} r & t' \\ t & r' \end{pmatrix} \quad (3.11)$$

of dimension  $2N \times 2N$ . The unitarity of the  $S$ -matrix ( $SS^\dagger = \mathbb{1}$ ) implies the current conservation. We define the reflection and transmission coefficient as  $\mathcal{R} = \sum_{a,b} |r_{ab}(\epsilon_{k_a})|^2$  and  $\mathcal{T} = \sum_{a,b} |t_{ab}(\epsilon_{k_a})|^2$ , respectively. They fulfill the relation  $\mathcal{R}(\epsilon_{k_a}) + \mathcal{T}(\epsilon_{k_a}) = N$ . The scattering wave functions satisfy the relation

$$\int d\mathbf{r} \Psi_{l,\epsilon,a}^{(+)*}(\mathbf{r}) \Psi_{l',\bar{\epsilon},\bar{a}}^{(+)}(\mathbf{r}) = \delta_{ll'} \delta(\epsilon - \bar{\epsilon}) \delta_{a\bar{a}} \quad (3.12)$$

and then constitute an orthonormal basis.

Now, we use the current density operator for the state  $l, \epsilon, a$

$$j_{l,a,\epsilon}^x(\mathbf{r}) = \frac{e\hbar}{2im_e} \left[ \Psi_{l,\epsilon,a}^{(+)*}(\mathbf{r}) \frac{\partial}{\partial x} \Psi_{l,\epsilon,a}^{(+)}(\mathbf{r}) - \Psi_{l,\epsilon,a}^{(+)}(\mathbf{r}) \frac{\partial}{\partial x} \Psi_{l,\epsilon,a}^{(+)*}(\mathbf{r}) \right] - \frac{e^2}{m_e} \mathbf{A}(\mathbf{r}) \Psi_{l,\epsilon,a}^{(+)*}(\mathbf{r}) \Psi_{l,\epsilon,a}^{(+)}(\mathbf{r}) \quad (3.13)$$

to determine the electrical current flowing along the  $x$ -axis in this system. In the absence of a magnetic field, we obtain

$$j_{l,a,\epsilon}^x(y) = \frac{e\hbar}{2im_e} \sum_{bb'} \frac{t_{ba}^*(\epsilon_{k_a}) t_{b'a}(\epsilon_{k_a})}{2\pi\hbar\sqrt{v_b v_{b'}}} (ik_b + ik_{b'}) \phi_b^*(y) \phi_{b'}(y) e^{i(k_b - k_{b'})x}. \quad (3.14)$$



Using the orthogonality property of  $\phi_b(y)$  when integrating over the  $y$  section, we obtain the current associated to the scattering state  $(l, a, \epsilon)$

$$I_{l,a,\epsilon} = \frac{e}{h} \sum_{b=1}^N \mathcal{T}_{ba}(\epsilon) \quad (3.15)$$

where  $h = 2\pi\hbar$  is Planck's constant and  $\mathcal{T}_{ba} = |t_{ba}|^2$ . Finally, the electrical current from reservoir  $l$  writes

$$I_l = \int_0^{+\infty} d\epsilon \sum_{a=1}^N 2\pi\hbar v_a \rho_a(\epsilon) I_{l,a,\epsilon} f(\mu_l, T; \epsilon) \quad (3.16)$$

where the factor  $2\pi\hbar v_a$  comes from the change of the normalization to a flux measured in particles arriving to the scatterer per unit time [84].  $f(\mu_l, T; \epsilon) = 1 / [1 + e^{(\epsilon - \mu_l)/k_B T}]$  is the Fermi-Dirac distribution associated to the reservoir  $l$  of chemical potential  $\mu_l$ .  $k_B$  is the Boltzmann constant and  $T$  is the temperature. The one-dimensional density of lead modes including the spin degeneracy factor is  $\rho_a(\epsilon) = 1 / (\pi\hbar v_a)$ . We finally obtain

$$I_l = \eta_l \frac{2e}{h} \int_0^{+\infty} d\epsilon \mathcal{T}(\epsilon) f(\mu_l, T; \epsilon) \quad (3.17)$$

where  $\mathcal{T}(\epsilon) = \sum_{ab} \mathcal{T}_{ba}(\epsilon)$  is the energy-dependent transmission coefficient. The current direction is taken into account through the prefactor  $\eta_l$  which is equal to 1 for  $l = 1$  and -1 for  $l = 2$ . Finally, the conductance of the system is

$$G(V, T) = \frac{I(V)}{V} = \frac{2e^2}{h} \frac{1}{eV} \int_0^{+\infty} d\epsilon \mathcal{T}(\epsilon) (f(\mu_1, T; \epsilon) - f(\mu_2, T; \epsilon)) \quad (3.18)$$

where  $V$  is the bias voltage applied on the system ( $eV = \mu_1 - \mu_2$ ). Considering the linear regime i.e. weak bias voltage  $\mu_1 \approx \mu_2$ , one can perform a Taylor expansion where the first order term of the conductance is

$$G(V \rightarrow 0, T) = \frac{2e^2}{h} \int_0^{+\infty} d\epsilon \mathcal{T}(\epsilon) \left( - \frac{\partial f}{\partial \epsilon}(\mu, T; \epsilon) \Big|_{\mu=\mu_1} \right) \quad (3.19)$$

where we have

$$- \frac{\partial f}{\partial \epsilon}(\mu, T; \epsilon) \Big|_{\mu=\mu_1} = \frac{1}{2k_B T} \frac{1}{1 + \cosh[(\epsilon - \mu_1)/k_B T]}. \quad (3.20)$$

Since quantum transport experiments are frequently performed at low temperature, one can be interested in another Taylor expansion in the temperature, which gives

$$G(V \rightarrow 0, T) = \frac{2e^2}{h} \frac{1}{2} \sum_{k=0}^{\infty} \frac{1}{(k_B T)^{2k}} \frac{\mathcal{T}(\epsilon)^{(2k)}|_{\epsilon=\mu_1}}{(2k)!} \int_{-\infty}^{+\infty} du \frac{u^{2k}}{1 + \cosh u} \quad (3.21)$$

where  $\mathcal{T}^{(n)}$  is the  $n$ -th derivative of the transmission coefficient with respect to the energy at  $\epsilon = \mu_1$ . Thus, we notice that the thermal corrections are non zero only when we have a non-vanishing even derivative of the transmission coefficient with respect to the energy. We note that in the extreme case where  $T = 0$ , the conductance writes

$$G(V \rightarrow 0, T = 0) = \frac{2e^2}{h} \mathcal{T}(\mu_1), \quad (3.22)$$

where  $2e^2/h = G_0$  is the quantum of conductance.

### 3.1.2 Multi-terminal systems

One can generalize the Landauer formula to multi-terminal systems [86]. In this part we note  $\mathcal{T}_{ab}^{\alpha\beta}$  the transmission probability from mode  $b$  in lead  $\beta$  to mode  $a$  in lead  $\alpha$ . Then, at low bias voltage, the current contribution out of the reservoir  $\alpha$

$$I^{\alpha\alpha} = \frac{2e}{h} \int d\epsilon [N^\alpha(\epsilon) - \mathcal{R}^\alpha(\epsilon)] f(\mu_\alpha, T; \epsilon), \quad (3.23)$$

where  $N^\alpha(\epsilon) = \sum_{a \in \alpha} 1$  is the number of open channels in terminal  $\alpha$  at energy  $\epsilon$  and  $\mathcal{R}^\alpha(\epsilon)$  is the reflection factor. The current contribution due to the contact between lead  $\alpha$  and  $\beta$  is

$$I^{\alpha\beta} = \frac{2e}{h} \int d\epsilon \mathcal{T}^{\alpha\beta}(\epsilon) f(\mu_\beta, T; \epsilon), \quad (3.24)$$

where  $\mathcal{T}^{\alpha\beta} = \sum_{ab} \mathcal{T}_{ab}^{\alpha\beta}(\epsilon)$  is the transmission factor. The total current flowing in lead  $\alpha$  writes

$$I_{\text{tot}}^\alpha = \frac{2e}{h} \int d\epsilon \left[ (N^\alpha(\epsilon) - \mathcal{R}^\alpha(\epsilon)) f(\mu_\alpha, T; \epsilon) - \sum_{\beta \neq \alpha} \mathcal{T}_{\alpha\beta}(\epsilon) f(\mu_\beta, T; \epsilon) \right] \quad (3.25)$$

In thermodynamic equilibrium, we have  $f(\mu_\alpha, T; \epsilon) = f(\mu_\beta, T; \epsilon)$ , which implies that  $I_{\text{tot}}^\alpha = 0$  and leads to

$$N^\alpha(\epsilon) = \mathcal{R}^\alpha(\epsilon) + \sum_{\beta \neq \alpha} \mathcal{T}_{\alpha\beta}(\epsilon). \quad (3.26)$$

## 3.2 Quantum transport simulations

Our quantum transport simulations are carried out numerically using a Python package called KWANT [87]. This package is based on a tight-binding model which, in a two-dimensional square lattice with site indices  $(i, j)$  and lattice parameter  $a$ , is described, in absence of a magnetic field, by the Hamiltonian

$$\mathcal{H} = \sum_{i,j} U_{i,j} |i, j\rangle \langle i, j| - t [ |i+1, j\rangle \langle i, j| + |i, j\rangle \langle i+1, j| + |i, j+1\rangle \langle i, j| + |i, j\rangle \langle i, j+1| ]. \quad (3.27)$$

$U_{i,j} = 4t + V(\mathbf{r}_{ij})$  corresponds to the onsite energy of the site  $(i, j)$  which has for coordinate  $(x, y) = (ai + aj)$ . The potential in the onsite term corresponds to the external electrostatic field of the system created by the gates, the impurities, and the Hartree part of the electron-electron interaction. We also define  $t = \hbar^2 / (2m^*a^2)$  the hopping term. In the absence of the potential term  $V$ , the Hamiltonian (3.27) leads to plane-wave solutions  $\psi(x, y) = e^{ik_x x} e^{ik_y y}$  and the dispersion relation

$$E(k_x, k_y) = 2t (2 - \cos(k_x a) - \cos(k_y a)). \quad (3.28)$$

The Fermi energy is then given by

$$E_F = 2t (1 - \cos(k_F a)). \quad (3.29)$$

This energy dispersion approximates the parabolic energy distribution when  $\lambda_F \gg a$ . We consider that the tight-binding approach is a good approximation for a continuous model when  $\lambda_F \geq 6a$ .

KWANT uses the scattering states corresponding to the Hamiltonian and computes the scattering matrix.

One can also add a magnetic field  $\mathbf{B}$  within the Landau gauge, using the Peierls substitution that consists to change the hopping term in the following way [88]

$$t_{ij}(\phi) = t e^{i\phi(x_i - x_j)(y_i + y_j)/2} \quad (3.30)$$

with  $\phi = Be a^2 / \hbar$  the magnetic flux per lattice cell.

In order to perform the quantum transport simulation with KWANT, one has first to define the system (see Fig. 3.2) which is composed of (i) (black dots) the scattering part (size, onsite energy with associated potential  $V(\mathbf{r}_{ij})$ , nearest neighbor interaction) and (ii) (red dots) the leads that connect the system. Then one can compute the scattering matrix for instance in order to obtain the conductance or directly compute the wave functions in order to obtain the PLDOS.

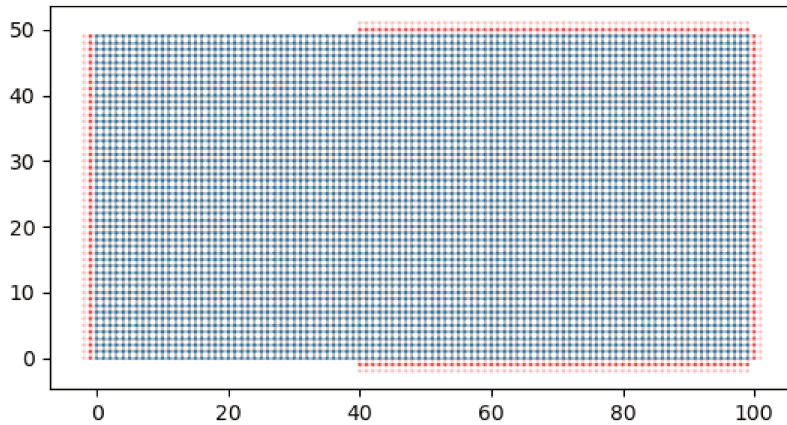


FIGURE 3.2: Lattice corresponding to a KWANT system. The blue dots correspond to lattice sites that are part of the scattering region. The red dots correspond to the leads.

### 3.3 Description of the heterostructure model

Among the several materials that allow to create a 2DEG, we will focus on semiconductor heterostructures made from GaAs/AlGaAs. They correspond to the stacking of layers of different semiconductor materials (see left panel of Fig. 3.3). This stacking is grown using the molecular beam epitaxy method [89, 90], which allows to obtain high quality heterojunctions, thanks to the atomic layer by layer growing. This precise stacking leads to the energy diagram depicted in the right panel of Fig. 3.3, which is at the origin of the confinement of the electrons in a two-dimensional plane. The electrons at the interface compensate the ionized dopants that are all located in a layer whose the distance from the 2DEG is denoted  $s$ . This distance between the 2DEG and the doping layer allows to reach a very high mobility of the electrons in the gas (The transport mean free path is typically of the order of micrometers), on short length scales we can thus consider the electronic transport as ballistic to some extent. We denote the density of ionized dopants by  $N_d$ . We note that only a part of the dopants are ionized and only a small proportion of the electrons are confined in the 2D plane, the other electrons go to the material surface [92].

The Fermi energy of the 2DEG in the material depends on the electron density  $N_s$  :

$$E_F = \frac{\hbar^2 \pi}{m^*} N_s \quad (3.31)$$

where  $m^*$  is the effective mass of the material in which is located the 2DEG. Despite the distance between the 2DEG and the ionized dopants, the latter affect the electrons due to their positive charge. The potential seen by the electron acts as an uncontrolled weak disorder it is impossible to choose or to know the position of these dopants. It affects the wave functions (and

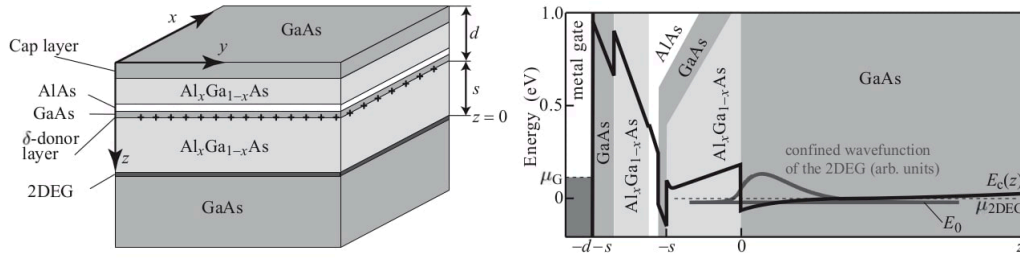


FIGURE 3.3: Left panel: Sketch of the heterostructure with detailed layer stacking. The energy diagram of this heterostructure is depicted in the right panel. In the latter figure, we observe the potential minimum at  $z = 0$  where the 2DEG is present. Figure from [91].

semi-classical trajectories) of the electrons and by consequence, the electronic transport properties of the material [93].

## 3.4 Potential simulation

### 3.4.1 Background disorder potential

As presented in Sec. 3.3, at equilibrium a part of the dopants is ionized and about 10% of the electrons arising from those dopants compose the 2DEG [92]. Hence, the electrons in the 2DEG are subjected to the electrostatic potential due to the positive charges of the ionized dopants. Then, the bare background Coulomb potential at position  $\mathbf{r}$  is

$$V_{\text{dop}}^{(\text{bare})}(\mathbf{r}) = -\frac{e^2}{4\pi\epsilon_r\epsilon_0} \sum_{i=0}^{n_{\text{dop}}} \frac{1}{|\mathbf{r}_i - \mathbf{r}|}, \quad (3.32)$$

where  $\epsilon_r = 12.9$  is the relative permittivity of GaAs,  $\epsilon_0$  is the dielectric constant and  $\mathbf{r}_i$  are the positions of the  $n_{\text{dop}}$  ionized dopants. Those positions are not known and cannot be found experimentally. In a theoretical modeling of the system, realistic configurations can be generated by placing ionized dopants at random positions. With a large number of random disorder configurations, this technique allows us to cover a large space of the possible configurations, including correlated distributions of the dopants as it is probably the case in real samples [83]. We note that this potential does not take into account the electron-electron interactions.

Considering the need to implement the disorder in the following sections, we have to take into account the amount of time required to simulate such a disorder. If one wants to directly use Eq. 3.32, one has to perform the sum over a very large number of ionized dopants ( $\approx 10^4$ ). However, using the Fourier transform, it is possible to sum only on a few hundred terms [91]. In this derivation, we consider only the fluctuating part of the potential. The Fourier transform of this fluctuating part is

$$V_{\text{dop}}^{(\text{bare})}(\mathbf{q}) = -\frac{e^2}{4\pi\epsilon_r\epsilon_0}C(\mathbf{q})\frac{2\pi e^{-qs}}{q} \quad (3.33)$$

in case where all the dopants are all located within a plane parallel to the 2DEG, and located at a distance  $s$  from the 2DEG.  $C(\mathbf{q})$  is a complex Fourier coefficient representing the Fourier transform of the charge distribution. Considering the influence of the Thomas-Fermi screening of the impurities by the electrons, the potential can be expressed as

$$V_{\text{dop}}(\mathbf{q}) = -4\pi E_{\text{Ry}}^* a_{\text{B}}^* C(\mathbf{q}) \frac{e^{-qs}}{q + q_{\text{TF}}}, \quad (3.34)$$

where we have the product of the effective Bohr-radius with the effective Rydberg energy is

$$E_{\text{Ry}}^* a_{\text{B}}^* = \frac{e^2}{8\pi\epsilon_r\epsilon_0}$$

and  $q_{\text{TF}} = 2/a_{\text{B}}^*$  is the Thomas-Fermi factor. Coming back to the real space, the potential writes

$$V_{\text{dop}}(\mathbf{r}) = -\frac{E_{\text{Ry}}^* a_{\text{B}}^*}{\pi} \int d\mathbf{q} \frac{e^{-qs}}{q + q_{\text{TF}}} C(\mathbf{q}) e^{-i\mathbf{q}\mathbf{r}}. \quad (3.35)$$

The random position of the positive charges is taken into account through the Fourier coefficients  $C(\mathbf{q})$ , which satisfy  $C(\mathbf{q}) = C^*(-\mathbf{q})$  in order to ensure that  $V_{\text{dop}}(\mathbf{r})$  is real. The real and the imaginary part are random numbers that follow a normal distribution with variance  $\sigma^2 = n_{\text{dop}}/2$ .

It becomes now clear that the potential is characterized by two parameters: the distance  $s$  that separates the 2DEG from the doping layer and the density of dopants  $N_{\text{d}} = n_{\text{dop}}/LW$  ( $L$  corresponds to the length and  $W$  to the width of the sample). In order to observe the influence of those two terms on the potential characterization, we focus on the auto-correlation value of the potential

$$\mathcal{C}(R) = \langle V(\vec{r})V(\vec{r} + \vec{R}) \rangle_{\vec{r}, \theta_R} = \langle V(\vec{r})V(\vec{r} + \vec{R}) \rangle_{\text{Conf}}. \quad (3.36)$$

If the sample is spatially infinite, the average on the space and  $\theta_R$  corresponds to the average on the configurations. For simplicity  $\langle \dots \rangle_{\text{Conf}} \rightarrow \langle \dots \rangle$ . When using the potential given by Eq. 3.35, we find

$$\mathcal{C}(R) = \frac{(E_{\text{Ry}}^* a_{\text{B}}^*)^2}{\pi^2} \int d^2\vec{q} \int d^2\vec{q}' \frac{e^{-(q+q')s}}{(q + q_{\text{TF}})(q' + q_{\text{TF}})} \langle C(\vec{q})C(\vec{q}') \rangle e^{-i(\vec{q}+\vec{q}')\cdot\vec{r}} e^{-i\vec{q}\cdot\vec{R}}. \quad (3.37)$$

The average product of the Fourier coefficients can be determined by the Fourier transform combined with the property of the randomness of the dopant location  $\langle C(\vec{r})C(\vec{r}') \rangle = N_{\text{d}}\delta(\vec{r} - \vec{r}')$ , which gives

$$\langle C(\vec{q})C(\vec{q}') \rangle = \int d^2\vec{r} \int d^2\vec{r}' \langle C(\vec{r})C(\vec{r}') \rangle e^{-i\vec{q}' \cdot \vec{r}'} e^{-i\vec{q} \cdot \vec{r}} = N_d (2\pi)^2 \delta(\vec{q} + \vec{q}') \quad (3.38)$$

We now plug this in the correlation formula 3.37, which gives

$$\mathcal{C}(R) = 8\pi (E_{\text{Ry}}^* a_{\text{B}}^*)^2 N_d \int_0^{+\infty} dq q \frac{e^{-2qs}}{(q + q_{\text{TF}})^2} J_0(qR), \quad (3.39)$$

where  $J_0(x) = (2\pi)^{-1} \int d\theta e^{ix \cos \theta}$  is the Bessel function of the first kind. From this auto-correlation function one can derive two important parameters: the correlation length  $\xi$  that we define through

$$\mathcal{C}(\xi) = \frac{\mathcal{C}(0)}{2} \quad (3.40)$$

and the root mean square of the potential (RMSP)

$$\Pi = \sqrt{\langle V^2 \rangle} = \sqrt{\mathcal{C}(0)} = E_{\text{Ry}}^* a_{\text{B}}^* \left[ 8\pi N_d \int_0^{q_c} dq q \frac{e^{-2qs}}{(q + q_{\text{TF}})^2} \right]^{1/2}. \quad (3.41)$$

One can obtain the analytic expression for the RMSP

$$\Pi^2 = E_{\text{Ry}}^* a_{\text{B}}^* 16\pi^2 N_d f(2q_{\text{TF}}s) \quad (3.42)$$

with

$$f(x) = e^x (1 + x) \Gamma(0, x) - 1 \quad (3.43)$$

where  $\Gamma(a, x)$  is the incomplete gamma function. In case of  $2q_{\text{TF}}s \gg 1$ , one can make the approximation  $f(x) \approx 1/x^2$  [91].

Since  $N_d$  is a prefactor in the formula (3.39) for  $\mathcal{C}(R)$ ,  $\xi$  depends only on  $s$  as observed in the lower panel of Fig. 3.4, while  $\sqrt{\langle V^2 \rangle}$  depends on both  $s$  and  $N_d$ . The dashed black lines in the upper panel of Fig. 3.4 represent the equi-root mean square of the potential. This means that two different sets of  $(s, N_d)$  can lead to the same value of  $\Pi$ . We note that for the plotted realistic range of  $s$  the correlation length is proportional to the latter.

Considering the simulation part, the potential is computed on a discrete lattice with parameter  $a$  of size  $n_x \times n_y = LW/a^2$ . The  $\mathbf{q}$ -space is discretized with step widths

$$\Delta q_x = \frac{2\pi}{L} \quad \text{and} \quad \Delta q_y = \frac{2\pi}{W}. \quad (3.44)$$

The discrete vectors of the two-dimensional  $\mathbf{q}$ -space are denoted  $\mathbf{q}_j$ . The integer  $j$  labels the vector such that  $-\mathbf{q}_j = \mathbf{q}_{-j}$  and  $\mathbf{q}_0 = \vec{0}$ . Then the potential on a discrete lattice writes

$$V_{\text{dop}}(\mathbf{r}) = -E_{\text{Ry}}^* a_{\text{B}}^* \frac{\Delta q_x \Delta q_y}{\pi} \sum_{j>0} \frac{e^{-q_j s}}{q + q_{\text{TF}}} C(\mathbf{q}_j) e^{-i\mathbf{q}_j \cdot \mathbf{r}}. \quad (3.45)$$

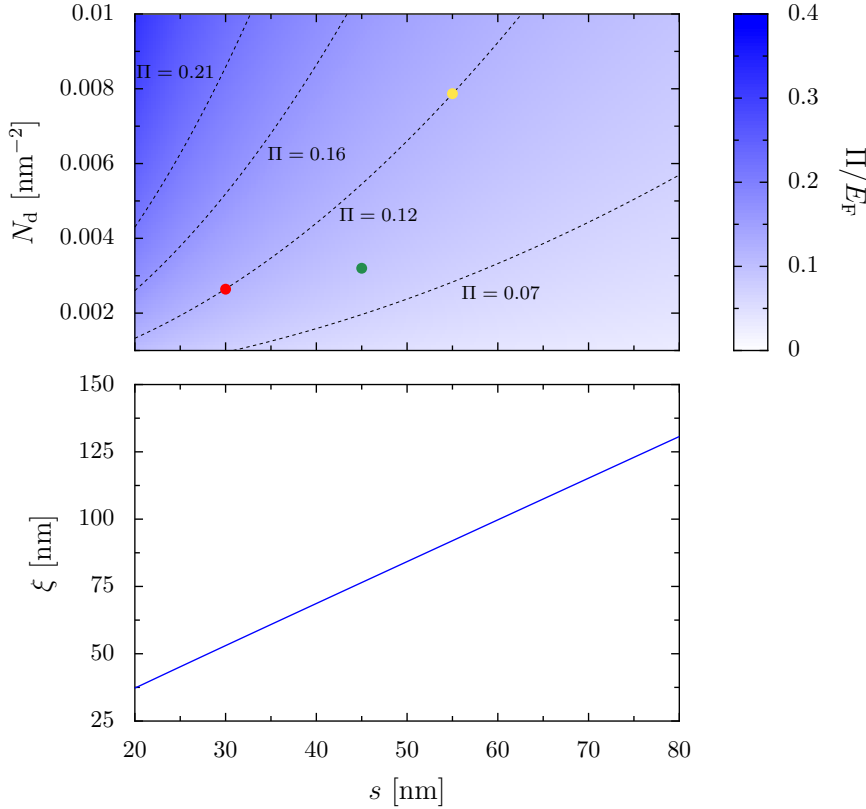


FIGURE 3.4: Upper panel: Root mean square of the potential as function of the density of dopants and the distance  $s$ . Lower panel: Correlation length as a function of the distance  $s$  between the two-dimensional electron gas and the doping layer. The black dashed lines depict the equi-root mean square of the potential value. The dots represents the parameters of the disorder potential represented in Fig. 3.5

One can rewrite the Fourier coefficients  $C(\mathbf{q}_j) = r_j e^{-i\phi_j}$  where  $r_j = |C(\mathbf{q}_j)|$  and  $\phi_j = \arg(C(\mathbf{q}_j))$ . This new definition of  $C(\mathbf{q}_j)$  allows us to express the background potential with a sum only over the positive values of  $j$

$$V_{\text{dop}}(\mathbf{r}) = -E_{\text{Ry}}^* a_{\text{B}}^* \frac{2\Delta q_x \Delta q_y}{\pi} \sum_{j>0} \frac{e^{-q_j s}}{q + q_{\text{TF}}} r_j \cos(\mathbf{q}_j \mathbf{r} + \phi_j) \quad (3.46)$$

Due to the exponential decay factor, we can fix a cutoff by considering that terms above this cutoff yield negligible contributions to the potential value. Then the sum is performed over all  $q_j \leq 3.5/s$ .

Examples of disorder potentials are depicted in Fig. 3.5. One can easily observe the influence of the distance  $s$  on the correlation length. Moreover, we can observe an illustration of the degeneracy of the RMSP in the figure. The potential of the middle and right panel have both different values of both,  $s$  and  $N_{\text{d}}$  (see red and yellow dots on Fig. 3.4), but the same RMSP.

One can also find the electrostatic potential by generating random (ionized) dopant positions and solving the Poisson equation numerically. This



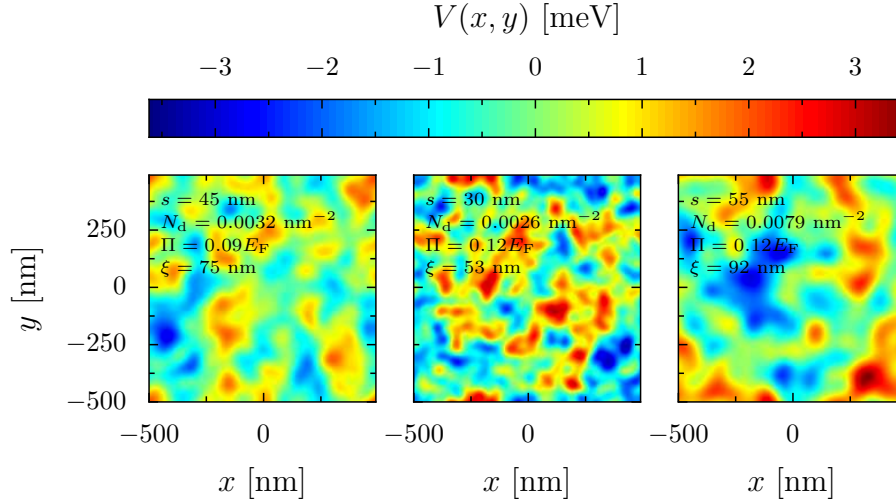


FIGURE 3.5: Three examples of potentials created by randomly chosen positions of the ionized dopants. The left panel corresponds to the green dot in Fig. 3.4, the middle and right panel are both on the same equi-potential root mean square line and are represented by red and yellow dots, respectively.

method will be used in the context of a more realistic model and will be detailed in chapter 6.

### 3.4.2 Simulation of the top gates

To perform the quantum transport simulations with quantum point contact (QPC) top gates, one has to describe the gate-induced electrostatic potential seen by the electrons. We use the model of Ref. [94], that defines the potential (without screening) of a finite gate rectangle defined by  $x_1 < x < x_2$  and  $y_1 < y < y_2$  as

$$\frac{U_{\text{rect}}(x_1, x_2, y_1, y_2)}{A_{\text{QPC}}} = g(x - x_1, y - y_1) + g(x - x_1, y_2 - y) + g(x_2 - x, y - y_1) + g(x_2 - x, y_2 - y), \quad (3.47)$$

where  $A_{\text{QPC}}$  is the potential value under the gate far from the edges and the function  $g(u, v)$  is defined as

$$g(u, v) = \frac{1}{2\pi} \arctan\left(\frac{uv}{sR}\right) \quad \text{with} \quad R = \sqrt{u^2 + v^2 + s^2}. \quad (3.48)$$

The QPC gate potential is composed of two rectangular gates with edges at  $x_1^{\text{left}}, x_2^{\text{left}}, y_1^{\text{left}}, y_2^{\text{left}}$  and  $x_1^{\text{right}}, x_2^{\text{right}}, y_1^{\text{right}}, y_2^{\text{right}}$ , such that

$$U_{\text{QPC}} = U_{\text{rect}}(x_1^{\text{left}}, x_2^{\text{left}}, y_1^{\text{left}}, y_2^{\text{left}}) + U_{\text{rect}}(x_1^{\text{right}}, x_2^{\text{right}}, y_1^{\text{right}}, y_2^{\text{right}}). \quad (3.49)$$

We note that this formula does not include the screening of the electrons.

### 3.4.3 Simulation of the tip

To simulate the tip, we used the approximation shown numerically already in the early days of scanning gate microscopy [95], in which the potential created by a charged AFM tip on the 2DEG can be expressed as

$$V_t(\mathbf{r}) = A_{\text{tip}} \times \frac{L_t^3}{(L_t^2 + |\mathbf{r} - \mathbf{r}_t|^2)^{3/2}}, \quad (3.50)$$

where  $\mathbf{r}_t$  is the coordinate of the tip in the 2DEG plane.  $L_t$  is the typical decay length of the tip potential and is of the same order of magnitude as the distance between the tip and the 2DEG.  $A_{\text{tip}}$  is the potential in the center. We note that a more precise expression for modeling the tip potential taking into account the Thomas-Fermi screening is given in Ref. [96]. However, the most important features concerning the tip modeling are the radius of the depletion disc and the potential tails. We also note that this formula does not include the screening of the electrons.

## 3.5 Conductance through a quantum point contact

We now consider a QPC [97]. The latter corresponds to a tiny constriction ( $\approx 100$  nm) in the 2DEG as depicted in Fig. 3.6 from the experiment of Ref. [1]. The constriction is created by applying a negative voltage to two metallic finger gates located on top of the heterostructure. The electrostatic field creates a depletion in the 2DEG, which means that the potential due to the gate is higher than the Fermi energy. Such an experiment has been realized for the first time in 1988 by Van Wees *et al.* [1] and Wharam *et al.* [98] and has shown the quantization of the conductance appearing at very low temperature (0.6 K in [1]). As depicted in Fig. 3.6, we can observe the stair-like evolution of the conductance while decreasing the magnitude of the gate voltage (The higher the magnitude of the voltage, the smaller the constriction.). The conductance plateaus are located at integer multiples of twice the conductance quantum.

This result can be easily understood using the two-terminal Landauer formula (3.22) and considering the QPC as a narrow ideal quantum wire. In the linear regime and the zero temperature limit, the transmission probability between a mode  $a$  and  $b$  is close to a delta function i.e.  $|t_{ab}|^2 = \delta_{ab}$ . Using Eq. 3.22, we obtain

$$G = \frac{2e^2}{h} N_{\text{qpc}} \quad (3.51)$$

where  $N_{\text{qpc}}$  is the number of modes propagating through the constriction, that is, the modes with transverse energy below  $E_F$ . One of the most important properties, that has been brought into light with this experiment and its theoretical understanding is the finite value of the conductance even for a perfect conductor.

In the following part we will apply the Landauer formula to an easy analytically tractable model called the "saddle point model" in order to observe

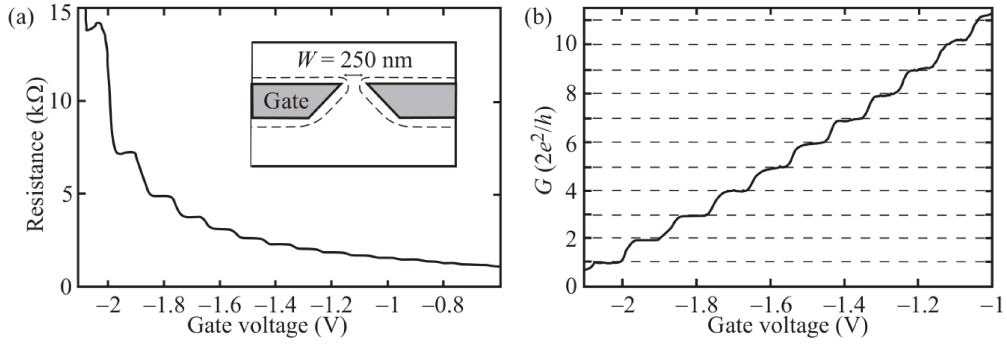


FIGURE 3.6: Experimental measurement of the resistance as function of the gate voltage (a). Representation of gates that create a quantum point contact (inset). Corresponding conductance as function of the gate voltage (b). Figure from [91] which corresponds to the results of [1].

the effects of the temperature on the conductance. In this model, we define the potential seen by electrons moving in a two-dimensional plane  $(x, y)$  with the QPC constriction in the  $y$  direction as

$$V(x, y) = V_0 - \frac{1}{2}m\omega_x^2x^2 + \frac{1}{2}m\omega_y^2y^2 \quad (3.52)$$

where  $V_0$  is the potential at the saddle point.  $\omega_x$  and  $\omega_y$  correspond to the curvature of the saddle in the  $x$  and  $y$  direction, respectively. The transmission probability from an incident mode  $a$  to an outgoing mode  $b$  writes [99, 100]

$$\mathcal{T}_{ba}(\epsilon) = \delta_{ba} \frac{1}{1 + e^{-\pi\epsilon_a}} \quad (3.53)$$

with the energy parameter

$$\epsilon_a = 2 [\epsilon - \hbar\omega_y(a + 1/2) - V_0] / \hbar\omega_x \quad (3.54)$$

The absence of channel mixing comes from the purely quadratic form of the potential<sup>1</sup>. The conductance of such a system is obtained by plugging this transmission into Eq. 3.22. Then we obtain the plot of the left panel Fig. 3.7 which represents the conductance as a function of the applied gate voltage at zero temperature. At finite temperature, the expression of the transmission is plugged into Eq. 3.21. The difference of conductance between the case at  $T = 0$  K and experimentally reachable temperatures of  $T = 100$  mK ( $T = 300$  mK) are represented in the upper (lower) right panel of Fig. 3.7. We notice that the thermal correction is not zero only outside of the conductance plateaus but are still small compared to the conductance values. Indeed, at  $T = 100$  mK, the order of magnitude of the correction is  $10^{-4}G_0$ , while for  $T = 300$  mK it

<sup>1</sup>This model is convenient because of the analytic form of the solutions. However, the quadratic potential is not bounded from below, and then it is unrealistic.

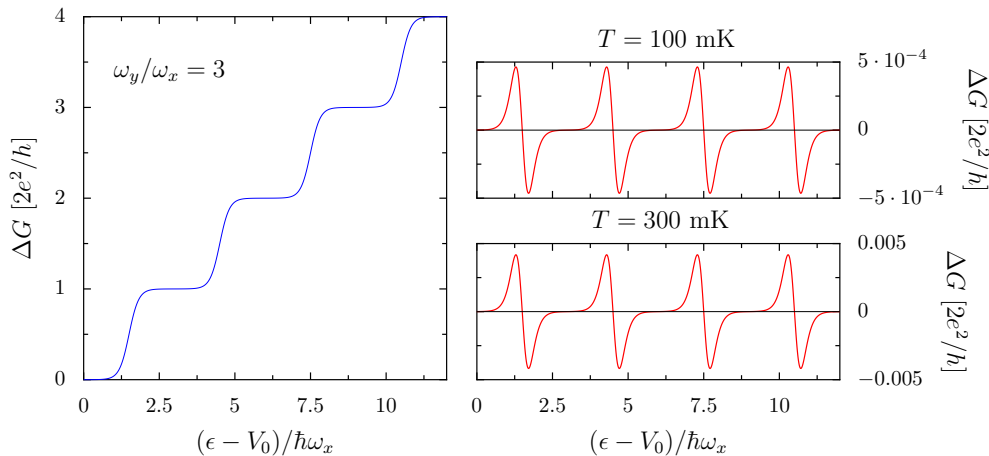


FIGURE 3.7: Left panel: Evolution of the conductance at  $T = 0$  mK as a function of the Fermi energy within the saddle point model where  $\omega_y/\omega_x = 3$ . Right: Difference between the conductance at  $T = 0$  mK and  $T = 100$  mK (upper panel) and  $T = 300$  mK (lower panel).

is  $10^{-3}G_0$ . Considering these small corrections, for all the studies performed in this thesis, we do not take into account the thermal corrections.

### 3.6 Scanning gate microscopy

During the last decades of the 20<sup>th</sup> century, electronic transport in 2DEGs has been the subject of a large number of investigations. However, it is only since the very beginning of the 21<sup>st</sup> century that the electron flow could be properly studied with a scanning probe technique called scanning gate microscopy (SGM) [25, 26]. In those first experiments, the SGM has been used to image the electron flow after passing through a QPC. It has shown the emergence of a branching pattern describing the electron flow. Studies have shown that this branching pattern is due to the presence of disorder in the heterostructure (see Fig. 3.8b) [26, 27, 101] even when the disorder potential is much weaker than the Fermi energy. Consequently, a distinct relationship emerges between the SGM signal and the disorder potential.

In order to image the electron flow, experimentalists use an atomic force microscope (AFM) tip on the system as depicted in Fig. 3.8a. The tip is located at a few tens of nanometers above the surface of the heterostructure and a negative voltage  $V_t$  is applied to it. The aim of the tip is to deplete the electron gas (in a circle shaped region of radius  $r_{\text{dep}}$ ) and thereby to change the transport properties of the 2DEG. A map of the spatially resolved transport properties is realized by measuring the conductance through the sample for different positions of the tip. The first SGM experiments pointed out the correspondence between the electron flow and the SGM signal. Indeed when the tip is depleting a region with an important electron flux, many electrons are scattered back to the QPC, resulting in a decrease in conductance.

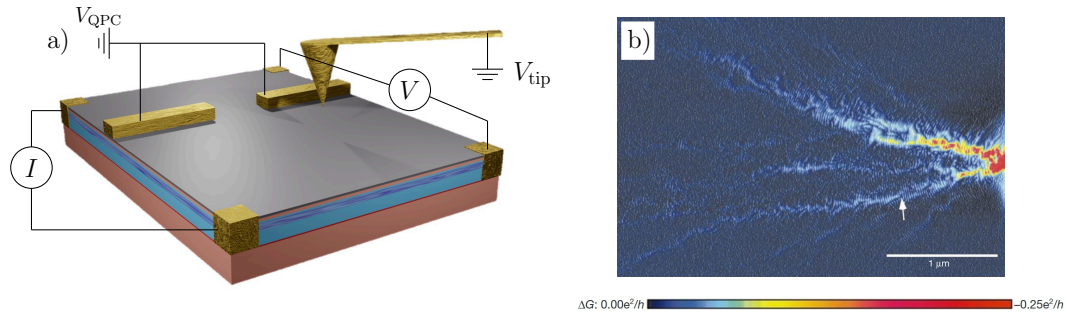


FIGURE 3.8: a): Sketch of SGM experiment with the QPC gate and the metallic tip. b): Electron flow through a QPC at  $T = 1.7$  K obtained by SGM [26]

Another interesting observation is the emergence of interference fringes in the branches. Several studies attempt to provide explanations of this phenomenon [27, 102]. The interference can be caused by three different processes where two of them are limited by the distance between the QPC and the tip that has to be smaller than the thermal coherence length. One of the explanations concerns the case where the QPC gate voltage is such that the conductance of the sample is not located on a plateau. In this situation, the wave is partially reflected and can interfere with the transmitted part that has been backscattered by the tip. This mechanism gives rise to circular-shaped interference patterns. In case of SGM experiments performed on the conductance plateau, another interference mechanism occurs which is characterized by a checkerboard pattern. In this mechanism, the directly backscattered wave interferes with waves that are reflected off the QPC. We note that the second described mechanism does not necessitate an hypothesis on the purity of the sample contrary to the first mechanism. The last proposed mechanism, that can occur at a distance from the QPC higher than the thermal coherence length, is based on the interference between the wave backscattered by the tip and the wave backscattered by impurities that cause backscattering of electrons. Thus, within this method, the interference fringes emerge only in presence impurities. For these three mechanisms the fringes are separated by a distance equal to half the Fermi wavelength.

The main advantage of the SGM is to study the electronic transport of an electron gas that is buried below an insulating layer under the free surface of the sample such that a scanning tunneling microscopy experiment cannot be used to access its properties. This buried electron gas is typically present in the heterostructures used to create high-mobility electron gases.

SGM has been used to study diverse nanostructured systems like QPC [25, 103], Aharonov–Bohm rings [104, 105], cavities [106–108] or even systems allowing to investigate the Braess Paradox [109]. Then, investigations have been performed to determine the possible existing link between the SGM image and the local density of states of the electrons arising from their wave function [93, 107, 110]. Indeed, it has been shown that under the following strong constraints: (i) low temperatures, (ii) in the case of perfect

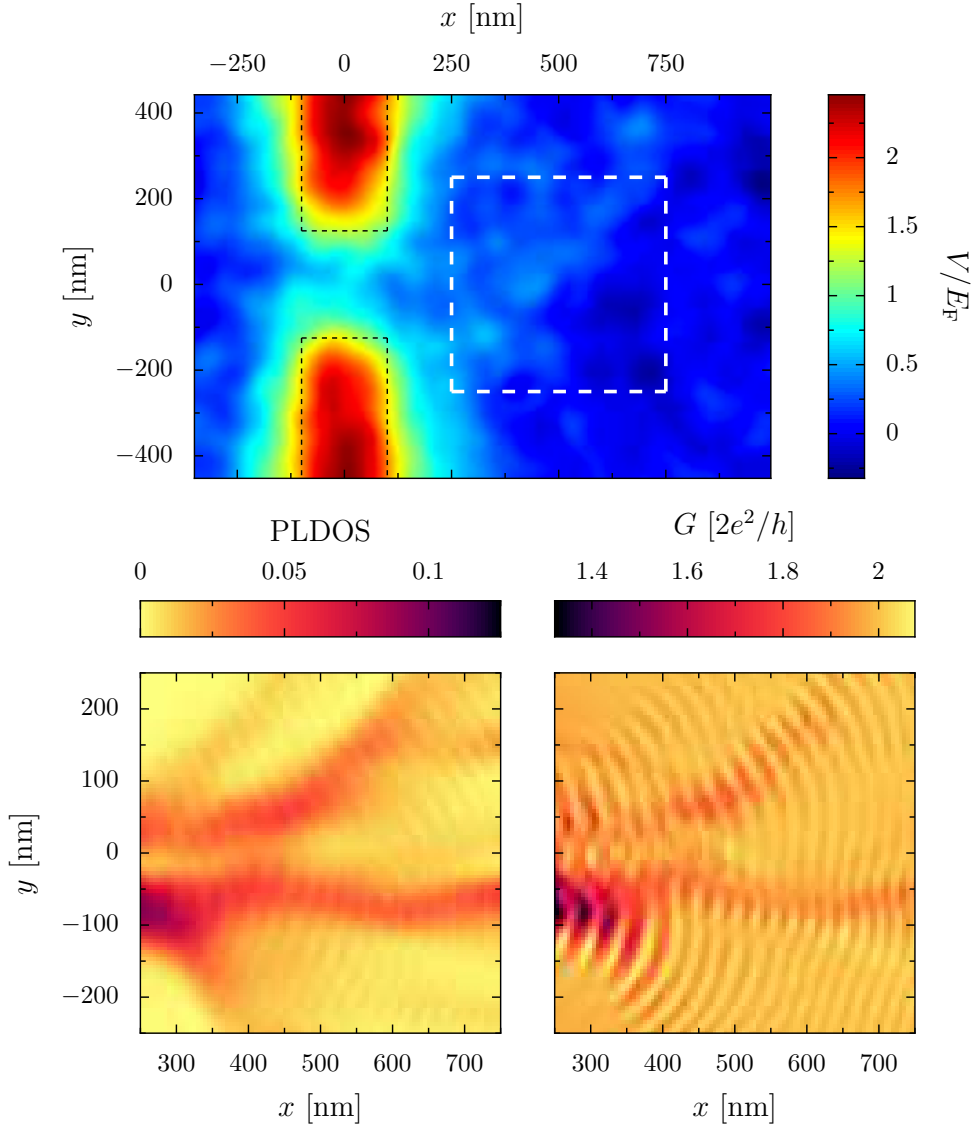


FIGURE 3.9: Sum of the disorder and QPC gate potential (scaled by the Fermi energy) seen by the electrons in the 2DEG for an example of disorder configuration (upper pannel). The location of the gates is depicted by the black dashed line. The lower pannels represent the PLDOS (left) and the SGM signal (right) in the scanning region (white dashed square in the upper panel), respectively. Both quantities are computed at  $E_F = 10$  meV

transmission (e.g., if the energy corresponds to a conductance plateau) and (iii) very weak disorder, the SGM response can be proportional to the partial local density of states (PLDOS) at the Fermi energy of the electrons arising from the QPC [93]

$$\rho_{\text{qpc}, E_F}(\mathbf{r}) = 2\pi \sum_{a=1}^{N_{\text{qpc}}} |\psi_{\text{qpc}^{(+)}, E_F, a}(\mathbf{r})|^2 \quad (3.55)$$

Fig. 3.9 illustrates an example of a numerical simulation where the SGM and the PLDOS have been computed under identical conditions and using the same disorder configuration. For this example, the QPC has been implemented using Eq. 3.49. The spatial position of the metallic gates is indicated by the black dashed lines.

In the example of Fig. 3.9, we can observe similar branch patterns in the two images. The main difference lies in the presence of interference fringes in the SGM map<sup>2 3</sup>.

---

<sup>2</sup>A weaker interference pattern is also observed in the PLDOS image, although it arises from artifacts of the model that inadequately capture the corners of the sample with respect to the leads.

<sup>3</sup>We also notice that the colormaps of the images are inverted. This is due to the explanation stating that when the tip is scanning a region with a high electron flux, it induces a decrease in the conductance.

## Chapter 4

# Global characterization of the disorder inside a heterostructure

The research presented in this chapter corresponds to the starting point of our application of AI to characterize the disorder in a heterostructure due to the ionized dopants. Since we did not know about the possibility of a neural network to extract information on the disorder from any electronic transport data, we started with the most basic problem, which is the prediction of a global characterization of the disorder. Therefore, it was then unnecessary to study a highly realistic system, but a simplified one that is described in more detail in Sec. 4.1. From this model, we generated random potential landscapes that have been used to compute the associated PLDOS images. Those data, required to train the neural network, are presented in Sec. 4.2. For the same reason that led to the usage of a simplified model for the samples, we used a standard CNN architecture that has been optimized to solve our problem, as detailed in Sec. 4.3. The prediction made by the neural network on the global parameters is discussed in Sec. 4.4. Finally, we took advantage of the simplification of the problem (i.e. the characterization by only two parameters of the disorder) to go beyond the black box model that is the neural network, trying to better understand the decisions of the AI as discussed in Sec. 4.5. The main results of this chapter are published in our Ref. [16].

### 4.1 Model of the sample

The aim of this section is to present a simplified model for nanostructured samples consisting of GaAs/AlGaAs heterostructures, where we consider only: (i) the source/drain electrodes, (ii) QPC gates, (iii) the disorder due to a doping layer and (iv) the Thomas-Fermi screening. For this study, we focus only on a region of the 2DEG located between the QPC and the drain electrodes of dimensions ( $L \times W = 1.28 \mu\text{m} \times 0.96 \mu\text{m}$ ). In this model, the QPC is represented as a narrow source electrode of width  $W_{\text{qpc}} = 100 \text{ nm}$  that corresponds to the QPC opening, see Fig. 4.1. We fixed the Fermi energy at 5.6 meV. This energy corresponds to  $\lambda_{\text{F}} = 65.5 \text{ nm}$  (which is a typical value for high-mobility heterostructures [107]), we then fixed the lattice parameter for our tight binding calculations to 5 nm, more than ten times smaller than the Fermi wavelength.



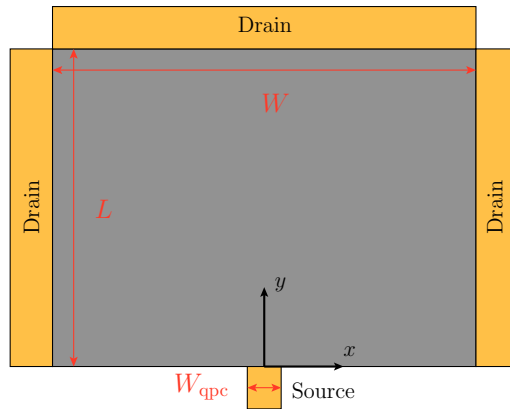


FIGURE 4.1: Sketch of the system geometry used for the simulations. The quantum point contact is simulated by the small source electrode.

The disorder corresponds to the electrostatic potential created by the ionized dopants that are randomly positioned in a two-dimensional plane parallel to the 2DEG and separated by a distance  $s$  from the 2DEG. The disorder potential is computed for the grey part of Fig. 4.1, implemented following the description of Sec. 3.4.1. The simulated potentials used in this chapter have the following properties: the distance  $s$  is between 40 nm and 70 nm and the dopant density is between  $6 \times 10^{11} \text{ cm}^{-2}$  and  $15 \times 10^{11} \text{ cm}^{-2}$ . An example of PLDOS-potential pair is shown in Fig. 4.2

This chapter is dedicated to the preliminary study of the project, where the main purpose is to determine the possibility for an artificial intelligence algorithm to extract properties concerning the disorder from spatial transport properties. As a consequence of this, to obtain the most realistic system is not the highest priority here, and we prefer a convenient one for a proof of principle study. This is the reason why we will not use the SGM as electronic transport data but the PLDOS, in order to be able to create a large dataset with limited numerical resources in a reasonable amount of time.

## 4.2 Presentation of the dataset

For the study, we simulated 72,000 samples. Each sample corresponds to a distinct microscopic disorder configuration, characterized by a different set of Fourier coefficients. The parameters  $s$  and  $N_d$  are randomly chosen from a uniform distribution within the ranges specified in Sec. 4.1. The input, which is the PLDOS, is computed through KWANT for all the samples and the output is either the set of parameters  $(s, N_d)$  or  $(\zeta, \Pi)$ . Examples of potentials with their corresponding parameters are depicted in Fig. 3.5, while the PLDOS with its associated potential is shown in Fig. 3.9. Moreover, thanks to the reflection symmetry of the system geometry with respect to the  $y$ -axis, one can perform data augmentation to double the number of samples by simply adding the reflected image of the disorder configuration and of the

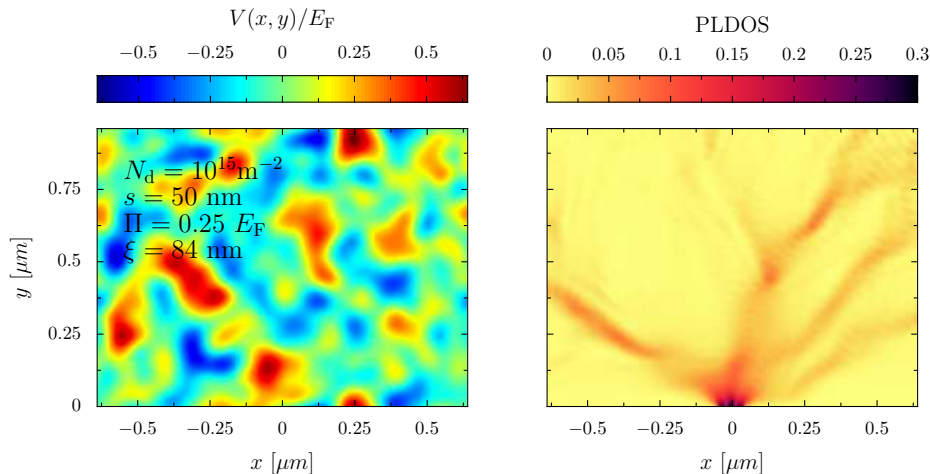


FIGURE 4.2: Left panel depicts a random Coulomb disorder potential that corresponds to the parameters present in the figure. The right panel depicts the PLDOS computed from that disorder. All the parameters used for the creation of these figures are given in Sec. 4.1.

PLDOS. The data augmentation is applied exclusively to samples present in the training set. As a result, the test set only contains samples that have real differences in disorder configuration. We end up with 132,000 samples in the training (and validation) set and 6,000 samples in the test set.

### 4.3 Architecture of the neural network

The main objective of this study is to determine whether an artificial intelligence algorithm can effectively characterize the global disorder within a heterostructure based on electronic transport properties as input data. Given the qualitative nature of this goal, achieving the highest possible quantitative precision, which may involve creating a highly complex neural network, is not of the highest importance.

Since we want to perform a regression on two parameters with an image (the PLDOS data) as input, one can start from a common CNN architecture used for classification and then tune the hyperparameters to solve our problem. This tuning is performed using a grid-search and the  $k$ -fold cross-validation (These methods are detailed in section 2.2.6). Our architecture, depicted in Fig. 4.3, initially consists of three convolution blocks. Each block is composed of two convolution layers followed by one pooling layer. Then, the last set of feature maps is *flattened* and is followed by two dense layers before the output layer. The grid search is used to determine the (i) number of filters, (ii) the size of the kernel, (iii) the number of neurons in the dense layer and (iv) the dropout rate. For all of these hyperparameters, we have selected a set of values that we intend to test. The table in Fig. 4.3 lists the top 6 configurations along with their average error obtained using a 10-fold cross-validation. We note that for the study, we used the set  $(\xi, \Pi)$  as output

Rank	Nb filters	Kernel size	Neurones	$\langle \text{MSE} \rangle$	$\sigma_{\text{MSE}}$
1	(16, 32, 64)	(5, 5, 3)	(100, 20)	0.114	0.004
2	(16, 32, 64)	(5, 5, 3)	(200, 50)	0.117	0.003
3	(16, 32, 64)	(5, 3, 3)	(200, 50)	0.120	0.005
4	(16, 32, 64)	(5, 3, 3)	(100, 20)	0.125	0.005
5	(8, 16, 32)	(5, 3, 3)	(100, 20)	0.130	0.007
6	(8, 16, 32)	(5, 5, 3)	(200, 50)	0.132	0.004

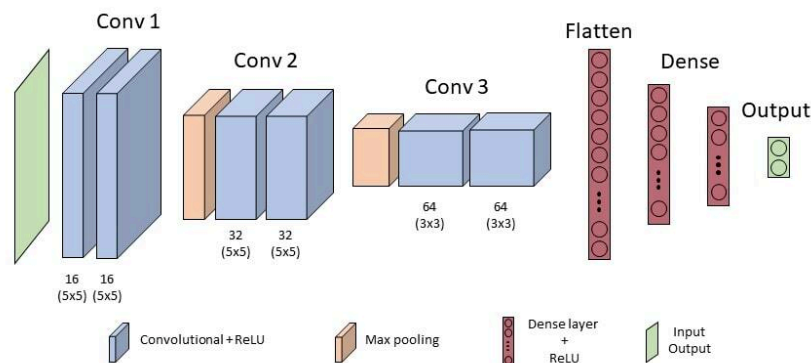


FIGURE 4.3: Top of the figure: List of the top 6 configurations of hyperparameters ranked by their average mean squared error (MSE) obtained using a 10-fold cross-validation. The number of filters and the kernels are represented by 3 numbers, which correspond to the three convolution blocks, respectively. The number of neurons is represented by two numbers that correspond to the number of neurons contained in the two dense layers, respectively. For each configuration, the dropout rate is 0. The best neural network architecture is represented on the bottom of the figure. This model contains about 5 million parameters.

parameters. The other set of parameters consistently yields higher errors, however the ranking remains unchanged.

The results presented in the table of Fig. 4.3 were obtained using 50,000 training (and validation) samples. Each training was performed with over 12 epochs.

Regarding the result of the grid-search, we obtain a set of parameters that seems to perform a bit better than the others. It is important to note that certain hyperparameters, such as batch normalization, batch size, optimizer, loss function, etc., were not included in this study due to the long time required for the procedure. Batch normalization, optimizer and loss function have been studied on less configurations and have shown a systematic difference in all the models (e.g. the optimizer *Adam* is always better than *RM-Sprop*). Concerning the batch size, we also decided to study it afterwards to

test more than three or four different values. Finally, we decided to use the optimizer *Adam* with a learning rate of  $10^{-3}$ , the *mean squared error* as loss function, a batch size of 64 and we do not use batch normalization.

After performing this whole study to tune the hyperparameters, we used our whole training set to train the neural network. However, we still keep some samples as a validation set to prevent overfitting during the training process. We have observed that increasing the number of epochs does not lead to further improvement in model performance. It comes now to test the neural network on unseen data.

## 4.4 Prediction of the global parameters

We now train and use the neural network with the optimized architecture described in Sec. 4.3. In the first scenario, the input is the PLDOS and the output is the two parameter set  $(N_d, s)$ , the dopant density and the distance between the doping layer and the 2DEG. When testing the performance on the test set, we notice a large difference of precision between the two parameters. Concerning  $s$ , on average<sup>1</sup>, the prediction differs from the expected value by 3.2 % while the average error of the parameter  $N_d$  is 14.9%. In the second scenario, we observe a significant improvement of the prediction accuracy when using the set  $(\zeta, \Pi)$  composed of the correlation length and the amplitude of the potential as output. Indeed,  $\zeta$  is predicted with an average error of 2.8 % and  $\Pi$  with an average error of 5.7 %. The reason of the difference of accuracy between the two sets of parameters is detailed below.

From these results, it clearly appears that the electronic transport information, in our case the PLDOS, contains information about the potential. In this study, the predicted information consists of a set of two numbers that characterizes the potential disorder. However, despite these findings, perfect accuracy is not achieved. Therefore, it would be interesting to investigate the source of deviation from exact values. This could be attributed to factors such as a lack of training data, an undersized neural network or the PLDOS does not give the possibility to characterize the potential with a higher accuracy.

Fig. 4.4 is obtained by performing 30 trainings for each training set size. Thanks to the power law shown in Fig. 4.4, it could be possible to improve the precision by increasing the dataset size. However, the quantity of additional examples for the training set would be enormous for only a small improvement. The coefficients of the power law for  $s$  and  $\zeta$  are almost identical (-0.141 and -0.135), which is not the case for  $N_d$  and  $\Pi$  where the coefficient of  $N_d$  is 50 % higher than the one of  $\Pi$ . Thus, the dataset size hardly explains the difficulty for the neural network to predict the parameter  $N_d$ .

Fig. 4.5 shows that the precision does not improve when using a neural network with more parameters. This suggests that the number of parameters in the neural network is not the limiting factor.

---

<sup>1</sup>The average corresponds to the mean absolute value error on all samples of the test set for ten different trainings.

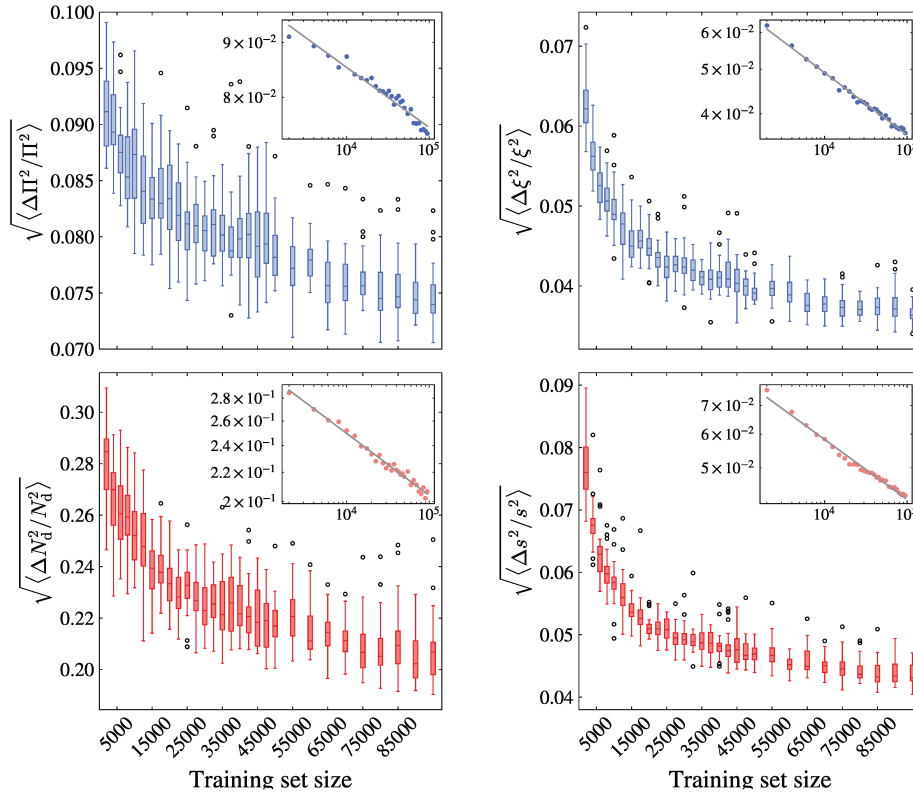


FIGURE 4.4: Error of the predicted parameters as a function of the training set size. The two upper (lower) figures in blue (red) represent the root mean squared error on the set of parameters  $\zeta, \Pi, (N_d, s)$ . The markers are box plots where the horizontal bars inside boxes correspond to the median values. The vertical edge of the boxes corresponds to quartiles, the upper (lower) limit of the whiskers correspond to 1.5 times the inter quartile range plus the value of the third (minus the value of the first) quartile. The white dots are considered as outliers data. The insets represent the median values in log-log scale where the grey dashed lines represent a power law fit.

Concerning the third hypothesis, we can compare the precision of a neural network trained with the PLDOS as input and a neural network trained directly with the potential landscape as input. Tab. 4.1 presents the average relative error for both cases. The average corresponds to the mean absolute error over the test set and over five trainings.

From Tab. 4.1, we observe a loss of information when studying the PLDOS instead of the potential. It is worth noting that we cannot expect better results than the one obtained with the potential as input.

In order to analyze the accuracy of the predictions made by the neural network, one can also use the confusion matrix. This object is commonly used for classification tasks. Nevertheless, by considering a small range of

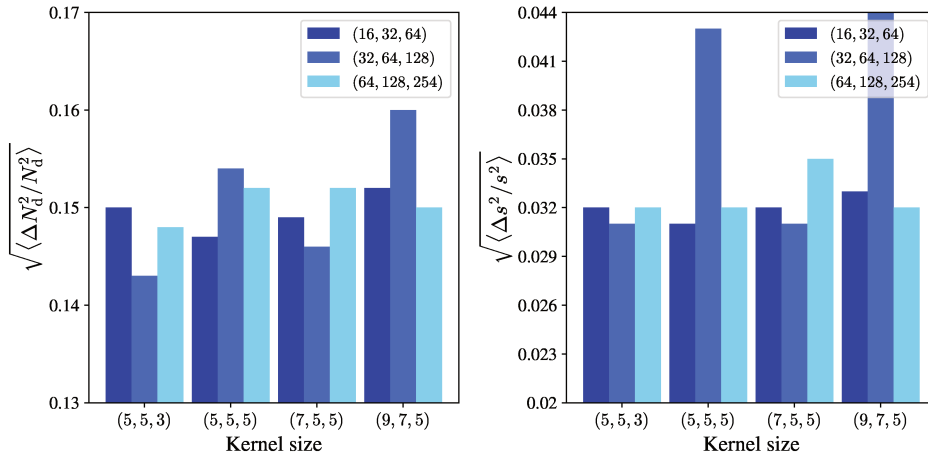


FIGURE 4.5: Average root mean squared error of a neural network (on 10 trainings) for a given number of feature maps (color code) and kernel size (horizontal axis). The left (right) panel corresponds to the density of dopants  $N_d$  (distance  $s$ ).

Input	Output			
	$N_d$ (%)	$s$ (%)	$\Pi$ (%)	$\zeta$ (%)
PLDOS	14.9	3.2	5.7	2.8
Potential	8.5	1.3	3.6	1.0

TABLE 4.1: Error made by a neural network with the architecture depicted in Fig. 4.3 when passing the potential or the PLDOS as input.

predicted and expected values as categories, we can use this tool for our regression problem. In Fig. 4.6, we show the dispersion of the predicted values. For the set of parameters  $(\zeta, \Pi)$ , the accuracy matrix has a pronounced diagonal. This shows that the accuracy peak of the predicted value is well centered on the expected value, except for very high values of  $\Pi$  where the neural network has a tendency to underestimate the value (upper panels of Fig. 4.6). However, when the output of the neural network is the set  $(N_d, s)$ , the dispersion of the prediction is quite large for the  $N_d$  parameter (lower panels). This difference of precision is also highlighted in Fig. 4.7, where the relative error on the two parameters  $N_d$  and  $s$  is plotted for the samples of the test set. In this figure, we also show the relative difference between the RMSP that corresponds to the expected parameters and the one of the predicted parameters. The predictions that correspond to a predicted value of the RMSP  $\Pi'$  equal to the expected one are depicted by the white dots. We observe that the latter predictions are well fitted by the black dashed line, that corresponds to the theoretical relative error on  $N_d$  as a function of the relative error on  $s$  for which we obtain the same value of  $\Pi$ . Indeed, using

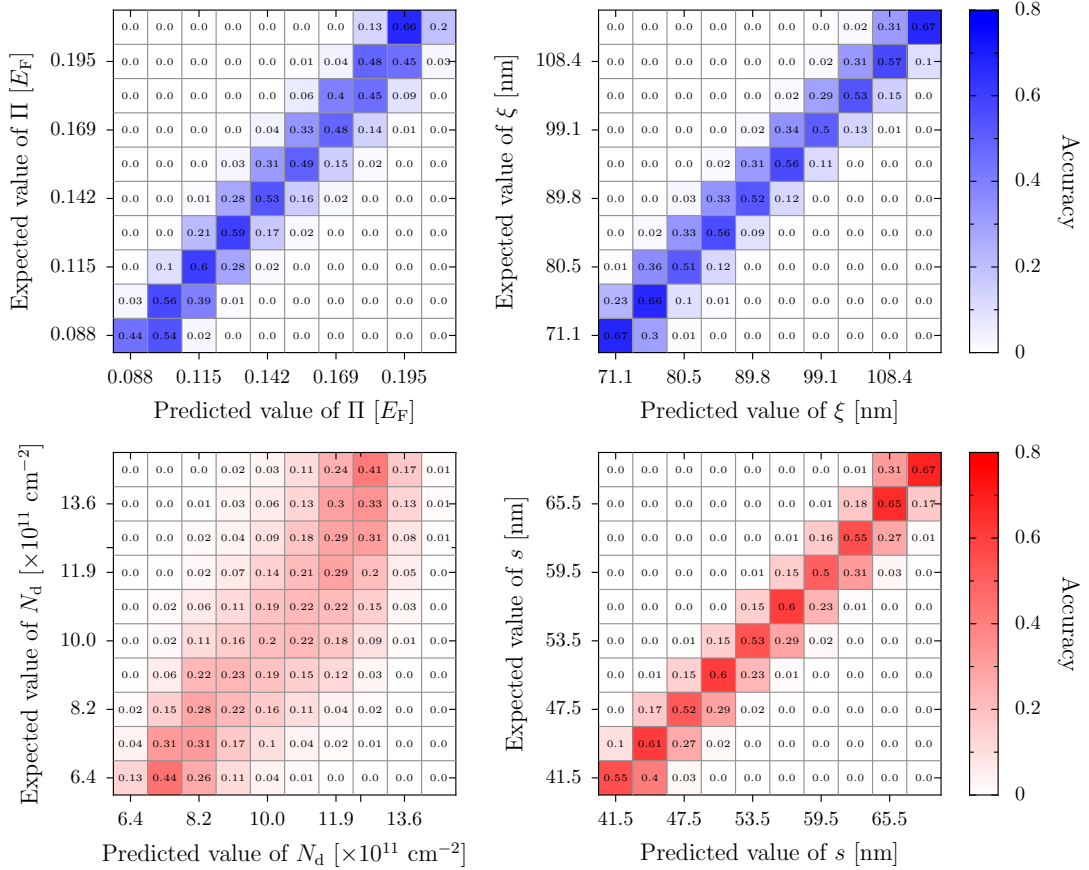


FIGURE 4.6: The two upper (lower) figures in blue (red) represent the confusion matrices for parameters  $\zeta$ ,  $\Pi$ , ( $N_d$ ,  $s$ ). Every element of the matrices corresponds to the percentage of samples from the test set that have been predicted as belonging to a certain value interval while in reality belonging to another or the same (in the diagonal) value interval.

Eq. 3.42 in the regime  $2q_{\text{TF}}s \gg 1^2$ , we obtain

$$N_d \propto \Pi^2 s^2, \quad (4.1)$$

Thus for a constant value of  $\Pi$  we have to lowest order in the deviations  $\Delta N_d$  and  $\Delta s$

$$\frac{\Delta N_d}{N_d} = 2 \frac{\Delta s}{s}. \quad (4.2)$$

It is thus clear from Fig. 4.7 that the error of prediction between the two parameters is correlated following the black dashed line. This implies that the neural network has a tendency to determine the parameters  $s$  and  $N_d$  such that the RMSP computed from the predicted parameters corresponds to the one computed from the expected parameters. Moreover, the parameter  $\zeta$  is proportional to  $s$  (see Fig. 3.4). We can conclude that the branch pattern of the PLDOS depends clearly on  $\Pi$  and  $\zeta$ : if  $\Pi$  is high, the branches will generally

<sup>2</sup> $q_{\text{TF}} = a^{-1}$  and  $s \in [8a, 12a]$  with the lattice parameter  $a = 5 \text{ nm}$

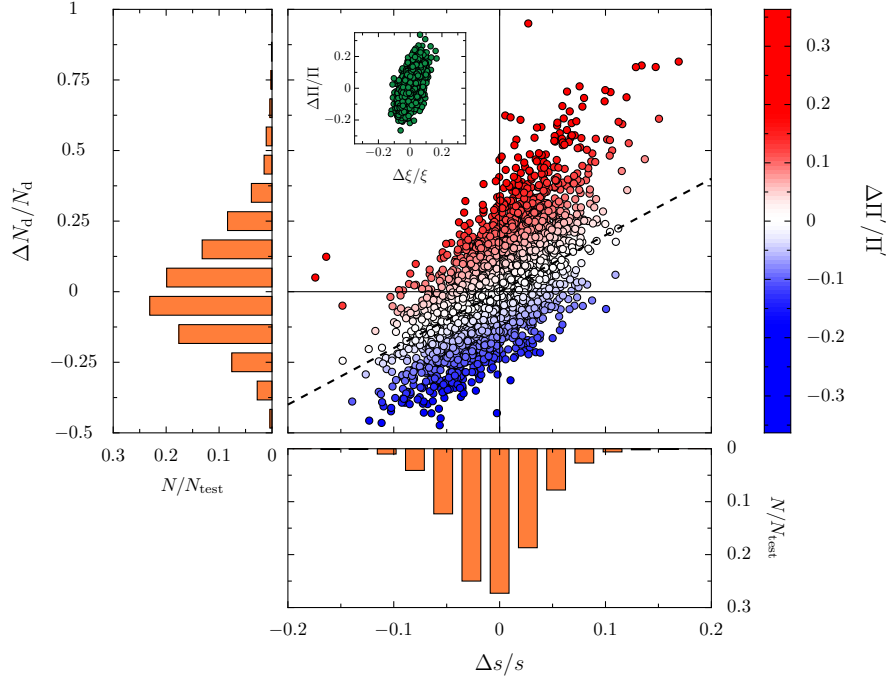


FIGURE 4.7: Distribution of the relative errors  $\Delta N_d/N_d$  of the dopant density and the distance  $\Delta s/s$  for all elements of the test set scaled by the number of samples in the test set  $N_{\text{test}}$  (orange histograms). The central figure depicts the individual values of both errors and thereby puts in evidence the correlation between the two parameters. The color code indicates the relative error on the calculated root mean square  $\Delta \Pi'/\Pi'$ . The black dashed line represents the linearized theoretical line in the  $\Delta N_d/N_d$  versus  $\Delta s/s$  plane for which the predicted root mean square corresponds to the expected root mean square. The inset represents the distribution of the relative errors of the potential root mean square  $\Delta \Pi/\Pi$  and the correlation length  $\Delta \xi/\xi$ .

be shorter while if  $\xi$  is high, the splitting will be less frequent. Then, the density of dopants can be determined only through the impact of the RMSP on the PLDOS, but the latter depends also on  $s$ . When using the set parameter  $\zeta$  and  $\Pi$ , we observe a consequent decrease of the error and of the correlation between the error of prediction of the two parameters. We conclude that AI can be able to indicate which are the best parameters to describe a system. Despite the previous discussion that highlighted the bias in the predictions, when examining the error distribution (orange histogram in Fig. 4.7), we observe a concentration of errors around 0 for both parameters.

The analysis of the neural network prediction led to a partial comprehension of the learning procedure. In the next section, we discuss in more detail what part of the input has been relevant for the predictions made by the neural network.



## 4.5 Beyond the black box

The comprehension of AI algorithms is currently an active research field [111–113]. The growing interest in this domain is explained by the importance to provide an idea on the degree of confidence of a model. Indeed, when it comes to use AI algorithms in real life, it is essential to understand the working procedure of the neural network<sup>3</sup>.

Before trying to understand the working procedure of a model, we first want to determine what part of the PLDOS image contains the information needed for the prediction. Our study consists in training a model with images where a part of the information is hidden and to observe how the precision varies on average as a function of the rate of hidden information. In the second part, we will focus on one model and try to determine what part of the image has been relevant for the prediction of the model. To achieve that, we use the method of the *Integrated gradient* [111] developed by Google. The latter method has the advantage to be easy to apply on a regression task.

### 4.5.1 Where is the information?

In this section, we study the impact of hiding a part of the input data on the average precision of the model that has been trained on those data. As it seems obvious that the form of the branches is important for the prediction, we are going to focus on two parameters. The first one is a cutoff value  $\eta$  under which the value of the PLDOS will be set to 0. Thus, the new PLDOS image created with the parameter  $\eta$  is

$$\bar{\rho}_{\text{qpc},\epsilon}(\mathbf{r}) = \rho_{\text{qpc},\epsilon}(\mathbf{r})\Theta(\rho_{\text{qpc},\epsilon}(\mathbf{r}) - \eta), \quad (4.3)$$

with  $\Theta$  the Heaviside function. It will allow to state on the importance of the region of space where the PLDOS is weak enough such that no branches are present. The other parameter  $\beta$  is a coefficient ranging from 0 to 1, that corresponds to the fluctuation of the PLDOS inside the branches. A value of 0 corresponds to the original image and a value of 1 corresponds to an image where the PLDOS is constant all over the branch. Concerning the images where we change the  $\beta$  parameter, one has first to define a cutoff value below which the PLDOS is too weak to be considered as a branch. However, there is no perfect method to determine this value. Therefore, we just fixed empirically this number to  $\eta = 0.01$  in order to keep as much of the branches as possible without obtaining too large branches. In the case of

---

<sup>3</sup>To illustrate the importance of that, we can mention the algorithm supposed to detect camouflaged enemy tanks asked by the US army. Researchers have trained a neural network with a set of pictures, half of them with the presence of a tank and the other half without. When the US army received the algorithm they noticed that the neural network seemed to answer randomly. It turns out that the images of the training set were biased by the weather. Pictures with tanks have been taken during cloudy days and pictures without tanks have been taken during sunny days. [114]. The neural network had learned to recognize clouds rather than tanks.

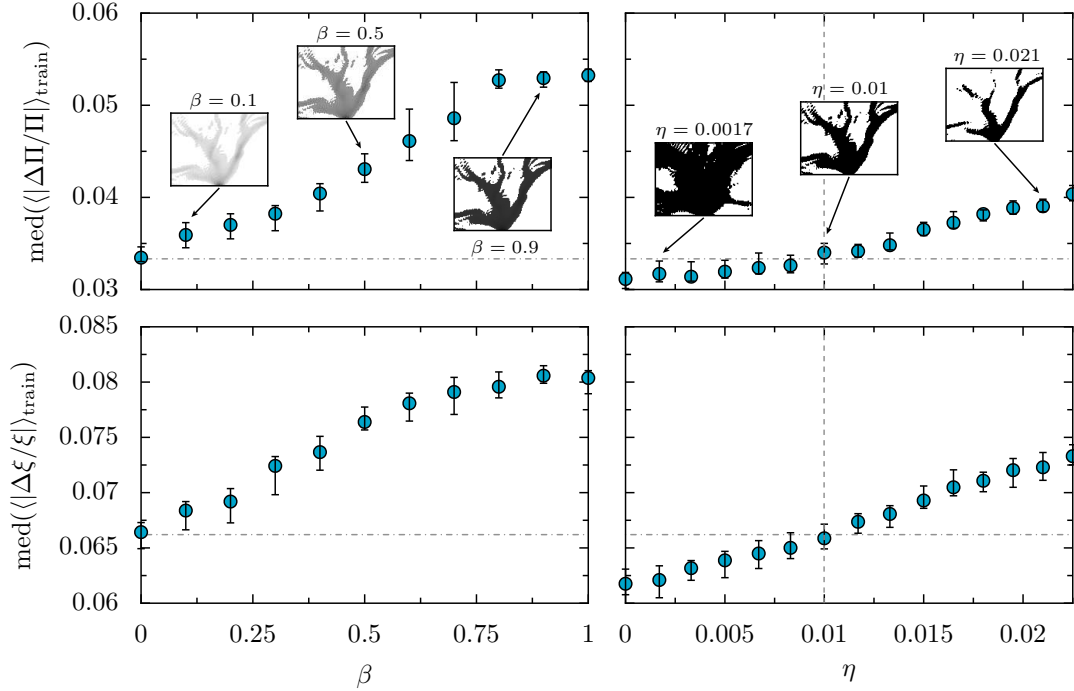


FIGURE 4.8: The blue dots denote the median of the average error of 30 trainings (50,000 samples). The error bars represent the first and third quartile. The upper (lower) panels correspond to the error in the RMSPII (correlation length  $\zeta$ ). On the left panels the varying parameter is the branch fluctuation  $\beta$  at fixed cutoff  $\eta = 0.01$ , while the cutoff  $\eta$  is varied in the right panels for fixed  $\beta = 0$ . The insets of the left panel correspond to examples of one input image that changes with  $\beta$ . In the insets of the right panel, we have an example of the variation of space concerned by the cutoff. The black part corresponds to the region above  $\eta$ . The gray dashed line corresponds to  $\eta = 0.01$  and the dashdotted ones correspond to the initial median error from which the impact of the two parameters  $\eta$  and  $\beta$  can be compared.

variation of the parameter  $\beta$ , the PLDOS is defined as

$$\tilde{\rho}_{\text{qpc},\epsilon}(\mathbf{r}) = [\rho_{\text{qpc},\epsilon}(\mathbf{r}) + (\max(\rho_{\text{qpc},\epsilon}(\mathbf{r})) - \rho_{\text{qpc},\epsilon}(\mathbf{r})) \beta] \Theta(\rho_{\text{qpc},\epsilon}(\mathbf{r}) - 0.01). \quad (4.4)$$

In the following part, it is important to keep in mind that for images where we vary the  $\beta$  parameter, the image already went through a cutoff filter.

The results of this study are presented in Fig. 4.8. This figure depicts the evolution of the average error on the test set for the two parameters  $\Pi$  and  $\zeta$  as a function of  $\beta$  for the left panels and  $\eta$  in the right panels. We first notice that the error of the two parameters evolves with the same tendency. Only the evolution of the magnitude is a bit different. The error increases more for the  $\Pi$  parameter than for  $\zeta$ . When comparing the impact of the two parameters, we have to keep in mind that for all the values  $\beta$  the image already felt a

cutoff  $\eta = 0.01$ . Then we have to ignore the range 0 to 0.01 of the parameter  $\eta$ . From the figure, it seems that increasing the cutoff has less impact than making the intensity of the branches uniform. This leads to say that the variation of the PLDOS inside the branch is a more important determining factor than the weak PLDOS signal in the rest of the space. However, we should be careful with this conclusion. Indeed, it is complicated to compare the rate of hiding information on the range of the two parameters previously described. In other words, the variation of  $\beta$  seems to have more impact on the accuracy than the variation of the cutoff.

Now that we have insights about the location of the information in the input images, in the next part, we study the working procedure of one neural network, using a tool developed by Google to explain the prediction of the neural network.

## 4.5.2 Integrated gradient method

One can distinguish two categories called: *interpretable* and *explainable* machine learning. An interpretable model has the property to be understandable by a human being just by knowing the parameters of the model (e.g. a decision tree is an interpretable model). By opposition, an explainable model needs additional tools to understand their working procedure. In our case, we are performing explainable machine learning with the so-called integrated gradient method.

The main principle of this method is to study the variation of the neural network response  $f_{\text{nn}}(x)$  (where  $x$  is the input image) while varying the pixel values  $x_i$ . However, due to the so-called *saturation problem* [115], one has to sum the gradient over the images  $x' + \alpha(x - x')$  for  $\alpha$  ranging from 0 to 1.  $x'$  is called the baseline image. The integrated gradient method is defined as

$$\phi_i^{\text{IG}}(f_{\text{nn}}, x, x') = (x_i - x'_i) \times \int_{\alpha=0}^1 \frac{\partial f_{\text{nn}}(x' + \alpha(x - x'))}{\partial x_i} d\alpha. \quad (4.5)$$

The baseline is an initial image that is supposed to contain no information that allows the neural network to process correctly the image (e.g. a white or black image). In Eq. 4.5, an issue appears due to the first term. If a pixel of the original image has a value close to the one of the baseline, the value of  $\phi_i^{\text{IG}}$  will be low no matter the importance of this pixel. The solution used to tackle this problem consists in defining the baseline image as the original image on which we apply a Gaussian noise. Nevertheless, the response highly depends on the randomness of the baseline. In order to decrease this dependency, we average the response over different baseline images which differ only by the random Gaussian noise. This corresponds to the expected gradient [115]

$$\phi_i^{\text{EG}}(f_{\text{nn}}, x; B) = \langle \phi_i^{\text{IG}}(f_{\text{nn}}, x, x') \rangle_{x' \in B} \quad (4.6)$$

where  $B$  is the set of baseline images. After training a model on 125,000 samples, one can compute the expected gradient for the two parameters. We note  $f_{\Pi}(f_{\tilde{\xi}})$  the function that corresponds to the neural network and that returns

the value of the parameter  $\Pi$  ( $\xi$ ). With the chosen model, one can observe in Fig. 4.9 that  $\phi_i^{\text{EG}}(f_{\Pi}, x; B)$  and  $\phi_i^{\text{EG}}(f_{\xi}, x; B)$  are quite similar. This property is present in all the trained models we observed. However, the following properties are specific to the studied model. First, we can be surprised by the asymmetry of the highlighted pixels. The right part of the image seems more important than the left part. Also, the upper left corner is systematically highlighted. Those two last properties are not easy to explain and may correspond to a local minimum in the training process that has been able to give good results while not all the parameters have been well used. If the observations discussed so far do not allow to understand how the model has determined the value of  $\Pi$  and  $\xi$ , the following could give us some insights. It appears that some highlighted pixels correspond to branches of weak electron density. This means that by changing the value of the density inside those branches, the result can be considerably modified.

Even if this last observation seems nice and convincing, we have to keep in mind that the integrated gradient method is not perfect and depends on external parameters besides the model itself. Moreover, some other models seem to focus on another part of the image (e.g.  $\approx 200$  nm after the QPC or the region around the main branch(es)) and the observations are based on the visual interpretation of a human being.

The presented methods have shown some element of comprehension on the location of information in the image and we learned a bit about the learning procedure of the model. However, the explainable artificial intelligence domain is still in expansion and the results given by the various existing methods cannot be blindly accepted.

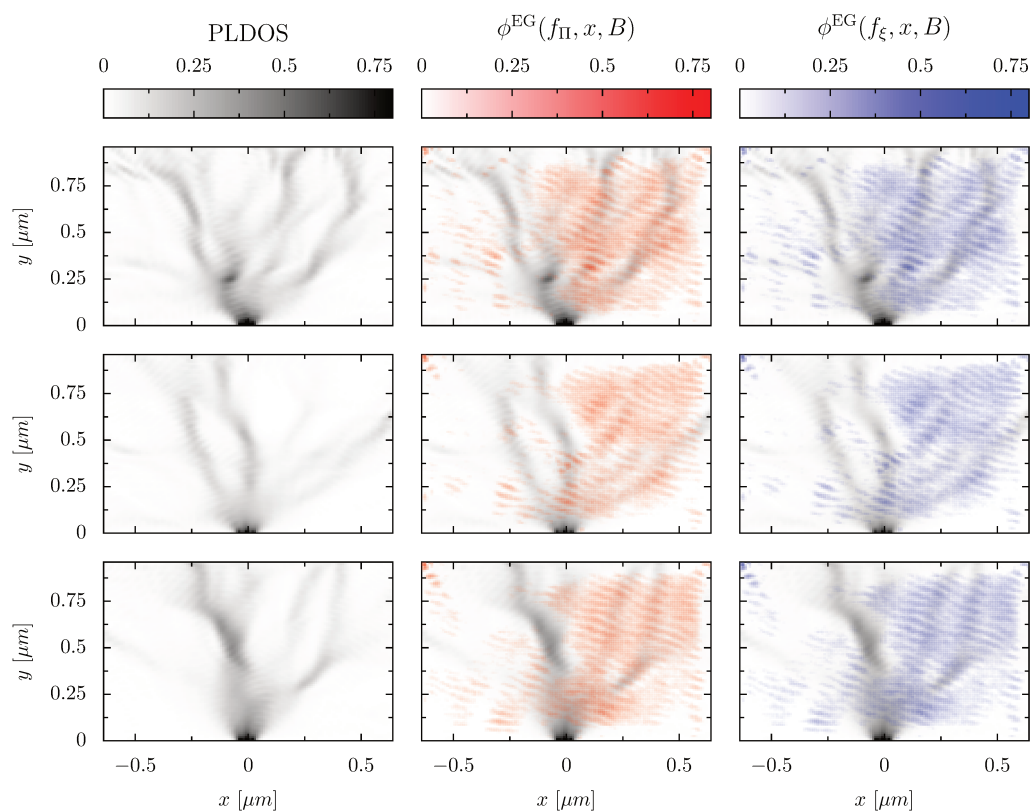


FIGURE 4.9: The left column corresponds to the input image. The middle (right) column shows the input image (in gray scale) with the underlying expected gradient response for the parameter  $\Pi$  ( $\xi$ ) (in color scale). Each line corresponds to a different example.

## Chapter 5

# Determination of the full potential landscape from the PLDOS

In chapter 4, we presented a proof of principle that basic information on the disorder potential can be extracted from the PLDOS. We now move forward to a more ambitious goal which consists in using an artificial intelligence approach to solve completely the inverse problem by determining the full potential landscape. In this section, we use the model as described in Sec. 4.1. The only difference is the assumption on our knowledge of the sample. Previously, the two experimental parameters  $s$  and  $N_d$  were the unknown parameters of the potential and thus the targets for the neural network. In our new study, we create the dataset with the same global properties as the studied sample. In other words,  $N_d$  and  $s$  are no longer unknown parameters of the system (In the dataset:  $s = 50$  nm and  $N_d = 1 \times 10^{12}$  cm<sup>-2</sup>). The remaining unknown feature is the set of positions of the dopants. In our model, the information on the dopant positions is hidden either in the potential landscape image or in its Fourier coefficients. We show in Sec. 5.1 that it is then possible to predict the potential by performing a regression on the Fourier coefficients, and in Sec. 5.2 that one can even directly obtain the value of the potential for each point of space. In order to give more information to the neural network, we also created an architecture that can receive three PLDOS images as input as discussed in Sec. 5.3. Finally, in Sec. 5.4, we present a study of a system without QPC which does not present a branch pattern in order to generalize our approach. The main results of this chapter are published in Ref. [16].

## 5.1 Regression on the Fourier coefficients

### 5.1.1 Architecture of the neural network

In this section we focus on determining the full potential disorder landscape by performing a regression on the Fourier coefficients. The neural network still takes the PLDOS image as input, but the neurons in the output layer correspond to all the Fourier coefficients used to create the potential. Each Fourier coefficient is represented by two output neurons, one for the real part and another for the imaginary part. Thus, the architecture of the current neural network is very similar to the one used previously to characterize the disorder potential with only two parameters. As depicted in the lower

Rank	Nb filters	Kernel size	Neurones	$\langle MSE \rangle$	$\sigma_{MSE}$
1	(128, 256, 256)	(5, 5, 3)	800	0.055	0.010
2	(128, 256, 256)	(5, 3, 3)	800	0.070	0.005
3	(128, 256, 512)	(5, 3, 3)	800	0.095	0.019
4	(64, 128, 256)	(7, 5, 3)	800	0.111	0.016
5	(64, 128, 256)	(5, 5, 3)	800	0.116	0.019
6	(64, 128, 256)	(5, 5, 3)	1600	0.138	0.018

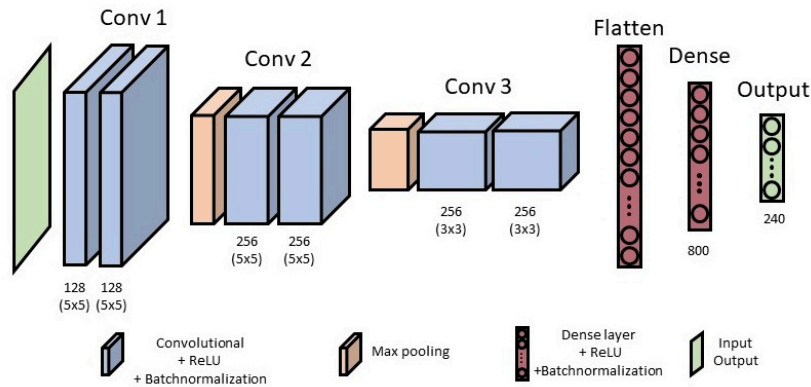


FIGURE 5.1: Top of the figure: List of the six best configurations of hyperparameters evaluated with 10-fold cross-validation. The numbers of filters and kernels are represented by 3 numbers, which correspond to the three convolution blocks, respectively. The number of neurons is represented by two numbers that correspond to the number of neurons contained in the two dense layers, respectively. For each configuration, the dropout rate is 0. The best neural network architecture is represented on the bottom of the figure. This model contains about 5 million parameters.

panel of Fig. 5.1, the first part of the network is composed of a sequence of convolution blocks. The output of the last convolution layer is flattened and connected to one dense layer before the output layer. We note a small difference concerning the number of dense layers with the global architecture of the model of Chap. 4. Indeed, the current model has only one hidden dense layer because the presence of two successive hidden dense layers leads often the training to be stuck in a local minimum.

In order to obtain high quality results, one has to use a large neural network. It is then difficult to perform the grid search and  $k$ -fold cross validation method due to the long training time and the high memory required. To perform the study, we used 80,000 samples that went through a standard

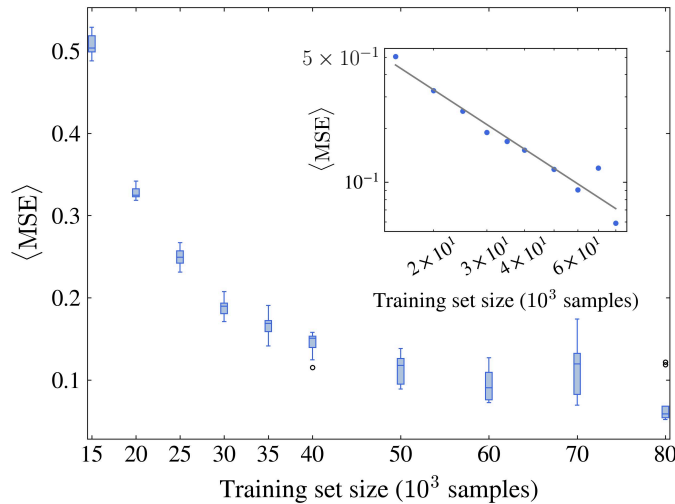


FIGURE 5.2: Mean squared error summed over all the Fourier coefficients in the prediction of the Fourier coefficients as a function of the training set size. The markers are box plots where the horizontal bars inside boxes correspond to the median values. The vertical edge of the boxes corresponds to quartiles, and white dots are outliers. The inset represents the median values in log-log scale where the grey dashed line represents a power-law fit.

normalization. Then, we empirically determined a range of hyperparameters that allows to have a satisfactory prediction. To perform a fine tuning, we choose 9 models that already give good results and evaluate them. To perform this evaluation we shuffle and split the dataset, keeping 90% for the training and 10% for testing. The average results of the latter procedure, repeated 5 times, are presented in Fig. 5.1.

The parameters that do not appear in the architecture of Fig. 5.1 are the learning rate set to  $10^{-4}$ , the optimizer for the gradient descent which is *Adam* and the chosen loss function is *root mean squared*. We also note the presence of batchnormalization which is especially efficient because of the normal distribution of the Fourier coefficient.

### 5.1.2 Accuracy of the predicted Fourier coefficients

Before training the neural network, we have performed data augmentation by using the symmetry as explained in Sec. 4.2. The symmetry operation on the Fourier coefficients is a switch of the sign of  $q_y$ . Thus, the inverse Fourier coefficients become

$$C^{\text{inv}}((q_x, q_y)_j) = C((q_x, -q_y)_j). \quad (5.1)$$

This data augmentation increases the number of samples up to 150,000 (135,000 samples for the training set, 5,000 for the validation set and 10,000



for the test set). Fig. 5.2 shows the mean squared error as a function of training set size. Since each point of the figure correspond to 30 trainings, we limited the figure to 80,000 samples in the training set, due to the computation time. Thus, for the largest training set size, we already reach a very high precision rate that would be complicated (and not interesting for most usages) to increase. From the power-law fit, we obtain an exponent of  $-1.24$ , meaning that to divide the mean squared error by two, one has to add 100,000 samples to the current training set.

The results presented above correspond to the mean squared error averaged over all the Fourier coefficients composed of real and imaginary part (i.e.  $1/N_{\text{coeff}} \sum_{N_{\text{coeff}}} |C(\mathbf{q})|^2$  with  $N_{\text{coeff}}$  the number of Fourier coefficients). Thus, it does not contain any information on the variation of the prediction error that can exist between Fourier coefficients associated to different vectors  $\vec{q}$  of the reciprocal space.

To study such a dependence, we compute the average error of all the samples on each Fourier coefficient individually. In the two upper panels of Fig. 5.3, the latter average error is depicted in colorscale for each Fourier coefficient located in their corresponding  $q$ -vector coordinates in the reciprocal space. First, we note the similarity of the average prediction error made for the real and imaginary parts by the neural network. The dependency of the error on the magnitude of the vector  $\vec{q}$  visible in the two upper panels is highlighted in the lower panel of Fig. 5.3 (blue dots). In that figure, one can observe the exponential increase of the error with  $|\vec{q}|$ . This can be explained by taking into account the decaying exponential factor in Eq. 3.46. Indeed, we note that the larger is the magnitude of the vector  $\vec{q}$ , the weaker is the influence of the corresponding term of the potential. Thus, it becomes difficult for the neural network to determine the Fourier coefficient that corresponds to a term that has almost no impact on the disorder configuration.

The accuracy of the neural network in the prediction of the Fourier coefficients with the PLDOS as input cannot be better than when using the potential itself as input. In the case where the accuracy is the same, that means that the PLDOS contains the whole information on the disorder. The lower panel of Fig. 5.3 depicts the evolution of the error with the magnitude of  $\vec{q}$  when the potential is used as input (red dots). When performing the exponential fit, we observe that reduce the pre-factor of the exponential by a factor 20 using the potential as input<sup>1</sup>. Similarly to Chap. 4, we observe a significant decrease of the error when using the potential as input and thus a loss of information on the disorder in the PLDOS image.

One can also evaluate the performance of the neural network by reconstructing the potential  $y_{\text{pred}}$  from the predicted coefficients and to compare it to the expected disorder potential  $y_{\text{exp}}$ . In order to calculate the difference between two images, we use the Pearson correlation coefficient (PCC)

$$r_P = \frac{\sum_{i=1}^{N_p} (y_{\text{pred},i} - \bar{y}_{\text{pred}}) (y_{\text{exp},i} - \bar{y}_{\text{exp}})}{\sqrt{\sum_{i=1}^{N_p} (y_{\text{pred},i} - \bar{y}_{\text{pred}})^2} \sqrt{\sum_{i=1}^{N_p} (y_{\text{exp},i} - \bar{y}_{\text{exp}})^2}}, \quad (5.2)$$

<sup>1</sup>The factor in the exponential is about the same in the two cases.

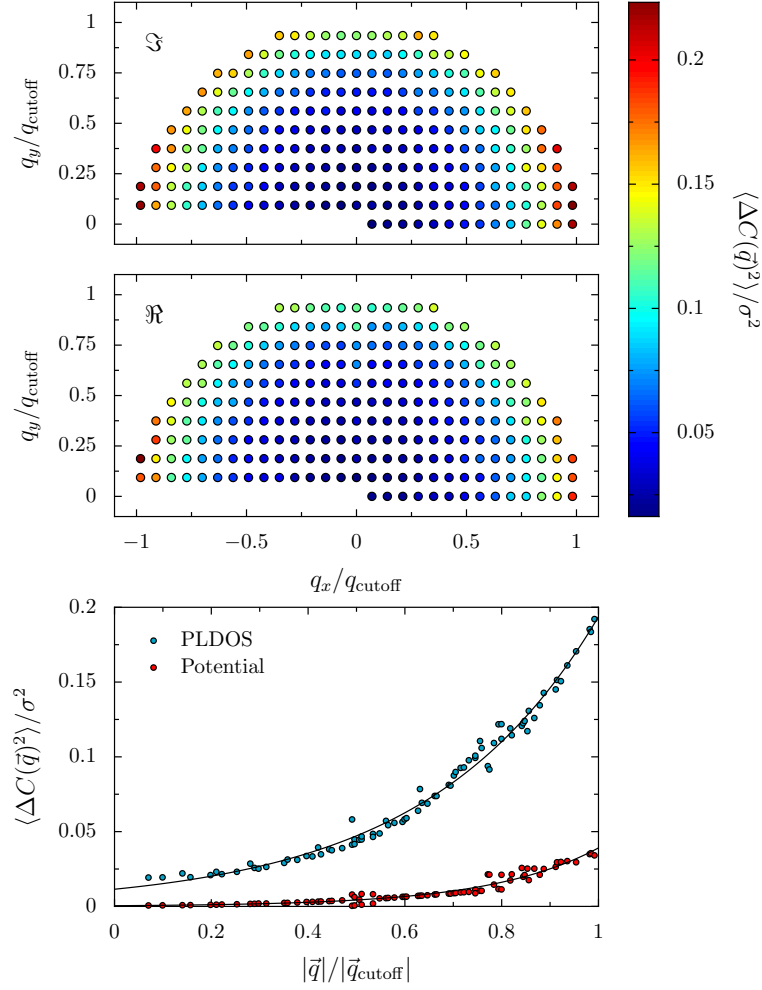


FIGURE 5.3: Average error of the real (imaginary) part of the Fourier coefficients located in their associate  $q$  vector coordinates in the reciprocal space depicted in the upper (middle) panel. The lower panel represents the average error performed on the Fourier coefficients with respect to the magnitude of their associate  $q$  vector. The black solid lines correspond to exponential fits.

where the sums run over the  $N_p$  pixels of the images while  $\bar{y}_{\text{pred}}$  and  $\bar{y}_{\text{exp}}$  are the mean values of the two images. Using the PCC to evaluate our best model<sup>2</sup> trained with 135,000 samples, we obtain the histogram of the PCC for the samples of the test set shown in the upper panel of Fig. 5.4. Fitting the histogram with a skew normal function

$$\gamma(x; \zeta, \omega, \theta, \beta) = \beta \frac{1}{\omega \sqrt{2\pi}} e^{-\frac{(x-\zeta)^2}{2\omega^2}} \left[ 1 + \text{erf} \left( \theta \frac{x-\zeta}{\omega \sqrt{2}} \right) \right], \quad (5.3)$$

where  $\text{erf}(x)$  is the error function, we find that the peak is at  $r_p = 0.987$ . Considering the distribution, we note that 99 % of the predictions will have

<sup>2</sup>best model over 10 trainings

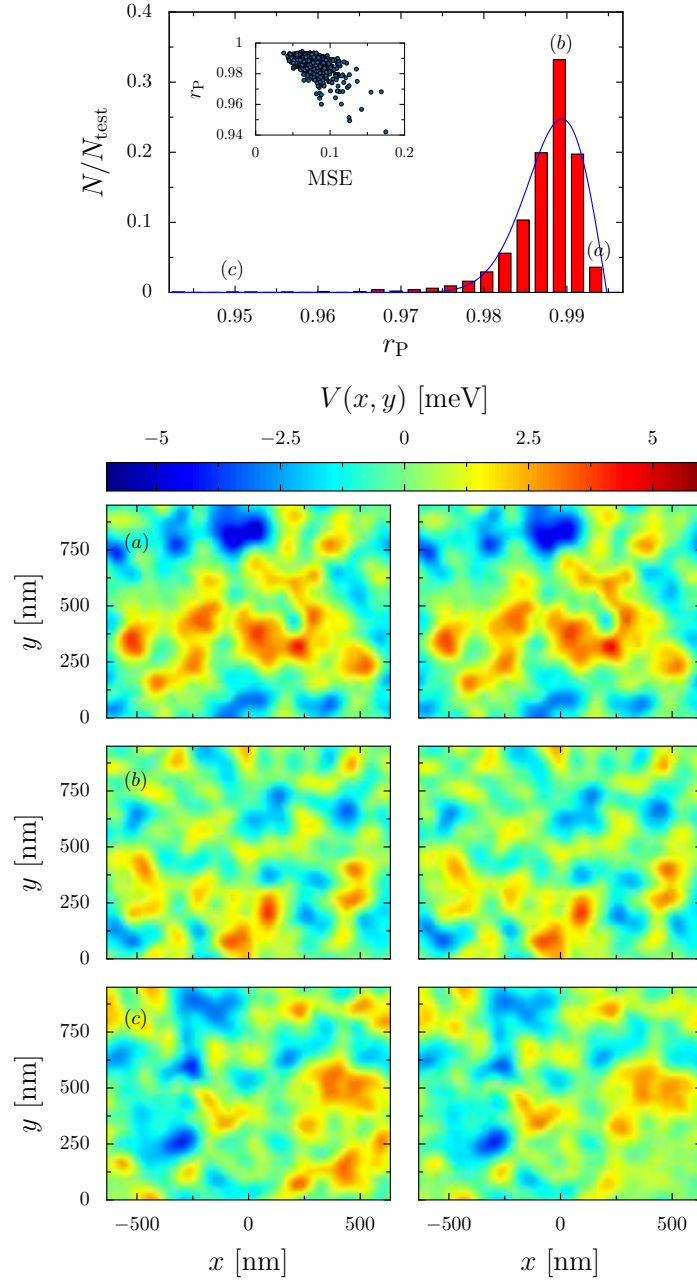


FIGURE 5.4: The upper panel represents the distribution of the correlation coefficient of samples from the test set between the expected potential and the one reconstructed from the predicted coefficients. The blue dashed line represents the fit of the distribution (5.3) for the parameters  $\zeta = 0.99$ ,  $\omega = 0.005$ ,  $\theta = 0.04$  and  $\beta = -0.067$ . The inset represents for each sample of the correlation coefficient described before and the error performed on the Fourier coefficients. The images of the lower panel are examples of expected (left) and predicted (right) potentials that correspond to a correlation coefficient indicated in the upper panel.

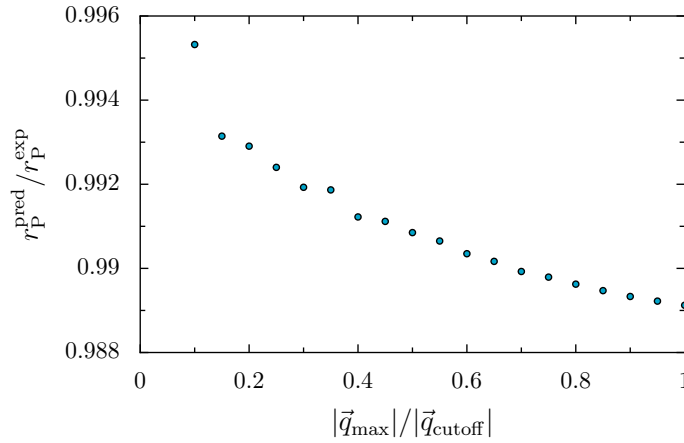


FIGURE 5.5: Ratio of the Pearson coefficients between the reconstructed potentials from the expected and predicted Fourier coefficients as a function of the maximum magnitude of the vector  $\vec{q}$  considered in the sum scaled by the actual cutoff.

a PCC higher than 0.976. Examples of potential reconstruction are depicted in the lower panel of Fig. 5.4 where the left (right) images represent the expected (predicted) potentials. The three examples have different PCC and correspond to the categories of accuracy labeled  $a$ ,  $b$  and  $c$  in the histogram. One can notice the reliability of the prediction even for the worst cases. It is also interesting to notice the correlation between the error in the coefficients and the PCC between the reconstructed potential and the expected one. The correlation between these quantities exists but is not as clear as one can expect. This is probably due to the difference of impact in the potential reconstruction of the Fourier coefficients associated to different magnitudes of  $\vec{q}$  while the MSE considers all the coefficients equally.

To sum up, the neural network has more trouble to predict Fourier coefficients associated to high  $|\vec{q}|$  values and in the same time those coefficients have a limited impact on the potential. Therefore, one can study which Fourier coefficients impact negatively the reconstructed potential. For this we compare the PCC  $r_{\text{P}}^{\text{exp}}$  measured between the potential computed from all the *exact* Fourier coefficients and computed with the *exact* Fourier coefficients associated to  $|\vec{q}| < |\vec{q}_{\max}|$  and the PCC  $r_{\text{P}}^{\text{pred}}$  measured between the potential computed from all the *exact* Fourier coefficients and computed with the *predicted* Fourier coefficients associated to  $|\vec{q}| < |\vec{q}_{\max}|$ . Fig. 5.5 shows the ratio of these two PCCs as a function of  $|\vec{q}_{\max}|$ . We notice the convex shape of the decreasing ratio, which indicates that the higher  $|\vec{q}|$ , the smaller is the impact of predicted Fourier coefficient. This means that the decreasing impact on the potential of a Fourier coefficient for a high  $|\vec{q}|$  is more important than the exponentially increasing error with  $|\vec{q}|$ . However, we notice that the ratio decreases in a very short range implying that the ratio is almost constant and thus that all predicted Fourier coefficients have almost the same impact on the reconstructed potential. Therefore, we do not expect increasing the precision of the neural network by a smart weighting of the loss function

depending on the Fourier coefficient represented by the output layer.

In this section, we have seen that one can determine the potential using a neural network that is trained to predict the Fourier coefficients. The advantage of this method is to obtain a reconstructed potential that has a physical meaning. However, this method works only for disorders that are described by Eq. 3.46. Moreover, since the end of the neural network is composed of dense layers, the training is very time consuming. In the following section we use a different neural network that allows to overcome the issues discussed above.

## 5.2 Encoder-decoder neural network

### 5.2.1 Architecture of the neural network

In this section, we tackle the same inverse problem of training a neural network to give the disorder potential from the PLDOS in a more direct way. Instead of performing a regression on the Fourier coefficients and then reconstruct the potential landscape, we directly perform the regression on the potential landscape image. The easiest way to solve this task with AI is to use a convolutional encoder-decoder (CED) (detailed in Sec. 2.3.3). The architecture of the network will be composed of two parts: The encoder, that decreases the resolution of the image (with the pooling layers) while the number of feature maps increases. Then, the decoder increases the size of the image (with the upsampling layers) while reducing the number of feature maps to finish with an image that has the same dimension as the input image. In order to determine the neural network architecture and the hyperparameters, we perform a  $k$ -fold cross-validation with 20,000 samples. We choose the range of the parameter values close to a satisfactory set of parameters. The 6 best configurations are detailed in the upper panel of Fig. 5.6. We note that hyperparameters like the number of filters in the convolution layers and the number of convolution layers do not appear in the table of Fig. 5.6 because they are identical for the 6 best configurations. As before, the amount of time required to perform the cross-validation on all parameters is too important. Then we fixed some parameters on common values: the learning rate is equal to  $10^{-3}$ , the chosen optimizer is *Adam*. We cannot ignore that the results presented in the table of Fig. 5.6 are very close and their difference is typically in the order of magnitude of the statistical error. Hence, the ranking gives just a rough idea of the best neural networks and allows us to choose a neural network architecture and the associated hyperparameters based on a quantitative information. We keep in mind that all of the presented configurations could give about the same results for the rest of the study. Nevertheless, the  $k$ -fold cross-validation has been performed on about 60 models and allowed to bring to light which ones are the best configurations.

The lower panel of Fig. 5.6 depicts the architecture of the selected configuration and includes the information that is not present in the table.

Rank	Kernel size	Dropout	Batch	$\langle r_P \rangle$	$\sigma_{r_P}$
1	(5, 5, 3)	(0.0, 0.0, 0.0)	8	0.984	0.001
2	(5, 5, 5)	(0.0, 0.0, 0.0)	8	0.983	0.003
3	(5, 5, 3)	(0.1, 0.1, 0.1)	8	0.983	0.001
4	(7, 5, 3)	(0.1, 0.1, 0.1)	8	0.982	0.002
5	(5, 5, 5)	(0.1, 0.1, 0.1)	8	0.981	0.003
6	(5, 5, 3)	(0.0, 0.0, 0.0)	16	0.980	0.004

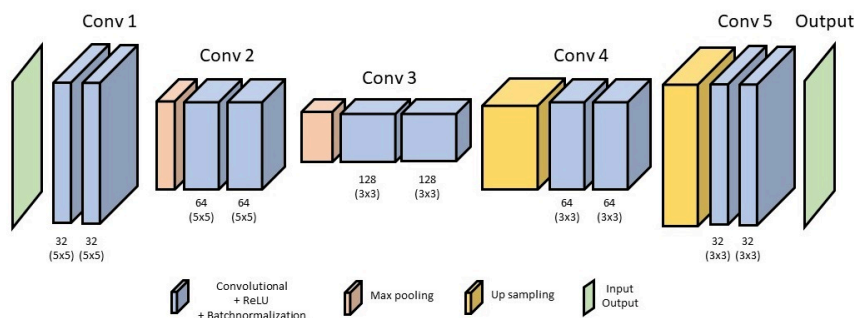


FIGURE 5.6: Top of the figure: List of the six best configurations of hyperparameters evaluated with 10-fold cross-validation. The size of the kernels is represented by 3 numbers, which correspond to the three convolution blocks, respectively. The dropout is also represented by 3 numbers that correspond to the value of the dropout layer located at the end of each convolution block. The best neural network architecture is represented at the bottom of the figure. Each box represents a layer and the meaning of the color of the box is depicted directly in the figure. This model contains about 550,000 parameters.

## 5.2.2 Average error in real space

In order to perform the following study, it is first important to know if the size of our training set is large enough to not expect a significant improvement in the quality of prediction of our neural network by adding a reasonable amount of training samples. To perform such a study, we train 20 neural networks with training set sizes from 15,000 samples to 90,000 samples and observe the evolution of the distribution of the averaged Pearson coefficients between the predicted potentials and the exact ones for the samples of the test set. We note that we doubled the size of our training set by performing the data augmentation exploiting the symmetry of the set up as described in the previous section.

In Fig. 5.7, we show the decrease of the error ( $1 - r_P$ ) of prediction performed by the neural networks with respect to the training set size. In the inset of the figure, we can observe the power-law of the median value. From

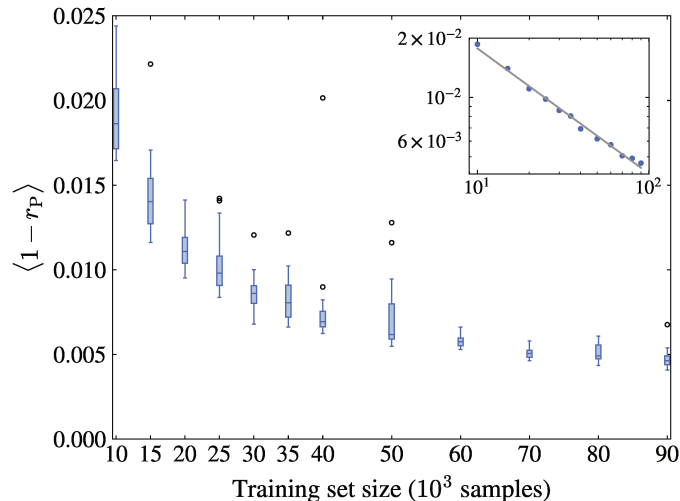


FIGURE 5.7: Error in the prediction of the potential ( $1 - r_P$ ) as a function of the training set size. The markers are box plots where the horizontal bars inside boxes correspond to the median values. The vertical edge of the boxes corresponds to quartiles and white dots are outliers. The inset represents the median values in log-log scale where the grey dashed line represents a power-law fit with exponent -0.63.

this power-law, we note that to divide the median of the error by two, we have to multiply the training set size by a factor 3. Considering the already impressive quality of prediction for a training set size of 90,000 samples while for the rest of the study we will use a training set of 150,000 samples and keeping in mind that the numerical resources required to reduce the error, we consider our training set as being big enough.

We can observe in Fig. 5.8, that a neural network trained with 145,000 samples, performs well and rather uniformly in the test set composed of 5,000 samples. Indeed, all the predicted potentials of the test have a PCC with the exact potential located between 99% and 100%. The histogram presented in the upper panel of the figure focuses on this narrow range and depicts an asymmetric distribution which is peaked at  $r_P = 0.997$ . The worst prediction of the test set has a correlation of  $r_P = 0.991$  with the expected disorder potential. Those two potentials (expected and predicted) are depicted in the lower parts of the figure and are labeled by the letter (*c*). Other typical examples of expected and predicted potentials that correspond to one of the best reconstruction and one located on the peak of the histogram are also depicted in the figure and labeled by the letter (*a*) and (*b*), respectively.

From this impressive quality of potential prediction, it seems possible to state that almost the whole information about the disorder is carried in the PLDOS data. Indeed, even when the PLDOS is locally very low, the small fluctuations are sufficient for the neural network to determine with a reasonable precision the underlying disorder potential. However, as shown in the left panel of Fig. 5.9, on average, the model has lower performance at the

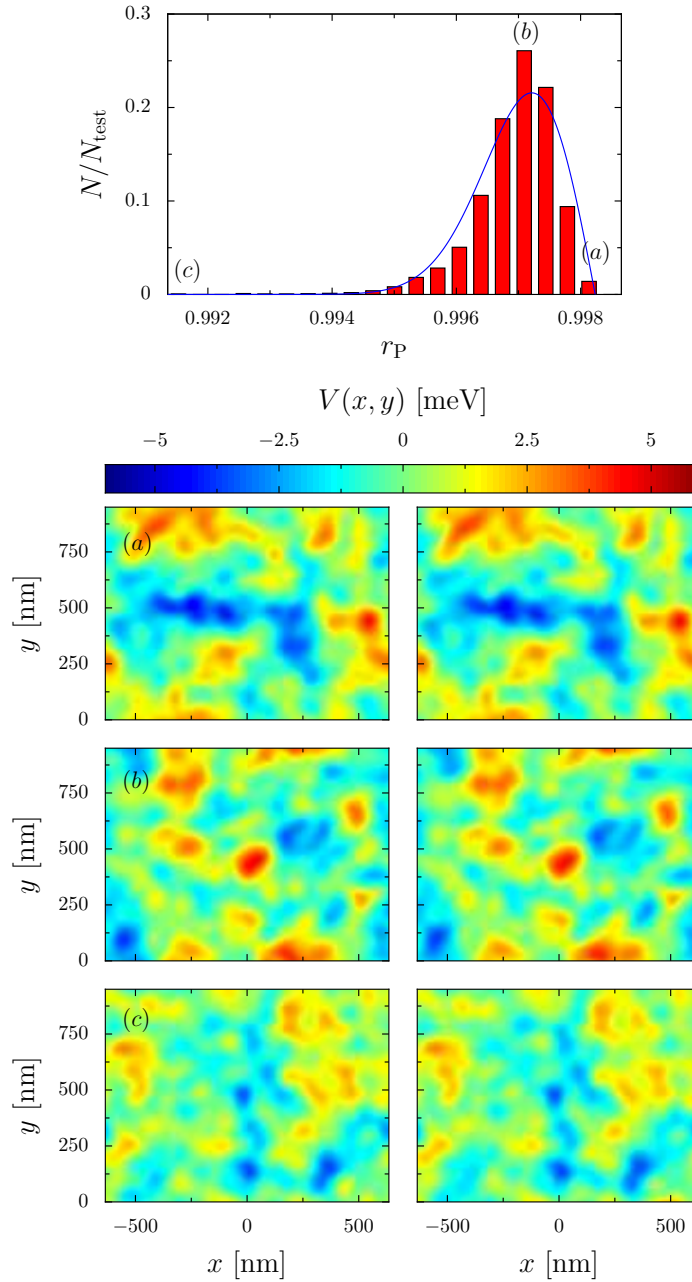


FIGURE 5.8: Upper panel: distribution of the correlation coefficient of samples from the test set between the expected potential and the one reconstructed from the predicted coefficients. The blue dashed line represents the fit of the distribution. The lower panels are examples of expected (left) and predicted (right) potentials that correspond to a correlation coefficient indicated in the upper panel.

edges of the sample. This is due to the combined effect of the absence of information beyond the edges of the sample and the weak average intensity of the PLDOS at this location. Then, we can state that the rate of information carried by the PLDOS on the disorder depends on the intensity of this latter quantum transport information. We also note that the region near the QPC



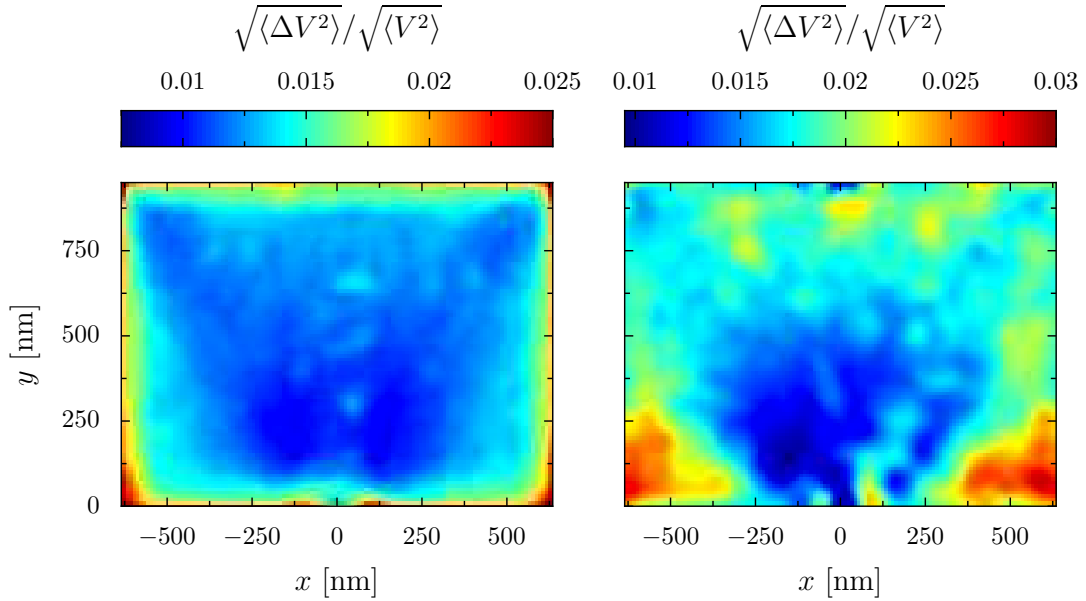


FIGURE 5.9: Root of the average squared local error in the real space potential (scaled by the root mean square of the potential used for the generation of samples), calculated by averaging over the test samples. The left and right panel corresponds to the methods of the encoder-decoder and the Fourier coefficients, respectively.

does not respect the latter statement. Indeed, the error is higher in this region, which we explain by the presence of a high intensity of the PLDOS no matter the disorder because of the proximity with the QPC.

In Fig. 5.9, we can also compare the mean spatial error on the sample averaged over the test set for the current method that uses a CED (left panel) and the method that performs a regression on the Fourier coefficients (right panel). Note that the range of the color code is not the same for the two images. The regression method that predicts the Fourier coefficients leads to about the same distribution of the error as the CED method, but with more variations in this distribution which is due to the non-local impact of the Fourier coefficients.

### 5.2.3 Variation of the prediction accuracy when varying the disorder characteristics

From the previous results, we can state that our artificial intelligence method using a convolutional encoder-decoder is able to determine almost the exact disorder potential from the PLDOS. This statement belongs only in the case where the neural network has been trained with PLDOS-potential pairs, where the potential characteristics used for the training are the same as the ones we want to determine. However, the disorder characteristics of a sample elaborated in a laboratory can have some deviations from the one expected. It is therefore interesting to determine the variation of the accuracy

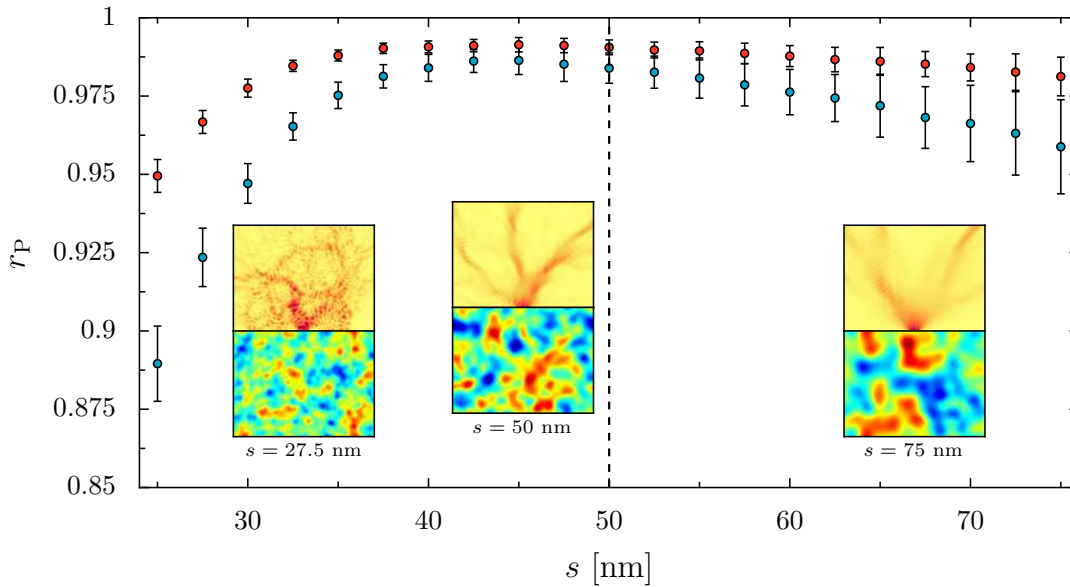


FIGURE 5.10: Average correlation evaluated for small test sets composed of 100 samples with different disorder characteristic  $s$ . The error bar depicts the standard deviation of the Pearson coefficient on a test set. The red (blue) dots depict the quality of a neural network that has been trained with samples which have a value  $s \in [40 \text{ nm}, 70 \text{ nm}]$  ( $s = 50 \text{ nm}$ ). The insets show examples of PLDOS-potential pairs for  $s = 27.5 \text{ nm}, 50 \text{ nm}, 75 \text{ nm}$

of our model with the deviation of the potential characteristics between the one used to train the model and the one used to test it. To perform such an analysis, we use the fact that the parameter that has the strongest impact on the potential landscape is the distance  $s$  between the 2DEG and the doping layer, as seen in Sec. 3.4.1. To study the dependence of the accuracy on that parameter, we created several test sets with different values of  $s$  which are each composed of 100 samples.

In Fig. 5.10, we plot the performance of two neural networks on the above described datasets. The neural network associated with the red dots corresponds to the neural network trained on samples that have a parameter  $s$  between 40 nm and 70 nm. This dataset is the one used in Sec. 4 but with the potential as output. The blue dots depict the performance of a neural network trained with samples having the parameter  $s$  equal to 50 nm. We can first observe the superiority of the neural network trained on a range of parameters  $s$  compared to the one trained with a single value of  $s$  even for that specific value  $s = 50 \text{ nm}$ . Unexpectedly, we observe that the neural network trained with the same global characteristic  $s = 50 \text{ nm}$  shows a peak of performance around  $s = 42.5 \text{ nm}$ . The other remark concerns the asymmetry of the average accuracy. Indeed, the performance of the neural network decreases faster for a decreasing  $s$  than for an increasing  $s$ . This can be explained by the increase of the pattern complexity of the potential landscape when  $s$  is decreasing due to the decrease of the correlation length of the potential. We

can also notice from the inset that the PLDOS does not behave the same way at very low  $s$  than at the other values of  $s$ . Indeed, due to the higher impact of the disorder at low  $s$ , the electrons are more localized spatially. Nevertheless, this decrease of performance at low  $s$  is not a severe issue considering that the studied heterostructures are used for their high-mobility properties meaning that the disorder is not supposed to be strong.

From this study, we can state that it is more interesting to train a neural network with samples that do not have fixed parameters. We also observed that the convolutional encoder-decoder is rather resistant to a small variation of the disorder characteristics. Even for neural networks trained with samples having the same characteristics, the loss of precision is not too important.

### 5.2.4 Image translation through the network

Besides all the previous study, it is difficult to get insights about the working procedure of the neural network. When the neural network is applied to data for which the output is unknown, we have no possibility to check the prediction of the neural network, it is therefore important that we can trust our model. However, it is very complicated to find a method to interpret the working procedure of the neural network as explained in Sec. 4.5. This is especially true when performing an image translation as we are doing. The first reason for that is the fact that the image translation performed by a neural network is not a common problem when the input and output images are very different<sup>3</sup>. Thus, in this section we restrict our study to the visualization and the interpretation of the image translation through the neural network.

---

<sup>3</sup>The image translation often performed with GANs concerns problems where the input and output images share the same pattern like transforming a satellite image to a road map.

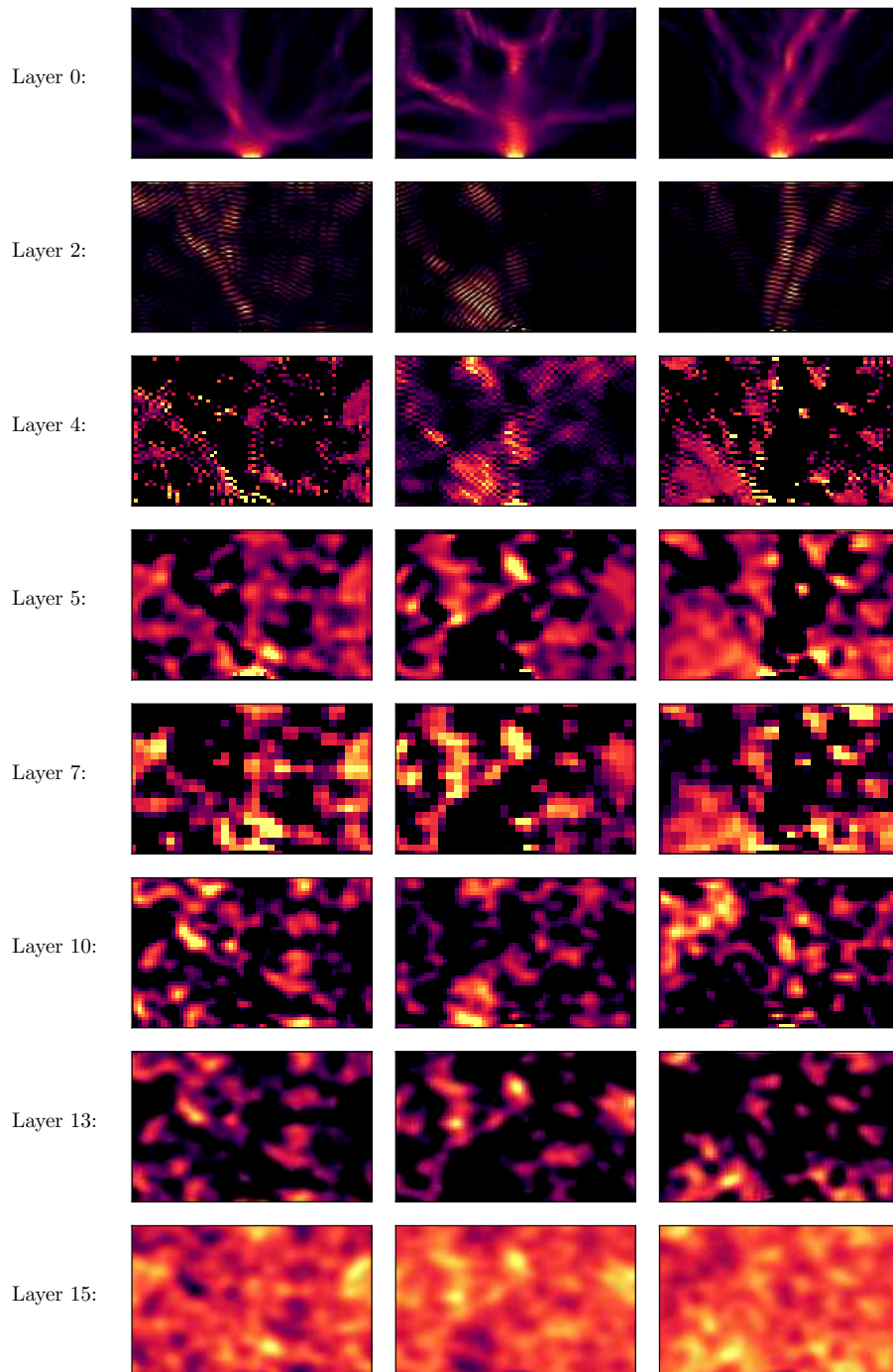


FIGURE 5.11: Example of three PLDOS images passing through the layers of the neural network. The top layer corresponds to the inputs, then each line corresponds to a layer of the neural network and depicts the feature maps having the best correlation coefficient with the expected output. Finally, the lowest line corresponds to the outputs.

Fig. 5.11 depicts this image translation for three PLDOS images of the test set. The input is represented in the first line, then each line corresponds to feature maps, of the indicated layer, which have the best PCC with the

expected output. This choice allows to depict only one feature map, even if it may not be the most relevant choice for layers close to the input. For aesthetic reasons, we did not show all the hidden layers, only the ones with significant changes are depicted. It is important to note that all the maps have their own color code. We can also remark expected general features like the variation of the resolution due to the max pooling and also the presence of a large number of pixels with the lowest value 0 in the hidden layer because of the ReLU activation function. We also note that minima and maxima can be inverted from one layer to another (layer 7 to layer 10) and considering all the feature maps, in a given layer, we can have feature maps with opposite minima and maxima. From Fig. 5.11, we also can observe a more interesting phenomenon. Indeed, we can see that at the beginning of the translation the PLDOS intensity becomes higher in areas that correspond to a minimum of the predicted potential<sup>4 5</sup>. From the figure, it is hard to have more insight on the prediction of the potential at regions where the fluctuation of the PLDOS is weak. We can just observe details arising in those regions while the image passes through the neural network. This information can arise from other feature maps. In any case, we have noticed earlier in Sec. 5.2.2, that the disorder potential is predicted with a good accuracy even when the PLDOS signal is weak.

Since we have no possibilities to have quantified results for this study, we decided not to investigate the interpretation of the prediction more than what we have already done.

### 5.3 Magnetic field

Up to now, we considered a neural network that takes one PLDOS image as input to determine the full potential disorder. However, nothing forces us to use only one PLDOS image to determine this disorder. Indeed, one can compute the PLDOS under different experimental conditions. For example, we can change the QPC opening or apply a magnetic field on the sample, resulting in different PLDOS images associated to the same disorder potential. In this section, we will use the magnetic field to obtain three different PLDOS images, one without magnetic field and two with a magnetic field that has the same magnitude but opposite signs<sup>6</sup> as depicted in Fig. 5.12. One of the objectives of this project is to observe the modification of the real space error as depicted in Fig. 5.9. Indeed, we previously stated that the average distribution of the error in the real space is correlated with the average

<sup>4</sup>From the results of Sec. 5.2.2 we can consider that the expected and predicted potential are identical for this study.

<sup>5</sup>This result arises from a more detailed study where all the feature maps of a large number of samples have been observed. It is therefore not biased by our choice of the feature map.

<sup>6</sup>In the case of the PLDOS, we can expect three different images, however that is not the case when performing an SGM experiment. Due to the time reversal symmetry, the conductance in a two-terminal setup has to be even in magnetic field, such that two SGM experiments with a magnetic field of the same magnitude but opposite sign would give the same result.

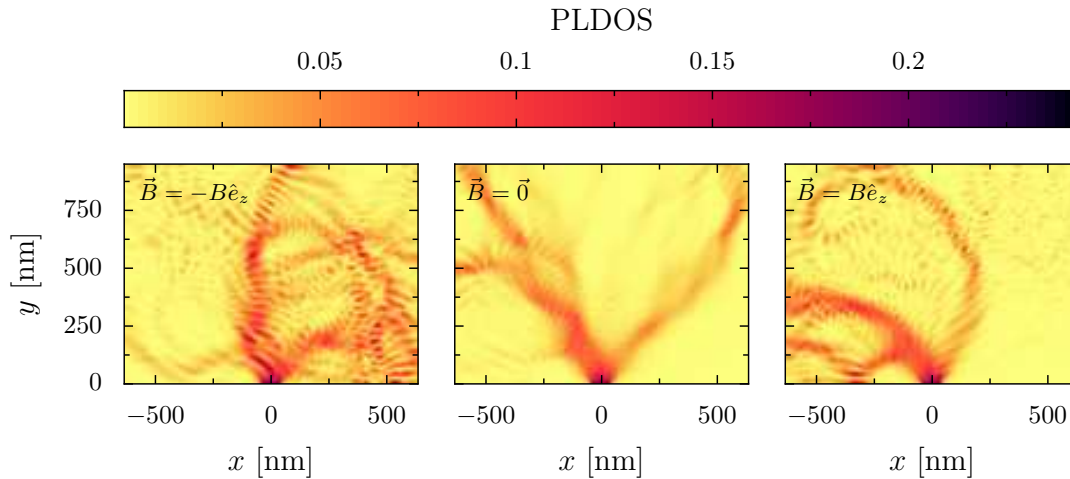


FIGURE 5.12: Example of three PLDOS images arising from the same disorder configuration but under the effect of a negative (positive) magnetic field in the left (right) panel, and no magnetic field in the middle panel.

distribution of the PLDOS. Therefore, we can expect that feeding the neural network simultaneously with the three PLDOS images described before can improve the accuracy especially on the edge of the sample.

To add the magnetic field in the quantum transport simulation we use the Peierls substitution as described in Sec. 3.2. The magnitude of the magnetic field  $B$  is chosen such that the branches of the electron flow are mainly present in the right or left region of the sample as depicted in Fig. 3.2. However, the magnetic field has to be low enough to avoid the quantum Hall regime. Thus, we fixed the cyclotron radius

$$r_c = \frac{\sqrt{2m^*E_F}}{|q|B} \quad (5.4)$$

to 420 nm which is about half the width of the simulated sample. Therewith, the magnitude of the magnetic field is equal to 0.15 T.

### 5.3.1 Three images input neural network architecture

To perform this study, we use a convolutional neural network that takes three images as input and gives one image as output. To simplify our study and allow us to compare the performance with the one-image approach, we use almost the same neural network architecture as before. The difference comes in the encoder part. Previously, we got a single encoder that takes one image as input and gives feature maps in the latent space. In this section, we have 3 encoders that take one image each, process the image and give feature maps in the latent space. All the feature maps from the 3 encoders are then merged and processed by the decoder that gives the predicted potential. The details of this architecture are depicted in Fig. 5.13. We note that this new

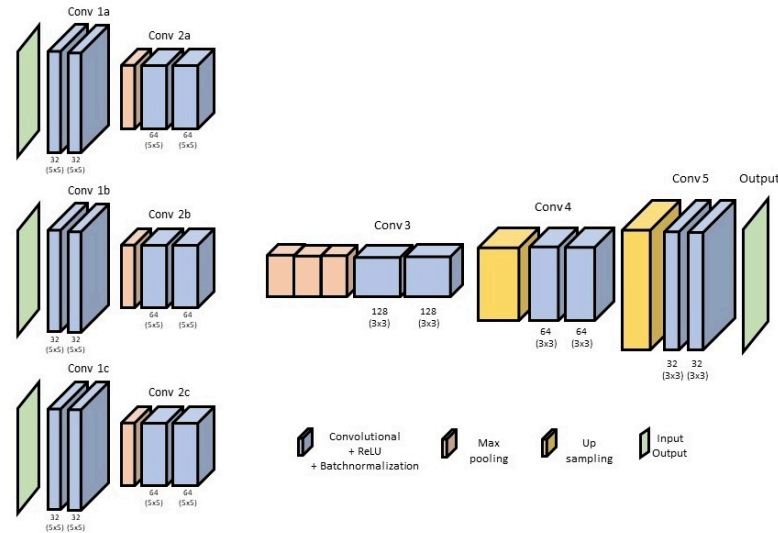


FIGURE 5.13: Architecture of the neural network with the three inputs. The end of the three branches is followed by the merging of the pooling layers. This model contains about 1.3 million parameters.

architecture does not change the training procedure. An important change is the number of parameters of the neural network. Indeed, the neural network with three inputs has about twice as many parameters as the single input neural network<sup>7</sup>.

### 5.3.2 Advantage of the multi-input neural network

Fig. 5.14 depicts the average performance over 20 trainings for the one input (red dots) and the three input (blue dots) neural network as a function of the training set size. We notice a large advantage for the 3 input neural network, especially for small training sets. While it could be tempting to use the difference of number of weights in the two models to explain such a difference, we observe in Fig. 5.14 that a single input neural network, with almost the same number of weights as the three input neural network<sup>8</sup> (green dots) does not perform better than the single neural network with less weights.

It appears that the difference of performance is not due to the larger number of parameters in the three input neural network but to the increase of information that has been used to determine the disorder potential. One can also observe the difference between the two neural networks on the error and its dependence on the position of the sample. As expected, when training single input neural networks, the distribution of the error depends on the presence or not of magnetic field as shown in the left and middle panels of Fig. 5.15. The lowest average error region seems to correspond to the

<sup>7</sup>This implies a 50% longer training time for the 3 input neural network.

<sup>8</sup>To increase the number of weights, we just increase the number of feature maps in the convolution layers.

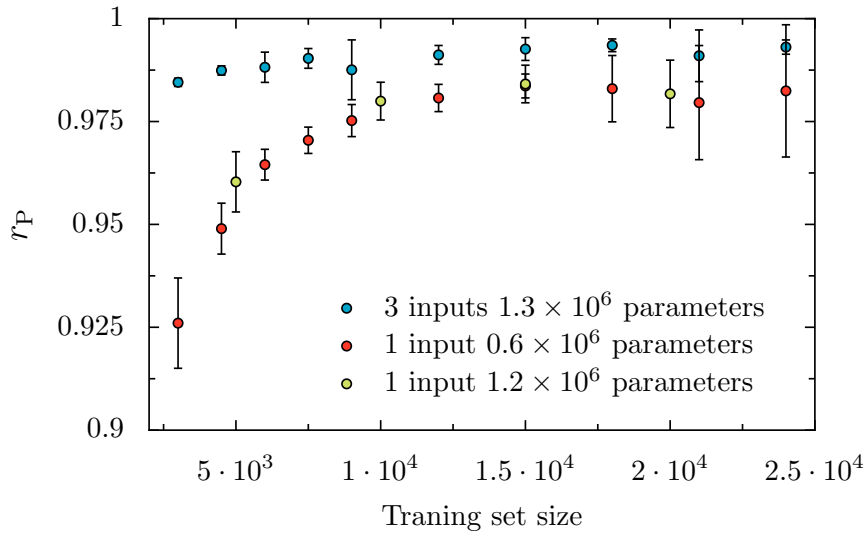


FIGURE 5.14: Evolution of the average Pearson correlation coefficient over 20 trainings with the size of the training set. Blue (red) dots represent the neural network with 3 inputs (1 input) depicted in Fig. 5.13 (Fig. 5.6). The green dots represent the performance of a single input neural network with a higher number of training parameters.

highest average PLDOS intensity. Therefore, when using a three input neural network which takes the PLDOS of the 2DEG for  $\vec{B} = -B\hat{e}_z, \vec{0}$  and  $B\hat{e}_z$ , the distribution of the error becomes more uniform (the increase of the error on the edge disappears) and leads to an increase in the accuracy of the prediction.

When applying this artificial intelligence method to a real sample, it can be interesting to collect electronic transport data in different experimental conditions to give as much information as possible to the neural network. In this study, we specifically chose the magnetic field in order to cover different regions of the space but other parameters can be used.

## 5.4 Potential prediction without branch pattern in the PLDOS

So far we have focused on one system, the 2DEG with electrons injected through a quantum point contact. The reasons why we choose this system are the large number of experiments using this setup and the presence of branches in the electron flow that depends on the disorder [101]. However, the PLDOS depends on the disorder no matter the presence of a branch pattern or not. It could then be interesting to see if the previously studied method works also for other kinds of systems. Therefore, we decided to study the simplest system that does not present a branch pattern which is the wire connected to a source and a drain on its two sides (see Fig. 5.16).



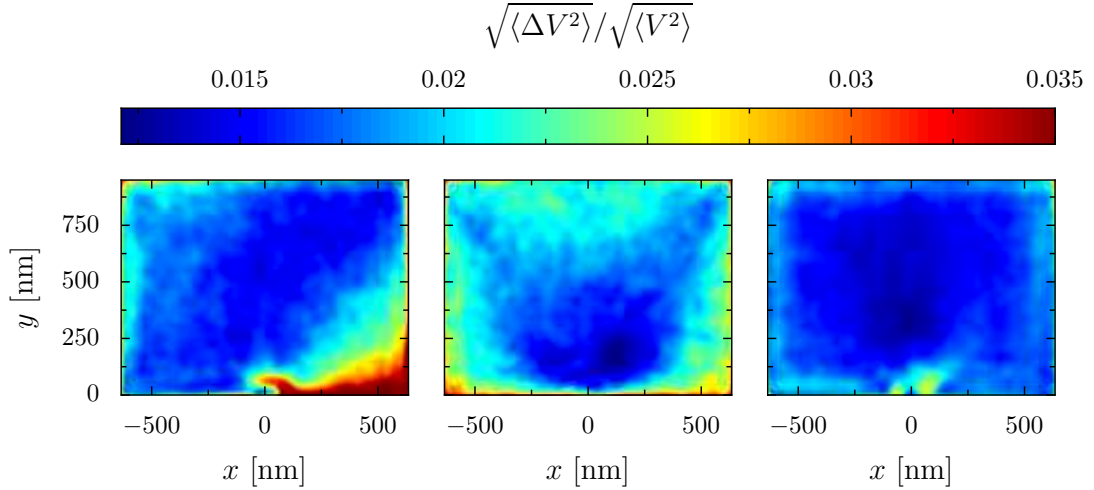


FIGURE 5.15: Root of the average squared local error in the real space potential (scaled by the root mean square of the potential used for the generation of samples), calculated from an average over the test samples. Left panel: single input neural network trained with PLDOS at  $\vec{B} = B\hat{e}_z$ , middle panel: single input neural network trained with PLDOS at  $\vec{B} = \vec{0}$  and right panel: 3 inputs neural network trained with PLDOS at  $\vec{B} = \vec{0}$  and  $\vec{B} = \pm B\hat{e}_z$ .

The gray scattering region has a dimension of  $500 \text{ nm} \times 1000 \text{ nm}$ . The distance between the 2DEG and the doping layer has been set to  $s = 35 \text{ nm}$ , which implies a small correlation length of the potential. The dopant density is fixed at  $1 \times 10^{12} \text{ cm}^{-2}$ . The Fermi energy of the 2DEG is  $2.1 \text{ meV}$ . An example of PLDOS computed for this system is depicted in the lower panel of Fig. 5.16. The dataset is composed of 36,000 samples.

For simplicity, we keep the same neural network architecture as the one depicted in Fig. 5.6. When training the neural network, we observe a higher probability for the neural network to be stuck in a non-optimized local minimum<sup>9</sup>. However, the results have about the same accuracy as the previous model of Sec. 5.2 when applied to a system with QPC. An example of the distribution of the error is depicted in Fig. 5.17. The average PCC is almost identical in both cases when averaging over several trainings.

<sup>9</sup>The phenomenon disappears when we decrease the number of filters on the convolution layers.

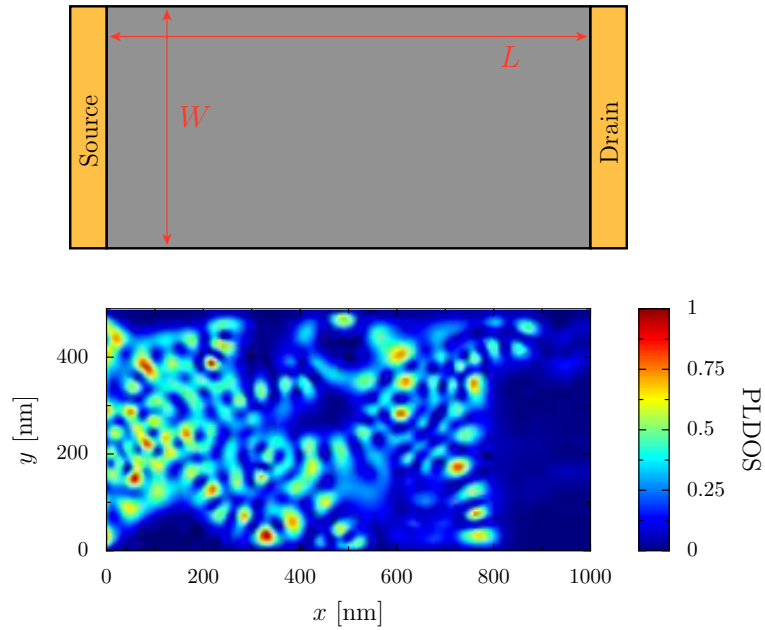


FIGURE 5.16: Top panel: system composed of the scattering region in gray and two electrodes in orange. Lower panel: example of PLDOS in such a system.

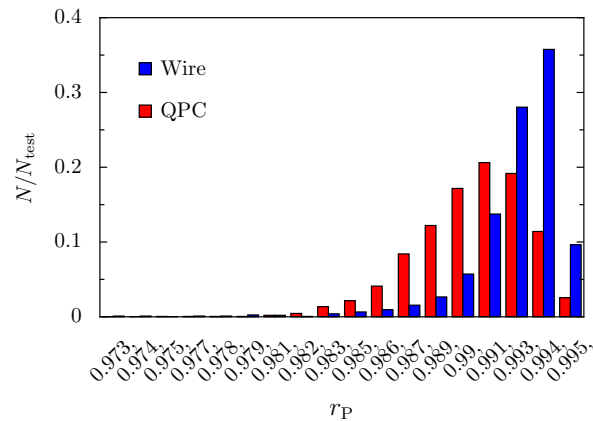


FIGURE 5.17: Pearson coefficient distribution over a test set of 2,000 samples. The blue (red) bar depicts the wire (QPC) system.

We conclude from this result that the disorder can also be extracted from transport data when there is no QPC and no branching.



## Chapter 6

# Realistic simulation of an experimental sample

We have shown in Chap. 4 and 5 that one can extract information about the disorder of a simulated heterostructure sample from its transport properties using an artificial intelligence. It becomes now essential to apply the previous result on a real sample. In order to obtain a result close enough to reality and in a reasonable amount of time, we chose a sample with convenient characteristics. The main expected characteristics of such a sample are the availability of very-low temperature SGM data in order to not be obliged to perform the conductance computation for several energy values and an SGM that can be resized with a reasonably small amount of pixels which implies a reasonable amount of conductance computations per SGM map.

In artificial intelligence, the performance of prediction of an algorithm depends on the quality of the training data. Thus, we use a new method to compute the electrostatic environment that is more realistic than before. We use a Poisson Thomas-Fermi solver (PTFS) that is developed in Grenoble in the theory group of the Pheliqs lab [116]. It allows to perform more precise simulations by taking into account the electrostatic interaction between the different charge sources. The quantum properties of the 2DEG are taken into account to compute the electrostatic potential seen by the electrons.

In this chapter, we describe the experimental sample in Sec. 6.1, followed by the description of the model used to simulate the sample in Sec. 6.2. The latter section is also dedicated to describe the approximations of the model and the comparison with the fast-to-compute model used to create a pre-training dataset in order to perform transfer learning. The last section 6.3 is dedicated to the presentation of the datasets used for the training. A part of the results present in this chapter arises from [117].

## 6.1 Presentation of the sample

### 6.1.1 Properties of the experimental sample

The chosen sample is the one used in Ref. [96], which has been measured in Louvain-La-Neuve. An electron microscopy image of the sample is shown in the left panel of Fig. 6.1. The 2DEG, located in an  $\text{Al}_{0.3}\text{Ga}_{0.7}\text{As}/\text{GaAs}$  heterostructure, has a density of  $n_s = 2.53 \times 10^{11} \text{ cm}^{-2}$  and a mobility of

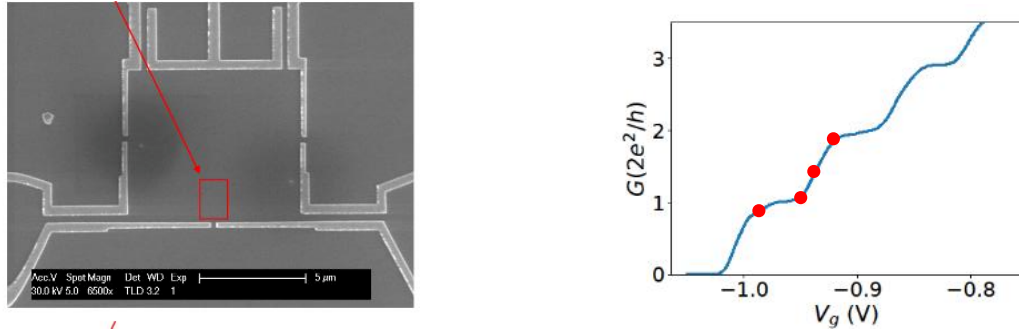


FIGURE 6.1: Left panel: Electron microscopy of the sample. The QPC corresponds to the gap between the lowest gate fingers and the red rectangle depicts the scanning zone. Right panel: Conductance of the studied sample as a function of the QPC gate voltage. The red dots depicts the value at which the SGM is performed. Data provided by B. Brun.

$\mu = 3.25 \times 10^6 \text{ cm}^2 \text{ V}^{-1} \text{ s}^{-1}$ . The 2DEG is placed 67 nm below the surface under the following stacking of layers (from 2DEG to surface of the sample):

- AlGaAs 30 nm (spacer)
- AlGaAs 15 nm (Si doped [ $4.8 \times 10^{24} \text{ m}^{-3}$ ])
- AlGaAs undoped 5 nm
- GaAs 7 nm (capping)
- Hafnium oxyde 10 nm

The Fermi wavelength that has been experimentally determined, using interference fringes, is about  $\lambda_F = 40 \text{ nm}$ , which is close to the value derived from  $n_s$ :  $\lambda_F = \sqrt{2\pi/n_s} = 49.8 \text{ nm}$ , assuming parabolic dispersion. The QPC is realized by applying a negative voltage on two metallic gates that have a width of 150 nm, an opening of 250 nm and a thickness of 105 nm. The SGM scan has been performed in a cryostat with a base temperature set to 100 mK. The metallic tip used for the SGM is placed 30 nm above the heterostructure and has a curvature radius of 50 nm. The voltage applied on the tip is maintained to  $-6 \text{ V}$  during the scan, creating a depletion radius  $r_{\text{dep}} = 60 \text{ nm}$ . A small excitation is also applied to the tuning fork at its bar resonance frequency ( $\sim 32 \text{ kHz}$ ), resulting in a vibrating tip. More details concerning the tip are presented in Ref. [118]. The scan is performed in a rectangular zone of dimension  $520 \text{ nm} \times 260 \text{ nm}$  located at 100 nm in front of the QPC and depicted by the red rectangle in the left panel of Fig. 6.1.

The conductance is measured with the four contacts lock-in technique at low frequency (77 Hz). For this sample and experimental set up, four SGM maps are measured at different QPC gate voltages that correspond to the following value of conductance:  $1.0 G_0$ ,  $1.3 G_0$ ,  $1.7 G_0$ ,  $2.0 G_0$  (red dots in right panel of Fig. 6.1). The experimental SGM maps are depicted in Fig. 6.2.

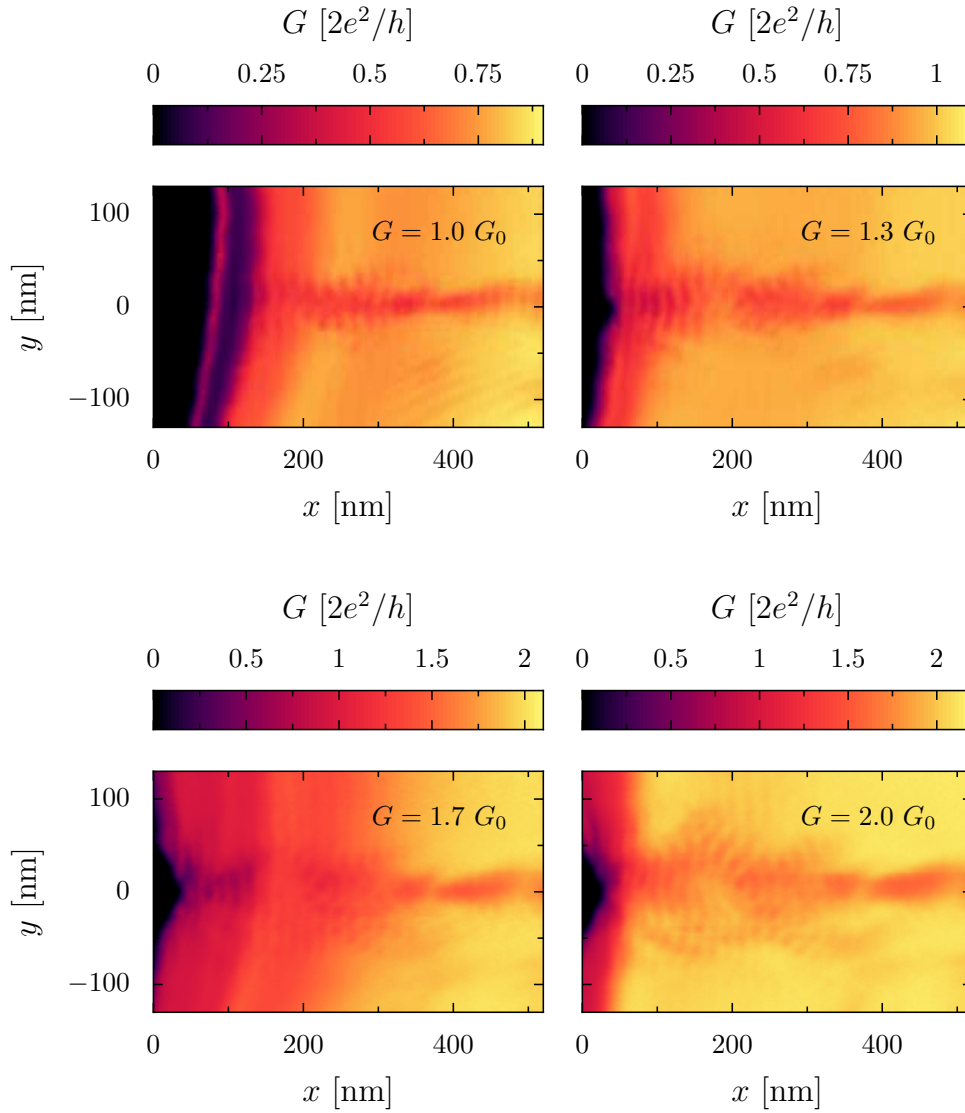


FIGURE 6.2: SGM maps experimentally measured. Note that the images are rotated by  $\pi/2$  compared to left panel of Fig. 6.1. Data provided by B. Brun.

### 6.1.2 Realistic model

To model the experimental sample described in Sec. 6.1.1, we consider the heterostructure (depicted in the left panel Fig. 6.3) as a stacking of four different layers in the vertical ( $z$ ) direction. The first one is the substrate that is represented as a block of dielectric material with a relative permittivity  $\epsilon_r = 12$ . Then, we have the 2DEG located at  $z = 0$ . The AlGaAs/GaAs stacking is modeled as another block of dielectric material ( $\epsilon_r = 12$ ) where the dopants are represented by positive charges. Their positions are randomly chosen from a Poisson distribution in a 15 nm thick region that begins 30 nm above the 2DEG. Finally, the Hafnium oxide is represented by a layer with a dielectric permittivity of  $\epsilon_r = 20$ . All the electrostatic computations are performed using the PTFS that is detailed in the next section.

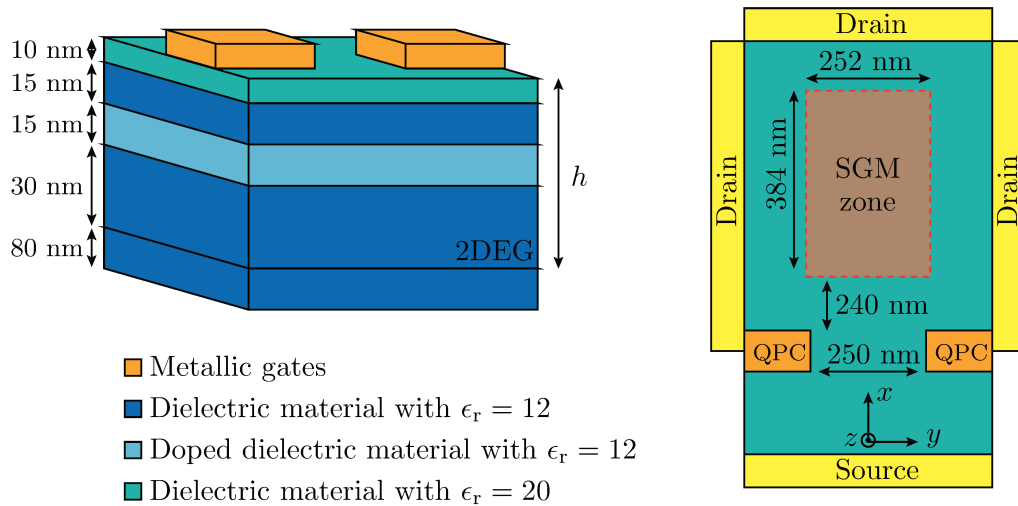


FIGURE 6.3: Sketch of the model that represents the experimental sample. The side view (left panel) highlights the layer stacking. The right panel depicts the top view of the sample and shows the location of the leads used for the quantum transport computation.

The two metallic gates that create the QPC are represented by two cuboid elements with a width of 150 nm, a thickness of 40 nm<sup>1</sup> and a length such that they cover the entire lateral width of the sample except for an opening of 250 nm in the center. The tip is represented by a metallic half sphere with a radius of 50 nm and located 30 nm above the sample surface. To perform the scan, the tip moves in a plane parallel to the one of the 2DEG. The scanning zone for the simulations is a rectangle of 252 nm  $\times$  384 nm that begins 240 nm after the QPC. The latter zone has been reduced as compared to the available experimental data since the conductance is often close to 0. This is due to the cross-talk which corresponds to the combined electrostatic effect of the tip and the QPC that closes the contact between the source and the drains when the tip is close to the QPC.

In our simulations of the quantum transport measurement, we approximate the low experimental temperature as 0 K. The leads are located as depicted in the right panel of Fig. 6.3. In order to avoid a discontinuity between the scattering region and the leads that can create reflection phenomena, we extend the edges of the sample by decreasing the potential smoothly to 0. It is also important to note that the quantum transport simulations are performed within an effective one-particle picture and do not take into account the electron-electron correlations beyond the mean field level.

<sup>1</sup>The thickness of the QPC gates does not impact the results

## 6.2 Electrostatic simulations

### 6.2.1 Presentation of the PESCADO python package

In order to have the most reliable model, we decided to use PESCADO a Poisson solver which gives more realistic simulations than those performed with the model of Sec. 4.1. This python package developed in Grenoble [116] allows to solve the Poisson equation

$$\Delta U = \frac{\varrho(U)}{\epsilon} \quad (6.1)$$

iteratively.  $U$  denotes the electric potential,  $\varrho(U)$  the charge density and  $\epsilon$  is the dielectric constant. The basic principle is to compute the electric potential  $U$  from the charge density and then compute the new charge density from the previously determined  $U$  until reaching a convergence for those two quantities. The charge density can be rewritten as  $\varrho(U) = en(\mu = eU)$ , where  $n(\mu)$  is the charge density that depends on  $\mu$ , the electrochemical potential. The electron density corresponds to the Integrated Local Density of States (ILDOS)

$$n(\mu) = \int^{\mu} \rho(E) dE \quad (6.2)$$

that depends on the local density of states  $\rho(E)$  which is obtained by solving the Schrödinger equation. However, in case of an infinite 2DEG discretized with a lattice parameter  $a$ , the density of states is constant and is equal to

$$\rho(E) = a^2 \frac{m^*}{\pi \hbar^2} \quad (6.3)$$

for positive values of  $\mu$ , otherwise the density of states is equal to 0, as depicted in Fig. 6.4. Thus, the ILDOS is equal to 0 for negative values of  $\mu$  and for positive values, we have  $n(\mu) = \rho\mu$ .

In practice, the space is discretized in sites. Depending on the local electrostatic properties, we observe three different kinds of site. The first one corresponds to a region of space where the charge value is fixed (e.g. the metallic gates) and is called Dirichlet. When we have region of space without charges ( $n(\mu) = 0$ ) like in the depleted regions of the 2DEG, we have what we call the Neumann sites. Finally, the last kind of site corresponds to region of space where PESCADO solves the Helmholtz equation. In the 2DEG, the sites are flexible and can either be Neumann or Helmholtz (depleted or carrying charges, respectively) depending on the electro-chemical potential as depicted in Fig. 6.4. For positive values of  $\mu$ , the sites carries a charge  $en(\mu)$ .

The algorithm works as follows: The electric potential  $U$  is computed from the initial charge distribution that we fixed. Then, the charge distribution is computed from the potential  $U$  determined previously. When computing this charge distribution, the flexible sites can change between Neumann or Helmholtz. The iteration is programmed to stop when all the sites have a fixed property (i.e. when two charge distribution computations do



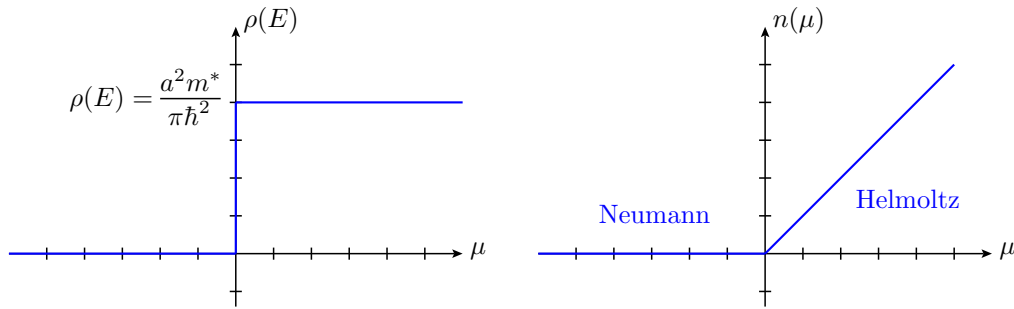


FIGURE 6.4: Right panel: evolution of the density of states as a function of the electro-chemical potential. Left panel: electron density as a function of the electro-chemical potential. For  $\mu < 0$  the site is depleted and has Neumann boundary conditions while for  $\mu > 0$ , the boundary condition becomes Helmholtz.

not change the property of any sites).

## 6.2.2 Electrostatics of the model

The electrostatic simulations are performed in three dimensions where the value of the charge and the potential are computed at each point of a pre-defined meshing. That meshing is characterized by the lattice parameter. Obviously, the accuracy of the computation increases when the lattice parameter decreases. To perform the computation, we use a lattice parameter that is not equal in the parallel and perpendicular direction, with respect to the 2DEG plane.

For simplicity the parallel lattice parameter  $a_{\parallel}$  is the same for the electrostatic simulation of PESCADO and the quantum transport computation of KWANT. Hence, this parameter has to be much smaller than  $\lambda_F$ . We also notice that this parameter has an impact on the both, PESCADO and KWANT computation time<sup>2</sup>. Therefore, we set  $a_{\parallel} = 6$  nm, which implies a difference of 2% for the conductance computation with our reference case  $a_{\parallel} = 5$  nm, as depicted in Fig. 6.5<sup>3</sup>. However, the CPU-time of a conductance calculation in the inset of Fig. 6.5 indicates that when using a parallel lattice parameter of 6 nm instead of 5 nm, we reduce the computation time in KWANT by almost a factor of 2. About the same factor of reduction is found for the PESCADO computation.

<sup>2</sup>The computation time of KWANT is more crucial than in the previous model of Sec. 4.1 since one sample does no longer need one computation for the PLDOS but several thousands computation, each corresponding to the conductance for one tip position. On the contrary, the potential is calculated only once for each disorder configuration, thanks to the approximations described in Sec. 6.2.4

<sup>3</sup>We do not consider lowering  $a_{\parallel}$  below 4 nm because of the enormous computational resources required by PESCADO for our sample size (not doable on a workstation with 32GB of memory, thus hard to perform the generation of thousands of samples in a reasonable amount of time, even with a cluster.).

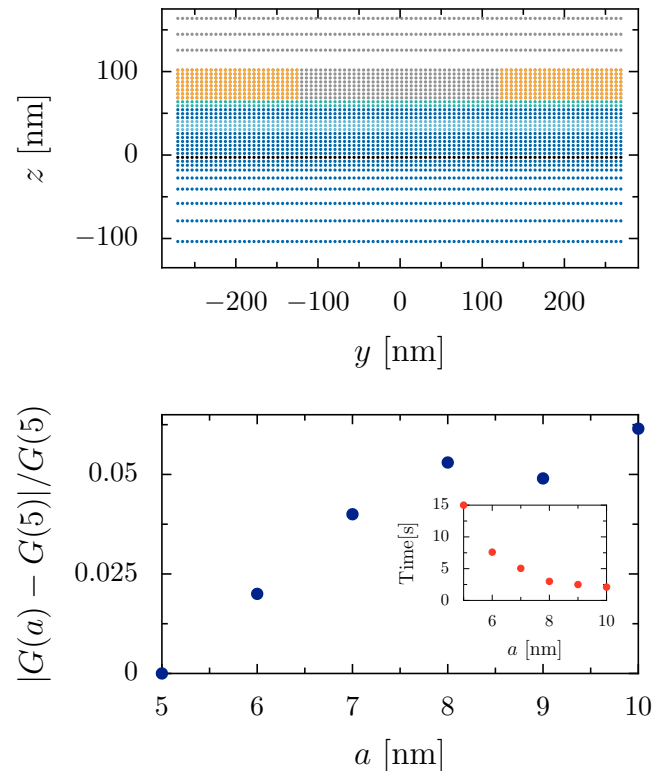


FIGURE 6.5: The upper panel depicts a slice of the sample in the  $y - z$  plane. Each point represents a site of the Poisson Solver. The color of the points is the same as in Fig. 6.3. The lower panel represents the relative variation of the conductance calculated for a sample without disorder but with QPC and tip for different values of the parallel lattice parameter. The inset depicts the time required to perform a conductance computation with KWANT on a workstation.

In the direction perpendicular to the 2DEG, we have a non-uniformity of the medium. The bottom of the sample is (i) the GaAs substrate (high thickness compared to the other layers. In the simulation this thickness is set to 125 nm in order to be converged enough), on which we have (ii) the stacking of the different layers and on top we have (iii) the vacuum or the gates (of thickness 100 nm). To take into account the different properties of the different regions, the perpendicular lattice parameter varies. In region (ii) (i.e.  $0 < z < h + h_{\text{QPC}}$ ), the lattice parameter is set to 5 nm, to reproduce as close as possible the different layer thicknesses and to allow a computation in a reasonable amount of time. The region (iii) (i.e.  $z > h + h_{\text{QPC}}$ ) is far from the 2DEG and has less impact on the potential seen by the electron gas. Thus, in this region the lattice parameter has been set to 20 nm. The region (i) (i.e.  $z < 0$ ) begins near the 2DEG and ends far from it. Thus, we decided to fix the lattice parameter as a quadratic function of  $z$ . To sum up, the lattice

parameter is defined as

$$a_{\parallel} = 6 \text{ nm}, \quad a_{\perp} = \begin{cases} 2m^2 + 3, & z < 0 \\ 5, & 0 < z < h + h_{\text{QPC}} \\ 20 & z > h + h_{\text{QPC}} \end{cases} \quad (6.4)$$

where  $m$  is an integer. A slice of the  $y - z$  plane of the sample showing the location of the sites is depicted in the upper panel of Fig. 6.5.

As presented in Sec. 6.1.2, the different elements to take into account in order to perform the electrostatic computation are the gates (The QPC and the tip are both regions of sites with Dirichlet boundary conditions.), the charges (that come from the ionized dopants) and the permittivity of the materials. Concerning the last element, the dielectric permittivity has been set following the model shown in Fig. 6.3. The dopants charges are placed in the doped region of the sample (see Fig. 6.3). Since the space is discretized for the computation, one has to fix the charge value for each cell of the doped region. To spread the charge in the cells, we use the Poisson distribution with the parameter  $\lambda$  that is equal to the average number of charges per cells<sup>4</sup>. The latter parameter is determined from the volumic density of dopants  $n_{\text{dop}}$ , with the relation  $\lambda = n_{\text{dop}} a_{\perp} a_{\parallel}^2$ , where  $a_{\perp}$  and  $a_{\parallel}$  are the lattice parameters perpendicular and parallel to the 2DEG, respectively. In our model, we consider that all the supplementary electrons of the dopants are located in the 2DEG. Thus, we obtain the following relation  $n_{\text{dop}} = n_s / d_{\text{dop}}$  where  $d_{\text{dop}}$  is the width of the doping layer, which is equal to 15 nm. Finally, we obtain a mean volumic dopant density  $n_{\text{dop}} = 1.69 \times 10^{23} \text{ m}^{-3}$ .

When adding the QPC gates on top of the heterostructure, even at zero QPC voltage the electron density of the 2DEG is not uniform anymore. To compensate this effect, we apply an offset voltage  $V_{\text{off}}$  on the gates. The value that allows to reach a uniform density is  $V_{\text{off}} = 0.162 \text{ V}$ .

Since we have to create a large amount of samples to train the neural network, we have to optimize the algorithm. For this purpose, we can reduce the number of iterations performed by PESCADO. Indeed, the algorithm stops the iteration process when all the sites stay with the same boundary conditions (i.e. depleted or not depleted) two times in a row, thus, one can start with an initial boundary condition distribution close to the expected one. This initial boundary condition can easily compute in the absence of disorder. The difference in the iteration process is depicted in Fig. 6.6. We observe that without an initial guess (first line) the number of iterations is much more important than with an initial guess (second line). On average, the initial guess allows to win 3 iterations. A second method to reduce the number of iterations is to stop them when the potential has already sufficiently converged. This implies to stop the process even when the number of sites that change boundary condition from the previous iteration  $\delta$  is not equal to 0 but still very low, as in the last iterations in Fig. 6.6. The sites that

<sup>4</sup>We also can be interested in having no disorder, which corresponds to a uniform charge density in the whole layer. In that case, all the cells carry a density of charge  $e\lambda$ .

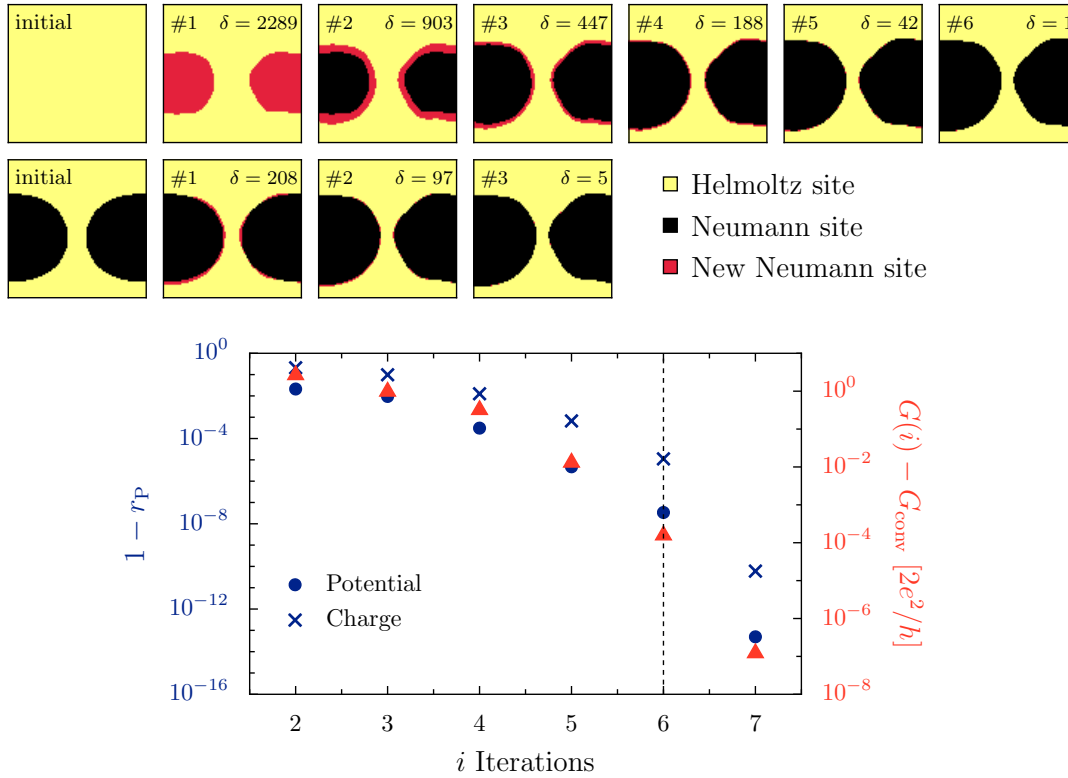


FIGURE 6.6: In the upper part, the images depict the boundary condition (black for Neumann, yellow for Helmholtz) of the sites in a region focused on the QPC. The upper line depicts all the iterations (until the last change of site boundary conditions) when we have no optimized initial guess. The sites that have changed their boundary condition from the previous iteration are highlighted by the red color and their number  $\delta$  is indicated. The second line corresponds to the iterations when having an initial guess. The lower panel depicts, in blue, the evolution of the correlation error  $1 - r_P$  of the potential and the charge distribution (left axis) compared to the converged ones, for an example of disorder configuration. In this example, we did not use the optimal initial condition. The evolution of the deviation of the conduction (right axis) between the sample at iteration  $i$  and the sample when it is completely converged is depicted in red.

are still changing are close to the depletion limit, but in practice, it just corresponds to a value of the potential that is close to  $E_F$  and not a critical change in the potential landscape. Therefore, we fixed that the iteration loop has to stop when the correlation of the potential and the charge distribution at iteration  $N$  is correlated at least at 99.99 % with the potential and the charge distribution found at iteration  $N - 1$ . In the lower panel of Fig. 6.6, we show the convergence of the charge distribution, the correlation and the deviation between the conductance measured at iteration  $N$  and the one of the sample

completely converged, for one example of disorder configuration<sup>5</sup>. We observe that the charge distribution converges less rapidly than the potential. The vertical black dashed line represents the iteration at which the algorithm fulfills the 99.99% of correlation condition. In this case, the iteration stops when the conductivity of the sample is equal to the conductivity of the converged one within  $10^{-4}G_0$ . On average, we observe that we perform one iteration less than what we would have done without this condition. This approximation leads to a decrease of the computation time by 25% on the PESCADO part, when starting from an initial guess.

When performing quantum transport simulations, the contact regions of the sample with the ideal leads are extended by 10 site lines on which the potential decays linearly to 0 in order to avoid reflection at the entrance of the leads.

With the defined geometry, and using the cluster of X. Waintal's group, composed of 23 nodes, each with 48 cores and 128 GB of RAM, we can compute in parallel 69 potentials in 2600 seconds<sup>6</sup>.

### 6.2.3 Calibration of the gate voltage

In order to perform the electrostatic computation, one has to fix the values of the gate potentials. The two gates used in this study are the ones of the QPC and the tip.

The value of the tip voltage  $V_{\text{tip}}$  is fixed to  $-6$  V for all experimental measurements (if we ignore the noise). However, due to the potentially small differences between the experiment and the simulation, we decided to set the tip voltage such that the depletion radius induced by the tip corresponds to the one of the experiment which is 60 nm. This condition on the depletion radius allows to take care of the effect of the tip on the 2DEG even if the combination of the announced tip voltage and tip distance (from the surface of the heterostructure) are slightly different from the real one. Thus, the transport properties computed from the simulated sample are as similar as possible to the ones of the experiment. In the right panel of Fig. 6.7, we plot the variation of the depletion radius as a function of the gate voltage applied to the tip which is located 30 nm above the surface of the heterostructure. The depletion radius is determined from the depleted area in the 2DEG which corresponds to the area of a unit cell multiplied by the number of sites without charge. The simulated data (red dots) are fitted by the black dashed line that corresponds to a square root function. Thus, we deduce that the depletion area is linear with the voltage of the tip. From the data shown in this figure, we obtain a depletion radius of 60 nm for  $V_{\text{tip}} = -5.77$  V which is close to the experimental value.

Concerning the QPC gate voltage  $V_g$ , it is fixed such that the dataset is composed of samples with a given range of values of unperturbed conductance. Since we want to apply our neural network to the experimental SGM

<sup>5</sup>This figure does not change much from one disorder configuration to another.

<sup>6</sup>Only 6 to 7 cores have been used in each node because 7 jobs use all the 128 GB of RAM. Calculation times have been estimated using the Pescado version dating from 18/07/2023.

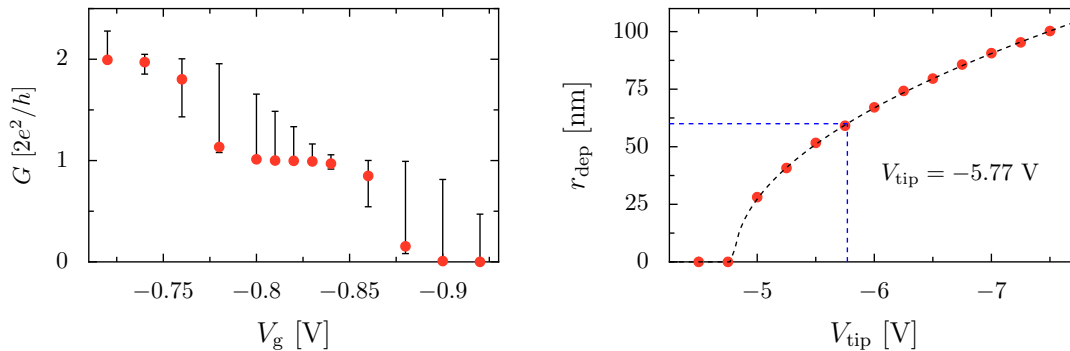


FIGURE 6.7: Left panel depicts the median (red dots) conductance over 30 samples with different disorder configurations as a function of the QPC gate voltage. The error bars depict the first and third quartile. Right panel represents the depletion radius as a function of the tip voltage. The black dashed line represents the fit which is a squared root function. The blue dashed lines represent the point where  $r_{\text{dep}} = 60$  nm.

map, we want a neural network that is trained on samples with a conductance included in the range that goes from  $1G_0$  and  $2G_0$ . However, we are interested in having a slightly higher weight of samples with conductance  $1G_0$ . Thus, we hope having a particularly good accuracy for the experiment close to  $G = 1G_0$ . From the information of the left panel of Fig. 6.7, we decide to use a QPC gate voltage of  $V_g = -0.82$  V. We note that this gate voltage does not take into account the offset voltage. Thus, the voltage set to the gate is equal to  $V_g + V_{\text{off}}$ .

#### 6.2.4 Approximation on the electrostatic potential for quantum transport simulations

The generation of simulated sample data is performed in two steps. First, we generate the electrostatic potential by using the PTFS from the Python package called PESCADO [116]. After generating a potential, we perform the electronic transport simulation with KWANT [87]. To compute one point of the SGM map, we need the potential that corresponds to a given disorder configuration (i.e. the dopant positions), the effect of the QPC and the effect of the tip located at a given position. To obtain the full SGM map, one has to compute the conductance for different tip positions (with the same disorder configuration). In other words, in a rigorous approach, we have to use for each of the tip positions the PTFS in each disorder realization. However, this is a very time consuming task, that sums up to an enormous amount of numerical resources for an entire dataset with a large number of realizations.

To tackle this problem, we have two possibilities. Method 1: We sum the potential induced by the tip  $U_{\text{tip}}$  on a sample without disorder (i.e. for uniform dopant density) to the potential of the QPC and the dopants  $U_{\text{QPC,dis}}$

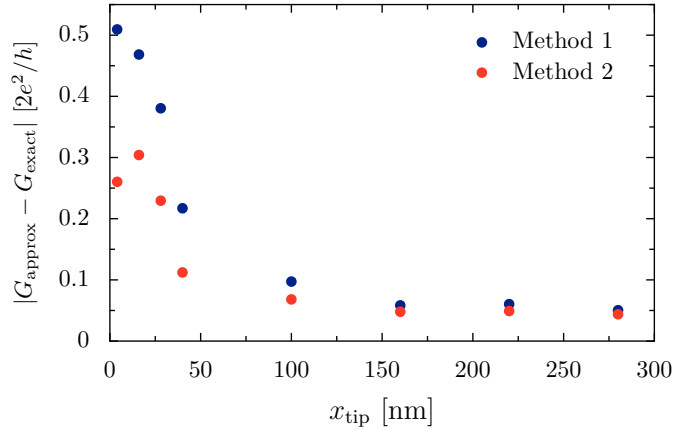


FIGURE 6.8: Absolute value of the difference between the exact method of potential computation and the approximated one as function of the tip position along the  $x$  axis. The position of the tip in the  $y$  axis is set to 0. The computation is performed for the two approximation methods and averaged over 23 configurations.

computed by PESCADO, meaning that the full potential with this method is

$$U_{\text{full}}^{(1)} = U_{\text{tip}} + U_{\text{QPC,dis}}. \quad (6.5)$$

Considering a data set of size  $N_{\text{sample}}$  and a number of tip positions  $N_{\text{tip}}$ , we have to use the PTFS  $N_{\text{sample}} + 1$  times (The +1 corresponds to the tip without disorder calculation). Considering a dataset composed of 2,650 samples, this method requires 28 hours of computational time.

For the method 2, the potential is computed in two parts. We compute the potential that includes the disorder and the QPC gates  $U_{\text{QPC,dis}}$ . In parallel, we compute the potentials of the QPC and the tip for all tip positions  $U_{\text{QPC,tip}}$  without disorder. Still without disorder, we also compute the potential  $U_{\text{QPC}}$  of the QPC alone. The complete potential that includes the disorder, the QPC and the tip is then approximated as

$$U_{\text{full}}^{(2)} = U_{\text{QPC,dis}} - U_{\text{QPC}} + U_{\text{QPC,tip}}. \quad (6.6)$$

In this approximation, the number of times that we have to use the PTFS is equal to  $N_{\text{sample}} + N_{\text{tip}} + 1$  (The +1 corresponds to the QPC without disorder calculation). Since we have 2795 tip positions which is about the number of samples in the dataset, it is a non-negligible increase of computation effort compared to method 1. However, it is still much lower than the exact method that would require to use the PTFS  $N_{\text{sample}} \times N_{\text{tip}}$  times. Considering a dataset composed of 2,650 samples, the method 2 requires 57 hours against about 9 years for the exact method. Those two methods are compared in Fig. 6.8 where we show the deviation of conductance computed using the potential with the tip within the method 1 and method 2 from the results of a full calculation without approximation. For those computations,

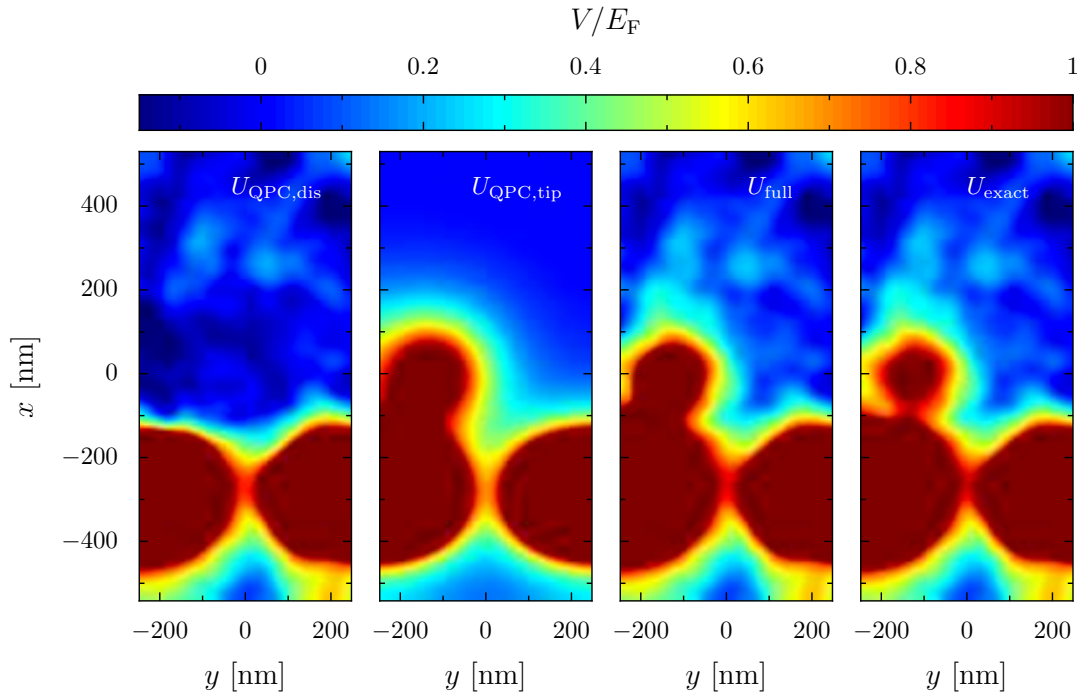


FIGURE 6.9: The two left panels are the two potential parts used to determine the potential  $U_{\text{full}}$  from the dopant configuration. The right panels depict the results of the 2<sup>nd</sup> approximation method  $U_{\text{full}}$  and the exact potential  $U_{\text{exact}}$  computed for the same disorder configuration.

we fixed  $y_{\text{tip}} = 0$  nm. Each point corresponds to a different  $x_{\text{tip}}$  position where  $x_{\text{tip}} = 0$  nm corresponds to the beginning of the scanning region. The data plotted in Fig. 6.8 correspond to an average over 23 disorder configurations. We observe that the difference between the two methods is more critical when the tip is closer to the QPC, which corresponds to the region where the coupling between the gates is most important. We note that we do not show the error bar from the statistics in the figure. This is due to the importance of the disparity of the results. In order to make a reasonable approximation for the potential calculation, we will perform the simulation using method 2.

An example of the method 2 is depicted in Fig. 6.9, where we illustrate the decomposition of the potential corresponding to Eq. 6.6. One can see small differences between the exact potential and the one computed in method 2. This figure depicts a case where the tip is close to the QPC and thus corresponds to one of the worst cases. Other examples are depicted in Fig. A.1. In these figures, the result of the approximation is very close to the exact computation.

In the following chapter, we use a verification method of a potential prediction by neural networks that consists in computing the SGM map from that potential for a given conductance  $G^{\text{target}}$  (without tip) and compare it to the experimental SGM data of that value of conductance. In this case, we have to reconstruct a potential that takes into account the disorder, the tip



and the QPC. Therefore, we have to change  $U_{\text{QPC,tip}}$ , by adjusting the QPC gate voltage to obtain the desired conductance (without tip). In this case,  $U_{\text{QPC,tip}}$  becomes  $U_{\text{QPC,tip}}(G^{\text{target}})$  which is determined by

$$U_{\text{QPC,tip}}(G^{\text{target}}) = U_{\text{QPC,tip}} + (\nu - 1)U_{\text{QPC}} \quad (6.7)$$

where the factor  $\nu$  is determined such that the conductance of  $\nu U_{\text{QPC}}$  is equal to  $G^{\text{target}}$ . Thus, it allows to compute the SGM map without using the PTFS no matter the desired conductance  $G^{\text{target}}$ .

When performing the SGM simulations from a predicted potential, we are facing a second issue. Indeed, the latter predicted potential corresponds only to a part of the full potential map<sup>7</sup>. Thus, we fix the disorder potential to 0 everywhere in the sample excepted for the scanned region where we put the predicted potential. In order to avoid the discontinuity of the potential, at the edges of the predicted potential we add 30 nm wide strips with a linear decay to reach 0 fluctuations. It is important to note that this technique assumes that no important impurities are present in the electron flux between the QPC and the scanning region.

## 6.2.5 Advantage of the PESCADO computation

Considering the significant increase of computation time of the potential using the PTFS, one has to test the improvement of the quality of the simulation by using PESCADO compared to a much faster-to-compute model. The latter model defined the full potential

$$U_{\text{full}}^{\text{f}} = U_{\text{dis}}^{\text{f}} + U_{\text{QPC}}^{\text{f}} + U_{\text{tip}}^{\text{f}} \quad (6.8)$$

as a simple sum of the different elements.  $U_{\text{dis}}^{\text{f}}$  corresponds to the disorder potential implemented using the Fourier transform as explained in Sec. 3.4.1, with the parameters of the current system.  $U_{\text{QPC}}^{\text{f}}$  is the potential of the gates and is implemented using Eq. 3.49. Finally,  $U_{\text{tip}}^{\text{f}}$  is the potential of the tip described in Eq. 3.50. The parameters  $A_{\text{tip}}$  and  $A_{\text{QPC}}$  are defined using the experimental information of the depletion radius of the tip and the conductance of the sample (without tip), respectively. However, the decay length of the tip  $L_{\text{t}}$ , which is a parameter of the tip potential of Eq. 3.50, is one other unknown parameter that is hard to determine. As an approximation, we fix this parameter equal to the distance tip-2DEG. In Fig. 6.10, we show the difference between the potential computed from Eq. 6.8 and the one computed from Eq. 6.6. In the left part of the figure, we show the potential of the QPC and the tip obtained with the two methods for two different tip positions (Each line corresponds to one of the tip positions) in absence of disorder<sup>8</sup>. We observe that the potential of the gate computed by PESCADO

<sup>7</sup>The neural network is trained to determine the potential only in the region where the tip scanned the sample

<sup>8</sup>In the absence of disorder Eq. 6.6 corresponds to  $U_{\text{full}}^{(2)} = U_{\text{QPC,tip}}$

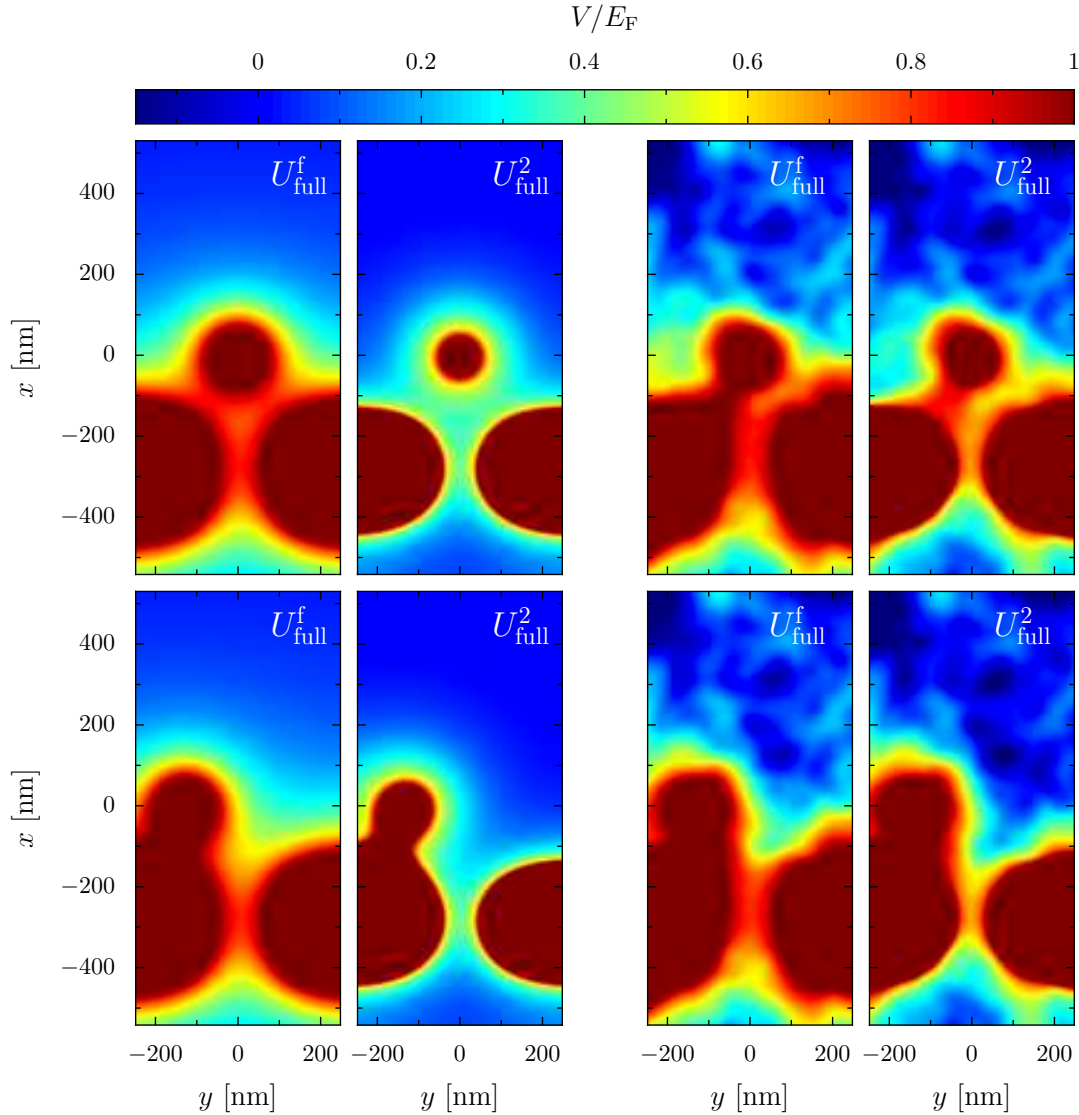


FIGURE 6.10: Example of potentials used for the SGM. The two lines represent different tip positions. The left part depicts the no disorder case while the right part shows the potential for randomly chosen disorder configuration. The two parts are composed of two columns where the left image corresponds to the fast-to-compute method while the right columns depict the potential computed with the approximative method 2.

decreases on a smaller length scale than the simple superposition of the potentials. This is due to the screening of the electrons that is taken into account only with PESCADO. Thus, when the tip is close to the QPC, the potential of the tip is more prominent with the fast-to-compute method. However, we have to be careful about this comparison because the parameter  $L_{\text{tip}}$  is approximately set. In the right part, the same tip positions are depicted, but we

add a randomly generated disorder configuration<sup>9</sup>. We note that in the presence of disorder the two methods give less different results. Nevertheless, the results are more realistic using the method 2.

### 6.3 Description of the dataset

In the previous section, we have shown that the method used to create the potential and quantum transport computation considerably increases the amount of time required for the creation of the training data. Thus, it is no longer possible to create a dataset that contains the same order of magnitude of samples than for the model of Sec. 4.1. However, to not decrease too much the accuracy of the neural network, we use transfer learning from a neural network trained on a dataset that is much faster to create. However, in order to be efficient, this dataset has to be such that the neural network solves a problem close to the one at hand.

In our case, this large dataset is composed of potentials computed as in Sec. 3.4.1 with the QPC gate implemented from Eq. 3.49. All the parameters are chosen such that they fit the experimental data. The gate voltage is set in order to have samples with a conductance that is between  $0.75G_0$  and  $1.25G_0$ . Since the disorder of Sec. 3.4.1 considers only the delta doping, we set the delta doping layer at the center of the doping layer of the real experiment, which corresponds to  $z = 42.5 \text{ nm}$ <sup>10</sup>. The density of ionized dopants in this layer is exactly the same as the electron density measured experimentally in the 2DEG. Using this potential generation method, we gain about a factor 50 in the time required for producing potentials.

Concerning the quantum transport information, we use the PLDOS because it is obtained in a single computation for the whole scan area while it still has a qualitative similarity with the SGM (as shown in Fig. 3.9). Then, the computation is about  $N_{\text{tip}}$  times faster than the one of the SGM. To perform this simulation, we use a lattice parameter of 6 nm. The source and the drain are placed as depicted in Fig. 6.3.

At the end we obtain a dataset for the pre-training, composed of 54,000 PLDOS-potential pairs, that is doubled using the data augmentation described in Sec. 4.2.

The realistic SGM dataset is composed of much less samples. Indeed, due to the significant difference in CPU-time required, we have been able to generate only 2,650 SGM-potential pairs in a reasonable amount of time. One example of this SGM-potential pair is depicted in Fig. 6.11. As for the pre-training dataset, we perform the data augmentation that allows to double the number of samples. Since the QPC gate voltage has been fixed, the

<sup>9</sup>We note that it is not possible to have the same disorder configuration with the two methods because for one method we select Fourier coefficients and for the other the dopant positions. For the comparison, we compute with PESCADO the potential of a sample without gates and use it as  $U'_{\text{dis}}$ .

<sup>10</sup>We note that this distance between the doping layer and the 2DEG corresponds to a correlation length of the potential of about 70 nm

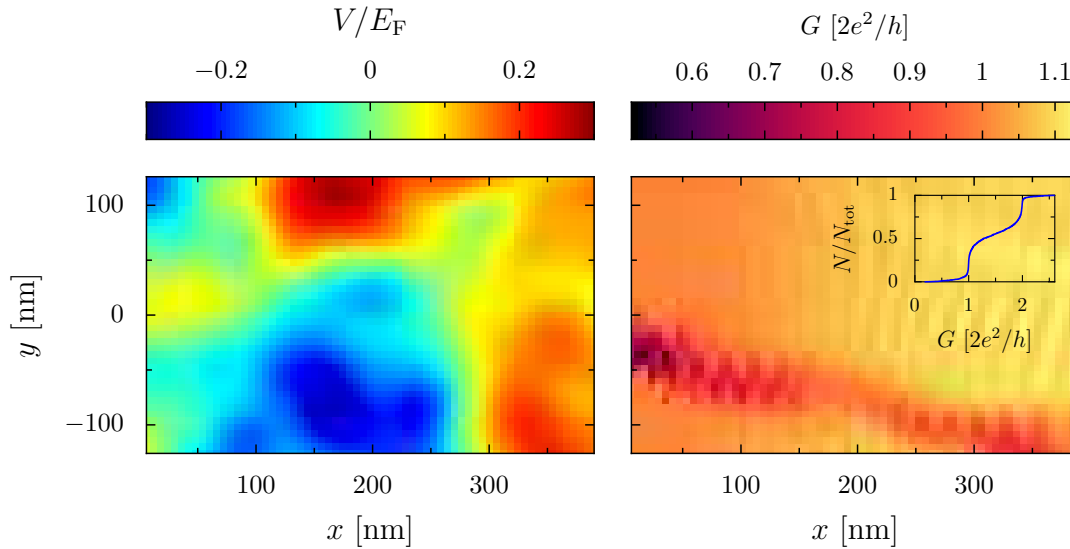


FIGURE 6.11: Example of a pair of training data. The SGM (input) is depicted in the right panel while the potential (output) is represented in the left panel. The inset represents the cumulative distribution of transmissions for samples of the dataset.

unperturbed conductance of the sample varies due to the disorder. The cumulative distribution of the conductances of the samples that compose the dataset is depicted in the inset of Fig. 6.11. We note that in Sec. 5.2.3, we have shown that varying a parameter in the dataset can lead to an increase in the neural network performance. In our case this parameter is the unperturbed conductance of the sample<sup>11</sup>.

<sup>11</sup>Since the scanning size is not very much larger than the correlation length, we also have small fluctuations in the disorder potential amplitude and correlation length from one sample to another.



## Chapter 7

# Disorder potential prediction for an experimental sample

Using the simulated experimental-like samples of the previous chapter, one can create and train a neural network to determine the disorder potential from an SGM image. In this chapter, we apply our method on the real sample of Ref. [96] and predict its disorder landscape with the best possible accuracy. However, it is not possible to evaluate directly the reliability of the prediction since the real disorder is not accessible. Thus, we tackle the two following challenges: creating a neural network with good performance and to evaluate indirectly its performance.

Sec. 7.1 is dedicated to the presentation of the neural network architecture. The performance of the neural network on the test set data is described in Sec. 7.2. The similarity between the simulated SGM data and the real ones is increased using a method described in Sec. 7.3. The proposed indirect prediction methods are discussed in Sec. 7.4. Finally, the disorder potential predicted by the neural network from the experimental SGM is shown in Sec. 7.5. A part of the results present in this chapter arises from [117].

## 7.1 Neural network architecture

The data described in Chap. 6 are used to train an artificial neural network that determines the potential landscape (output) from the SGM-response (input). However, in view of the large disorder configuration space, a dataset that contains only 2650 samples can be considered as small. To increase the performance of our neural network, we use the method of transfer learning. For this purpose, we use the large dataset composed of PLDOS-potential pairs described in Sec. 6.3. Thus, the neural network is already trained to perform a task close to the wanted one and the main difference is supposed to be located in the encoder part due to the difference of input only.

Since the data are 2D arrays (for the input and the output), we use a neural network with the same kind of convolutional encoder-decoder structure as the one described in Sec. 5.3. The architecture has been determined with a grid search evaluated through an 8-fold cross-validation over 2450 samples. Before directly training on the dataset composed of SGM-potential pairs, we first trained the neural network on the large dataset composed of the 50,000 PLDOS-potential pairs. For each neural network architecture, we performed

Rank	Number of filters	Kernel size	Batch	BatchNorm	Frozen layers	$\langle r_P \rangle$	$\sigma_{r_P}$
1	(64, 128, 256)	(3, 3, 3)	4	True	Alm. all	0.9213	0.0043
2	(64, 128, 256)	(3, 3, 3)	4	False	Half	0.9198	0.0037
3	(64, 128, 256)	(3, 3, 3)	8	True	Alm. all	0.9198	0.0038
4	(64, 128, 256)	(3, 3, 3)	4	False	None	0.9196	0.0042
10	(64, 128, 256)	(5, 3, 3)	8	False	Half	0.9121	0.0040
20	(32, 64, 128)	(3, 3, 3)	16	False	None	0.9196	0.0042

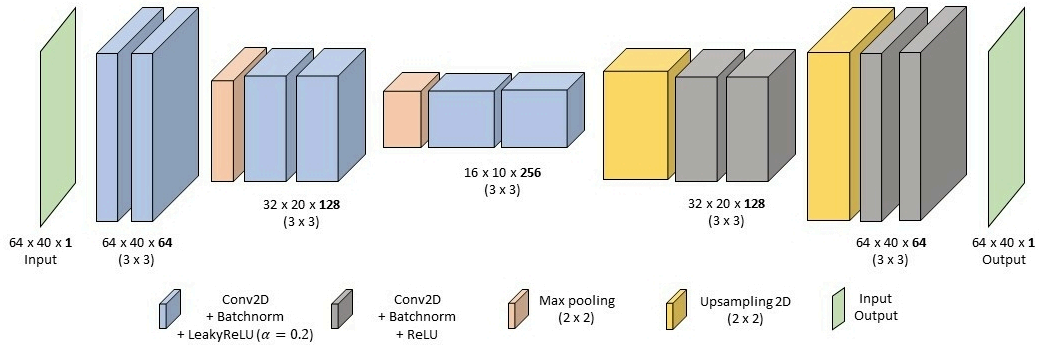


FIGURE 7.1: Top of the figure: List of different configurations of hyperparameters evaluated with 8-fold cross-validation, including the four best ones. The size of the kernels and the number of filters are represented by 3 numbers, which correspond to the three convolution blocks, respectively. The number of frozen layers during the second training is given in the sixth columns. Half means that the second half of the neural network is frozen (i.e. the decoder) and almost all (Alm. all) means that almost all the layers are frozen except the first convolution block. The best neural network architecture is represented on the bottom of the figure. Each box represents a layer and the meaning of the color of the box is depicted directly in the figure. This model contains about 1.7 million parameters.

such a pre-training and then perform the 8 trainings on SGM-potential data. The different tested hyperparameters with the cross-validation are (i) number of filters, (ii) kernel size, (iii) utilization of batchnormalization, (iv) presence of dropout layer, (v) number of trainable layers after the pre-training and (vi) the batch size. This grid search led to the evaluation of several tens of models.

From the results of the grid search, we observe that the six best architectures are very close. The difference between those models is present in the presence of Batchnormalization layers and the number of frozen layers for the transfer learning. Thus, the choice of the neural network among these six best architectures is quite arbitrary considering the statistic error. It is interesting to note that the number of frozen layers has almost no impact on the

final average performance. The retained model from the grid search corresponds to the 1<sup>st</sup> model of the table of Fig. 7.1. The details of the architecture are given in the lower panel. The selected optimizer is *Adam* and the loss function is the *binary cross-entropy*. We note that for the training we use the following callbacks: *Reducing learning rate on plateau*, *Early stopping* and also the *ModelCheckpoint* to save only the model that corresponds to the epoch where the latter model has performed the best on the validation data.

When comparing the performance of the described model with the performance of the same model but with an adversarial training (see Sec. 2.4), we do not observe a significant difference. However, we suspect training issues from the discriminator. Indeed, even if the results of the discriminator seem to be correlated with the mean squared error, the output changes in a very small range. Therefore, we decide to use a simple encoder-decoder.

## 7.2 Performance of the neural network

### 7.2.1 Pre-training

We call pre-training, the training with the large PLDOS dataset before performing the transfer learning with SGM data. The aim of this preliminary step is to obtain a neural network able to predict the potential landscape from local quantum properties. Thus, when pre-training the neural network, we do not aim to reach the best possible accuracy. However, we have to be careful to not underestimate this step. Indeed, one has to train over a sufficiently large number of epochs in order to reach the convergence. We also observed that the size of the pre-training set has no important impact beyond a few thousands of samples.

In order to quantify the accuracy of the models, we compute the average PCC<sup>1</sup>  $\bar{c}_{ep}$  between the expected and predicted disorder potential over all samples of the test set.

When performing the pre-training with the 50,000 PLDOS-potential pairs with data augmentation, the performance of the model is above 99 % when the input consists of the PLDOS data. This performance is consistent with the results presented in Sec. 5.2.2.

### 7.2.2 Performance on simulated SGM data

In this section the models are tested over 200 SGM-potential samples that have not been used during the cross-validation. We note that without the transfer learning method, the training of the neural network does not converge with a training set containing less than 2,000 samples (or 1,000 with data augmentation).

---

<sup>1</sup>In this chapter, we denote the PCC between two images  $F^{(i)}$  and  $F^{(j)}$  by  $c_{ij}$ . In particular, when computing the average PCC between expected and predicted disorder over a test set, we use the symbol  $\bar{c}_{ep}$ .



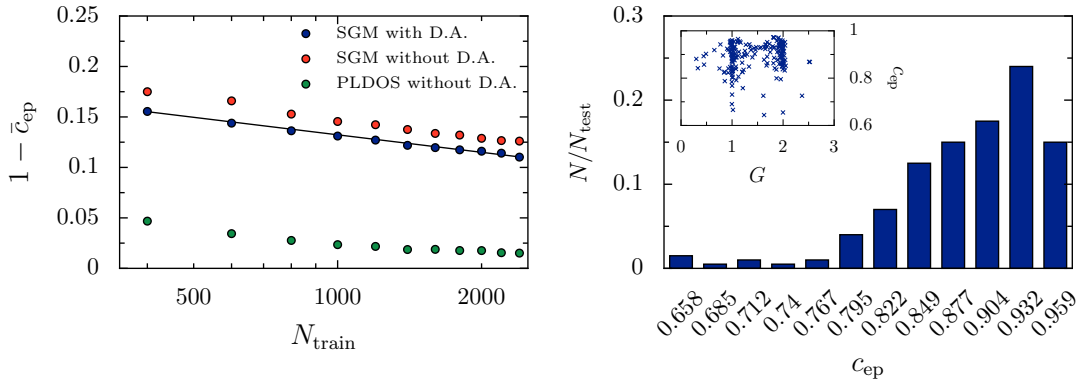


FIGURE 7.2: Left panel: evolution of the average correlation factor (over 30 trainings with a different splitting between train and test set) with the training set size  $N_{train}$ . Blue (red) dots represent the cases where the neural network is fed with SGM as input and with transfer learning and with (without) data augmentation. The black solid line corresponds to algorithmic fit with the coefficient  $-0.305$ . The green dots correspond to the case where the PLDOS is used as input in both training and testing. In the three cases, the architecture of the neural networks is the same. Right panel: distribution of the correlation predicted on the test set in the case of training with SGM and data augmentation for  $N_{train} = 2450$ . The inset represents the Pearson correlation coefficient as a function of the conductance (without tip) for all the 200 samples of the test set.

The left panel of Fig. 7.2 depicts the average PCC between the expected and the predicted potential over the test set  $\bar{c}_{ep}$  as a function of the number of training samples. Focusing only on the blue dots that correspond to a neural network trained with data augmentation and with SGM as input, we can observe the slow improvement of the accuracy of the prediction with training set size due to the logarithmic evolution<sup>2</sup>. Even if we consider a large training set of 10,000 samples (which implies a large amount of time to create such a training set), we cannot expect to reach a significantly better accuracy than with the current training set composed of about 2,500 samples. It is therefore not useful to waste numerical resources to create a huge dataset. However, we can still perform data augmentation that will not have a huge impact on the performance (especially when the dataset is already important) but it is a costless operation. When comparing the training with (blue dots) and without (red dots) the data augmentation in the left panel of Fig. 7.2, we observe that the accuracy of the neural network for a given training set size without the data augmentation is the same as in the case where only half of the data has been used for the training but performing the data augmentation. Thus, the proposed data augmentation that artificially doubles the number of samples is as efficient as creating a dataset with twice the number of samples.

<sup>2</sup>We consider the logarithmic evolution of the precision in a range that goes to reachable dataset sizes and not infinity.

Besides computing the SGM response of the 2650 disorder potentials simulated with the method of Chap. 6, we have also computed their PLDOS. Therefore, one can compare, in Fig. 7.2, the evolution of the accuracy with the training set size when using the SGM (blue dots) and the PLDOS (green dots) as input in training and testing. In the presented data, the neural network with SGM is pre-trained on a large PLDOS dataset while it is not the case for the neural network trained directly with PLDOS. It clearly appears that unveiling the potential from the PLDOS is an easier task than from the SGM. This phenomenon could be induced by the loss of locality of the signal within the SGM technique that is performed with a tip that has a large depletion radius.

For the case of SGM input with data augmentation, the distribution of the PCC between the expected potential and the real one  $c_{ep}$  is depicted in the right panel of Fig. 7.2 where we can observe an asymmetric distribution, similar to the one presented in Chap. 5, with a mean correlation  $\bar{c}_{ep} = 0.90$ . The asymmetric distribution indicates that a poor prediction accuracy is a rare event. The inset of Fig. 7.2 shows a plot of the  $c_{ep}$  versus the unperturbed conductance of the sample  $G$ . It appears that the accuracy of our model is independent of  $G$ .

In the following sections, the starting point will be the use of SGM as input with pre-training and data augmentation.

### 7.3 Smoothing of simulated SGM response to approach experimental data

During the scanning of the sample with the tip, a small excitation is applied to the tuning fork, resulting in a vibrating tip. Consequently, the depletion spot at the 2DEG level experiences a fast "breathing" motion, and the experimental SGM image in the end is a convolution of the different depletion spot radii. This effect is at the origin of a smoothing of the conductance fringes that we can observe by comparing a typical simulated SGM-response without tip vibration (Fig. 6.11) and the experimental SGM maps (Fig. 6.2). This could account for the blurring of the experimental maps requiring the application of a Gaussian filter on the simulated SGM-response to reproduce the data<sup>3</sup>.

In order to take this issue into account, we train our neural network with a new additional data augmentation which works as follows: Before feeding the neural network with the simulated data to train it, we apply a Gaussian filter which corresponds to the convolution of the SGM image with a Gaussian bump of width  $\sigma$  that is randomly chosen from a normal distribution with a width  $\eta_\sigma$ . The width of the Gaussian filter is different for all the samples and changes for each epoch.

<sup>3</sup>The main source of difference between the simulation and the experiment in the SGM-response is the vibrating tip effect. To a lesser extent, the non-zero temperature, during the experimental measurements, is also responsible for the difference between the experiments and the simulations.

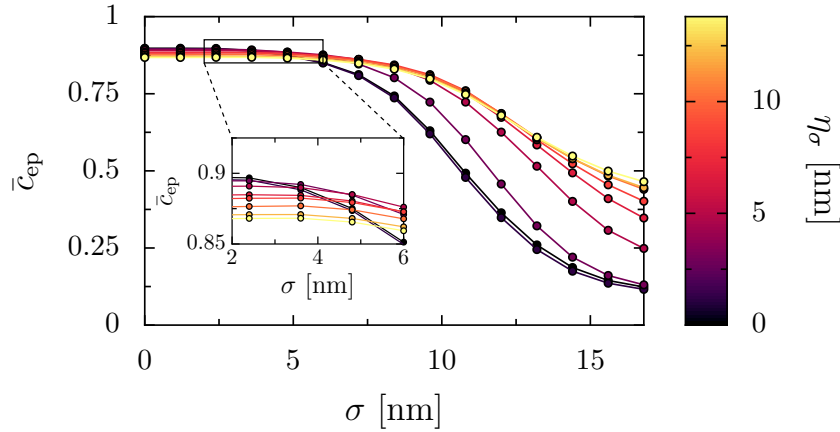


FIGURE 7.3: Average correlation over the test set and over 40 trained models of the predicted disorders with the exact one as a function of the width  $\sigma$  of the Gaussian filter applied on the test set. Each color represents a blurring data augmentation with a different  $\eta_\sigma$  that is given by the colorcode. The data correspond to the dots and the lines are just guides to the eye. The inset represents a zoom of the main panel for low  $\sigma$  values.

While the application of a Gaussian filter on the simulated data allows to get closer to the appearance of the experimental data, it seems important to evaluate the predictive power of the neural network on the simulated test set that has also been blurred with a Gaussian filter  $\sigma$ . Fig. 7.3 represents the evolution of the prediction performance with the parameter  $\sigma$ . The latter evaluations have been performed by averaging the PCC between the expected and predicted potential  $\bar{c}_{ep}$  over all the samples of the test set. Due to the statistic variations, we perform another average over the performance of 40 trained models. This procedure has been repeated for different values of  $\eta_\sigma$  that corresponds to the colorcode of the data. We observe that the precision at low  $\sigma$  value decreases with increasing value of  $\eta_\sigma$ . However, the drop of precision appears for larger value of  $\sigma$  when using a data augmentation with a large value of  $\eta_\sigma$ . This dropping takes place for values of width  $5.5 \text{ nm} < \sigma < 10 \text{ nm}$  depending on the value of  $\eta_\sigma$ . Even if there is no obvious possibility to estimate the value of  $\sigma$  that fits the best with the experimental data, we observe in Fig. B.1 of App. B that  $\sigma \approx 6 \text{ nm}$  approximates the vibrating tip effect.

## 7.4 Evaluation of prediction quality without the expected potential

### 7.4.1 Quantitative method

When applying the neural network to an experimental SGM map to determine the disorder of a semi-conductor heterostructure, there is no possibility

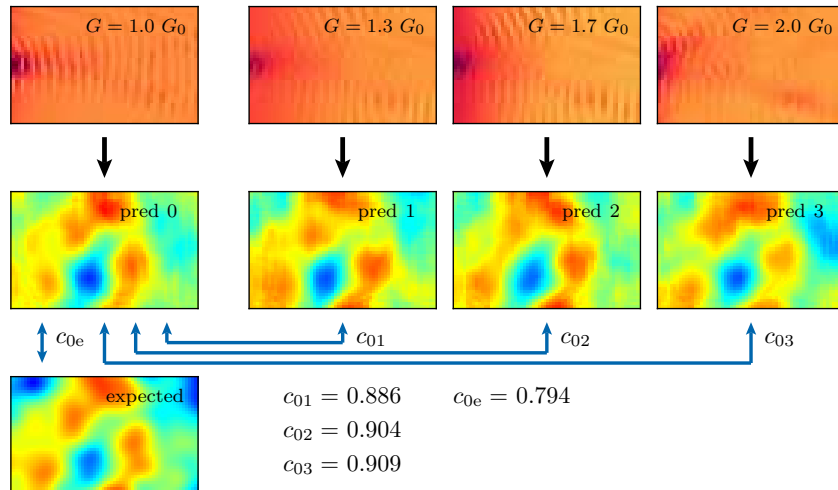


FIGURE 7.4: The four upper panels depict simulated SGM images of a sample with fixed disorder but under different gate voltages with the indicated unperturbed conductances. The black arrows represent the passage through the neural network. The four panels below the SGM images correspond to the potential predictions of the neural network. The lower image is the exact disorder potential that corresponds to all four SGM images. For the example  $i = 0$ , the blue arrows link images among which are computed the Pearson correlation coefficients  $c_{ij}$  and  $c_{ie}$ .

to compare the prediction to the real disorder. To get an estimate of the prediction quality, we propose an indirect method to determine the reliability of the predicted disorder.

The indirect evaluation consists in the comparison of the predictions from several SGM scans that are obtained from the same sample under different experimental conditions. In our case, the varying parameter is the gate voltage of the QPC, with four different values that are available. The sample disorder, and thus the expected prediction from the SGM maps, should remain the same if the sample stayed in the cryostat at low temperature during the whole series of measurements. It is then possible to compute this PCC  $c_{ij}$  between two predictions  $i$  and  $j$ , as depicted in Fig. 7.4 for the example  $i = 0$  and  $j = \{1, 2, 3\}$ . We denote  $\bar{c}_i = 1/3 \sum_{j \neq i} c_{ij}$ , the average correlation of the prediction  $i$  with the others that can be used as an indicator for the quality of the predictions. When dealing with simulated data, we can also compute the correlation coefficient  $c_i^e$  between the predicted and the expected potential. Therefore, based on simulated data, we establish a link between  $c_{ie}$  and  $\bar{c}_i$ . An example of this method is depicted in Fig. 7.4 for  $i = 0$ .

Due to the randomness of the initialization of a neural network, one can observe that neural networks with the same architecture and trained with the same dataset can give slightly different results. Therefore, we have to choose a set of predictions (composed of the 4 potentials determined by a

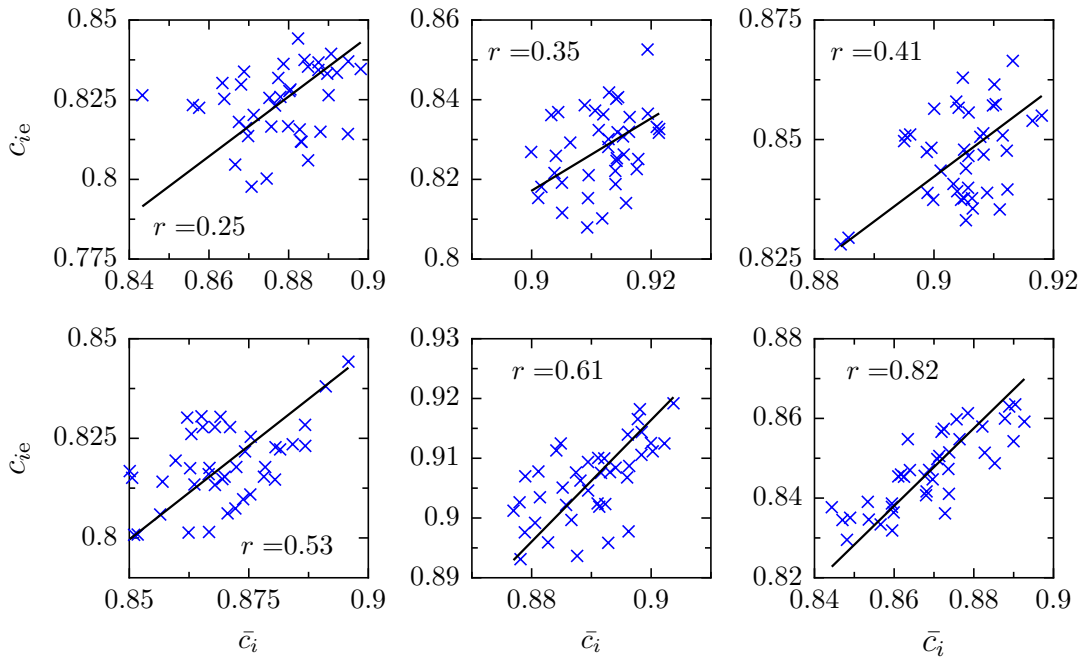


FIGURE 7.5: The real Pearson correlation coefficients  $c_{ie}$  as a function of the one computed indirectly  $\bar{c}_i$ . The six panels show the results for different simulated samples. The crosses correspond to the differently initialized neural networks. The black solid lines correspond to linear fits of the data. The coefficient  $r$  gives the correlation between  $c_{ie}$  and  $\bar{c}_i$ .

neural network from the 4 experimental SGM maps) arising from neural networks that have been trained on the same dataset but with a different initial state. In order to study the impact of the choice of the neural network, we trained 40 neural networks and evaluate them by computing the quantity  $c_{ie}$  and  $\bar{c}_i$  for 20 samples and for the 40 training. The panels in Fig. 7.5 depict an ensemble of points  $(c_{ie}, \bar{c}_i)$  that corresponds to the 40 trainings (one training corresponds to a cross). Each panel represents a different sample. Depending on the sample, we observe a more or less important correlation  $r$  between the quantities  $c_{ie}$  and  $\bar{c}_i$ . In average, the correlation is about 40% over 20 tested samples. Therefore, when predicting a disorder with several neural networks, we select the one with the highest values of  $\bar{c}_i$ . However, the differences are small and we do not expect a significant improvement on the prediction quality.

Considering only the model that provides the best performance for the 40 trainings, we now study the relation between the two coefficients  $c_{ie}$  and  $\bar{c}_i$ . From the 4 SGM maps computed for the 4 unperturbed conductance values for each of the 20 disorder configurations without taking care of the blurring filter, we obtain 80  $(c_{ie}, \bar{c}_i)$  points presented in the left panel of Fig. 7.6. It clearly appears that the two quantities are correlated. Thus, one can perform the linear fit to establish the relation

$$c_{ie} = \alpha \bar{c}_i \quad (7.1)$$

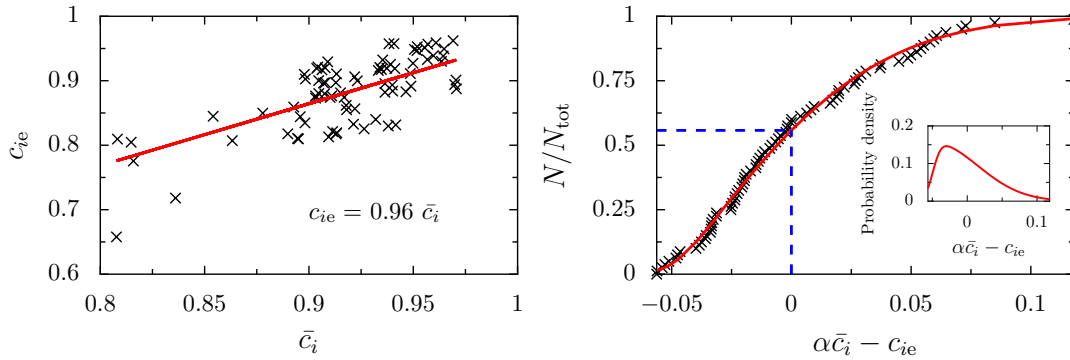


FIGURE 7.6: Left panel: the correlation of the predicted potential with the exact one for 20 different simulated samples and four different gate voltages, plotted versus the average correlation with the prediction for the three other gate voltages (see Fig. 7.4 for an example). The red line represents a linear fit  $c_i^e = \alpha \bar{c}_i$  to the data with  $\alpha = 0.96$ . Right panel: cumulative density function of the deviation between the real correlation coefficient  $c_{ie}$  and the one obtained by averaging  $c_{ij}$ . The black crosses represent the distribution obtained from the first 20 samples of the test set. The red curve represents the cumulative skew normal function  $\Phi$  of Eq. (7.2) with  $x = \alpha \bar{c}_i = c_{ie}$  and parameters  $\zeta = -0.049$ ,  $\omega = 0.064$ , and  $\theta = 6.996$  that fits the data. The inset represents the corresponding skew normal function. The dashed blue line represents the point of the fit where  $\alpha \bar{c}_i = c_{ie}$ .

where  $\alpha = 0.96$ . In addition, the right panel of Fig. 7.6 depicts the cumulative distribution function of the indirect evaluation of the PCC  $\alpha \bar{c}_i$  around the real one  $c_{ie}$ . The skewed distribution of the predicted correlation around the real one (see Fig. 7.6) is very well fitted by the skew normal distribution

$$\gamma(x; \zeta, \omega, \theta) = \frac{1}{\omega \sqrt{2\pi}} e^{-\frac{(x-\zeta)^2}{2\omega^2}} \left[ 1 + \operatorname{erf} \left( \theta \frac{x-\zeta}{\omega \sqrt{2}} \right) \right], \quad (7.2)$$

where  $\operatorname{erf}(x)$  is the error function. The solid red line in the right panel of Fig. 7.6 corresponds to the associated cumulative density function. Although the linear fit between  $c_{ie}$  and  $\bar{c}_i$  does not correspond to a perfect matching between the two quantities and is not correctly defined for the point  $\bar{c}_i = 1$ , we observe (in the right panel of Fig. 7.6) that a large proportion (84%) of the indirect PCC evaluation deviate by less than 5% from the real PCC.

The value of  $\alpha < 1$  can be explained by the absence of branches in a given area of the scanning zone, no matter the gate voltage of the QPC. Indeed, in the absence of SGM signal in a given region, one expects that the precision of the disorder prediction decreases in that zone [16]. Therefore, the prediction can be the same when passing SGM with different gate voltages through the neural network, but it can nevertheless be different from the real disorder.

When considering the blurring data augmentation, the value of  $\alpha$  that we

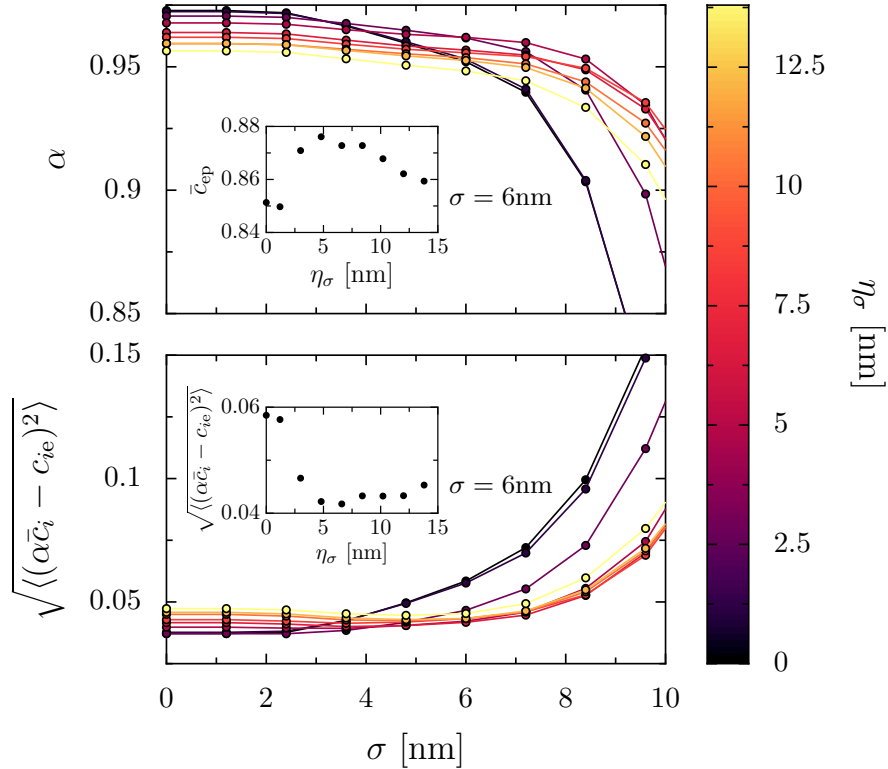


FIGURE 7.7: Upper panel: Evolution of the average coefficient  $\alpha$  over the 40 models as a function of the blurring width  $\sigma$  applied to the test set data for different values of training data augmentation  $\eta_\sigma$  that is depicted by the colorcode. The (upper) inset represents the average correlation between expected and predicted disorder potential over all the test set samples and over the 40 models for the case  $\sigma = 6$  nm. Lower panel: root mean squared error between the estimated correlation and the real one computed using the  $\alpha$  coefficient of the upper panel. The inset focuses on  $\sigma = 6$  nm.

consider depends on the estimated  $\sigma$  value and the parameter  $\eta_\sigma$  used for the data augmentation. We studied the effect of  $\eta_\sigma$  used for the training data on test set samples blurred with the fixed estimated value  $\sigma$ . The upper panel of Fig. 7.7 shows that. While  $\alpha$  is higher for low  $\eta_\sigma$  at low estimated  $\sigma$  the dependence is inverted for high values of  $\sigma$ . Focusing on the estimated value  $\sigma = 6$  nm that corresponds to the inset of the upper panel of Fig. 7.7, we observe that the highest average prediction  $\bar{c}_{ep}$  is reached when using  $\eta_\sigma = 4.8$  nm. We note that the distribution used for the data augmentation, which is a Gaussian centered in 0 and with a width  $\eta_\sigma$  is an arbitrary choice. The lower panel of Fig. 7.7 shows the evolution of the root mean squared of the difference between the expected and predicted accuracy  $\sqrt{\langle (\alpha \bar{c}_i - c_{ie})^2 \rangle}$  as a function of the estimated width  $\sigma$ . We observe the opposite behavior than for the factor  $\alpha$ . Focusing on  $\sigma = 6$  nm, we observe that the root mean squared error for  $\eta_\sigma = 4.8$  nm corresponds to one of the lowest values. Therefore, we perform the data augmentation for  $\eta_\sigma = 4.8$  nm.

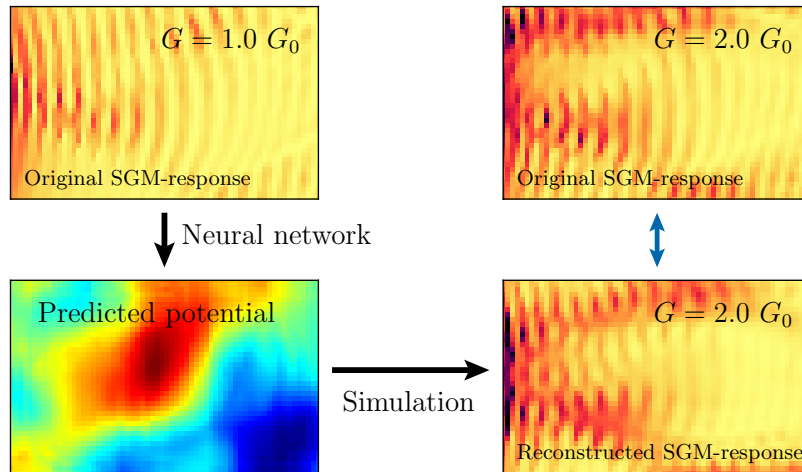


FIGURE 7.8: Qualitative verification method. The two upper panels represent the original SGM-response. The potential in the lower left panel is predicted from the SGM at  $G = 1 G_0$  by a neural network. The lower right panel represents the reconstructed SGM at  $G = 2 G_0$  from the predicted potential. The two right panels are the SGM-response that have to be compared.

### 7.4.2 Qualitative method

It is also possible to compute the SGM signal from the predicted potential and to compare the result to the original experimental data. To avoid to be skewed by the model used to compute the quantum transport properties, one can compute the SGM in a different experimental condition (e.g. adding a magnetic field or changing the gate voltage) and compare to the experimental data corresponding to this new experimental condition. An example of this procedure is depicted in Fig. 7.8 for a difference of unperturbed conductance.

However, as shown in Fig. 7.9, we observe no obvious correlation between the similarity of the potentials (the predicted one and the expected one) and the similarity of the SGM images (the SGM computed from the predicted potential and from the expected potential), both evaluated through the Pearson correlation coefficient (5.2). The hypotheses to explain these results are: (i) The lack of information on the disorder between the QPC and the scanning zone (for the SGM reconstruction, we consider no potential fluctuation due to the disorder there) that can lead to a large difference between the original SGM signal and the reconstructed one, even if the predicted potential is correct. And (ii), the Pearson coefficient compares two images pixel per pixel, meaning that a small shift of a branch or a phase difference of the interference fringes can lead to a significant decrease in the correlation coefficient. Thus, the value of the coefficient is mainly dependent on much less relevant features like the cross-talk<sup>4</sup> and leads to a better  $c_{\text{ep}}^{\text{SGM}}$  for samples whose unperturbed conductance is not on a quantization plateau as shown

<sup>4</sup>A possibility to avoid this phenomenon could be to use the structural similarity index measure (SSIM), however using this index does not allow to observe a clear correlation between the potential similarities and the SGM similarities.



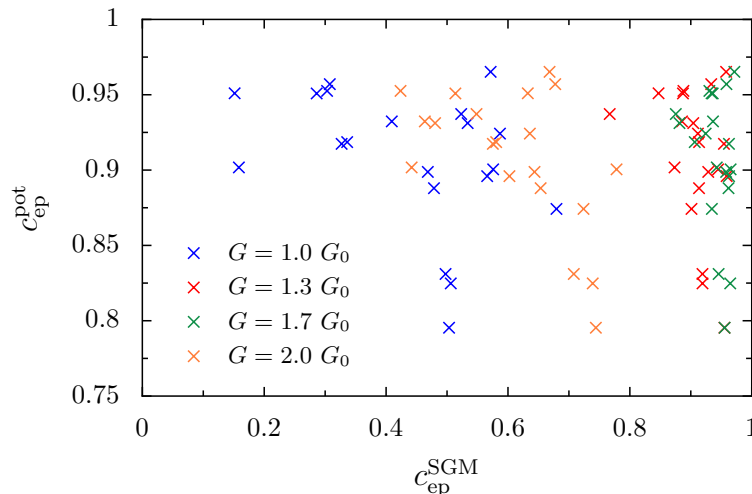


FIGURE 7.9: Correlation coefficient between the expected and predicted potential versus the correlation coefficient between the original and the reconstructed SGM response.

in Fig. 7.9. This evaluation method cannot give any quantitative information but can be considered qualitatively.

## 7.5 Prediction from real images

Having trained the neural network using the Gaussian filtering together with data augmentation and transfer learning methods as described above, optimized and tested on the set of simulated data, we are finally in position to use it for the prediction of the disorder potential in the experimental sample of Ref. [96].

The experimental SGM-responses (reshaped to the same resolution as the simulated ones<sup>5</sup>) is presented in the first line of Fig. 7.10.

The disorder potential predictions made by the neural network are depicted in the second line of Fig. 7.10, and the associated value  $\bar{c}_i$  is given in the images of the predictions. We take the results with the highest correlations coefficient  $\bar{c}_i = 0.905$ , which is the data associated to the conductance  $G = 1.7 G_0$ .

In order to ensure the reliability of the results, we plot the results of the selected model only (i.e. we do not perform the average on the 40 models). Fig. 7.11 represents the performance of the model on the test set for different blurring width. The  $\alpha$  coefficient determined for this model is 0.97. Considering the latter coefficient, we can observe the distribution of the error between the predicted and expected correlation coefficient in case of  $\sigma = 6$  nm.

Using Eq. 7.1 with  $\alpha = 0.97$ , we finally estimate the correlation between the predicted potentials of the third column in Fig. 7.10 and the real one equal

<sup>5</sup>Due to the limitation of computational time, we have not performed simulations with the same resolution as the one of the experiments.

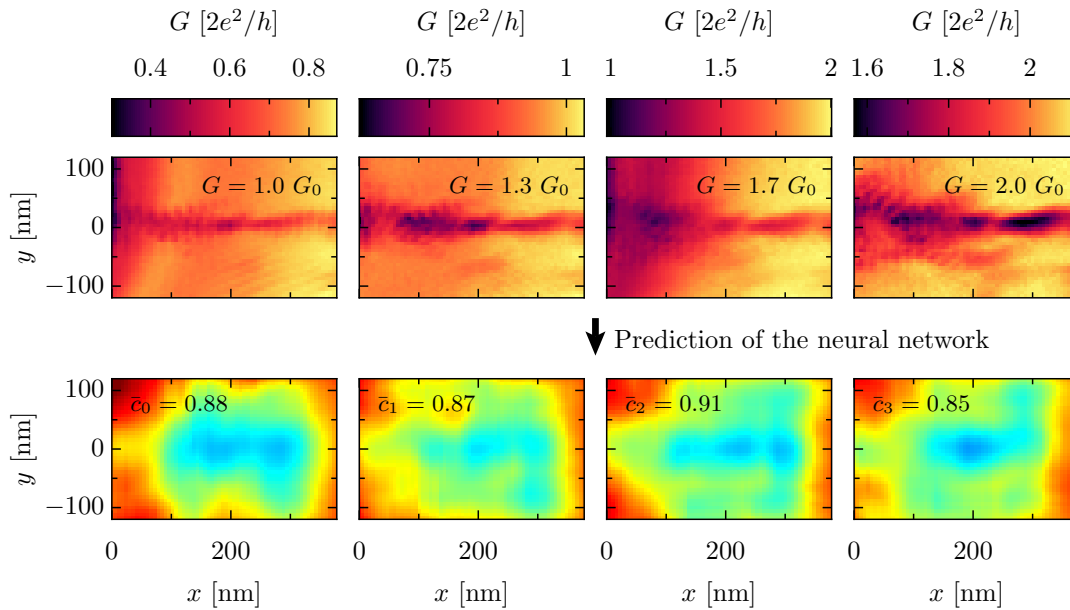


FIGURE 7.10: The top line represents the four experimental SGM maps obtained at the indicated QPC conductance values. The second line shows the corresponding predictions of the disorder potential obtained from the trained neural network when fed with the experimental SGM data as input.

to about 87 %. We note that the variation between the predicted potential and the real one is of the order of the variation between the four predictions.

It is possible to perform a qualitative verification of the relevance of the predicted potential by reconstructing the SGM response from the predicted potential of the four SGM maps corresponding to different unperturbed conductance values. From each of the predicted potentials, we computed the SGM for the four different values of unperturbed conductance<sup>6</sup>  $G$ . The results are depicted in Fig. 7.12<sup>7</sup>. We note that despite the difference of the predicted potentials, all the reconstructed SGM maps corresponding to the same conductance value are very similar. Thus, we can suggest that the variation between the predicted potentials has no important impact on the branch pattern. With this method we are not able to say which predicted potential is the best one. However, from the appearing similarities between the original and reconstructed SGM, we conclude that the predicted potentials are close to the real one.

<sup>6</sup>In reality, the unperturbed conductance values used for the simulation are a bit different from the original ones. Indeed, the unperturbed conductance indicated in the experimental SGM scans can be slightly modified by the presence of the tip setup. Thus, we performed the simulation for the following values of unperturbed conductance  $0.85 G_0$ ,  $1.0 G_0$ ,  $1.85 G_0$  and  $2.1 G_0$ .

<sup>7</sup>In some cases this method is not reliable. However, from the SGM images it appears that the electron flow has not felt a significant impact from the disorder outside the scanning zone.

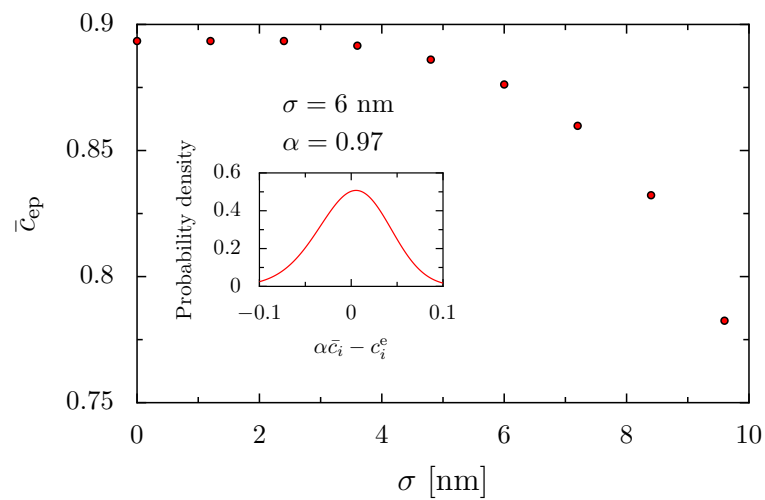


FIGURE 7.11: Precision of the model selected for the potential prediction for different values of blurring  $\sigma$ . The distribution of the error between the predicted and expected correlation coefficient is depicted in the inset considering the  $\alpha$  value corresponding to  $\sigma = 6$  nm. Using the fit (7.2) with  $\zeta = 0.031$ ,  $\omega = 0.050$  and  $\theta = -1.167$ .

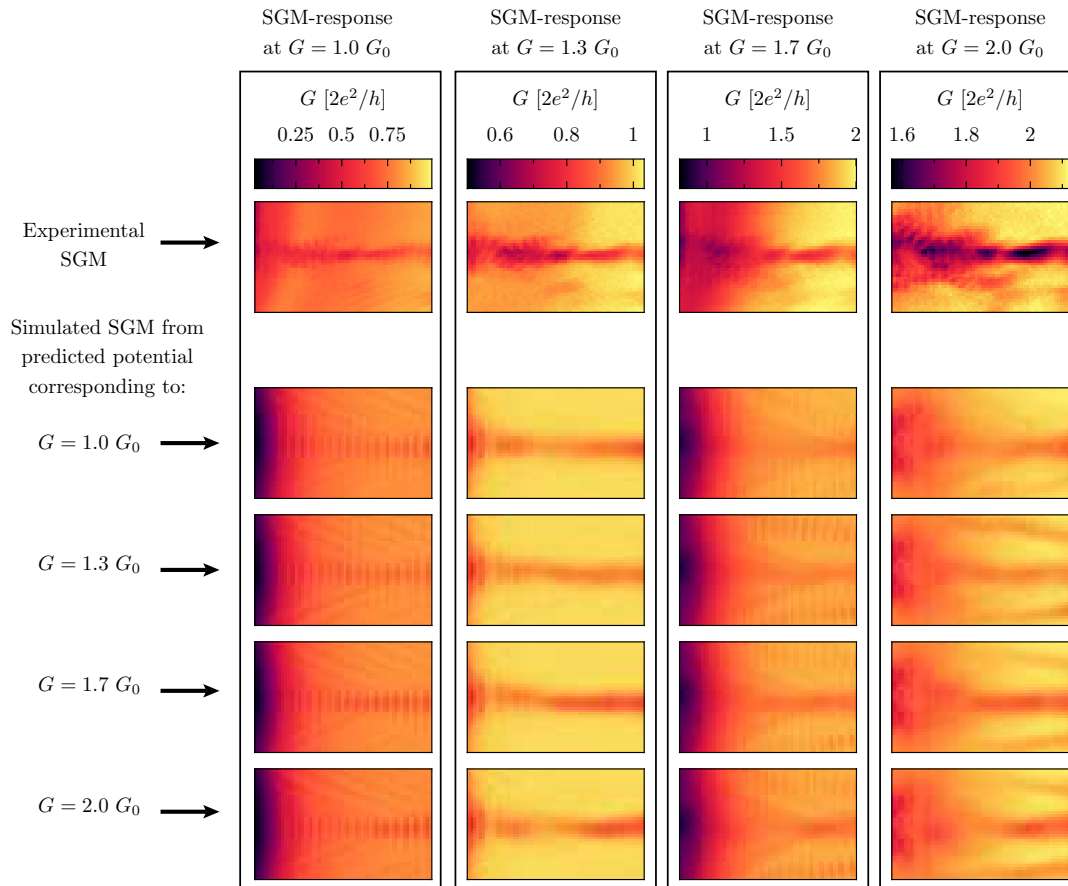


FIGURE 7.12: The top line represents the four experimental SGM maps obtained at the indicated QPC conductance values. The four lower lines represent the SGM responses reconstructed from the potentials predicted from the SGM measured at the conductance  $G$  indicated in the left part of the figure. A Gaussian filter of  $\sigma = 6$  nm is applied on all the simulated SGM maps. Each column corresponds to a different value of unperturbed conductance and has a different color code as indicated on the top of the column.



## Chapter 8

# Conclusions

The work presented in this manuscript follows the chronological order of research conducted over the past three years. Due to the uncertainty to obtain positive results about the extraction of the disorder information with machine learning methods from quantum transport information, we initially addressed a simplified problem. Encouraged by the positive outcomes of the initial study, we moved on to the more challenging problem of extracting all the information about the disorder potential. Finally, we determined the complete disorder potential landscape of a real sample. The latter study was conclusive enough to determine the disorder with a reasonable precision and also to propose suggestions for the experimental part of the study as well as an optimized protocol concerning the creation and the evaluation of a deep learning algorithm able to determine the disorder.

### 8.1 Proof of principle

Due to the absence of intuition about the possibility of extracting the disorder from quantum transport information, we initiated a preliminary study focused on a simplified problem. The first simplification involves utilizing a transport-related signal that is non-measurable experimentally. This approach aims to accelerate dataset generation and limit CPU-time to a reasonable amount for a proof of principle. Indeed, the PLDOS allowed to perform the simulation fast enough to create a complete dataset in a few weeks. This property of the PLDOS gave us a valuable flexibility for the preliminary studies.

#### Global parameters

The first step of the investigation was the extraction of global parameters that characterize the disorder from the PLDOS (Chap. 4). We observed a significant difference in the prediction quality for the two sets of parameters that we tried to recover. The set composed of uncorrelated parameters, namely the amplitude and the correlation length of the potential fluctuations, has been easier to determine than the other set composed of correlated parameters which are the experimentally relevant ones, the density of dopants and their distance from the 2DEG (see Fig. 4.7). We also took advantage of the simplicity of the problem to apply explainable machine learning techniques

to understand the working procedure of the neural network. However, it has been complicated to perform a quantitative study in that direction, using images like the ones present in Fig. 4.9, we only obtain an intuition about the part of the PLDOS image that seems to be most relevant for the prediction of the disorder.

## Complete disorder determination

In order to determine the complete disorder potential landscape, in Chap. 5 we kept using the PLDOS as input for our neural network. However, we modified the output to be either the Fourier coefficients used to create the corresponding disorder or directly the disorder landscape in real space.

Concerning the Fourier coefficients, using a large training set and an optimized network architecture, we observed a correlation between the impact of the term associated to one Fourier coefficient in the potential landscape and the accuracy of the neural network for the prediction of that Fourier coefficient. When reconstructing the potential landscape from the predicted Fourier coefficients, we obtained an average correlation between the predicted and expected potential of about 98.7% (see Fig. 5.4).

When using directly the real-space value of the disorder potential as output, using an appropriate and optimized model with convolutional encoder-decoder architecture, we observed that the neural network is even more efficient with an average performance of 99.7 % (see Fig. 5.8). Based on these remarkable results, we can deduce that almost all the information about the disorder is captured within the PLDOS. Averaging the spatial accuracy, we noticed a small decrease of performance on the edges of the potential image (see Fig. 5.9), which corresponds to the location where the electron density is weaker on average and an absence of information on the PLDOS beyond the edge of the image. Taking advantage of the simplicity of the neural network, we observed the image translation through the neural network, as depicted in Fig. 5.11, with the hope to understand its working procedure. However, no quantitative information could be extracted from this study. Additionally, it has been shown that constructing a dataset composed of samples with properties that vary within a parameter range, including the desired property of interest but not necessarily identical to the target sample, can lead to improved efficiency.

## Addition of information for the disorder determination

In order to increase the performance of the neural network (especially for small training sets), one can feed the neural network with several PLDOS images computed from the same disorder but corresponding to different experimental conditions. Thus, the neural network has more information to process. In this study, we observed that using PLDOS data computed for different values of the magnetic field significantly increases the precision. This enhancement was particularly pronounced at the edge of the sample, resulting in nearly uniform average spatial precision, as shown in Fig. 5.15.

## 8.2 Disorder determination for an existing sample

The last part of the study extends the machine-learning approach towards a prediction of the disorder in a real sample from experimentally accessible SGM data. This study confronts two fundamental challenges: predicting the disorder using a small dataset that contains less local data than the PLDOS, and evaluating our disorder prediction in the absence of knowledge about the actual disorder.

Regarding the dataset, we made a particular effort to simulate a real-life sample as realistically as possible. Chap. 6 is dedicated to the presentation of these simulations. Thus, we expect that the deviation between the disorders shown to the neural network and the real one are small enough to not have a significant impact on the prediction quality. Moreover, we used transfer learning, from a PLDOS-disorder pairs dataset, and data augmentation, using the symmetry of the problem, to cope with the small dataset size. We also took into account the vibrating tip effect present in the experimental data by blurring, with different intensity, the simulated SGM images used for the training of the neural network. When using those data to train the neural network, we observed an average correlation of 90 %.

The proposed indirect evaluation method consists in evaluating the results through the average Pearson coefficient computed between the disorders predicted from experimental SGM data of the same sample but performed within different experimental conditions (i.e. the predicted disorder should be the same). An example of this method is shown in Fig. 7.4. Analysis of simulated data allowed to observe a clear correlation between the latter average Pearson coefficient and the experimentally unavailable Pearson coefficient between the expected and predicted potential.

From the described dataset, and the indirect evaluation method, it has been possible to predict the disorder from the four experimental SGM images, each performed at a different gate voltage of the QPC. The four predictions share identical features and the best prediction has been estimated having a correlation coefficient of about 87% with the real disorder. These predictions are depicted in Fig. 7.10. When computing the SGM associated to the predicted disorders, one can observe qualitative similarities in the branching pattern. This correspondence corresponds to an additional proof of the reliability of the predicted disorder (see Fig. 7.12).

## 8.3 General method for unveiling the disorder potential in a heterostructure

The main result presented in this manuscript is the methodology for determining the disorder in a heterostructure.

For the experimental part, we recommend to perform the SGM experiments at the lowest possible temperature and avoiding as much as possible the vibrating tip effect. We suggest to perform the SGM in different experimental conditions that will be used as input of the neural network and for



the validation of the result. The resolution of the SGM image should be consistent with the estimated correlation length of the potential. The aim of this proposition is to avoid the waste of time by obtaining images with an excessive resolution that do not contain significantly more information than an appropriate lower resolution.

In order to be efficient in the creation of the dataset, we suggest to use transfer learning from PLDOS-disorder pairs for pre-training the neural network and data augmentation. This approach allows us to use a reasonable amount of SGM-disorder pairs for the last training step. While we recommend to use several inputs for the neural network, it is important to keep a reasonably small number of simulated data due to the significant computational power required for generating simulated SGM data under various experimental conditions. One has also to ensure that for the pre-training, the properties of the PLDOS and SGM-response remain similar under the given experimental conditions, which is not the case in a magnetic field, for instance.

Concerning the neural network architecture, we demonstrated that a simple encoder-decoder can be efficient enough to obtain high-quality results. However, we did not explore all the state-of-the-art architectures to solve the problem and it is possible that a more sophisticated neural network can improve the results.

The evaluation of the reliability of the predicted disorder is a difficult issue. Here, we propose a method based on the average correlation coefficient between the prediction of the disorder originating from different SGM maps measured on the same sample. To use this method, one has to establish the link between this coefficient and the actual correlation between the real and predicted disorder using simulated data. In parallel, one has to evaluate the average error of this method. It is also possible in some cases to compute the SGM-response from the predicted disorder in another experimental condition. However, this method does not allow to quantify the prediction accuracy.

## 8.4 Perspectives

In order to develop a machine learning algorithm for determining disorder in heterostructures with QPC nanostructures under more general conditions, one could train a neural network with a large dataset composed of various QPC shapes, a complete range of correlation lengths of the disorder, and other relevant factors. In addition to such an algorithm, it would be interesting to propose another algorithm which can create a relatively small dataset (a few hundred samples) based on predefined nanostructure geometry and heterostructure information. Thus, it would be possible to create a dataset in a relatively short amount of time (a few days) and perform transfer learning from the pre-trained neural network. Such a program could be at the origin of a better comprehension of the experiment and more generally the disorder configuration in high-mobility 2DEGs.

### **Predictive simulation of a real sample**

The disorder prediction combined to the recent advancements in the field of predictive charge distribution of a GaAs-based nanoelectronic device [119] would allow to simulate almost exactly the electrostatic potential of a real sample. Such a performance could prove to be a powerful tool for a deeper comprehension of the physics of nanodevices. Indeed, it could be possible to quantitatively simulate the main effects that led to the experimental observation, using a model of quantum transport including the precise microscopic disorder realization of the sample used in the experiment.

### **Disorder analysis before performing the experiment**

The rapid prediction of disorder configurations in real samples can serve as a valuable tool for experimentalists. They can observe the disorder of a cooled sample and decide to keep the observed configuration if the latter allows to perform the experiment in the best conditions. However, if the disorder configuration is not satisfying enough to perform the experiment, one can perform a new thermal cycle by warming up and then cooling down the sample to obtain a new disorder configuration.

### **Study of the dopant distribution**

By training an algorithm with several kinds of distributions concerning the dopant location, one can determine the distribution present in the sample. In the study, we considered only random location without any correlation. Other hypotheses exist and state that the dopant positions are correlated due to the Coulomb interaction [83].

### **Application to other kinds of systems**

SGM experiments have also been performed on graphene samples. The conductance of such systems behaves in a different manner than that of a heterostructure when using an invasive tip [120]. This results in a large difference between SGM images obtained from those two systems. Nevertheless, it can be interesting to perform a similar study with graphene in order to determine the disorder. It is important to consider that the origin of the disorder in these systems is fundamentally distinct.



## Appendix A

# Examples of potential construction

The approximation corresponding to Eq. 6.6 allows to perform the electrostatic computation in a relatively short amount of time. A detailed example of the approximation method is depicted in Fig. 6.9 and represents a situation where the tip is close to the QPC, which corresponds to one of the worst cases. In Fig. A.1, other examples of potentials computed with and without this approximation are shown. In the depicted examples, we observe that the results given by the approximation method are close to the exact ones.

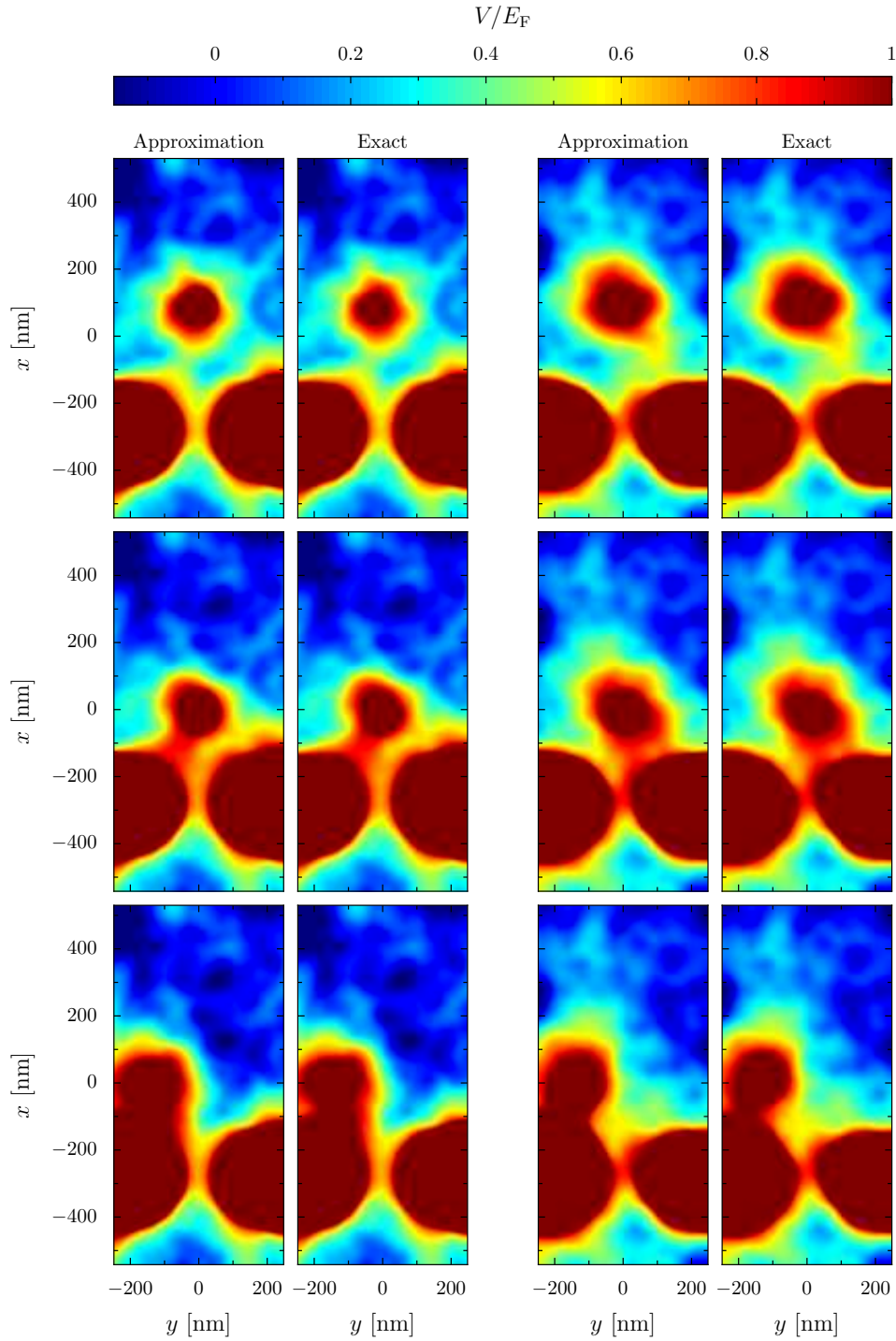


FIGURE A.1: Examples of potentials used for the SGM simulations. Each line represents a different tip position. Two different disorder realizations are depicted in the two columns, each composed of two columns where the left image corresponds to the approximate potential and the right one represents the exact one.

## Appendix B

# Gaussian Filter examples

Examples of simulated SGM maps computed from a predicted potential of the real sample are depicted in Fig. B.1, where each line (except the first one) corresponds to a different width of the Gaussian filter. We observe that blurring with a width  $\sigma = 6$  nm seems to be the most realistic choice.

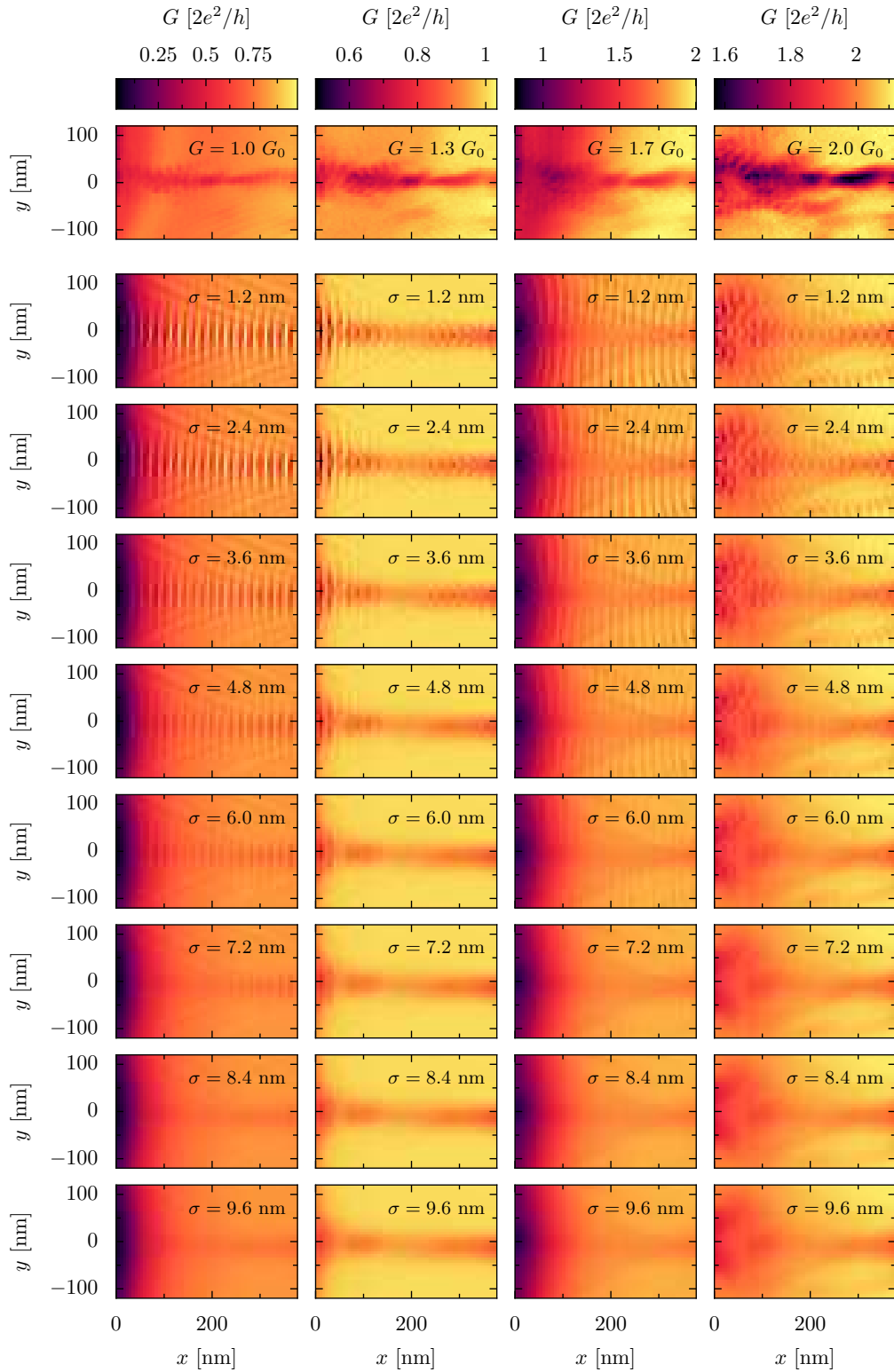


FIGURE B.1: The top line shows experimental SGM data for four different QPC conductance values. The other lines show examples of images obtained by applying Gaussian filters with different widths on the reconstructed SGM response.

## Appendix C

# Résumé en français

### C.1 Introduction

Grâce à l'évolution des techniques de fabrication d'échantillons électroniques, les structures de ces derniers ont vu leur taille diminuer jusqu'à atteindre la taille du micron. Cette miniaturisation a entraîné l'apparition des propriétés quantiques des électrons dans les mesures de courant à basse température. Un type d'échantillon fréquemment utilisé dans le domaine du transport quantique sont les hétérostructures semi-conductrices contenant un gaz d'électron bi-dimensionnel à haute mobilité. Dans le cadre de l'étude menée durant la thèse, nous nous focalisons uniquement sur ce type de matériaux. Pour de tels échantillons, il a été observé une fluctuation des propriétés électroniques entre différents échantillons ayant cependant suivi le même processus de fabrication. Cette fluctuation est due aux propriétés microscopiques de l'échantillon. En effet, les électrons du gaz proviennent de dopants ionisés distribués de façon non contrôlée dans une des couches du matériau. Lors de la création de ces échantillons, seule la densité de dopants peut être fixée par les expérimentateurs mais leur position est désordonnée et inconnue. L'ensemble de ces dopants ionisés crée un champ électrostatique qui dépend de la configuration et de la localisation des dopants. Ce champ électrostatique crée donc un potentiel électrostatique qui influe sur le comportement des électrons. *In fine*, les propriétés électroniques sont affectées par ce potentiel électrostatique irrégulier qui provient du désordre inconnu des dopants.

L'objectif des travaux de recherche présentés dans cette thèse est de développer une méthode pour déterminer ce désordre. Pour cela, nous utilisons des algorithmes d'intelligence artificielle qui utilisent des données de transports électroniques qu'il est possible de mesurer expérimentalement sur les échantillons mentionnés précédemment. Ces données sont issues de l'expérience de microscopie à grille locale (MGL). Cette dernière consiste à perturber localement le champ électrostatique à l'aide d'une pointe fine placée au-dessus de l'échantillon et sur laquelle une tension négative est appliquée. Une fois la perturbation appliquée, on mesure une quantité appelée conductance (inverse de la résistance). Ce procédé est appliqué pour un grand nombre de positions de pointe. Lorsque l'on reporte les données de conductance sur le plan où la pointe s'est déplacée, nous obtenons une image de MGL. L'objectif est donc de déterminer la valeur du champ électrostatique créé



par les dopants ionisés à chaque position où la pointe a été placée ce qui correspond aussi à une image. L'algorithme d'intelligence artificielle utilise l'image MGL comme donnée d'entrée dans le but de déterminer l'image du potentiel de désordre.

Les motivations concernant la détermination du désordre sont multiples. Tout d'abord, une parfaite connaissance du désordre couplé à une modélisation réaliste de l'environnement électrostatique permettrait de pouvoir effectuer des simulations de transport électroniques sur de vrais échantillons. Il serait ainsi possible de vérifier certaines théories en les comparant aux mesures expérimentales ce qui n'est pas possible actuellement. De plus, la détermination du désordre permettrait d'en apprendre davantage sur les propriétés du désordre comme par exemple la mise en évidence d'une corrélation entre les positions des dopants ionisés. Une dernière application, qui pourrait avoir un impact considérable sur la partie expérimentale, est le "choix" du désordre. Plus précisément, il s'agirait pour l'expérimentateur de déterminer le désordre présent dans l'échantillon et de voir si ce dernier pourrait être gênant ou non pour l'expérience. Dans le cas où le désordre ne serait pas favorable à l'expérience, l'expérimentateur pourrait alors réchauffer l'échantillon à température ambiante et le refroidir à quelques centaines de millikelvin ce qui aurait pour impact de changer la configuration du désordre.

Afin de déterminer le désordre présent dans une heterostructure sur laquelle a été effectuée une mesure de MGL, nous avons commencé par déterminer quelques caractéristiques du désordre sur des échantillons simulés puis nous avons graduellement complexifié le problème pour atteindre l'objectif fixé. Afin de résumer le travail effectué, nous allons en premier lieu poser les bases d'intelligence artificielle et de transport quantique nécessaire à la pleine compréhension du projet dans la section C.2. Dans la section C.3, nous aborderons la première partie de l'étude qui a servi de preuve de principe concernant la réalisation d'un tel projet et a été à l'origine d'une première publication [16]. L'utilisation de notre algorithme sur un échantillon réel sera détaillée dans la section C.4. Finalement, les conclusions du projet seront présentées dans la section C.5.

## C.2 Pré-requis

### C.2.1 Description du système

L'échantillon étudié est une heterostructure semi-conductrice composée de GaAs/AlGaAs. Cet échantillon est constitué d'un empilement de différents matériaux, avec une interface contenant un gaz d'électrons libre dans un espace bi-dimensionnel. Ces électrons proviennent de dopants ionisés qui se situent à une distance donnée du gaz d'électrons permettant ainsi d'obtenir une forte mobilité dans le gaz d'électrons. Le champ électrostatique créé par les dopants ionisés affecte le comportement des électrons.

En plus d'être affectés par le champ électrostatique provenant des dopants, les électrons sont aussi soumis aux champs des grilles placées à la surface

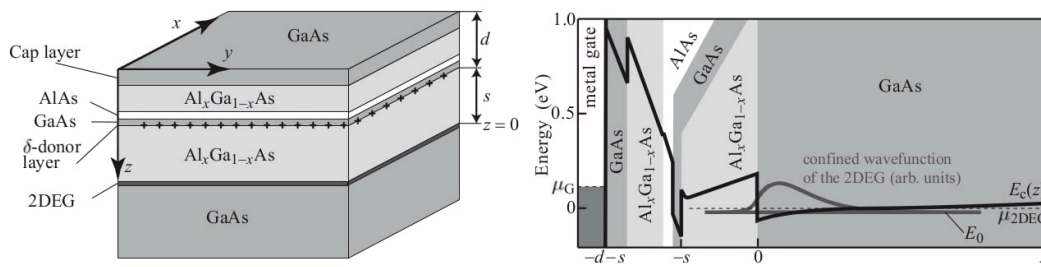


FIGURE C.1: Panneau de gauche : Schéma de l'hétérostructure avec un empilement détaillé des couches. Le diagramme d'énergie de cette hétérostructure est représenté dans le panneau de droite. Nous observons le minimum de potentiel à  $z = 0$  où se trouve le 2DEG. Figure tirée de [91].

de l'hétérostructure. Dans toute notre étude, nous nous focalisons sur un système contenant un point de contact quantique (PCQ). Pour réaliser un PCQ dans le gaz d'électrons, il est nécessaire d'utiliser des grilles positionnées comme montré sur l'encart de la figure C.2. Sur ces grilles est appliqué un potentiel électrostatique négatif afin de repousser les électrons du gaz bidimensionnel créant ainsi ce qui est appelé des zones de déplétions. Les électrons sont donc obligés de passer par une zone de constriction qui se situe entre les régions déplétées. De tels systèmes ont été l'objet d'un grand nombre d'études dont notamment célèbre pour avoir mis en évidence la quantification de la conductance à la fin des années 80. En observant le panneau de droite de la figure C.2, nous observons cette quantification à travers l'évolution en "escalier" de la conductance en fonction de la tension appliquée aux grilles du PCQ. En l'absence de quantification, l'évolution de la conductance n'aurait pas dû présenter de sauts. En effet, plus la tension de grille baisse (en valeur absolue), plus la constriction dans le gaz devient grande et plus la conductivité augmente de manière proportionnelle. La structure en escalier, qui provient de la quantification de la conductance, apparaît car lorsque l'on baisse la tension de grille (en valeur absolue), la conductance reste constante jusqu'à un certain seuil où elle bondit brusquement. Cet effet est dû à la quantification de l'énergie transverse des électrons dans le PCQ.

La dernière grille qui est utilisée pour l'expérience est la pointe qui scanne la surface de l'échantillon. Sur cette pointe, on applique une tension négative qui vient perturber la trajectoire des électrons. Cette perturbation crée une bosse de potentiel sur laquelle diffuse les électrons. Sur la figure C.3, nous pouvons observer une image de scan de MGL, à savoir la conductance en fonction de la position de la pointe, où nous observons un motif de branche. Sur ces branches, nous observons une forte diminution de la conductance par l'effet de pointe. Ceci s'explique facilement si l'on associe ces branches au flux des électrons. En effet, si la pointe est positionnée au-dessus d'un flux d'électrons important et créant une réflexion de ce flux d'électrons vers le PCQ, alors le courant à travers le PCQ aura diminué de façon conséquente

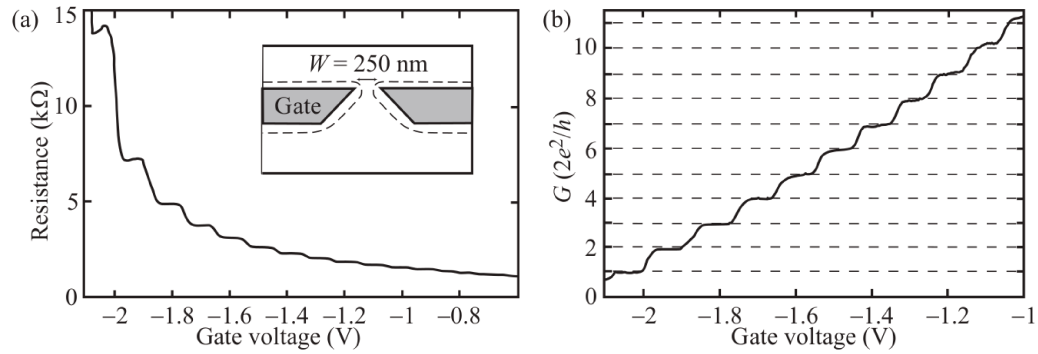


FIGURE C.2: Mesure expérimentale de la résistance en fonction du voltage de grille du point de contact quantique (a). Représentation des grilles qui créent le point de contact quantique (encart). Conductance associée en fonction du voltage de grille du point de contact quantique (b). Figure extraite de [91] qui correspond aux résultats de [1].

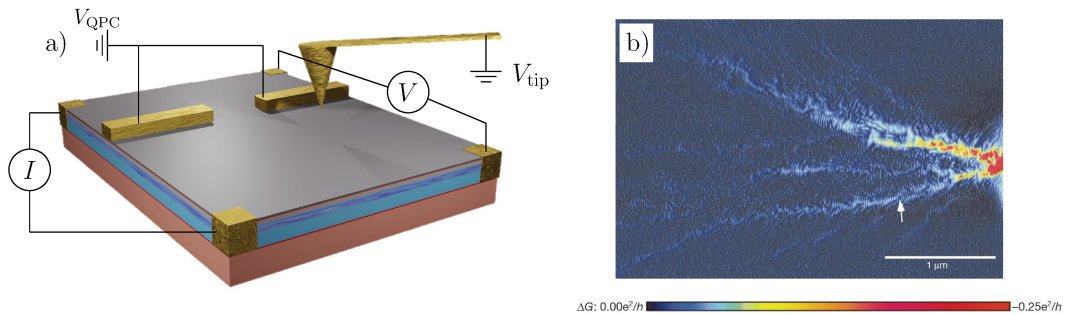


FIGURE C.3: a): Schéma de l'expérience de microscopie de grille locale pour un système contenant un point de contact quantique. b): Résultat de l'expérience de microscopie à grille locale à la température  $T = 1.7$  K. Les données sont extraites de [26]

ayant pour effet une baisse de la conductance. Notons aussi que la structure des branches qui apparaît sur l'image de MGL dépend fortement des positions des dopants ionisés.

Afin de déterminer le potentiel électrostatique créé par les dopants ionisés et subi par les électrons, nous utilisons une intelligence artificielle qui prendra comme information d'entrée l'image MGL.

## C.2.2 Deep learning

Dans cette section nous allons nous intéresser à une catégorie précise d'intelligence artificielle appelée "deep learning". Le deep learning correspond à l'apprentissage de réseaux neuronaux profonds. Le block élémentaire d'un réseau de neurones est appelé le perceptron et peut être représenté comme sur la figure C.4. Sur cette figure, il est possible de voir qu'il existe différentes couches



L'entraînement d'un réseau de neurones consiste à déterminer l'ensemble des poids  $\{\mathbf{w}\}$  qui permettent au réseau de neurones de faire correspondre la bonne sortie à la bonne entrée. Pour cela, il faut premièrement initialiser les poids du réseau de neurones, ce qui se fait souvent en les choisissant aléatoirement. Par la suite, nous allons montrer au réseau de neurones des entrées avec les sorties qui correspondent. Pour chaque exemple, le réseau de neurones calcule de combien il s'est trompé à l'aide d'une fonction appelée fonction coût. Ensuite, il évalue la variation de cette fonction coût en fonction de la variation des poids. Pour finir, il redéfinit ses poids de manière à minimiser le plus possible la fonction coût. Ce processus est répété jusqu'à atteindre un ensemble de poids optimal. Étant donné que pendant l'entraînement, nous montrons au réseau de neurones à la fois l'entrée et la sortie, nous parlons d'apprentissage supervisé.

Dans le cadre de notre étude, nous allons traiter des images comme données d'entrée. Afin d'analyser des images, nous utilisons des couches dites convolutionnelles. La particularité de ces couches est que contrairement aux couches décrites précédemment, appelées couches denses, les couches convolutionnelles ne connectent pas tous les neurones de la couche  $l$  avec tous les neurones de la couche  $l - 1$ . Ceci est dû au fait que chaque couche convolutionnelle contient un paquet d'images et les neurones correspondent aux valeurs de tous les pixels de toutes les images. Ainsi, le nombre de connexions devient rapidement gigantesque. Dans les couches convolutionnelles, les connexions entre les couches sont très peu denses, mais sont placées de façon très optimisée.

Il arrive aussi fréquemment qu'on utilise une première partie du réseau de neurones composé de couches convolutionnelles de façon à traiter une image puis d'aplatir la dernière couche convolutionnelle de façon à mettre des couches denses pour déterminer un ou plusieurs scalaires.

## C.3 Preuve de principe

### C.3.1 Modèle

Afin de faire une preuve de principe concernant la possibilité de déterminer des informations sur le désordre à partir de données de transport, nous allons utiliser un modèle du système étudié, qui sera simplifié de façon à générer des données dans un court intervalle de temps. Pour cette étude, le potentiel correspond à celui d'une couche de dopants de largeur infinitésimale située à une distance  $s$  du plan contenant le gaz d'électrons. Au lieu de sommer la contribution électrostatique de tous les dopants, ce qui revient à calculer  $10^4$  termes pour chaque pixel de l'image de potentiel, nous considérons la transformée de Fourier de ce potentiel définie par

$$V_{\text{dop}}(\mathbf{r}) = -E_{\text{Ry}}^* a_{\text{B}}^* \frac{\Delta q_x \Delta q_y}{\pi} \sum_{j>0} \frac{e^{-q_j s}}{q + q_{\text{TF}}} C(\mathbf{q}_j) e^{-i\mathbf{q}_j \cdot \mathbf{r}}. \quad (\text{C.3})$$

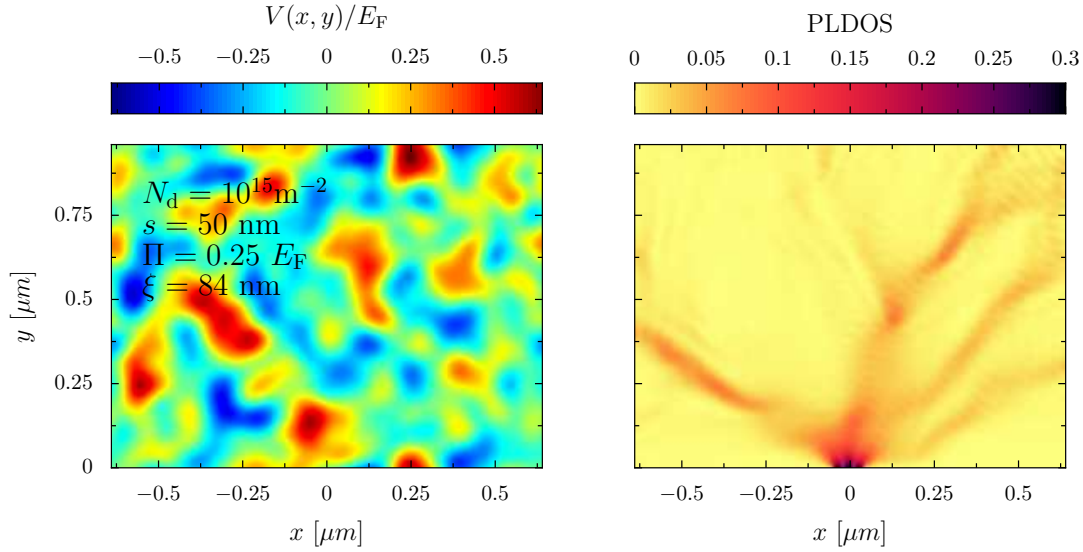


FIGURE C.5: Le panneau de gauche représente un potentiel de désordre de Coulomb choisi aléatoirement qui correspond aux paramètres présents dans la figure. Le panneau de droite représente la DPLE calculée à partir de ce désordre.

$E_{\text{Ry}}^*$  et  $a_{\text{B}}^*$  sont respectivement l'énergie de Rydberg et le rayon de Bohr effectif. Les coefficients de Fourier  $C(\mathbf{q}_j)$  sont des nombres complexes aléatoires qui sont choisis suivant une loi normale dont l'écart-type dépend de la densité de dopants  $N_{\text{d}}$ . Cette méthode nous permet d'effectuer sur 250 fréquences<sup>1</sup>. Notons aussi que cette méthode en compte l'écrantage des électrons présents dans le gaz à travers le terme  $q_{\text{TF}}$ . Un exemple de ce potentiel est présent sur le panneau gauche de la Fig. C.5. D'après la formule (C.3), le potentiel peut être défini par les paramètres  $s$  et  $N_{\text{d}}$ , mais il est aussi possible de caractériser ce potentiel par une longueur de corrélation  $\xi$  définie comme

$$\mathcal{C}(\xi) = \frac{\mathcal{C}(0)}{2}, \quad (\text{C.4})$$

où  $\mathcal{C}(R)$  est la fonction d'auto-corrélation  $\mathcal{C}(R) = \langle V(\vec{r})V(\vec{r} + \vec{R}) \rangle$  du potentiel, et par l'amplitude des fluctuations de potentiel calculé à travers la racine de la valeur moyenne du potentiel au carré (RMPC)

$$\Pi = \sqrt{\langle V^2 \rangle}. \quad (\text{C.5})$$

Sachant que nous souhaitons uniquement faire une preuve de principe, nous utilisons une information de transport, qui est beaucoup plus rapide à simuler que l'expérience de MGL, et qui correspond à la densité partielle locale d'états (DPLE). En effet, afin d'obtenir une image de DPLE un seul calcul suffit contre un nombre de calculs égal au nombre de pixels pour la MGL.

<sup>1</sup>Dû à la décroissance exponentielle du poids des coefficients de Fourier, l'image de potentiel a convergé après avoir pris en compte 250 fréquences.

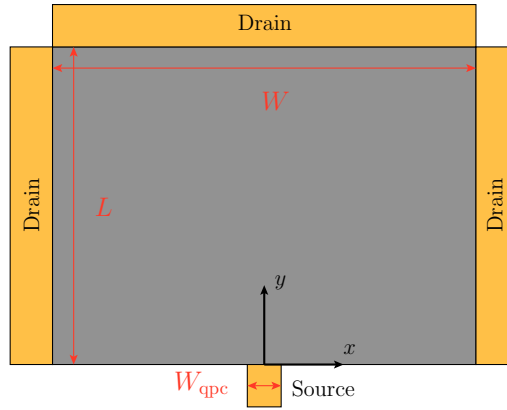


FIGURE C.6: Schéma de la géométrie du système utilisée pour les simulations. Le point quantique est simulé par la petite électrode de source.

Tous les calculs sont obtenus dans le cadre du modèle de liaisons fortes qui est implémenté par le package python appelé KWANT [87]. Par souci de rapidité des calculs, nous nous plaçons à température nulle. Le système du PCQ est modélisé par la figure C.6. Un exemple de DPLE calculée à partir d'un potentiel de désordre est présent sur le panneau de gauche de la figure C.5.

### C.3.2 Caractérisation du potentiel par deux paramètres

En guise d'échauffement dans l'extraction d'informations du potentiel de désordre, nous allons tout d'abord tenter d'obtenir un ensemble de paramètres qui caractérise le désordre, à savoir  $(N_d, s)$  ou  $(\Pi, \zeta)$ . Ces paramètres seront donc les sorties du réseau de neurones alors que l'entrée sera l'image de DPLE. Après avoir entraîné le réseau de neurones avec 140,000 échantillons, nous évaluons le réseau de neurones avec des échantillons que le réseau de neurones n'a jamais vus durant l'entraînement. En ce qui concerne l'ensemble  $(N_d, s)$ , le réseau commet une erreur relativement importante de 15% sur la densité de dopants et une erreur de 3% sur la distance  $s$ . Nous constatons que l'erreur faite sur ces deux quantités est corrélée d'après la figure C.7 où chaque point représente un échantillon utilisé pour l'évaluation. Cette corrélation s'explique par l'écart de la RMPC correspondant aux paramètres prédits par rapport à la RMPC calculée depuis les paramètres souhaités. En effet, on observe sur la figure C.7 que la pente de la corrélation est superposée aux échantillons pour lesquels aucune erreur n'a été faite sur la RMPC. Ce résultat est surprenant car il montre que l'algorithme a appris par lui-même à reconnaître les paramètres naturels du problème (On peut considérer que les paramètres  $s$  et  $\zeta$  sont similaires car ils sont proportionnels.). Ainsi en utilisant maintenant l'ensemble de paramètres  $(\Pi, \zeta)$ , le réseau de neurones a beaucoup plus de facilité à déterminer ces paramètres. En effet, dans ce cas l'erreur sur  $\zeta$  est légèrement inférieure à 3% et d'environ 5% pour le paramètre

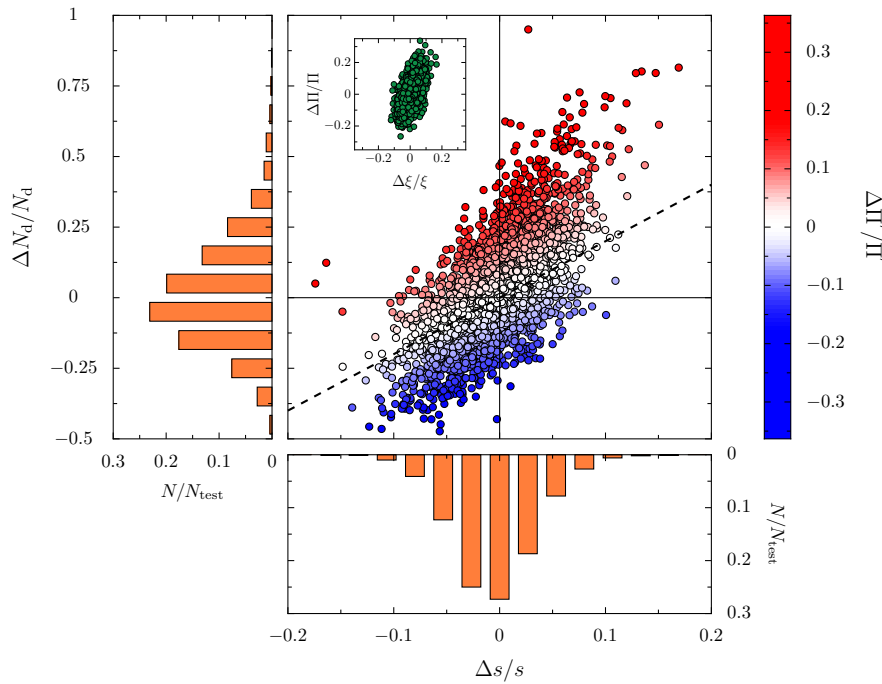


FIGURE C.7: Distribution des erreurs relatives  $\Delta N_d/N_d$  de la densité de dopants et de la distance  $\Delta s/s$  pour tous les éléments de l'ensemble de test, mises à l'échelle par le nombre d'échantillons de l'ensemble de test  $N_{\text{test}}$  (histogrammes orange). La figure centrale représente les valeurs individuelles des deux erreurs, mettant ainsi en évidence la corrélation entre les deux paramètres. Le code couleur indique l'erreur relative sur la racine carrée moyenne calculée  $\Delta \Pi/\Pi$ . La ligne en pointillés noirs représente la ligne théorique linéarisée dans le plan  $\Delta N_d/N_d$  versus  $\Delta s/s$  pour laquelle la racine carrée moyenne prédite correspond à la racine carrée moyenne attendue. L'encart représente la distribution des erreurs relatives de la racine carrée moyenne du potentiel  $\Delta \Pi/\Pi$  et de la longueur de corrélation  $\Delta \xi/\xi$ .

II. De plus, la corrélation entre les erreurs commises sur ces deux paramètres à très largement diminué (voir encart de la figure C.7).

### C.3.3 Détermination de l'image du potentiel observé par les électrons

Nous allons maintenant nous attaquer à un problème plus compliqué qui consiste à déterminer la carte du potentiel électrostatique subi par les électrons. Pour cela, nous utilisons un réseau de neurones particulier qui permet de donner une image en sortie tout en prenant une image en entrée. Ce réseau s'appelle un encodeur-décodeur convolutionnel. Nous entraînons le réseau de neurones avec comme donnée d'entrée, l'image de DPLE et en image de sortie la carte du potentiel (Un exemple du jeu de données correspond



à l'image C.5). Cette fois-ci, nous testons le réseau de neurones en calculant le coefficient de corrélation entre l'image prédite et l'image souhaitée. Ce coefficient de corrélation est défini comme

$$r_p = \frac{\sum_{i=1}^{N_p} (y_{\text{pred},i} - \bar{y}_{\text{pred}}) (y_{\text{exp},i} - \bar{y}_{\text{exp}})}{\sqrt{\sum_{i=1}^{N_p} (y_{\text{pred},i} - \bar{y}_{\text{pred}})^2} \sqrt{\sum_{i=1}^{N_p} (y_{\text{exp},i} - \bar{y}_{\text{exp}})^2}}, \quad (\text{C.6})$$

où les sommes portent sur les  $N_p$  pixels des images, tandis que  $\bar{y}_{\text{pred}}$  et  $\bar{y}_{\text{exp}}$  sont les valeurs moyennes des deux images. La distribution des coefficients de corrélations pour tous les échantillons utilisés pour l'évaluation du réseau de neurones est présentée sur la figure C.8. On peut apercevoir que tous les potentiels prédits ont une corrélation supérieure à 99% avec les potentiels souhaités. Cette impressionnante précision nous permet d'obtenir une prédiction qui est toujours très proche de la réalité même dans le pire des cas comme présenté dans l'exemple c) de la figure C.8. Notons que ce réseau de neurones est relativement rapide à entraîner car il est composé uniquement de couches convolutionnelles. Ainsi, cette architecture se prête parfaitement à la détermination du potentiel depuis une image contenant des données de transport.

Nous avons aussi montré qu'il était possible d'améliorer la performance de tels réseaux de neurones en utilisant plusieurs images de données de transport en entrée qui correspondent au même désordre, mais qui sont obtenus dans des conditions expérimentales différentes. Dans l'étude, nous avons fait changer la valeur du champ magnétique présent durant l'expérience simulée.

## C.4 Application de la méthode à un vrai échantillon expérimental

### C.4.1 Présentation de l'échantillon et du modèle

Afin d'utiliser notre méthode, nous avons choisi un échantillon qui permet d'être simulé dans un temps raisonnable. En l'occurrence, il s'agit d'un échantillon qui a été mesuré à très basse température (pour pouvoir être approximé comme température nulle) et qui est plutôt petit ce qui impacte positivement le temps de calcul. En revanche, étant donné qu'il s'agit d'un vrai échantillon, il n'est pas possible d'utiliser la DPLE en guise de données de transport car elle n'est pas accessible expérimentalement. Nous devons donc utiliser les images MGL qui sont plus longues à simuler. En plus de cela, nous allons aussi utiliser des simulations du potentiel qui sont plus réalistes en résolvant l'équation de Poisson qui prendra aussi en compte l'écrantage de Thomas-Fermi. Dans le cas de l'échantillon choisi, nous avons 4 cartes de MGL qui ont été obtenues dans des conditions expérimentales différentes qui sont des valeurs de voltages de grilles de PCQ différentes. Ces cartes de MGL sont présentées sur la figure C.9. Afin de réduire au maximum le temps de simulation, nous avons fait plusieurs approximations comme dans l'implémentation du potentiel de la pointe par exemple, et nous avons

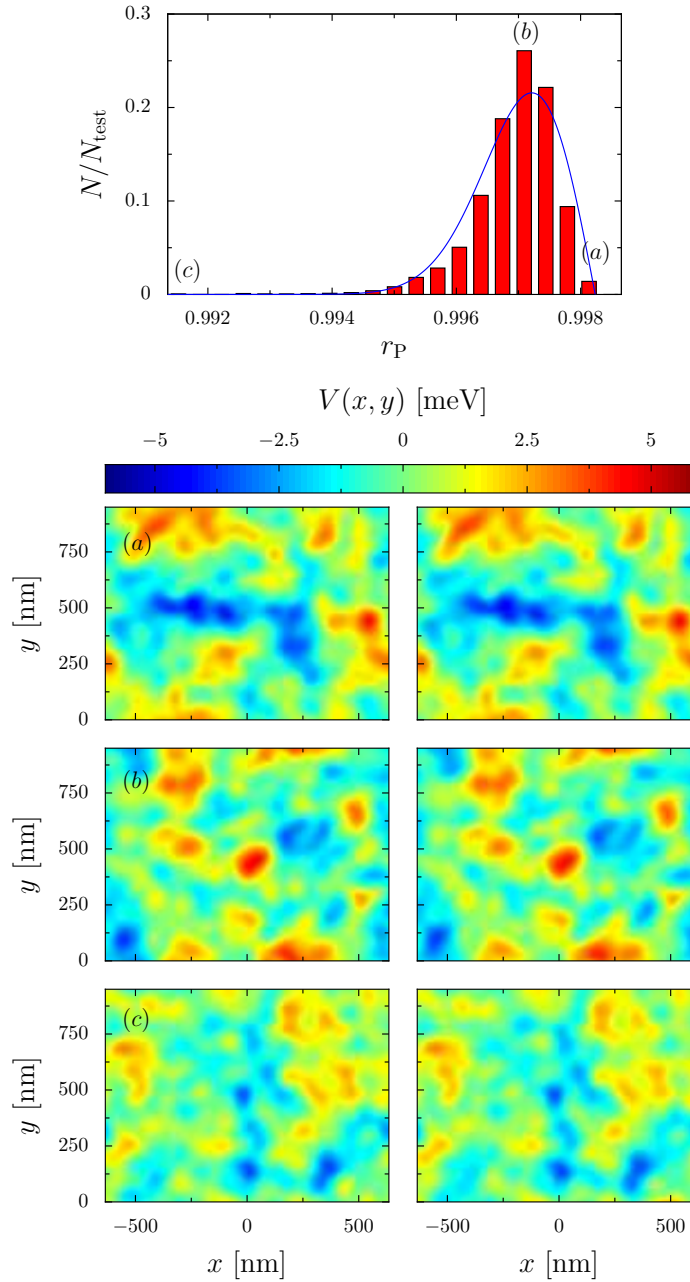


FIGURE C.8: Panneau supérieur : distribution du coefficient de corrélation des échantillons de l'ensemble de test entre le potentiel attendu et celui prédit par le réseau de neurones entraîné. La ligne bleue en pointillés représente l'ajustement de la distribution. Les panneaux inférieurs sont des exemples de potentiels attendus (à gauche) et prédits (à droite) correspondant à un coefficient de corrélation indiqué dans le panneau supérieur.

aussi utilisé un ordinateur parallèle très puissant où nous avons pu utiliser 23 nœuds composés chacun de 48 cœurs. Malgré tout cela, le temps de simulation ne nous a pas permis de créer un ensemble de données de plusieurs dizaines de milliers d'échantillons comme dans l'étude de la preuve de principe.

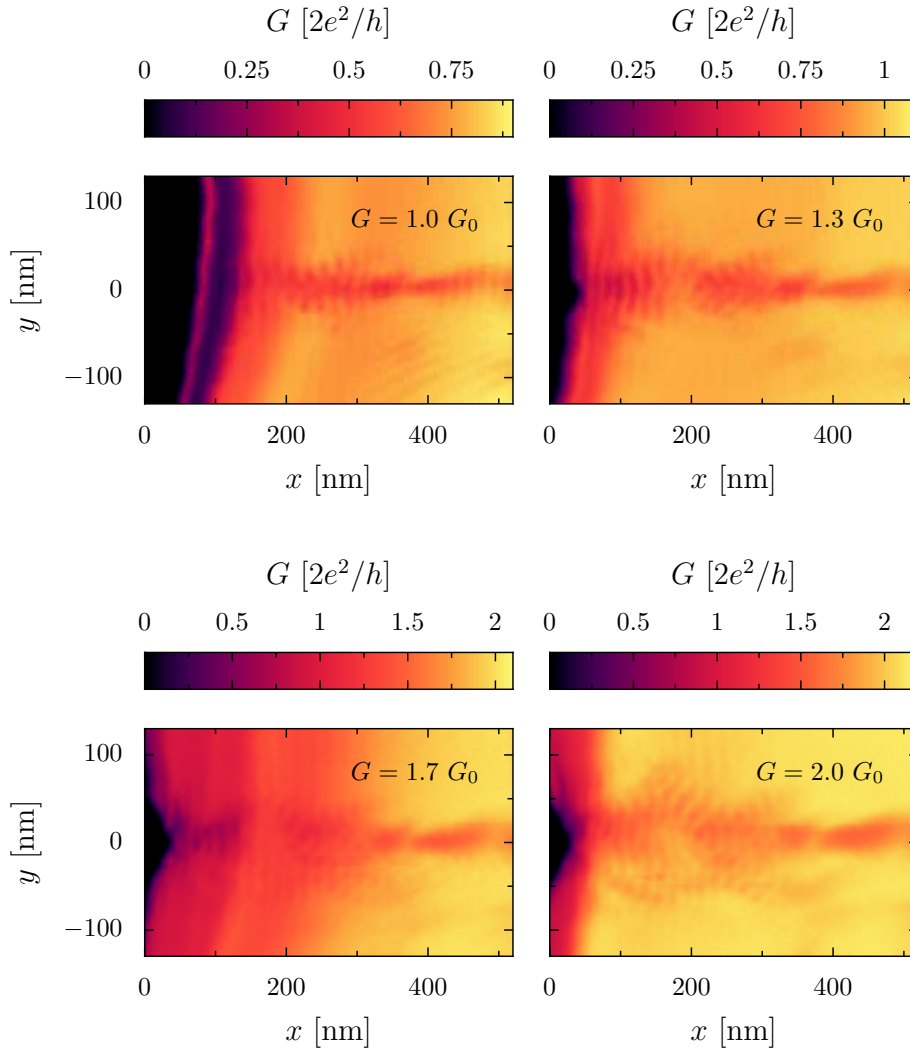


FIGURE C.9: Cartes SGM mesurées expérimentalement. Données fournies par B. Brun.

Ici, nous avons simulé 2,650 échantillons au total. Ainsi, il a fallu utiliser quelques astuces pour entraîner un réseau de neurones avec aussi peu de données. Les deux astuces utilisées sont l'augmentation de données (en utilisant la symétrie du système on peut doubler le nombre de données) et l'apprentissage par transfert. Ce dernier correspond à entraîner un réseau de neurones avec un grand ensemble de données qui est très proche de l'ensemble de données finales. Puis nous utilisons ce réseau de neurones pré-entraîné pour l'entraîner cette fois sur l'ensemble de données finales qui contient moins d'échantillons.

## C.4.2 Résultats

Après avoir entraîné le réseau de neurones avec les astuces décrites ci-dessus, nous allons prédire le potentiel et déterminer la précision de la prédiction. Cette dernière étape est en réalité compliquée. En effet, il n'est pas possible

de comparer le désordre prédit au désordre présent dans l'hétérostructure car ce dernier n'est pas accessible par l'expérience. Nous allons alors déterminer la fiabilité de la prédiction d'une autre manière en utilisant une méthode basée sur la comparaison de prédictions différentes validée avec des données simulées. Dans cette méthode, nous utilisons le fait que les potentiels de désordre des quatre cartes de MGL sont identiques car la seule différence provient de la tension appliquée sur la grille. Il est donc possible d'évaluer la qualité de la prédiction en comparant la ressemblance des potentiels prédits pour chacune des cartes de MGL. L'image C.10 correspond à un exemple où l'on tente d'estimer la qualité de la prédiction 0. Nous pouvons y voir qu'il est possible de calculer 3 coefficients de corrélation entre la prédiction 0 d'un côté et les prédictions 1, 2 et 3 de l'autre côté. En moyennant les 3 coefficients, nous obtenons ce que nous pouvons appeler le coefficient moyen. En parallèle, puisqu'il s'agit d'échantillons simulés, nous pouvons calculer le coefficient de corrélation exact entre la prédiction et le potentiel souhaité. Ainsi pour chaque configuration de désordre, il est possible d'obtenir 4 paires de coefficients exacts et coefficients moyens. En étudiant plusieurs dizaines de configurations de désordres, nous avons pu observer une claire corrélation entre ces quantités. Il est donc possible de déterminer le coefficient exact à partir du coefficient moyen. En utilisant cette méthode, la qualité de la prédiction du potentiel de désordre présent dans l'hétérostructure est estimée à 70%.

Afin d'augmenter la précision du réseau de neurones, nous allons prendre en compte la présence de la vibration de la pointe durant les mesures. En effet, ces vibrations ont changé la façon dont la pointe est venue perturber le transport des électrons. Ainsi, la carte de MGL obtenue par l'expérience correspond à une convolution de cartes MGL correspondant aux différentes perturbations engendrées par la pointe. Afin de simuler la présence de la vibration de la pointe, nous pouvons à nouveau faire une augmentation de données en convoluant l'image de MGL avec des Gaussiennes de différentes largeurs. Avec ce nouvel élément pris en compte dans l'entraînement, il est désormais possible d'obtenir des nouvelles prédictions plus précises car elles sont maintenant estimées comme étant justes avec une corrélation autour de 85%. Ces prédictions sont présentées sur la figure C.11.

## C.5 Conclusion

En raison de l'incertitude d'obtenir des résultats positifs pour l'extraction d'informations sur le désordre à partir de données de transport quantiques avec des méthodes d'apprentissage automatiques, une étude préliminaire a été entreprise en se concentrant sur un problème simplifié. Encouragés par les résultats positifs de cette étude initiale, nous avons abordé le problème plus complexe de l'extraction de toutes les informations sur le potentiel de désordre. Enfin, nous avons déterminé le paysage complet du potentiel de désordre d'un échantillon réel.

Dans cette démarche, une première étape a consisté à extraire les paramètres globaux caractérisant le désordre à partir de la DPLE. Cette étude a révélé

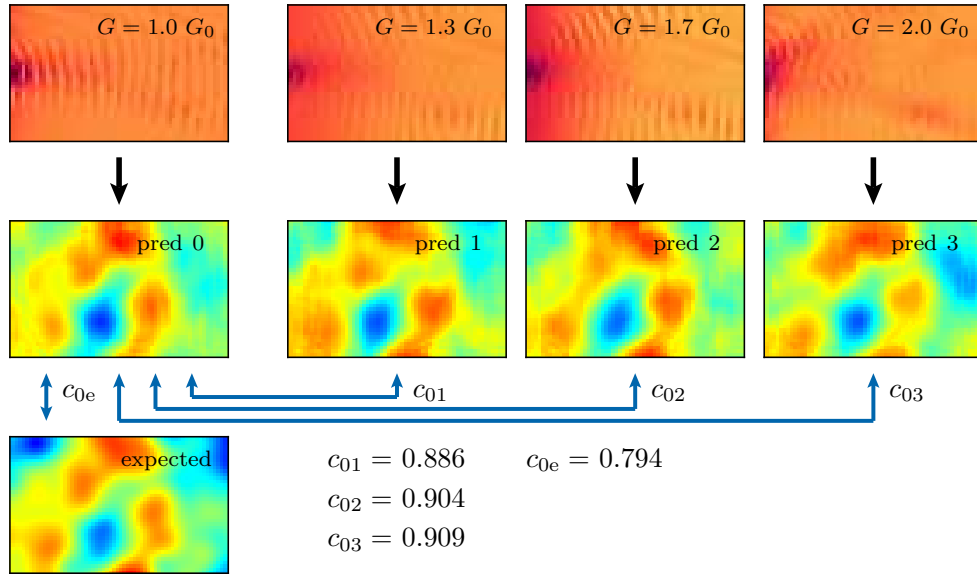


FIGURE C.10: Les quatre panneaux supérieurs représentent des images SGM simulées d'un échantillon avec un désordre fixe mais sous différentes tensions de grille avec les conductances non perturbées indiquées. Les flèches noires représentent le passage à travers le réseau de neurone. Les quatre panneaux en dessous des images de MGL correspondent aux prédictions de potentiel du réseau neuronal. L'image inférieure est le potentiel de désordre exact qui correspond à ces quatre images de MGL. Pour l'exemple  $i = 0$ , les flèches bleues relient les images parmi lesquelles sont calculés les coefficients de corrélation  $c_{ij}$  et  $c_{ie}$ .

des différences significatives dans la qualité des prédictions entre deux ensembles de paramètres, l'un non corrélé et l'autre corrélé.

La détermination complète du paysage de désordre a été réalisée en utilisant la DPLE comme entrée pour un réseau neuronal. Les résultats obtenus étaient remarquables, avec une corrélation moyenne de près de 99,7% lors de l'utilisation du paysage de désordre en espace réel en sortie. Nous avons aussi montré la possibilité d'augmenter la précision du réseau de neurones en fournissant à ce dernier plusieurs informations de transport correspondant au même désordre mais obtenues dans des conditions expérimentales différentes.

Pour finir, une méthodologie pour déterminer le désordre dans une hétérostructure a été développée. Grâce aux méthodes proposées, il a été possible de déterminer le désordre avec une précision très satisfaisante. Afin d'améliorer la précision, des recommandations pour les expériences SGM ont été fournies, suggérant des conditions expérimentales optimales, la résolution d'images appropriées et l'utilisation de transfert d'apprentissage. Une méthode pour l'évaluation de la fiabilité des prédictions sans connaître le potentiel exact a également été proposée.

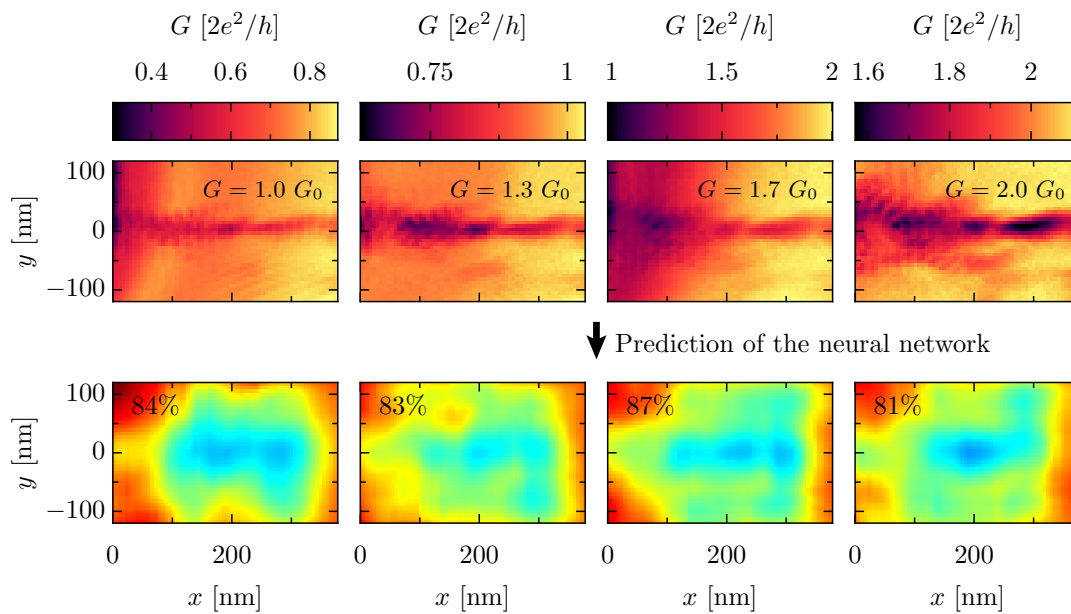


FIGURE C.11: La ligne supérieure représente les quatre cartes SGM expérimentales obtenues aux valeurs de conductance QPC indiquées. La deuxième ligne montre les prédictions correspondantes du potentiel de désordre obtenues à partir du réseau de neurone entraîné lorsqu'il est alimenté par les données de MGL expérimentales en entrée.



# Bibliography

- [1] B. J. van Wees, H. van Houten, C. W. J. Beenakker, J. G. Williamson, L. P. Kouwenhoven, D. van der Marel, and C. T. Foxon, “Quantized conductance of point contacts in a two-dimensional electron gas”, *Phys. Rev. Lett.* **60**, 848–850 (1988) (cited on pages 1, 35–36, 124).
- [2] R. A. Webb, S. Washburn, C. P. Umbach, and R. B. Laibowitz, “Observation of  $\frac{h}{e}$  aharonov-bohm oscillations in normal-metal rings”, *Phys. Rev. Lett.* **54**, 2696–2699 (1985) (cited on page 1).
- [3] S. Datta, “Conductance from transmission”, in *Electronic transport in mesoscopic systems*, Cambridge Studies in Semiconductor Physics and Microelectronic Engineering (Cambridge University Press, 1995), pages 48–116 (cited on pages 1, 23).
- [4] M. Yamamoto, S. Takada, C. Bäuerle, K. Watanabe, A. D. Wieck, and S. Tarucha, “Electrical control of a solid-state flying qubit”, *Nature Nanotechnology* **7**, 247–251 (2012) (cited on pages 1–2).
- [5] J. R. Petta, A. C. Johnson, J. M. Taylor, E. A. Laird, A. Yacoby, M. D. Lukin, C. M. Marcus, M. P. Hanson, and A. C. Gossard, “Coherent manipulation of coupled electron spins in semiconductor quantum dots”, *Science* **309**, 2180–2184 (2005), eprint: <https://www.science.org/doi/pdf/10.1126/science.1116955> (cited on page 1).
- [6] B. Sacépé, M. Feigel’man, and T. Klapwijk, “Quantum breakdown of superconductivity in low-dimensional materials”, *Nature Physics* **16**, 734–746 (2020) (cited on page 1).
- [7] J. Martin, N. Akerman, G. Ulbricht, T. Lohmann, J. H. Smet, K. von Klitzing, and A. Yacoby, “Observation of electron–hole puddles in graphene using a scanning single-electron transistor”, *Nature Physics* **4**, 144–148 (2008) (cited on page 1).
- [8] S. Feng, P. A. Lee, and A. D. Stone, “Sensitivity of the conductance of a disordered metal to the motion of a single atom: implications for  $1/f$  noise”, *Phys. Rev. Lett.* **56**, 1960–1963 (1986) (cited on page 1).
- [9] P. A. Lee and A. D. Stone, “Universal conductance fluctuations in metals”, *Phys. Rev. Lett.* **55**, 1622–1625 (1985) (cited on page 1).
- [10] T. J. Thornton, M. Pepper, H. Ahmed, G. J. Davies, and D. Andrews, “Universal conductance fluctuations and electron coherence lengths in a narrow two-dimensional electron gas”, *Phys. Rev. B* **36**, 4514–4517 (1987) (cited on page 1).



- [11] I. J. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, <http://www.deeplearningbook.org> (MIT Press, Cambridge, MA, USA, 2016) (cited on pages 1, 7, 10, 16).
- [12] Y. Saito, K. Shin, K. Terayama, S. Desai, M. Onga, Y. Nakagawa, Y. M. Itahashi, Y. Iwasa, M. Yamada, and K. Tsuda, “Deep-learning-based quality filtering of mechanically exfoliated 2d crystals”, *npj Computational Materials* **5**, 124 (2019) (cited on page 1).
- [13] N. M. van Esbroeck, D. T. Lennon, H. Moon, V. Nguyen, F. Vigneau, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. D. Briggs, D. Sejdinovic, and N. Ares, “Quantum device fine-tuning using unsupervised embedding learning”, *New Journal of Physics* **22**, 095003 (2020) (cited on page 1).
- [14] J. M. Ede, “Deep learning in electron microscopy”, *Machine Learning: Science and Technology* **2**, 011004 (2021) (cited on pages 1–2).
- [15] S. Akers, E. Kautz, A. Trevino-Gavito, M. Olszta, B. E. Matthews, L. Wang, Y. Du, and S. R. Spurgeon, “Rapid and flexible segmentation of electron microscopy data using few-shot machine learning”, *npj Computational Materials* **7**, 187 (2021) (cited on page 1).
- [16] G. J. Percebois and D. Weinmann, “Deep neural networks for inverse problems in mesoscopic physics: characterization of the disorder configuration from quantum transport properties”, *Phys. Rev. B* **104**, 075422 (2021) (cited on pages 1, 41, 55, 103, 122).
- [17] J. R. Taylor, J. D. Sau, and S. D. Sarma, *Machine learning majorana nanowire disorder landscape*, 2023, eprint: [arXiv:2307.11068](https://arxiv.org/abs/2307.11068) (cited on pages 1–2).
- [18] T. Kacprzak and J. Fluri, “Deepplss: breaking parameter degeneracies in large-scale structure with deep-learning analysis of combined probes”, *Phys. Rev. X* **12**, 031029 (2022) (cited on page 2).
- [19] J. Chardin and P. Bianchini, “Predicting images for the dynamics of stellar clusters (pi-DOC): a deep learning framework to predict mass, distance, and age of globular clusters”, *Monthly Notices of the Royal Astronomical Society* **504**, 5656–5670 (2021), eprint: <https://academic.oup.com/mnras/article-pdf/504/4/5656/38103969/stab737.pdf> (cited on page 2).
- [20] R. Sadre, C. Ophus, A. Butko, and G. H. Weber, “Deep learning segmentation of complex features in atomic-resolution phase-contrast transmission electron microscopy images”, *Microscopy and Microanalysis* **27**, 804–814 (2021) (cited on page 2).
- [21] T. Ando, “Self-consistent results for a GaAs/Al<sub>x</sub>As heterojunction. I. subband structure and light-scattering spectra”, *Journal of the Physical Society of Japan* **51**, 3893–3899 (1982) (cited on page 2).
- [22] W. G. van der Wiel, S. De Franceschi, J. M. Elzerman, T. Fujisawa, S. Tarucha, and L. P. Kouwenhoven, “Electron transport through double quantum dots”, *Rev. Mod. Phys.* **75**, 1–22 (2002) (cited on page 2).

- [23] H. van Houten and C. Beenakker, “Quantum point contacts”, *Physics Today* **49**, 22–27 (1996) (cited on page 2).
- [24] R. Dingle, H. L. Störmer, A. C. Gossard, and W. Wiegmann, “Electron mobilities in modulation-doped semiconductor heterojunction superlattices”, *Applied Physics Letters* **33**, 665–667 (1978) (cited on page 2).
- [25] M. A. Topinka, B. J. LeRoy, S. E. J. Shaw, E. J. Heller, R. M. Westervelt, K. D. Maranowski, and A. C. Gossard, “Imaging coherent electron flow from a quantum point contact”, *Science* **289**, 2323–2326 (2000) (cited on pages 2, 37–38).
- [26] M. A. Topinka, B. J. LeRoy, R. M. Westervelt, S. E. J. Shaw, R. Fleischmann, E. J. Heller, K. D. Maranowski, and A. C. Gossard, *Coherent branched flow in a two-dimensional electron gas*, 2001 (cited on pages 2, 23, 37–38, 124).
- [27] M. P. Jura, M. A. Topinka, L. Urban, A. Yazdani, H. Shtrikman, L. N. Pfeiffer, K. W. West, and D. Goldhaber-Gordon, “Unexpected features of branched flow through high-mobility two-dimensional electron gases”, *Nature Physics* **3**, 841–845 (2007) (cited on pages 2, 37–38).
- [28] D. L. Craig, H. Moon, F. Fedele, D. T. Lennon, B. van Straaten, F. Vigneau, L. C. Camenzind, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, D. Sejdinovic, and N. Ares, “Bridging the reality gap in quantum devices with physics-aware machine learning”, *Phys. Rev. X* **14**, 011001 (2024) (cited on page 2).
- [29] C. R. da Cunha, N. Aoki, D. K. Ferry, and Y.-C. Lai, “A method for finding the background potential of quantum devices from scanning gate microscopy data using machine learning”, *Machine Learning: Science and Technology* **3**, 025013 (2022) (cited on page 2).
- [30] C. da Cunha, N. Aoki, D. Ferry, A. Velasquez, and Y. Zhang, “An investigation of the background potential in quantum constrictions using scanning gate microscopy and a swarming algorithm”, *Physica A: Statistical Mechanics and its Applications* **614**, 128550 (2023) (cited on page 2).
- [31] T. M. Mitchell, *Machine learning* (McGraw-Hill, New York, 1997) (cited on page 5).
- [32] A. Yang, W. Zhang, J. Wang, K. Yang, Y. Han, and L. Zhang, “Review on the application of machine learning algorithms in the sequence data mining of dna”, *Frontiers in Bioengineering and Biotechnology* **8**, 10.3389/fbioe.2020.01032 (2020) (cited on page 5).
- [33] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli,

- and D. Hassabis, “Highly accurate protein structure prediction with alphafold”, *Nature* **596**, 583–589 (2021) (cited on page 5).
- [34] D. Rolnick, P. L. Donti, L. H. Kaack, K. Kochanski, A. Lacoste, K. Sankaran, A. S. Ross, N. Milojevic-Dupont, N. Jaques, A. Waldman-Brown, A. Luccioni, T. Maharaj, E. D. Sherwin, S. K. Mukkavilli, K. P. Kording, C. Gomes, A. Y. Ng, D. Hassabis, J. C. Platt, F. Creutzig, J. Chayes, and Y. Bengio, *Tackling climate change with machine learning*, 2019, eprint: [arXiv:1906.05433](https://arxiv.org/abs/1906.05433) (cited on page 5).
- [35] E. Bedolla, L. C. Padierna, and R. Castañeda-Priego, “Machine learning for condensed matter physics”, *Journal of Physics: Condensed Matter* **33**, 053001 (2020) (cited on page 5).
- [36] J. Schmidt, M. R. G. Marques, S. Botti, and M. A. L. Marques, “Recent advances and applications of machine learning in solid-state materials science”, *npj Computational Materials* **5**, 83 (2019) (cited on page 5).
- [37] M. Krenn, J. Landgraf, T. Foesel, and F. Marquardt, “Artificial intelligence and machine learning for quantum technologies”, *Phys. Rev. A* **107**, 010101 (2023) (cited on page 5).
- [38] V. Walter, C. Ruscher, O. Benzerara, C. M. Marques, and F. Thalmann, “A machine learning study of the two states model for lipid bilayer phase transitions”, *Phys. Chem. Chem. Phys.* **22**, 19147–19154 (2020) (cited on page 6).
- [39] B. S. Rem, N. Käming, M. Tarnowski, L. Asteria, N. Fläschner, C. Becker, K. Sengstock, and C. Weitenberg, “Identifying quantum phase transitions using artificial neural networks on experimental data”, *Nature Physics* **15**, 917–920 (2019) (cited on page 6).
- [40] J. Carrasquilla and R. G. Melko, “Machine learning phases of matter”, *Nature Physics* **13**, 431–434 (2017) (cited on page 6).
- [41] E. P. L. van Nieuwenburg, Y.-H. Liu, and S. D. Huber, “Learning phase transitions by confusion”, *Nature Physics* **13**, 435–439 (2017) (cited on page 6).
- [42] N. Claussen, B. A. Bernevig, and N. Regnault, “Detection of topological materials with machine learning”, *Phys. Rev. B* **101**, 245117 (2020) (cited on page 6).
- [43] A. Blania, S. Herbig, F. Dechert, E. van Nieuwenburg, and F. Marquardt, *Deep learning of spatial densities in inhomogeneous correlated quantum systems*, 2022, eprint: [arXiv:2211.09050](https://arxiv.org/abs/2211.09050) (cited on page 6).
- [44] J. F. Rodriguez-Nieva and M. S. Scheurer, “Identifying topological order through unsupervised machine learning”, *Nature Physics* **15**, 790–795 (2019) (cited on page 6).
- [45] D. Bondarenko and P. Feldmann, “Quantum autoencoders to denoise quantum data”, *Phys. Rev. Lett.* **124**, 130502 (2020) (cited on page 6).

- [46] N. M. van Esbroeck, D. T. Lennon, H. Moon, V. Nguyen, F. Vigneau, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. D. Briggs, D. Sejdinovic, and N. Ares, “Quantum device fine-tuning using unsupervised embedding learning”, *New Journal of Physics* **22**, 095003 (2020) (cited on page 6).
- [47] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: a brief survey”, *IEEE Signal Processing Magazine* **34**, 26–38 (2017) (cited on page 6).
- [48] R. S. Sutton and A. G. Barto, *Reinforcement learning*, 2nd edition, Adaptive Computation and Machine Learning series (Bradford Books, Cambridge, MA, Nov. 2018) (cited on page 6).
- [49] R. Porotti, A. Essig, B. Huard, and F. Marquardt, “Deep reinforcement learning for quantum state preparation with weak nonlinear measurements”, (2021), [arXiv:2107.08816 \[quant-ph\]](https://arxiv.org/abs/2107.08816) (cited on page 7).
- [50] F. Marquardt, “Machine learning and quantum devices”, *SciPost Phys. Lect. Notes*, **29** (2021) (cited on page 7).
- [51] H. Moon, D. T. Lennon, J. Kirkpatrick, N. M. van Esbroeck, L. C. Camenzind, L. Yu, F. Vigneau, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, D. Sejdinovic, E. A. Laird, and N. Ares, “Machine learning enables completely automatic tuning of a quantum device faster than human experts”, *Nature Communications* **11**, 4161 (2020) (cited on page 7).
- [52] V. Nguyen, S. B. Orbell, D. T. Lennon, H. Moon, F. Vigneau, L. C. Camenzind, L. Yu, D. M. Zumbühl, G. A. D. Briggs, M. A. Osborne, D. Sejdinovic, and N. Ares, “Deep reinforcement learning for efficient measurement of quantum devices”, *npj Quantum Information* **7**, 100 (2021) (cited on page 7).
- [53] M. Raissi, P. Perdikaris, and G. Karniadakis, “Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations”, *Journal of Computational Physics* **378**, 686–707 (2019) (cited on page 7).
- [54] K. Beer, D. Bondarenko, T. Farrelly, T. J. Osborne, R. Salzmann, D. Scheiermann, and R. Wolf, “Training deep quantum neural networks”, *Nature Communications* **11**, 808 (2020) (cited on page 7).
- [55] J. Romero, J. P. Olson, and A. Aspuru-Guzik, “Quantum autoencoders for efficient compression of quantum data”, *Quantum Science and Technology* **2**, 045001 (2017) (cited on page 7).
- [56] A. Pepper, N. Tischler, and G. J. Pryde, “Experimental realization of a quantum autoencoder: the compression of qutrits via machine learning”, *Phys. Rev. Lett.* **122**, 060501 (2019) (cited on page 7).
- [57] S. Skansi, *Introduction to deep learning: from logical calculus to artificial intelligence*, 1st edition (Springer Publishing Company, Incorporated, 2018) (cited on page 7).

- [58] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity", *The bulletin of mathematical biophysics* **5**, 115–133 (1943) (cited on page 7).
- [59] J. Fan, L. Fang, J. Wu, Y. Guo, and Q. Dai, "From brain science to artificial intelligence", *Engineering* **6**, 248–252 (2020) (cited on page 8).
- [60] A. F. Agarap, *Deep learning using rectified linear units (relu)*, 2018, eprint: [arXiv:1803.08375](https://arxiv.org/abs/1803.08375) (cited on page 9).
- [61] V. Nair and G. Hinton, "Rectified linear units improve restricted Boltzmann machines Vinod Nair", in, Vol. 27 (June 2010), pages 807–814 (cited on page 9).
- [62] A. L. Maas, "Rectifier nonlinearities improve neural network acoustic models", in (2013) (cited on page 9).
- [63] S. G. K. Patro and K. K. Sahu, *Normalization: a preprocessing stage*, 2015, eprint: [arXiv:1503.06462](https://arxiv.org/abs/1503.06462) (cited on page 10).
- [64] H. J. Kelley, "Gradient theory of optimal flight paths", *Ars Journal* **30**, 947–954 (1960) (cited on page 11).
- [65] D. P. Kingma and J. Ba, *Adam: a method for stochastic optimization*, 2014, eprint: [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (cited on pages 12–13).
- [66] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization", *Journal of Machine Learning Research* **12**, 2121–2159 (2011) (cited on page 12).
- [67] T. Tieleman and G. Hinton, *Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude*. 2012 (cited on page 13).
- [68] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Y. Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, *TensorFlow: large-scale machine learning on heterogeneous systems*, Software available from [tensorflow.org](https://tensorflow.org), 2015 (cited on page 13).
- [69] F. Chollet et al., *Keras*, <https://github.com/fchollet/keras>, 2015 (cited on page 13).
- [70] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", *Journal of Machine Learning Research* **15**, 1929–1958 (2014) (cited on page 15).
- [71] P. Refaeilzadeh, L. Tang, and H. Liu, "Cross-validation", in *Encyclopedia of database systems*, edited by L. LIU and M. T. ÖZSU (Springer US, Boston, MA, 2009), pages 532–538 (cited on page 15).

- [72] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: a large-scale hierarchical image database”, in 2009 IEEE conference on computer vision and pattern recognition (Ieee, 2009), pages 248–255 (cited on page 16).
- [73] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks”, in *Advances in neural information processing systems*, Vol. 25, edited by F. Pereira, C. Burges, L. Bottou, and K. Weinberger (2012) (cited on page 16).
- [74] W. Zhang, J. Tanida, K. Itoh, and Y. Ichioka, “Shift-invariant pattern recognition neural network and its optical architecture”, in Proceedings of annual conference of the Japan Society of Applied Physics (Montreal, CA, 1988), pages 2147–2151 (cited on page 16).
- [75] K. Fukushima, “Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position”, *Biological Cybernetics* **36**, 193–202 (1980) (cited on page 16).
- [76] O. Ronneberger, P. Fischer, and T. Brox, *U-net: convolutional networks for biomedical image segmentation*, 2015, eprint: [arXiv:1505.04597](https://arxiv.org/abs/1505.04597) (cited on page 19).
- [77] K.-S. Oh and K. Jung, “GPU implementation of neural networks”, *Pattern Recognition* **37**, 1311–1314 (2004) (cited on page 19).
- [78] J. Kosaian and A. Phanishayee, *A study on the intersection of gpu utilization and cnn inference*, 2022, eprint: [arXiv:2212.07936](https://arxiv.org/abs/2212.07936) (cited on page 20).
- [79] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, *Generative adversarial networks*, 2014, eprint: [arXiv:1406.2661](https://arxiv.org/abs/1406.2661) (cited on page 20).
- [80] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, *Image-to-image translation with conditional adversarial networks*, 2016, eprint: [arXiv:1611.07004](https://arxiv.org/abs/1611.07004) (cited on page 20).
- [81] H. Bartolomei, M. Kumar, R. Bisognin, A. Marguerite, J.-M. Berroir, E. Bocquillon, B. Plaçais, A. Cavanna, Q. Dong, U. Gennser, Y. Jin, and G. Fève, “Fractional statistics in anyon collisions”, *Science* **368**, 173–177 (2020), eprint: <https://www.science.org/doi/pdf/10.1126/science.aaz5601> (cited on page 23).
- [82] S. Takada, H. Edlbauer, H. V. Lepage, J. Wang, P.-A. Mortemousque, G. Georgiou, C. H. W. Barnes, C. J. B. Ford, M. Yuan, P. V. Santos, X. Waintal, A. Ludwig, A. D. Wieck, M. Urdampilleta, T. Meunier, and C. Bäuerle, “Sound-driven single-electron transfer in a circuit of coupled quantum rails”, *Nature Communications* **10**, 4557 (2019) (cited on page 23).
- [83] E. Buks, M. Heiblum, Y. Levinson, and H. Shtrikman, “Scattering of a two-dimensional electron gas by a correlated system of ionized donors”, *Semiconductor Science and Technology* **9**, 2031 (1994) (cited on pages 23, 30, 115).

- [84] R. A. Jalabert, “Mesoscopic transport and quantum chaos”, *Scholarpedia* **11**, revision #154013, 30946 (2016) (cited on pages 23, 26).
- [85] D. Ryndyk, *Theory of quantum transport at nanoscale* (Springer International Publishing, 2016) (cited on page 23).
- [86] M. Büttiker, Y. Imry, R. Landauer, and S. Pinhas, “Generalized many-channel conductance formula with application to small rings”, *Phys. Rev. B* **31**, 6207–6215 (1985) (cited on page 27).
- [87] C. W. Groth, M. Wimmer, A. R. Akhmerov, and X. Waintal, “Kwant: a software package for quantum transport”, *New Journal of Physics* **16**, 063065 (2014) (cited on pages 28, 87, 128).
- [88] R. Peierls, “Zur theorie des diamagnetismus von leitungselektronen”, *Zeitschrift für Physik* **80**, 763–791 (1933) (cited on page 28).
- [89] A. Y. Cho, “Recent developments in molecular beam epitaxy (MBE)”, *Journal of Vacuum Science and Technology* **16**, 275–284 (1979) (cited on page 29).
- [90] T. De Jong, W. Douma, S. Doorn, and F. Saris, “Modulation doping structures in silicon by molecular beam epitaxy and off-line ion implantation”, *Materials Letters* **1**, 157–162 (1983) (cited on page 29).
- [91] T. Ihn, *Semiconductor nanostructures* (Oxford Press, 2009) Chap. 9 (cited on pages 30, 32, 36, 123–124).
- [92] E. Buks, M. Heiblum, and H. Shtrikman, “Correlated charged donors and strong mobility enhancement in a two-dimensional electron gas”, *Phys. Rev. B* **49**, 14790–14793 (1994) (cited on pages 29–30).
- [93] O. Ly, R. A. Jalabert, S. Tomsovic, and D. Weinmann, “Partial local density of states from scanning gate microscopy”, *Phys. Rev. B* **96**, 125439 (2017) (cited on pages 30, 38–39).
- [94] J. H. Davies, I. A. Larkin, and E. V. Sukhorukov, “Modeling the patterned two-dimensional electron gas: Electrostatics”, *Journal of Applied Physics* **77**, 4504–4512 (1995), eprint: [https://pubs.aip.org/aip/jap/article-pdf/77/9/4504/8044741/4504\\_1\\_online.pdf](https://pubs.aip.org/aip/jap/article-pdf/77/9/4504/8044741/4504_1_online.pdf) (cited on page 34).
- [95] M. A. Topinka, “Imaging coherent electron wave flow through 2-d electron gas nanostructures”, PhD thesis (Harvard, 2002), pages 59–63 (cited on page 35).
- [96] A. Iordanescu, S. Toussaint, G. Bachelier, S. Fallahi, C. G. Gardner, M. J. Manfra, B. Hackens, and B. Brun, “Accurate characterization of tip-induced potential using electron interferometry”, *Applied Physics Letters* **117**, 193101 (2020) (cited on pages 35, 77, 95, 106).
- [97] T. J. Thornton, M. Pepper, H. Ahmed, D. Andrews, and G. J. Davies, “One-dimensional conduction in the 2d electron gas of a gaas-algaas heterojunction”, *Phys. Rev. Lett.* **56**, 1198–1201 (1986) (cited on page 35).

- [98] D. A. Wharam, T. J. Thornton, R. Newbury, M. Pepper, H. Ahmed, J. E. F. Frost, D. G. Hasko, D. C. Peacock, D. A. Ritchie, and G. A. C. Jones, “One-dimensional transport and the quantisation of the ballistic resistance”, *Journal of Physics C: Solid State Physics* **21**, L209 (1988) (cited on page 35).
- [99] M. Büttiker, “Quantized transmission of a saddle-point constriction”, *Phys. Rev. B* **41**, 7906–7909 (1990) (cited on page 36).
- [100] H. A. Fertig and B. I. Halperin, “Transmission coefficient of an electron through a saddle-point potential in a magnetic field”, *Phys. Rev. B* **36**, 7969–7976 (1987) (cited on page 36).
- [101] K. R. Fratus, R. A. Jalabert, and D. Weinmann, “Energy stability of branching in the scanning gate response of two-dimensional electron gases with smooth disorder”, *Phys. Rev. B* **100**, 155435 (2019) (cited on pages 37, 73).
- [102] M. P. Jura, M. A. Topinka, M. Grobis, L. N. Pfeiffer, K. W. West, and D. Goldhaber-Gordon, “Electron interferometer formed with a scanning probe tip and quantum point contact”, *Phys. Rev. B* **80**, 041303 (2009) (cited on page 38).
- [103] B. Brun, F. Martins, S. Faniel, B. Hackens, G. Bachelier, A. Cavanna, C. Ulysse, A. Ouerghi, U. Gennser, D. Mailly, S. Huant, V. Bayot, M. Sanquer, and H. Sellier, “Wigner and kondo physics in quantum point contacts revealed by scanning gate microscopy”, *Nature Communications* **5**, 4290 (2014) (cited on page 38).
- [104] H. Sellier, B. Hackens, M. G. Pala, F. Martins, S. Baltazar, X. Wallart, L. Desplanque, V. Bayot, and S. Huant, “On the imaging of electron transport in semiconductor quantum structures by scanning-gate microscopy: successes and limitations”, *Semiconductor Science and Technology* **26**, 064008 (2011) (cited on page 38).
- [105] B. Hackens, F. Martins, T. Ouisse, H. Sellier, S. Bollaert, X. Wallart, A. Cappy, J. Chevrier, V. Bayot, and S. Huant, “Imaging and controlling electron transport inside a quantum ring”, *Nature Physics* **2**, 826–830 (2006) (cited on page 38).
- [106] R. Steinacher, C. Pörtl, T. Krähenmann, A. Hofmann, C. Reichl, W. Zwerger, W. Wegscheider, R. A. Jalabert, K. Ensslin, D. Weinmann, and T. Ihn, “Scanning gate experiments: from strongly to weakly invasive probes”, *Phys. Rev. B* **98**, 075426 (2018) (cited on page 38).
- [107] C. Gold, B. A. Bräm, M. S. Ferguson, T. Krähenmann, A. Hofmann, R. Steinacher, K. R. Fratus, C. Reichl, W. Wegscheider, D. Weinmann, K. Ensslin, and T. Ihn, “Imaging signatures of the local density of states in an electronic cavity”, *Phys. Rev. Res.* **3**, L032005 (2021) (cited on pages 38, 41).



- [108] K. R. Fratus, C. L. Calonnec, R. A. Jalabert, G. Weick, and D. Weinmann, "Signatures of folded branches in the scanning gate microscopy of ballistic electronic cavities", *SciPost Phys.* **10**, 069 (2021) (cited on page 38).
- [109] M. G. Pala, S. Baltazar, P. Liu, H. Sellier, B. Hackens, F. Martins, V. Bayot, X. Wallart, L. Desplanque, and S. Huant, "Transport inefficiency in branched-out mesoscopic networks: an analog of the braess paradox", *Phys. Rev. Lett.* **108**, 076802 (2012) (cited on page 38).
- [110] F. Martins, B. Hackens, M. G. Pala, T. Ouisse, H. Sellier, X. Wallart, S. Bollaert, A. Cappy, J. Chevrier, V. Bayot, and S. Huant, "Imaging electron wave functions inside open quantum rings", *Phys. Rev. Lett.* **99**, 136807 (2007) (cited on page 38).
- [111] M. Sundararajan, A. Taly, and Q. Yan, *Axiomatic attribution for deep networks*, 2017, eprint: [arXiv:1703.01365](https://arxiv.org/abs/1703.01365) (cited on page 50).
- [112] S. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, 2017, eprint: [arXiv:1705.07874](https://arxiv.org/abs/1705.07874) (cited on page 50).
- [113] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should i trust you?": explaining the predictions of any classifier, 2016, eprint: [arXiv:1602.04938](https://arxiv.org/abs/1602.04938) (cited on page 50).
- [114] *Machine learning and unintended consequences*, <https://www.lesswrong.com/posts/5o3CxyvZ2XKawRB5w/machine-learning-and-unintended-consequences>, Accessed: 2023-06-12, Sept. 2011 (cited on page 50).
- [115] P. Sturmfels, S. Lundberg, and L. Su-In, "Visualizing the impact of feature attribution baselines", *Distill*, **10.23915/distill.00022** (2020) (cited on page 52).
- [116] P. Armagnat, A. Lacerda-Santos, B. Rossignol, C. Groth, and X. Waintal, "The self-consistent quantum-electrostatic problem in strongly non-linear regime", *SciPost Phys.* **7**, 031 (2019) (cited on pages 77, 81, 87).
- [117] G. J. Percebois, A. Lacerda-Santos, B. Brun, B. Hackens, X. Waintal, and D. Weinmann, "Reconstructing the potential configuration in a high-mobility semiconductor heterostructure with scanning gate microscopy", *SciPost Phys.* **15**, 242 (2023) (cited on pages 77, 95).
- [118]  $\mu$ Masch, "afm conductive tips" (cited on page 78).
- [119] E. Chatzikyriakou, J. Wang, L. Mazzella, A. Lacerda-Santos, M. C. d. S. Figueira, A. Trellakis, S. Birner, T. Grange, C. Bäuerle, and X. Waintal, "Unveiling the charge distribution of a gaas-based nanoelectronic device: a large experimental dataset approach", *Phys. Rev. Res.* **4**, 043163 (2022) (cited on page 115).
- [120] X. Chen, G. Weick, D. Weinmann, and R. A. Jalabert, "Scanning gate microscopy in graphene nanostructures", *Phys. Rev. B* **107**, 085420 (2023) (cited on page 115).



# Quantum transport in two-dimensional systems : Artificial intelligence applied to material science

## Résumé

Les hétérostructures représentent l'un des dispositifs les plus largement utilisés pour générer un gaz d'électrons bidimensionnel (2DEG). Cependant, les propriétés exactes de transport des électrons au sein de tels dispositifs ne sont pas complètement contrôlables et prévisibles. Cela est principalement dû à la présence de dopants ionisés, nécessaires à la création du gaz d'électrons, qui impactent le mouvement des électrons par le biais d'interactions coulombiennes, entraînant ainsi un potentiel de désordre. Dans cette étude, nous introduisons une méthode pour déterminer ce désordre en utilisant des données de transport locales obtenues à partir d'expériences de microscopie à grille locale (SGM). La correspondance entre les données capturées dans une image SGM et le potentiel de désordre est réalisée grâce à l'utilisation d'un algorithme de deep learning. Nous avons démontré que ce problème inverse peut être résolu, et avons déterminé le potentiel de désordre au sein d'une véritable hétérostructure expérimentale. Dans ce manuscrit, nous présentons en détail la méthodologie employée, la rendant ainsi reproductible pour d'autres dispositifs.

Mots-clés : physique mésoscopique, gaz d'électrons bidimensionnel, microscopie à grille locale, désordre, hétérostructure, points de contacts quantiques, deep learning, intelligence artificielle

## Résumé en anglais

Heterostructures represent one of the most widely employed devices for generating a two-dimensional electron gas (2DEG). However, the precise transport properties of electrons within such devices remain neither completely controllable nor predictable. This is mainly due to the presence of randomly located ionized dopants, that are necessary for creating the electron gas. However, due to the Coulombic interactions, the electron motion is impacted, which results in a disorder potential. In this study, we introduce a method to determine this disorder potential using local transport information obtained from scanning gate microscopy (SGM) experiments. The mapping between the data captured in an SGM image and the disorder potential is achieved through the utilization of a deep learning algorithm. We have demonstrated that this inverse problem can be solved, and we have been able to determine the disorder potential within a real experimental heterostructure. In this manuscript, we detail the employed methodology, making it replicable for other devices.

Keywords: mesoscopic physics, two-dimensional electron gas, scanning gate microscopy, disorder, heterostructure, quantum point contacts, deep learning, artificial intelligence