



UNIVERSITY *of* LIMERICK

O L L S C O I L L U I M N I G H

Automated Atomic Resolution Open-Source Analysis of Two-Dimensional Entities

Eoghan Noel O'Connell

Department of Physics

Bernal Institute

A thesis presented for the degree of
Doctor of Philosophy

University of Limerick

Supervisor: Prof. Ursel Bangert

Internal Examiner: Dr. Ning Liu

External Examiner: Prof. Rafal Dunin-Borkowski

Submitted to the University of Limerick, June 2020

Automated Atomic Resolution Open-Source Analysis of Two-Dimensional Entities

Eoghan Noel O'Connell

Abstract

Since the first images of single atoms acquired by Crewe and co-workers, scanning transmission electron microscopy (STEM) has become a powerful tool for analysing materials from the sub-atomic to the micron scale. No other technique to date provides the same directly resolved information in structure and chemistry at this spatial resolution. Reliably analysing such data has recently been brought into the open-source sphere for material science. The recent development of Python-based open source electron microscopy software packages has not only made post-processing analysis accessible to all but has also fostered international large-scale collaborations.

In this thesis, we implement established methods as well as new routines for the analysis of atomic resolution STEM data of two-dimensional (2D) entities: monolayer transition metal dichalcogenides (TMDs) and ferroelectric domain walls (DWs). Thus, the 2D entities are investigated both perpendicular and parallel to the electron beam. The developed tools can be used interchangeably between both structures and are therefore extensible for the analysis of other material features at atomic resolution. The tools are contained in the TEMUL toolkit and are integrated with the widely used Hyperspy and Atomap Python packages.

Initially, low energy ion implantation is used to manipulate the atomic structure of 2D TMDs. We report the first successful substitutional ion implantation into monolayer TMDs at 10 eV and quantify the resulting $\text{MoS}_{1.87}\text{Se}_{0.13}$ alloy with annular dark field (ADF) STEM and electron energy loss spectroscopy (EELS). The alloying of the pristine MoS_2 monolayer shifted its band gap by 0.09 eV, confirming the successful manipulation of the band structure of monolayer TMDs via low energy ion implantation. At similar implantation energies (10-25 eV), Au ions produced \sim 3-4 nm diameter nanoparticles (NPs) on MoS_2 and MoSe_2 .

Se implanted MoS_2 monolayers are then utilised for the development of iterative analysis routines for the detection and tracking of atom dynamics in ADF STEM. By utilising fast image simulation, the atomic structure of the ion-implanted monolayers were quantified for large datasets. These open-source routines are not limited to 2D TMDs. Furthermore, the automated routines are compared with manual analysis of the lattice, exposing vacancy swapping and adatom movement. The

effect of the 60 keV electron beam on Se implanted monolayer MoS₂ is investigated with the developed routines. We quantify the damaging effect of the total electron dose and dose rate, and hypothesise a threshold dose rate of $\sim 4.25 \text{ pC nm}^{-2} \text{ s}^{-1}$, above which the monolayer MoS₂ lattice is damaged irreversibly.

Finally, 2D DWs in ferroelectric materials are investigated. Open-source, flexible functions, also included in the TEMUL toolkit, are created for the analysis of sub-atomic shifts and polarisation in such materials. We successfully apply these functions for the quantitative characterisation of charged DWs and vortexes at atomic resolution in ferroelectric boracites, lead titanates and lithium niobate.

Declaration

I hereby declare that this thesis is entirely my own work except where indicated in the Publications & Collaborations, and that all sources used in researching it are fully acknowledged and all quotations properly identified. The thesis has not been submitted, in whole or in part, by me or another person, for the purpose of obtaining any other academic award in this university, or to any other university or higher education institute.

Eoghan Noel O'Connell

This research was conducted with the financial support of the Irish Research Council under the Postgraduate Government of Ireland grant GOIPG/2015/2410.

Acknowledgements

So many people have helped me get to this stage. To my wonderful supervisor Uschi, thank you for your patience, excitement, guidance, and warmth. Thank you, Alan, for the years of empathy in the face of malfunctioning microscopes.

To everyone in the TEMUL group, I cannot articulate how much I value having you as colleagues and friends. Kalani, I am so happy we had so much time together. Thank you for your solidness, for the constant help you gave me, and for bringing me out on the river. Eileen, thanks for being you. Maybe without knowing it you always helped me by showing me that one can have a normal and productive life under such stress. Shelly, thank you for your constant enthusiasm. You have changed my PhD and lifted me up, as you do everyone around you. Thanks to Mike, Eoin, Jenn, and Samad for all the banter, as well as every master's and final year student we have had, I have learned so much from you all.

To Andy, thank you for coming to us on the lonely Fridays for a chat and for your help and guidance throughout the years. Dave, I only wish we could work together more closely, thanks for the chats and the climbing. To Gearóid, thanks for your quick laughter and calming presence. To everyone in the Department of Physics, thank you for the patience and support you have given me over the last 9 or so years. Thanks Shayon, Matthew, Sarah, Cian, Josephine, John Kelly and John Sweeney for the chats and help. Thanks as well to everyone on the third floor and in the Bernal Institute.

Jess, I hope you know how much I value you. Thank you for your constant companionship and for the long ranting chats we continue to have. To Pierce and Colin, I always know I can talk to you. You have helped more than you know. Claude, Siobhán, and Tess, thank you for the banter. Dave and Stuart thank you for always being there. To Oisin and Jason, and everyone I have met in the UL OPC, thank you for exploring the world with me and for giving me another family.

To my family, Noel, Margaret, Ciaran and Emer. Thank you for always caring and for helping me to relax. Linda, ich liebe dich. Your unwavering support is the reason I am here writing this.

Dedication

To my parents, Noel and Margaret, for always supporting us in everything we do.

Publications & Collaborations

Journal publications

- Bangert, U., Stewart, A., **O'Connell, E.**, Courtney, E., Ramasse, Q., Kepaptoglou, D., Hofsäss, H. Amani, J., Tu, J. S., Kardynal, B.
Ion-beam modification of 2-D materials - single implant atom analysis via annular dark-field electron microscopy
Ultramicroscopy, 176, p. 31-36, 2017
- **O'Connell, E.N.**, Hennessy, M., Courtney, E., Ramasse, Q.M., Kardynal, B., Hofsaess, H., Ghorbani-Asl, M., Krasheninnikov, A., Bangert, U.
Utilizing an Open-Source Workflow for the Analysis of Atom Dynamics in Two-Dimensional Materials
Microscopy and Microanalysis, 25, p. 136-137, 2019
- Conroy, M., Moore, K., **O'Connell, E.N.**, Courtney, E., Harvey, A., Cochard, C., Guy, J., McQuaid, R., Jones, L., Downing, C., Whatmore, R., Gregg, M., Bangert, U.
Investigating Ferroelectric Domain and Domain Wall Dynamics at Atomic Resolution by TEM/STEM in situ Heating and Biasing
Microscopy and Microanalysis, 25, p. 1882-1883, 2019
- Conroy, M., Moore, K., **O'Connell, E.N.**, McConville, J.P.V., Lu, H., Chaudhary, P., Lipatov, A., Sinitskii, A., Gruverman, A., Gregg, J.M., Bangert, U.
Atomic-Scale Characterization of Ferro-Electric Domains in Lithium Niobate-revealing the Electronic Properties of Domain Walls
Microscopy and Microanalysis, 25, p. 576-577, 2019
- Conroy, M., Moore, K., **O'Connell, E.N.**, Jones, L., Downing, C., Whamore, R., Gruverman, A., Gregg, J. M., Bangert, U.

Probing the Dynamics of Topologically Protected Charged Ferroelectric Domain Walls with the Electron Beam at the Atomic Scale
Accepted in Microscopy and Microanalysis, 2020

- Hadjimichael, M., Li, Y., Zatterin, E., Chahine, G.A., Conroy, M., **O'Connell, E.N.**, Ondrejkovi, P., Marton, P., Hlinka, J., Bangert, U., Leake, S., Zubko, P.
Direct imaging of artificially-layered ferroelectric domain supercrystals
Submitted to Nature materials, 2020
- Moore, K., Conroy, M., **O'Connell, E.N.**, Cochard, C., Mackel, J., Harvey, A., Bell, A.J., Gregg, M.J., Bangert, U.
Highly Charged 180 ° Head-to-Head Domain Walls in PbTiO₃
Submitted to Nature Communications Physics, 2020
- Moynihan, E., Rost, S., **O'Connell, E.**, Ramasse, Q., Friedrich, C., Bangert, U.
Plasmons in MoS₂ studied via Experimental and Theoretical Correlation of Energy Loss Spectra
Journal of Microscopy, 2020
- Wang, F., Kinloch, I.A., Wolverson, D., Tenne, R., Zak, A., **O'Connell, E.**, Bangert, U., Young, R.J.
Strain-induced phonon shifts in tungsten disulfide nanoplatelets and nanotubes
2D Materials, 4, 2016

Collaborations

The results presented in this thesis are due both to the author and a number of collaborators. The contributions of each collaborator is stated below for each Results chapter.

Results - Atomic Characterisation of Ion Implanted 2D TMDs: Preparation of two-dimensional (2D) transition metal dichalcogenides (TMDs) was carried out by the author in the University of Limerick (UL), and in the Forschungszentrum Jülich with the help of Dr. Florian Winkler and Dr. Jhih-Sian Tu. Low energy ion implantation was completed with the ADONIS implanter by Manuel Auge,

Dr. Julian Alexander Amani and Prof. Hans Hofsäss (Georg-August-Universität Göttingen). Imaging of Se and Au ion-implanted TMDs was carried out on the UL Titan transmission electron microscope (TEM) by the author, with the help of Dr. Alan Harvey, Kalani Moore, Eoin Moynihan, Michael Hennessy, and Eileen Courtney. These images are presented for Au implanted TMDs. Simulations of Au implanted TMDs were carried out by Eileen Courtney. Imaging of Se implanted MoS₂ at 60 keV were acquired in SuperSTEM, Daresbury, U.K., by Prof. Quentin Ramasse and Dr. Demie Kepaptsoglou. The TMD ion implantation project is a collaboration with the groups of Prof. Beata Kardynal (Peter Grünberg Institute (PGI-9), Forschungszentrum Jülich) and Prof. Hans Hofsäss (Georg-August-Universität Göttingen).

Results - Atom Dynamics in 2D TMDs: Movies of Se implanted MoS₂ at 60 keV were acquired in SuperSTEM, Daresbury, U.K., by Prof. Quentin Ramasse. Density functional theory (DFT) calculations of Se antisites and vacancies in monolayer MoS₂ were carried out by Dr. M. Ghorbani-Asl and Dr. A.V. Krasheninnikov. DFT calculations of Au NPs on monolayer TMDs were carried out by Dr. Pierre Cazade.

Results - Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls: Details **Redacted**.

Presentations and Awards

Conference Oral Presentations

- European Microscopy Society 2016, Lyon, France – 2D Materials Session
- Microscience Microscopy Congress 2017, Manchester, U.K. – 2D Materials Session
- International Microscopy Congress 2018, Sydney, Australia – 2D Materials Session
- Microscopy Society of Ireland Symposium 2018, Sligo IT – (Pres. By E. Courtney)
- Microscopy Society of Ireland Symposium 2019, UCD, Dublin – Open-Source Session
- Microscopy & Microanalysis 2019 Meeting, Portland U.S. – (Pres. By E. Courtney)
- PyCon Ireland 2019, Dublin - Scientific Session
- Microscopy Society of Ireland Symposium 2020, TCD, Dublin – Microscopy For ICT
- Invited contribution: Short Informal Maths (SIM) talk, Department of Mathematics, University of Limerick

Conference Poster Presentations

- Graphene Week 2016, Warsaw, Poland
- European Microscopy Society 2016, Lyon, France
- Institute of Physics Ireland Postgraduate Conference: Belfast 2016, Dublin 2017, Limerick 2018
- Microscopy Society of Ireland Symposium 2016, DIT, & 2019, UCD Dublin
- SuperSTEM Summer School 2018, Daresbury, U.K.

Awards

- Best Poster Award in Material Science at Microscopy Society of Ireland Symposium 2016, DIT & 2019, UCD Dublin
- Best Poster Award at SuperSTEM Summer School 2018, Daresbury, U.K.

Contents

Abstract	ii
Declaration	iii
Acknowledgements	iv
Dedication	v
Publications & Collaborations	vi
Presentations and Awards	ix
Table of Contents	xiii
List of Figures	xxxii
List of Tables	xxxiv
Code Segments	xxxvi
List of Programs	xxxvii
Abbreviations	xxxviii
1 Introduction	1
2 Literature Review	5
2.1 Characterisation via Electron Microscopy and Spectroscopy	6
2.1.1 Single Atom ADF STEM & EELS	7
2.1.2 Electron-Matter Interactions	22
2.2 Two-Dimensional Materials and Transition Metal Dichalcogenides .	29
2.2.1 Properties of 2D TMDs	29
2.2.2 Manipulating the Properties of 2D TMDs	33
2.2.3 Ion Implantation of 2D Materials	41

CONTENTS

2.3	Ferroelectric Materials and Domain Walls	47
2.3.1	Ferroelectric Domains and Domain Walls at Atomic Resolution	47
3	Methodology	50
3.1	2D Material Sample Preparation	51
3.1.1	Experimental 2D Stamping Setup	55
3.2	Low Energy Ion Implantation	59
3.2.1	Ion Implanter Setup	59
3.2.2	Ion Implanter Sample Holder	61
3.3	Experimental and Simulation Methods	62
3.3.1	Acquiring Images and Spectra	62
3.3.2	Image Simulation and DFT Calculations	65
3.4	Signal and Data Analysis	67
3.4.1	Element Quantification in HR ADF STEM Images	67
3.4.2	Methods for Analysing Images	72
3.4.3	Methods for Atom Position Analysis	78
3.4.4	Line Intensity Profile	83
3.4.5	Nanoparticle Identification	83
3.4.6	Model Visualisation and Data Formats	84
3.4.7	Methods for Analysing Atomic Movement and Polarisation .	86
3.4.8	Multidimensional Data Analysis	92
3.4.9	Analysis Tools	96
3.4.10	Python and its EM Packages	99
4	Results - Atomic Characterisation of Ion Implanted 2D TMDs	104
4.1	Se Ion Implantation in MoS ₂	105
4.1.1	Microscopy of Se implanted MoS ₂	106
4.1.2	Spectroscopy of Se implanted MoS ₂	127
4.2	Au Ion Implantation on TMDs	139
4.2.1	Distribution of Au NPs on TMDs	140
4.2.2	Comparison of Au Implants with Image Simulation	144
4.3	Chapter Conclusions - Atomic Characterisation of Ion Implanted 2D TMDs	147
5	Results - Atom Dynamics in 2D TMDs	148
5.1	Open Source Development of Automated Atomic Resolution Image Series Analysis Methods	148
5.1.1	Refinement of Atomic Resolution Images via Image Simulation	149
5.1.2	Model Refiner (MR) Tool	160

CONTENTS

5.2 Atom Dynamics in Implanted TMDs with the MR Tool	170
5.2.1 Image Series Automated Analysis Steps	170
5.2.2 Analysis of Lattice Energetics, Ad-atoms and Vacancies with the MR Tool	175
5.2.3 Effect of Electron Beam on TMD Monolayers with the MR Tool	186
5.3 Chapter Conclusions - Atom Dynamics in 2D TMDs	196
6 Results - Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls	197
6.1 Tools for Visualisation of Subatomic Shifts	198
6.1.1 Plot Polarisation Vectors	199
6.1.2 Find Polarisation Vectors	203
6.1.3 Atom Deviation from a Straight Line Fit	206
6.1.4 Combine Atom Deviations from all Zone Axes	209
6.1.5 Get the Average Polarisation in n Unit Cells	211
6.1.6 Delete Atom Planes from Sublattice Zone	213
6.1.7 Material Property Maps: Strain, Angle of Zone and Lattice Spacing Ratio	215
6.2 2D Domain Walls in Ferroelectric Materials	219
6.2.1 PbTiO ₃ Domain Wall Junctions	220
6.3 Chapter Conclusions - Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls	226
7 Conclusions & Future Directions	227
7.1 Overall Conclusions	227
7.2 Future Directions	229
References	232
A Appendix	272
A.1 Literature Review	272
A.1.1 Ferroic Domains	272
A.2 Methodology	273
A.3 Results 1	275
A.3.1 Low loss EELS	277
A.4 Results 2	279
A.5 Results 3	282
A.5.1 PTO Junction	282

CONTENTS

A.6 TEMUL Code	284
A.6.1 Z Contrast	284
A.6.2 Model Refinement Algorithms	295
A.6.3 Model Refiner	310
A.6.4 Some functions of the Polarisation Module	331

List of Figures

1.1	Imaging of two-dimensional (2D) entities perpendicular and parallel with respect to the incident electron beam.	2
2.1	First image of single atoms. Uranium atoms on a thin carbon film. 210 nm full scale. Taken from [22].	7
2.2	BF, ABF, ADF and ADF simulation of few-layer MoSe ₂ imaged at 300 kV (UL Titan). Scale bars are 6.6 Å.	8
2.3	(a) Graphic of relevant detectors used to collect the electron beam signal transmitted through a sample in a STEM. These are bright field (BF), annular bright field (ABF), annular dark field (ADF) and high angle ADF (HAADF) Taken from [37]. (b) Basic angles of an ADF STEM detector. Q_{obj} is the reciprocal radius of the objective aperture. θ_0 and q_0 are the inner collection angle (rad) and inner collection vector (\AA^{-1}) of the ADF detector, respectively, and are equivalent. Taken from [29].	8
2.4	(a) Histogram of intensities recorded in a hBN monolayer. (b) Overlay of the sorted elements from (a) on the hBN monolayer MAADF STEM image. Red is Boron, yellow is Carbon, green is Nitrogen and blue is Oxygen. Taken from [21]. (c) Cross-sectional ADF STEM image of a Janus monolayer of MoSe ₁ S ₁ , taken from [64]. (d-i) ADF STEM image and (d-ii) identified elemental map of Se doped monolayer MoS ₂ (Red is Mo, dark green is S ₂ , bright green is Se ₁ S ₁ , white is Se ₂). Taken from [65].	11
2.5	(a) Comparison between an experimental ADF STEM image and simulated intensities of different atomic column thicknesses in an Au foil and the resulting counts of each sorted into a histogram (y-axis is frequency of atom count). Taken from [45]. (b) Raw ADF STEM of a Pb NP. (c) Number of atoms in each atomic column of the Pb NP, defined by the Gaussian mixture model in (d). Taken from [115].	17

LIST OF FIGURES

2.6	(a) Raw HAADF STEM image of a CsI ionic chain in a double walled CNT. EELS signal of the (b) Cs and (c) I M _{4,5} edges. (d) HAADF image of six atoms in the ionic chain. Line profiles over the Cs and I atoms show in (e) that the I have higher ADF intensity. (a-e) Taken from [100]. (f) HAADF STEM images and (g) CL-EELS L _{2,3} edges of tetravalent (top, red/purple) and trivalent (bottom, green) Si atom in a graphene lattice with their respective atomic models. (f, g) Taken from [52]. (h) Experimental and (i) simulated energy-loss signal from LA-TA + LO-TO phonons (50-200 meV) in a hBN lattice at atomic resolution. Scale bars 2.45 Å. (h, i) Taken from [33].	18
2.7	(a) HAADF STEM image of a CVD-grown MoS ₂ /MoSe ₂ monolayer interface. (b) Exciton intensity across the interface. (c) Exciton and S L _{2,3} edge intensities across the interface. Taken from [78]. . .	19
2.8	(a) Classification of radiation damage in an electron microscope, separated into elastic and inelastic scattering. Taken from [160] (b) Table description given above the table. Blank entries are unknown or not relevant to the mechanism. Taken from [161].	23
2.9	(a) Schematic for the DFT calculation of chalcogen sputtering from the bottom (exit-beam) side of a TMD monolayer. (b) DFT calculation of the displacement threshold energies E_d and vacancy formation energies. (a,b) taken from [166]. (c) Sputtering rates of the chalcogen atom per unit cell as a function of TEM primary beam energy for MoS ₂ , WS ₂ , MoSe ₂ , and WSe ₂ . Taken from [168].	24
2.10	3D, image (top) and cross-sectional views of the structure of the TMD MoS ₂ . The image (top) view is the imaging view, orthogonal to the electron beam. (TMD Bonding) The bonding of an octahedral TMD structure showing the distribution of valence electrons. Taken from [220].	30
2.11	(a) Optical microscope image of monolayer and few-layer MoS ₂ . The large holes are 1.5 μm in diameter. (b) PL signal of the area in (a). (c) PL peak position of the relevant peaks as a function of the number of MoS ₂ layers. (d) Photoconductivity as a function of photon energy for monolayer and bilayer MoS ₂ . (a-d) Taken from [204]. ARPES spectra of (e) monolayer, (f) bilayer, (g) trilayer and (h) eight-layer MoSe ₂ . (e-h) Taken from [231].	31

LIST OF FIGURES

2.12 (a) Graphic depicting the various possible heterostructures available due to 2D material stacking [260]. (b) Atomic model of the structure imaged in (c), from which LED structures were built [261]. Gr_T and Gr_B are the top and bottom graphene layers, respectively. Scale bar is 5 nm.	34
2.13 (a) ADF STEM image of MoS_2 - MoSe_2 lateral monolayer heterostructure interface. Adapted from [78]. (b) ADF STEM image and (c) atomic model of WS_2 / MoS_2 lateral monolayer heterostructure interface. Adapted from [79]. Scale bars are 5 Å.	35
2.14 The relative probability of introducing substitutional B or N atoms as compared to other defects studied by Ahlgren <i>et al.</i> Figure taken from [307].	41
2.15 (a) Atomic resolution false colour ADF STEM image of N implanted monolayer graphene. The region in the yellow rectangle is magnified in (b). (c) EELS spectrum image of the region in (b). (d) EELS signal near the N implant. Figure taken from pre-print of [19].	43
2.16 (a) TEM image of Polarisation of Ti/Zr (red) and O (blue) atomic columns from PbO (yellow) across a DW in PbZrTiO_3 . Taken from [336]. ADF STEM image showing (b) the overall polarisation and (c) the individual Ti atomic column polarisation of the upper region of a vortex-like DW structure in $\text{PbTiO}_3/\text{SrTiO}_3$ multilayers (taken from [333]). (d) Method of polarisation mapping in PbTiO_3 -like structures (taken from [335]). (e) Polarisation of Nb atomic columns across a DW in LiNbO_3 (taken from [337]).	48
3.1 The micro-mechanical exfoliation technique for producing graphene and other 2D materials. Top row: Adhesive tape is used to exfoliate the several layers of graphite placed on a photoresist. Bottom left: The tape with graphitic flakes is then pressed against the glass substrate. Bottom right: Flakes remain on the substrate, producing high-quality graphene flakes ideal for electronic measurements [349].	52
3.2 Three sample preparation methods used for the creation of 2D nano-structures on various substrates. Combined with the deterministic stamping method in Figure 3.3, these methods form the basis on the sample preparation used during this work.	53

LIST OF FIGURES

3.3	(a)(i) Dissolving of a hydrophilic polymer in a water bath. (ii) Floating of a hydrophobic polymer/2D flake stack, which is ‘fished’ out with a glass slide. (iii) Glass slide/hydrophobic polymer/2D flake stack inverted and positioned over a rigid substrate/2D flake stack. The glass slide stack is then positioned and stamped on the rigid substrate, creating a heterostructure. (b) Deterministic stamping setup under an optical microscope.	54
3.4	(a) A render of the Solidworks design for the deterministic transfer of 2D flakes onto TEM grids and other substrates. (b) Picture of the laboratory set-up with the transfer stage fixed beside the microscope.	55
3.5	A list of the elements available for implantation using the ADONIS ion implanter. Sources available at the time of writing are highlighted.	59
3.6	(a) Schematic overview of the ion implantation device used throughout this project for the ion implantation of 2D materials. The special deceleration stage is highlighted with a red box and its schematic is presented in (b). The TEM grid substrate is placed on the substrate position and the defocusing lenses slow the incoming ions from 30 keV to several tens of eV. Original images courtesy of Prof. Hans Hofsäss.	60
3.7	(a) Picture and (b) Solidworks render of the ion implanter sample holder and tip. (c) Picture and (d) Solidworks view of the ion implanter tip.	60
3.8	(a) Schematic of a scanning transmission electron microscope (STEM), showing the ADF detectors and EELS spectrometers. Adapted from [376]. The signals resulting from fast (incident high-kV) electrons interacting with a sample in a STEM. Taken from [28].	63
3.9	Gaussian fitting of the ADF Z-contrast intensities from the transition metal (circle) and chalcogen (triangle) sublattices from a Se ion implanted MoS ₂ monolayer. The residuals from the Gaussian fitting are shown below. Overlap between the Gaussian curves can be used to estimate uncertainty of element configuration assignment.	69

LIST OF FIGURES

3.10 Image alignment of two images each represented by a 7x7 array of pixels. The central pixel in each image is marked by an orange circle for reference. (a-b) Initially, the image shift between image 1 and image 2 is calculated (often by cross correlation between the discrete Fourier transforms. (c-d) The calculated image shift is applied to image 2 to produce (d). Note the not a number (NaN) values in the bottom and right extremes of image 2. (e) The images are stacked by taking the mean of the pixel values. The NaN values represent lost information due to the alignment process.	73
3.11 Mean Squared Error (MSE) and Structural Similarity Index (SSM) measured between two example images. These methods can be used to compare any two images, such as an experimental STEM image and its simulation.	76
3.12 Images of the (a) raw, (b) local background, and (c) subtraction of raw - background image data for each Pb atom in PbTiO ₃ . The top and bottom rows show a basic example with pixel intensity values and an experimental example, respectively. This data is analysed further in Figure 6.14.	82
3.13 Pixel Classification + Object Classification step-by-step method used in the program Ilastik to analyse NPs on 2D materials.	85
3.14 An atomic model created from HR STEM data visualised in Vesta (left) and ASE (right).	85
3.15 ADF STEM simulation of ideal (a) and perturbed (b) atomic columns in an MoS ₂ monolayer. (b) and (e) show the simulated input model overlaid on the simulation. (c) and (f) show the same atomic position in each model marked as a red dot. The measured difference between their positions is the atomic movement, represented as the vector arrow.	86
3.16 (a) Atom positions (red) overlaid on a test dataset. (b) Atom planes from an example zone axis constructed from the atom positions. The 0-3 atom planes are labelled. (c) Atom positions highlighted along an atom plane, labelled 3.0-3.4.	88
3.17 A graphical example of polarisation across a single domain wall in an atomic lattice. The domain wall is represented by the white rectangular band. A best straight line fit along a zone axis is shown by the green line. In the left domain the atom positions are shifted down from this straight line fit, and are therefore polarised in the upward direction. Vice versa for the right domain.	89

3.18 (a) Atom deviation from several zone axes, each labelled with example (u, v) components. These components are added to form ‘Vector Addition’: the final combined atom deviation from the ‘ideal’ position. (b) An experimental example of the process of vector addition from (a). The vectors shown as unit vectors, with the magnitude displayed in a contour map, to properly present the turning of the structure.	91
3.19 The first 6 PCA decomposition (a) loadings (maps) and (b) factors (signals) of the spectrum image (SI) in (a). Inset (b)(i) is a PCA scree plot, also known as a variance plot, showing the first 5 components (triangles) used to produce the final PCA denoised SI. Inset (ii) is a zoom of the 40-100 eV energy range of the decomposition factors.	94
3.20 Experimentally acquired ADF image and PCA denoised spectrum image (SI) of a region of monolayer Se implanted MoS ₂ split into its five FastICA components from the PCA components in Figure 3.19.	95
3.21 (a) Python IDE Spyder, with Python code (solid blue), variable explorer (dotted green), and iPython output console (dashed blue). The ‘image.plot()’ of the dummy data is output in (b).	100
4.1 (a) Cross-sectional view and image (top) view of monolayer TMD. (b) Before and after ion implantation of a monolayer TMD. The top (A) and bottom (C) chalcogen sublayers, and middle TM layer (B), are shown.	105
4.2 Cross-sectional view of various element configurations of the atomic columns in monolayer MoS ₂ after low energy Se ion implantation. Not an exhaustive list.	105
4.3 Raw HAADF STEM images of monolayer MoS ₂ (a) before and (b) after 10 eV Se ion implantation.	106
4.4 Cropped region of a double Gaussian filtered ADF STEM (a) experimental and (b) simulated image (scale bars are 6.3 Å). A line intensity profile is taken from the triangle marker to the circle marker in both and overlaid in (b). (c) Line profile of a similar region while varying the inner ADF limit (outer ADF limit = 185 mrad).	108

LIST OF FIGURES

- 4.5 (a) A high-resolution ADF STEM image of pristine MoS₂. The red (top) and blue boxes represent a clean and contaminated region, respectively. (b) A PDF of the atom intensities from the regions in (a). The contaminated (contam) region is shifted higher due to the hydrocarbon contamination on the surface of the pristine MoS₂. The peak positions (Peak Pos.) due to the chalcogen atoms (A, B) and transition metal atoms (C, D) are labelled and numbered inset. (Scalebar is 3 nm) 109
- 4.6 Annular dark field (a) experimental and (b) simulated image of Se implanted monolayer MoS₂. (c) Energy map of the Se implants found in the chalcogen sublattice by combining element quantification and DFT calculations. Se₁ and Se₁S₁ atomic column configurations appear as bright yellow (high energy) and blue (low energy), respectively. Scalebars are 3 nm. 111
- 4.7 (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of a Se implant in the chalcogen sublattice. In (a) the Se implant is indicated by the blue rings corresponding to 0.8 eV. This is confirmed by the atom model in (b) showing the Se₁S₁ configuration. This higher chalcogen intensity is marked in the intensity line profile in (e). The large neighbour intensity is likely a Mo₁S₁ intensity, though it is incorrectly found as a Mo₁Se₁ configuration which would be corrected by using the Model Refiner tool. Scalebars are 5 Å. 112
- 4.8 (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of a Se implant in the chalcogen sublattice. In (a) the Se implant is indicated by the blue rings corresponding to 0.8 eV (the bright yellow energies are examined in Figure 4.9). This is confirmed by the atom model in (b) showing the Se₁S₁ configuration. This higher chalcogen intensity is marked in the intensity line profile in (e). Scalebars are 5 Å. 112
- 4.9 (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of several Se implants in the chalcogen sublattice. In (a) the Se implants are indicated by the bright yellow rings corresponding to 6.7 eV. This is confirmed by the atom model in (b) showing the Se₁ configuration. These higher chalcogen intensities are marked in the intensity line profile in (e). Scalebars are 5 Å. 112

LIST OF FIGURES

4.10 (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of a Se implant in the chalcogen sublattice. In (a) the Se implants are indicated by the yellow and blue rings cor- responding to 6.7 and 0.8 eV. This is confirmed by the atom model in (b) showing the Se_1 and Se_1S_1 configurations. The Se intensities are marked in the intensity line profile in (e). The intensity due to a single sulphur atom is also shown (S_{int}). Scalebars are 5 Å.	113
4.11 Annular dark field (a) experimental and (b) simulated image of Se implanted monolayer MoS_2 . The red arrows mark the location of a possible Se surface adatom. (c) Energy map of the Se adatoms found in the background sublattice (centre of hexagon) by combin- ing element quantification and DFT calculations. Scalebars are 10 Å. This cropped area is shown in context in Figure A.5 in Appendix Subsection A.3.	114
4.12 (a) Atom model, (b) experimental ADF image and (c) image sim- ulation of a Se adatom in the background sublattice after ion im- plantation. The adatom found with the energy map and indicated in Figure 4.11(c) is confirmed by the atom model in (a) showing the Se_1 configuration in the centre of the hexagon. The Se adatom intensity is marked in the intensity line profile in (d). (e) Experi- mental image and Simulation of the same adatom as in (a-d), with the line profile tracing in the vertical direction. Scalebars are 5 Å. .	116
4.13 Annular dark field (a) experimental and (b) simulated image of Se implanted monolayer MoS_2 . The dark contrast in each image in- dicates a hole. (c) The difference between the experiment (a) and simulation (b), wherein the hole becomes a bright region. (d) The energy map of the sulphur vacancies S_0 and S_1 . Scalebars are 10 Å. This cropped area is shown in context in Figure A.6 in Appendix Subsection A.3.	118
4.14 (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of hole in monolayer MoS_2 . The yellow region in (a) indicates double sulphur vacancies ($\text{S}_0 \sim 11.8$ eV) while the blue region indicates single vacancies ($\text{S}_1 \sim 5.9$ eV). The experiment and simulation match well except in the hole itself. The background intensity in the experimental image is likely due to hydrocarbon contamination. Scalebars are 5 Å.	118

LIST OF FIGURES

- 4.15 (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of hole in monolayer MoS₂. The yellow region in (a) indicates double sulphur vacancies ($S_0 \sim 11.8$ eV) while the blue region indicates single vacancies ($S_1 \sim 5.9$ eV). (e) The line profile comparison between experiment and simulated intensities. Scalebars are 5 Å. 118
- 4.16 (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of a single sulphur vacancy (S_1) in monolayer MoS₂. In (a) the S vacancy is indicated by the blue and purple rings corresponding to 5.9 eV. The line profile across the vacancy (e) demonstrates a good match between simulation and experiment. Scalebars are 2.5 Å. 119
- 4.17 Double Gaussian Filtered ADF STEM (a) experimental and (b) simulated image of pristine monolayer MoS₂. (c) Atom Model created after element quantification and used to simulate the data in (b). 121
- 4.18 Double Gaussian Filtered ADF STEM (a) experimental and (b) simulated image of 10 eV Se ion implanted monolayer MoS₂. The created alloy can be approximated by the chemical formula MoS_{1.87}Se_{0.13}. (c) Atom Model created after element quantification and used to simulate the data in (b). 121
- 4.19 Examples of clean (dashed yellow) and contaminated (red) regions in monolayer MoS₂ analysed both before (a) and after (b) 10 eV Se ion implantation. 125
- 4.20 (a) Line trace from point A to point B over several atoms in a 10 eV Se implanted MoS₂ monolayer. The intensity profile from this line (averaged over 8 pixels) is shown inset. (b) A plot of HAADF intensity versus each element in the periodic table up to Z=52. The average single element intensities for S, Se and Mo from all line profiles taken from the image in (a) are overlaid, resulting in a $Z^{1.6}$ scaling relationship. 125
- 4.21 (a) HAADF STEM overview image of Se implanted monolayer MoS₂. The highlighted region is enlarged in (b), where a substitutional Se implant is marked by the red arrow. Other Se-like intensities are shown by the orange and blue arrows. (c-d) Component 6 and 3 of a PCA denoised EELS spectrum image of the region in (b) are assigned as sulphur and selenium signals, respectively. The S and Se signals are false coloured and overlaid in (e). The PCA factors are displayed in Figure 4.22. 128

4.22	The 3rd, 4th and 6th components of the PCA denoised EELS spectrum image scanned over the highlighted region in Figure 4.21(a). The maps (left) are created from the signals (right) by integrating the signal over a 50 eV energy window after the onset of the relevant element edge. The 3rd and 6th components are likely the Se and S signals, respectively.	129
4.23	(a) Experimental HAADF STEM image of the EELS scan region in Figure 4.21 above. (b) Automatic and refined image simulation of the same region. (c) Refined atomic model used to produce the simulation in (b). (d) Cross-sectional view of the MoS ₂ monolayer lattice showing the automatically found substitutional Se implants. The red, orange and blue arrows indicate locations of Se atoms, as detected by the refinement of the atomic model.	130
4.24	EELS spectrum image (SI) over the atoms in the ADF image. FastICA analysis of the SI produces three relevant component maps: Molybdenum N _{2,3} , Sulfur L _{2,3} and Selenium M _{4,5} . The cross-sectional models display a likely lattice configuration. The areas in (i), (ii) and (iii) are discussed in the main text; they display interesting results due to the electron/ion beam.	131
4.25	(a) Experimental HAADF STEM image of the EELS scan region in Figure 4.24 above. (b) Automatic and refined image simulation of the same region. (c) Refined atom model used to produce the simulation in (b). (d) Cross-sectional view of the atom model. Scalebars 2 Å.	133
4.26	ADF and FastICA selenium M _{4,5} component map. Areas (i), (ii) and (iii) are described in the main text. (a) Experimental HAADF STEM image of the EELS scan region in Figure 3.19 above. (b) Automatic and refined image simulation of the same region. (c) Refined atomic model used to produce the simulation in (b). Scale bars are 2 Å.	134
4.27	ADF and SI of an area of (a) pristine and (b) Se implanted monolayer MoS ₂ . (c) The pristine (black line) and implanted (red line) signals from the regions of interest (red boxes) in (a) and (b), respectively.	137
4.28	Cross-sectional graphic of ¹⁹⁷ Au ⁺ ion implantation forming (a) substitutional or (b) nanoparticles (NPs) on a monolayer TMD before and after implantation.	139

LIST OF FIGURES

- 4.29 (a) Low magnification HAADF STEM image of 10 eV Au implanted MoSe₂ flakes. The dashed red rectangle is magnified in (b, c). (b) HAADF and (c) ABF image of Au implanted MoSe₂ flakes over the carbon quantifoil substrate (left) and suspended over vacuum (right). 140
- 4.30 ADF STEM images (a,c) before and (b,d) Au NP identification with the Ilastik program. The different shapes identified were grouped into four main categories: Single Nanoparticle (yellow), Dimer (blue), Trimer (red), and NPs from regions not analysed in the current frame (turquoise). 141
- 4.31 (a) Five regions of the ADF STEM image shown above in Figure 4.29. (b) Probability density and (c) ADF intensity brightness of the Au NPs diameter for each region in (a). 142
- 4.32 (a) Percentage coverage of Au NPs after 10 eV implantation into different thicknesses of MoSe₂ flakes. The five regions are presented in Figure 4.31. (b) Simultaneously acquired ADF image and EDX Au spectrum of three Au NPs on a thin (3-4 layer) MoSe₂ flake. (c) Atomic resolution ADF image taken at 300 kV of an Au NP on a thin (3-4 layer) MoSe₂ flake. Scale bars are 2 nm. 143
- 4.33 (a, i) Intensity profile of the experimental ADF STEM image in (a, ii). (a, iii) Low magnification overview of Au NPs on monolayer MoS₂. The red rectangle is magnified in (a, ii). (b-d, i) Intensity profiles of the simulated ADF STEM images in (b-d, ii) and (b-d, iv). (b-d, ii) Simulated images of the models in (b-d, iii). (b-d, iv) Simulated images of the models in (b-d, v). (b-d, iii) and (b-d, v) Plan-view (top-left) and 3D view of the Au NP on monolayer MoS₂. Scale bar of (a, iii) is 5 nm, all other scale bars are 1 nm. 145
- 5.1 The DG filtered Experimental image, final Simulated image, and final Atom Model for five frames of an image series of Se implanted MoS₂, analysed using the algorithms and tools designed in this chapter. Scale bars are 1 nm. 149
- 5.2 3D render of a test atomic model (a) before and (b) after intensity refinement shown in Figure 5.3. The vacancies were found by the Model Refiner tool using the `image_difference_intensity` function. 152

- 5.3 (a) Example cubic Ti sublattice (test experimental data) with several vacancies. All atom positions are labelled with the Ti_2 element configuration as seen in the ‘0: Initial State’ row of the table ‘Number of Atoms Counted in Sublattice by Model Refiner’. (b) Simulation images for each intensity refinement. (c) The resulting temporary difference sublattices for each intensity refinement. These are the sublattice image minus the simulation image. Simulation Image 3 can be thought of as a correct simulation of the test experimental data in (a). (d) The distribution of the atom position intensities of each difference image in (c). The mean (red dot) and first four standard deviations (green dots) are plotted on the histograms for reference. 153
- 5.4 Change of the element configurations as a function of (a) the intensity refinement and (b) the element configurations. The simple method `refiner.plot_element_count_as_bar_chart` in Code Segment 5.1 will produce these plots. 155
- 5.5 (a) Example of cubic Ti sublattice with three vacancies (one labelled with blue arrow). All atom positions are labelled with the Ti_2 element configuration as seen in the ‘0 Initial State’ row of the (e) Refinement History table. (b) Example simulation image. Results of each comparison between the example sublattice and simulation image are tabulated in (e). (c) The resulting sublattice after the position refinement (one vacancy position is labelled with blue arrow). (d) The difference (diff) image resulting from subtracting the (b) simulation image from the (a) example sublattice image, and vice versa for diff_inv. (e) Table of the element refinement after one position refinement and two intensity refinements. 158
- 5.6 Measure of the mean square error (MSE) and structural similarity index (SSM) between the Model Refiner’s reference (experimental) image and comparison (simulated) image. These values are automatically tracked by the Model Refiner (see Code Segment 5.4). . . 164
- 5.7 (a) Atom lattice, consisting of three sublattices (blue, yellow, green) of a region of Se implanted monolayer MoS_2 . (b) The constructed Model Refiner count of the elements configurations in the three sublattices. (c) The area with which the Model Refiner will calibrate any simulations can be set by the user (highlighted by the red box). This region has no adatoms or dopants. Scale bar is 1 nm. 165

LIST OF FIGURES

- 5.8 (a) Automatic double-Gaussian filtered HAADF STEM image of Se implanted MoS₂. Using the Model Refiner tool, this image is automatically calibrated and set as the Model Refiner's reference image. (b-d) Fast image simulation within the Model Refiner using the PyPrismatic program. The simulations are completed with an interpolationFactor (iF) of (b) 100, (c) 16 and (d) 16 with auto-filtering. The time taken for each simulation on a standard laptop (no GPU) is also noted under each image. Scale bars are 1 nm. 165
- 5.9 3D .cif model visualised with the ASE package (a) before refinement, (b) after intensity refinement, and (c) after position refinement. The red arrows highlight regions where the refinements have changed the model's structure to better match the experimental image in Figure 5.8(a). Further intensity refinement is required to better match these new positions with the experiment. Visualisation of the elements in the Model Refiner initially and after each refinement as (d) a bar chart and (e) a table of the element counts. 166
- 5.10 State of the Model Refiner's reference image (Experiment) and comparison image (Simulation) after the refinements logged in Figure 5.9. The too-bright atoms at the top-left and top-right of the Simulation will be corrected by further refinements. Scale bars are 1 nm. 167
- 5.11 Frames 0 (t=60s), 50 (t=127s), and 98 (t=190s) of an experimental movie. The Atom Model and Simulation for the PF and AV atom position methods are shown on the left and right of the experiment, respectively. Scale bars are 1 nm. 172
- 5.12 The (a) structural similarity index (SSM), (b) mean square error (MSE), (c) number of simulations and Model Refiner (MR) refinements, and (d) difference between the theoretical and experimental Z-contrast exponent n for the averaged image (AV) and previous frame (PF) atom position methods. (a-b) The SSM and MSE are calculated between the experimental frame and final simulation after the MR refinements. (c) Each MR refinement, and therefore simulations, were limited to 8 in this calculation. (d) The difference between the theoretical and experimental exponent (n_{diff}) is shown for each method. The experimental Z-contrast is calculated as described in Subsection 2.1(ADF STEM). The theoretical Z-contrast exponent is calculated as 1.86975. 172

- 5.13 DFT calculations of formation energies of substitutional and ad-atom Se in monolayer MoS₂. The difference between the energy of the left ‘A’ columns and right ‘B’ columns is calculated by the formula $\Delta E = E_A - E_B$. The DFT calculations were carried out by Dr. M. Ghorbani-Asl and Dr. A.V. Krasheninnikov. 176
- 5.14 Manual analysis of a Se-vacancy swap in three consecutive frames of an Se implanted MoS₂ monolayer image series. (a) Double Gaussian filtered frames 39, 40 and 41 of the image series with the blue (left) and red (right) arrows indicating the atomic columns in which the Se-vacancy swap occurs. (b) Two possible energy paths for the Se-vacancy swap over the three frames. The energies - blue for left, red for right atomic columns - are taken from the DFT calculations in Figure 5.13 above. Scale bars are 3 Å. 177
- 5.15 Model Refiner (MR) automatic analysis of the region from Figure 5.14 above. The final simulation and atom model produced by the MR algorithms are displayed under their respective experimental images; frames 39, 40 and 41. The blue (left) and red (right) arrow labelling is consistent with Figure 5.14 above. Scale bars are 3 Å. 177
- 5.16 Crops of five experimental frames of an automatically analysed image series using the MR tool. The simulation and atom model of each frame crop are below the respective experimental frames. The red arrows in frames 9, 18, 28 and 42 highlight a possible adatom, as do the blue arrows in frame 18. The orange arrows in frames 32 and 42 point to possible movement of a lattice Mo atom. Each frame number is also the number of seconds after the scan begins. Scale bars are 5 Å. 179
- 5.17 Crops of five experimental frames of an automatically analysed image series using the MR tool. The simulation and atom model of each frame crop are below the respective experimental frames. The red arrows in each frame highlight a possible Mo₁ adatom. Each frame number is also the number of seconds after the scan begins. Scale bars are 5 Å. 182
- 5.18 (a) Manual tracking of the location over which an adatom sits in Figure 5.17. (b) The percentage of the frames over which the adatom sits for each location. Centre refers to the centre of the MoS₂ hexagon. S: chalcogen column, Mo: transition metal column 183

LIST OF FIGURES

5.19	Cross-sectional and image (top) views of an Mo surface adatom (blue atom with red border) over the (a) centre of the hexagon, (b) chalcogen and (c) transition metal position.	184
5.20	(a) Total number of Mo atoms and (b) lattice Mo vacancies per squared nm for each frame in the movie. The lattice hole size per squared nm is overlaid on right y-axis of both (a) and (b). The upward and downward arrows in (a) and (b) indicate the frames at which the dwell time is increased and decreased, respectively. (c) Experimental images and final model refiner image simulations of frames 0, 25, 50, 75 and 99 of the movie. Scale bars are 1 nm.	186
5.21	(a) Lattice Mo vacancies per squared nm as a function of time for low (monochromated) dose rate movies of Se implanted MoS ₂ . Initial frames of (b) movie 004 (dashed red line), movie 003 (solid blue line), and movie 005 (dotted black line). Scale bars are 9 Å.	189
5.22	(a) Lattice Mo vacancies per squared nm as a function of time for high (non-monochromated) dose rate movies of Se implanted MoS ₂ . Initial frames of (b) movie 008 (dash-dotted purple line), movie 012 (dashed yellow line), movie 014 (solid cyan line), and movie 013 (dotted green line). Scale bars (b-c) 9 Å, (d-e) 10 Å.	189
5.23	(a) Lattice Mo vacancies as a function of the total cumulative dose for each non-monochromated, high dose rate, movie. Comparable time periods are presented in Figure 5.24(a). (b) Average lattice Mo vacancies as a function of the dose rate for each non-monochromated, high dose rate, movie.	190
5.24	(a) Lattice Mo vacancies as a function of time for each movie. (b) Average lattice Mo vacancies as a function of the dose rate for each movie.	190
5.25	Fit of the average Mo lattice vacancies as a function of electron beam dose rate for a total time of 130 s. The equation of the line is $y=0.029e^{0.256x}$	192
5.26	Fit of the average Mo lattice vacancies as a function of electron beam dose rate for a total cumulative dose of 1370 $pC\ nm^{-2}$. The equation of the line is $y=0.057e^{0.185x}$	192

5.27	(a) Number of lattice Mo vacancies counted per square nm for movie 008 (dash-dot purple line) and movie 010 (solid teal line). (b) The difference, n_{diff} , between the theoretical and experimental Z-contrast exponent n for movies 008 and 010. The mean n_{diff} for each is given in the legend. (c-d) Initial frames used in the MR analysis for movies 008 and 010, respectively. (e) Frame 325 (time=425 s) of movie 010. This frame is of the same area and \sim 293 seconds after the final MR analysed frame in (a). Scale bars are 1 nm.	194
6.1	Plots of polarisation in an example atomic resolution image showing (a) a basic vector plot, (b) scaled vectors (nm) overlaid on the example image, (c) colormap of the vectors, (d) colormap of the scaled vectors (nm), (e) colormap of unit vectors, and (f) a contour map of unit vectors. The atom position data for all images (a-f) is the same. The script in Code Segment 6.1 above produces these figures.	202
6.2	(a) Atomic movement calculated by the distance from the ideal, mid-point positions of each of the four atoms in the bright sublattice to the actual positions of the second sublattice (see Code Segment 6.2 Line 34). (b) The opposite vectors of (a) by reversing the <code>find_polarisation_vectors</code> parameters (see Code Segment 6.2 Line 42).	205
6.3	(a) Example sublattice with every second column in the bottom half of the image offset in y, creating a zigzag pattern. (b) Visualisation of the deviation of each atomic column from a horizontal straight line fit.	207
6.4	(a 0-4) The example sublattice's five zone axes. (b 0-4) The resulting atomic column deviations from each zone axis. Code Segment 6.4 will produce these plots.	208
6.5	The combined atomic column deviations from (a) all zone axes and (b) zone axes 0, 1, 2, and 3 from Figure 6.4 above.	210
6.6	Visualisation of vector data in an atomic resolution image with each vector representing (a) an atomic column and (b) the average of several vectors in a region.	212
6.7	Atom planes of the 0 zone axis (vertical lines) of an example sublattice (a) before and (b) after atom plane deletion with the function <code>delete_atom_planes_from_sublattice</code>	214

LIST OF FIGURES

6.8 (a) Example sublattice with atom y coordinate offsets in the lower half of the image. (b) First zone axis of the sublattice in (a) showing the zig-zag offset pattern. (c) Strain map of the first zone axis, calculated with the <code>get_strain_map</code> function.	217
6.9 (a) Example sublattice with atom y coordinate offsets in the lower half of the image. (b) Third zone axis of the sublattice in (a) showing the zig-zag offset pattern. (c) Angle map of the atom planes in the third zone axis, calculated with the <code>rotation_of_atom_planes</code> function.	218
6.10 (a) First and (c) second zone axis of the sublattice in Figure 6.9(a). Lattice spacing map of the atom spacings along the atom planes in the (b) first and (d) second zone axis, calculated with the function <code>ratio_of_lattice_spacings</code> . (e) Ratio map of the lattice spacing maps from (b) and (d).	218
6.11 (a) Atomic resolution ADF STEM image of a domain wall junction in PTO. The charged needle point (CNP) domain causes the area of high strain i.e., the PTO junction. The rectangular regions inset are magnified in (b), (c) and (d). (b) Sub-Ångström position refinement with Atomap of the bright Pb (red) atoms and Ti (blue) atoms. (c, d) Polarisation mapping of the H-H domain wall with the <code>polarisation.py</code> module of the TEMUL toolkit. Scale bars (b-d) are 1 nm.	220
6.12 (a) Quantified strain map, (b) atom plane angle map, and (c) lattice spacing c/a ratio map of the PTO junction from Figure 6.11(a) using the <code>polarisation.py</code> module of the TEMUL toolkit. Inset in (b) is the zone axis 1 direction showing bending across DWs. Scale bars are 8 nm.	222
6.13 Line profiles (a, c, e) from their respective (b) quantified strain map, (d) atom plane angle map and (d) lattice spacing c/a ratio map across the H-H domain wall. The line profiles and maps were visualised with the <code>polarisation.py</code> and <code>signal_plotting.py</code> modules of the TEMUL toolkit, respectively. The chosen line profile positions are shown in the appendix in Figure A.15. Scale bars are 8 nm.	222
6.14 Line intensity profiles along two parts of the (a) raw and (b) local background subtracted images from the central PTO DW junction in Figure 6.11(a). The top (solid red) and bottom (dashed blue) line profiles trace over the junction and the right side domain, respectively.	224

LIST OF FIGURES

A.1 (a) Exfoliation of a 2D single crystal with elastomer tape. (b) Stamping of the 2D flake onto PDMS (or another) polymer for ‘dry-stamping’. (c) Metal block with a acceptor substrate, usually a 3 mm TEM grid (here a piece of Si wafer is shown). On the right is the glass slide/polymer/2D flake stack. (d) Stamping of the 2D flake onto the acceptor substrate under the microscope, with the viscoelastic contact effect visible on the background screen. (e) Alternative view of (d). (f) Overview of the entire stamping set-up.	273
A.2 (a) Wafer spin-coat holder designed and used during the project. (b) Thermal tap/o-ring/polymer stack after cutting and fishing the polymer from the Si/SiO ₂ substrate. (c) Searching for the transferred flake under an optical microscope. This metal slide is then placed upside-down over the TEM grid and stamped.	273
A.3 (a) Solidworks view of the base-plate for a double DENS in-situ heating chip. (b) Solidworks view of the two clamp-plates with a DENS in-situ TEM grid in place (blue). (c) Machined single DENS in-situ ion implanter holder.	274
A.4 Scaling ratio R vs Z-contrast scaling exponent n (written here as P) for the four most common TMDs. A zoom in of the 1-2 region in x is shown on the right.	274
A.5 Full figure of HAADF image 017 with the green rectangle marking the cropped image on the right. The Cropped Image is used in Subsection 4.1.1(Regions affected by Se Ion Implantation).	275
A.6 Full figure of HAADF image 017 with the green rectangle marking the cropped image on the right. The Cropped Image is used in Subsection 4.1.1(Vacancies and Hole Damage).	275
A.7 Energy maps of various vacancies near the hole in Figure A.6(Cropped Image) above. The S ₀ S ₁ Energy Map only presents the energy above pristine MoS ₂ caused by double and single sulphur vacancies. The S ₀ S ₁ Se ₁ Energy Map presents the energy above pristine MoS ₂ caused by double and single sulphur vacancies, and the single sulphur vacancies in a Se ₁ column. The All Chalcogen Energy Map presents the energy above pristine MoS ₂ caused any chalcogen column with a any combination of sulphur vacancies, such as: S ₀ , S ₁ , Se ₁ , Se ₁ S ₁ and Se ₂ . The S ₀ S ₁ Energy Map is shown in Subsection 4.1.1(Vacancies and Hole Damage).	276

LIST OF FIGURES

A.8 (Clockwise starting in the top left) Overview ADF image of pristine monolayer MoS ₂ . The green region of interest is scanned in the ADF and SI on the right. The red regions of interest are summed and the spectrum shown on the bottom, where the LL EELS background is subtracted and signal shown (dark green).	277
A.9 (Clockwise starting in the top left) Overview ADF image of implanted monolayer MoS ₂ . The green region of interest is scanned in the ADF and SI on the right. The red regions of interest are summed and the spectrum shown on the bottom, where the LL EELS background is subtracted and signal shown (dark green).	278
A.10 Three frames of double-Gaussian filtered HAADF STEM images (column 1) of Se implanted MoS ₂ . The Se-vacancy swap is visualised via the integrated intensity in column 2. The histograms in column 3 show the intensities integrated in column 2.	279
A.11 Lattice Mo vacancies as a function of the total cumulative dose for movies 001, 003, 004 and 005. Movie 001 is not included in the main section due to alignment issues with; the image area was drifting over a large hole. This made quantification unreliable.	280
A.12 Average Z-contrast exponent n_{diff} (theoretical minus experiment) for each movie analysed with the Model Refiner tool in subsection 5.2.3. Relevant to comparisons in Subsection 5.2.3.	280
A.13 Histogram intensities of the chalcogen sublattice (orange, lower intensities) and transition metal sublattice (blue) for each movie analysed in Subsection 5.2(Atom Dynamics in Implanted TMDs with the MR Tool).	281
A.14 (a) Atom lattice of the region near a PTO DW junction. Red dots are Pb atomic columns and blue dots are Ti atomic columns. (b) Atomic shift of the Ti sublattice. The high-magnitude vectors displayed in the left are due to off zone axis imaging.	282
A.15 (a) start and end positions of the line profiles used in subsection 6.2.1(PTO Domain Wall Junctions). (b) Histogram of the angles calculated by <code>rotation_of_atom_planes</code> for Figure 6.9. The groupings show the angles in each domain.	282

List of Tables

2.1	Summary of analysis of single atoms via high-resolution (scanning) transmission electron microscopy in 2D and other materials.	12
2.2	Table taken from [278].	37
2.3	Table taken from [286].	38
2.4	Summary of ion implantation into 2D and low-D materials.	42
3.1	Each main toolkit and program used throughout the project, with their respective use cases and frequency of use.	96
4.1	Statistics of the chalcogen (Chal.) positions occupied by vacancies (Vac.) or substitutional (Subst.) Se and before and after 10 eV Se ion implantation for automatically analysed regions (example regions are shown in Figure 4.17 and Figure 4.18). The percentage, number per square nm, and number per unit cell are shown. The total area analysed was \sim 875 nm ² (21188 chalcogen positions) and \sim 725 nm ² (16996 chalcogen positions) before and after implantation, respectively.	122
4.2	Percentage of chalcogen positions occupied by vacancies before and after implantation for clean and contaminated regions. The average is also displayed. The percentage of chalcogen positions occupied by Se atoms due to 10 eV ion implantation, as well as the percentage of Se ions retained by the lattice from the ion beam are shown. The total area analysed manually was \sim 45 nm ²	126

LIST OF TABLES

4.3 LL EELS measurements of band structure properties of pristine and Se doped MoS ₂ . Numbers from literature are given under <i>Literature</i> . E _{BG} : optical band gap energy (onset). E _{exc,A} , E _{exc,B} , E _{exc,C₁,C₂} : A, B, C ₁ and C ₂ exciton peak energies, respectively. Uncertainty in experimental reading for pristine: ±0.026 eV, implanted: ±0.046 eV. *requires further fitting, higher uncertainty. ‡Bilayer implanted region not as accurate (smaller region). Literature values taken from [78, 234, 413, 414].	136
4.4 Mean, median and mode diameter (nm) and surface area coverage (%) of the Au NPs in each region shown in Figure 4.31(a).	144
5.1 The attributes, explanation (Python type), and how to assign the attributes of the Model Refiner. Not all attributes are shown.	160
5.2 The methods, explanations of method, and action taken by each method, of the Model Refiner class. These methods (functions) can be called easily, as seen in Subsection 5.1.2(Using the Model Refiner on Experimental Data).	161
6.1 The main functions of the <code>polarisation.py</code> module [13]. A link, brief explanation and action taken by each function is shown (Hspy refers to Hyperspy).	198

List of Code Segments

3.1	Mean Squared Error implementation from [1] in the <code>signal_processing</code> module.	77
3.2	Example code for analysing the atomic deviation from a straight line fit (see Figure 3.17) in the first zone axis and the vector addition of several zone axes.	89
5.1	The element configurations of a dummy cubic structure with vacancies refined by comparison with an dummy simulation via the intensity refinement algorithm. The outputs (Out:) for each step can be seen in both Figure 5.3 and Figure 5.4.	155
5.2	The element configurations of a dummy cubic structure with vacancies refined by comparison with an dummy simulation via the position refinement algorithm. The outputs (Out:) for each step can be seen in Figure 5.5	159
5.3	Se implanted MoS ₂ Model Refiner example part 1. The Model Refiner object is constructed and labelled ‘refiner’. Three sublattices are found via Atomap. The elements present in the image are found and visualised in Figure 5.7(b). The calibration criteria is set as shown in Figure 5.7(c).	163
5.4	Se implanted MoS ₂ Model Refiner example part 2. The Model Refiner can track the sublattice elements and simulate with PyPrismatic as shown in Figure 5.7 and Figure 5.8, respectively. Refinement of the elements is completed by using the image difference intensity and position functions. The user can quickly inspect the effect of the refinements with a bar chart or table shown in Figure 5.9(d, e). If a refinement is mistakenly called, one can revert to a previous version of the Model Refiner with the <code>previous_refiner_instance</code> method (Line 44).	168
5.5	Example loop for the Model Refiner (MR) tool for step 7 in the analysis steps outlined in subsection 5.2.1.	174

LIST OF CODE SEGMENTS

6.1	Example script using the <code>plot_polarisation_vectors</code> function. The outputs from this code is displayed in Figure 6.1	200
6.2	Example script using the <code>find_polarisation_vectors</code> function. The outputs from this code is displayed in Figure 6.2	203
6.3	Example script using the function <code>atom_deviation_from_straight_line_fit</code> . The outputs from this code are displayed in Figure 6.3	206
6.4	Example script using the function <code>plot_atom_deviation_from_all_zone_axes</code> . The outputs from this code is displayed in Figure 6.4	207
6.5	Example script using the function <code>combine_atom_deviations_from_zone_axes</code> . The outputs from this code is displayed in Figure 6.5	209
6.6	Example script using the <code>get_average_polarisation_in_regions</code> function. The outputs from this code is displayed in Figure 6.6 . . .	211
6.7	Example script using the <code>delete_atom_planes_from_sublattice</code> function. The outputs from this code is displayed in Figure 6.7 . . .	213
6.8	Example script using the <code>get_strain_map</code> , <code>rotation_of_atom_planes</code> and <code>ratio_of_lattice_spacings</code> functions. The outputs from these functions are displayed in Figure 6.8, Figure 6.9 and Fig- ure 6.10, respectively.	216
A.1	Some of the developed Z-contrast functions of the TEMUL toolkit discussed in Chapter 3 and in Chapter 4.	284
A.2	Some of the developed algorithm functions of the TEMUL toolkit discussed in Chapter 5.	295
A.3	The developed Model Refiner class of the TEMUL toolkit discussed in Chapter 5.	310
A.4	Some of the developed polarisation functions of the TEMUL toolkit discussed in Chapter 6.	331

List of Programs

Atomap is a Python package for atomic position finding, refining and analysis.

Atomic Simulation Environment is a Python package for visualisation of atomic models, first principle calculations and file conversion.

Dr. Probe is a program for image simulation of electron microscopy signals.

Gatan Microscopy Suite/Digital Micrograph is a program for acquisition and analysis of electron microscopy data.

Hyperspy is a Python package for multidimensional data analysis in the field of electron microscopy.

Ilastik is an interactive learning and image segmentation toolkit.

ImageJ/Fiji is an extensible image analysis program.

Jupyter Notebook is a browser-based open-source software for interactive computing across dozens of programming languages.

Matplotlib is Python's main data plotting package.

Numpy and Scipy are core Python packages for scientific computing.

Pandas is a Python package for the analysis of data structures, similar to MS Excel.

Prismatic is a CPU/GPU program for fast image simulation of scanning transmission electron microscopy (STEM) data. **PyPrismatic** is the Python packaging of Prismatic.

Python is an open-source programming language used extensively in scientific research and data analysis.

Rigid Registration is a Python package for alignment of series of images.

Scikit-Image/skimage is an image analysis Python library.

Smart Align is a program for advanced image registration.

SolidWorks is a computer aided design (CAD) program.

TEMUL toolkit is a set of image and data analysis tools for electron microscopy data, developed during this thesis.

Velox is a program for acquisition and analysis of electron microscopy data.

Vesta is a program for visualisation of atomic models.

Abbreviations

2D	two-dimensional
ADF	annular dark field
CNT	carbon nano-tubes
CL	core-loss
CVD	chemical vapour deposition
DFT	density functional theory
DW	domain wall
EELS	electron energy loss spectroscopy
HR	high-resolution
LL	low-loss
MSE	mean square error
PL	Photoluminescence
PMMA	poly(methyl-methacrylate)
SRIM	stopping range of ions in matter
SSM	structural similarity index
(S)TEM	(scanning) transmission electron microscopy
TDS	thermal diffuse scattering
TEMUL	transmission electron microscopy University of Limerick
TMD	transition metal dichalcogenide
UHV	ultra-high vacuum

Chapter 1

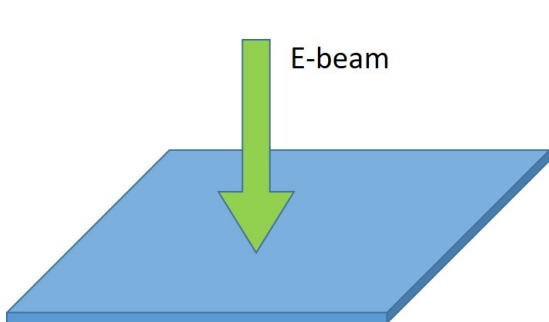
Introduction

Automated analysis of complex high-resolution annular dark field scanning transmission electron microscopy (ADF STEM) images has advanced considerably in recent years. With the excellent open-source Python facilities in HyperSpy [2, 3], and Atomap [4], as well as the advancements in reliability and reproducibility of atomic resolution data [5, 6], one can reasonably say that the field requires automated methods for analysing such data. Concerning data analysis, recent advances have not only yielded new methods but also allowed a multitude of users to collaborate on a large scale. In the coming years, we believe this will positively change the attitude toward data and data analysis in the field of electron microscopy. With regard to instrumentation improvements, we are now in an era of unparalleled resolution [7, 8].

However, analysis of large datasets of non-trivial atomic resolution data is not yet so advanced that one can simply click a button and assume the program will give accurate results. This thesis aims to edge the field closer to that objective [9]. In this thesis, we will utilise advancements in data analysis, reliable atomic resolution imaging and fast simulation for two systems: ion-implanted two-dimensional transition metal dichalcogenides (2D TMDs) and two-dimensional domain walls (2D DWs) in ferroelectric materials. We therefore examine 2D entities both perpendicular and parallel to the electron beam, as shown graphically in Figure 1.1.

In Chapter 2(Literature Review), the current state of the art in single atom microscopy analysis are discussed in detail. The properties of 2D TMDs are discussed, along with methods used to manipulate such properties. The literature review ends with an examination of ferroelectric materials, and how researchers have employed atomic resolution ADF STEM for analysis of ferroelectric domains and 2D DWs. Following this, Chapter 3(Methodology) examines the methods utilised throughout the project for sample preparation, low energy ion implanta-

2D Material (MoS_2)



2D Domain Wall

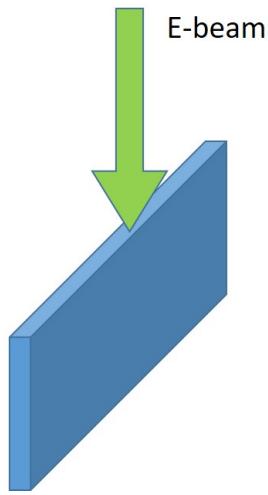


Figure 1.1: Imaging of two-dimensional (2D) entities perpendicular and parallel with respect to the incident electron beam.

tion, electron microscopy and data analysis. Throughout the methodology chapter, we discuss our development of Python-based methods for the automated analysis of atomic-resolution images.

There are three results chapters in this thesis. Chapter 4(Atomic Characterisation of Ion Implanted 2D TMDs) discusses the first low energy ion implantation of 2D TMD monolayers [10]. In the first half of the chapter (Subsection 4.1), single substitutional Se atoms are identified in monolayer MoS_2 via high-angle ADF STEM and electron energy loss spectroscopy (EELS). The developed Python-based workflow for analysis and quantification of elements with ADF STEM data, described in the methodology, is implemented in subsection 4.1.1(Microscopy of Se Implanted MoS_2). In the second half of Chapter 4, the low energy ion implantation of Au into both MoS_2 and MoSe_2 is analysed via HAADF STEM. Subsection 4.2(Au Ion Implantation) presents the formation of Au nanoparticles (NPs) on the surface of both monolayer and few-layer MoS_2 and MoSe_2 .

The second results chapter, Chapter 5(Atom Dynamics in 2D TMDs), is split into two main sections: Subsection 5.1, in which we develop automated atomic resolution image series analysis routines, subsequently applying them to movies of MoS_2 monolayers in Subsection 5.2(Atom Dynamics in Implanted TMDs with the MR Tool). Subsection 5.1(Open Source Development of Automated Analysis Methods) describes the development of two algorithms: the intensity refinement algorithm in Subsection 5.1.1 and the position refinement algorithm in Subsection 5.1.1. Both algorithms are based on the iterative comparison between an

experimental and simulated image, which is now possible on large datasets due to the fast image simulation software Prismatic [11, 12]. The Model Refiner Python class (MR tool) is presented in subsection 5.1.2. The MR tool allows the user to access the above algorithms, as well as many visualisation and tracking functions. It is based on the sublattice data structure of Atomap, and is therefore streamlined to work with Atomap, HyperSpy and PyPrismatic. These routines are available as part of the open-source TEMUL toolkit [13] (TEMUL: Transmission Electron Microscopy University of Limerick). The second part of Chapter 5 is given to the analysis of atom dynamics and electron beam effects in image series (movies) of Se implanted MoS₂ monolayers. First, the automated analysis workflow partly described for basic element quantification in subsection 3.4.1(Element Quantification in HR ADF STEM Images) and applied in Chapter 4(Atomic Characterisation of Ion Implanted 2D TMDs) is expanded upon in subsection 5.2.1(Image Series Automated Analysis Steps). Then, hole formation, surface adatom movement and vacancy swapping are examined in subsection 5.2.2(Analysis of Lattice Energetics) via automated and manual approaches. Finally, subsection 5.2.3(Effect of Electron Beam on TMD Monolayers) presents the quantification of Mo vacancies due to the electron beam in movies of Se implanted MoS₂ monolayers. We hypothesise a detrimental threshold dose rate and the total dose required to damage the MoS₂ lattice irreversibly.

In Chapter 6 we turn our attention to the analysis of 2D DWs in ferroelectric materials, such as single crystal boracites [14–16], PbTiO₃ [17], and **Redacted**. Subsection 6.1(Tools for Visualisation of Subatomic Shifts) is dedicated to the development of atomic shift and polarisation analysis, as well as visualisation functions, in the polarisation.py module of the TEMUL toolkit [13]. These functions are also streamlined with Atomap and Hyperspy, and can be used to understand sub-Ångström atomic shifts and characterise polarisation in atomic resolution images. Finally, in Subsection 6.2(2D Domain walls), these functions are used to understand 2D DWs in the above-mentioned ferroelastic materials.

To understand and map the dynamics of 2D entities, be it monolayer TMDs or ferroelectric DWs, one requires reliable atomic resolution imaging and streamlined analysis tools. The development of easy-to-use, open-source Python code which is streamlined with widely-used Python packages is a core aspect of this work. The vast majority of the source code is in the TEMUL GitHub repository (TEMUL toolkit) [13]. The most relevant code to this thesis is included in Subsection A.6. It is available to the public and free to use. Adapted code from methods in the literature or other programs, such as Gatan’s Digital Micrograph (DM), is discussed

in the methodology. New methods for the analysis of atomic columns in 2D TMDs and polarisation in 2D DWs are discussed in the first sections of Chapter 5(Atom Dynamics in 2D TMDs) and Chapter 6(Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls), respectively.

Redacted.

Chapter 2

Literature Review

This chapter will discuss the use of high-resolution electron microscopy for the analysis of two-dimensional (2D) transition metal dichalcogenides (TMDs) and 2D domain walls in ferroelectric materials carried out in Chapter 4-Chapter 5 and Chapter 6, respectively.

Subsection 2.1(Characterisation via Electron Microscopy and Spectroscopy) will detail literature in which single atom dopants were identified and individual atomic columns imaged via scanning transmission electron microscopy (STEM). The general theory behind electron-matter interactions, and the implementation of quantitative STEM will then be discussed.

Subsection 2.2 (Two-Dimensional Materials and Transition Metal Dichalcogenides) reviews the current state of manipulation of 2D TMDs and the application of ion implantation for atom-by-atom control over material properties.

Subsection 2.3(Ferroelectric Materials and Domain Walls) briefly summarises the material properties of proper and improper ferroelectric materials and, with reference to Subsection 2.1(Characterisation via Electron Microscopy and Spectroscopy), discusses the current state-of-the-art of atomic-column analysis near 2D domain walls (DWs).

2.1 Characterisation via Electron Microscopy and Spectroscopy

Transmission electron microscopy (TEM) allows for highly localised studies of the electronic and structural properties of 2D materials. This was demonstrated in a series of studies of B and N implanted graphene. First, the individual dopant atoms were spatially resolved via annular dark field (ADF) scanning transmission electron microscopy (STEM), followed by in-depth analysis of their bonding and band-structure via electron energy loss spectroscopy (EELS) [19, 20]. Utilising a combination of EELS and density functional theory (DFT), the authors revealed localised p-type and n-type doping at atomic resolution, successfully opening a band gap in the zero band gap semiconductor graphene. An earlier study by Krivanek *et al* had shown ADF STEM to be the ideal method for atom-by-atom analysis, wherein they quantified single B, C, N, and O atoms in a hexagonal boron nitride (hBN) monolayer [21]. These results are promising for future studies into MoS₂ and other 2Ds, as electronic and structural properties, such as defect detection, can be evaluated on the atomic scale. Table 2.1 presents many studies which show the effective use of high-resolution TEM and STEM for the study of individual atoms. In this section, we discuss the use of high-resolution TEM, STEM and EELS for the analysis of structures on the atomic scale.

2.1.1 Single Atom ADF STEM & EELS

Since the first image of single Uranium and Thorium atoms by Crewe, Wall and Langmore [22], many developments in hardware and software have been realised. Due to aberration correction, it is now almost routine that single atoms and atomic columns can be imaged in mid- and high-level TEMs. By 2010, 1.1 Å[21] and 0.5 Å[23] probe sizes were used for single atom analysis. As summarised for 2D materials in table Table 2.1, scanning TEM and EELS have been utilised a great deal for the unambiguous imaging of single atoms and investigation of their properties.

ADF STEM

The majority of studies shown in Table 2.1 used annular dark field (ADF) STEM to verify the presence of native and impurity atoms in 2D materials. In electron microscopy, ADF is synonymous with Z-contrast imaging [22, 24–30], wherein the intensity collected by the annular detector increases as a function of the atomic number Z . Originally, Z-contrast imaging referred to a technique in which the ADF image (elastic scattering signal I_{elas}) and energy-loss (inelastic scattering signal I_{inel}) image are captured simultaneously. The ratio of I_{elas}/I_{inel} was calculated and is proportional to Z

and independent of the sample thickness [22, 29]. The first single atoms (U and Th) were imaged by Crewe *et al* using this technique, as shown in Figure 2.1 [22, 31]. However, the method did not apply well to other samples due to strong diffraction effects in both the ADF and energy-loss images [24]. Using larger inner detector angles, Treacy *et al* found a significant decrease of Bragg diffraction in images. They proposed that by choosing an inner detector angle of 2.5 \AA^{-1} (~ 100 mrad), the majority of the coherent diffraction contributions would be excluded [24], leaving only the incoherent thermal diffuse scattering (TDS) effects [32].

As a side note, Hage *et al* very recently verified experimentally the nature of Z-contrast imaging as inelastic scattering associated with phonon excitation [33]. This is consistent with a model for the quantum excitation of phonons (QEP) [34].

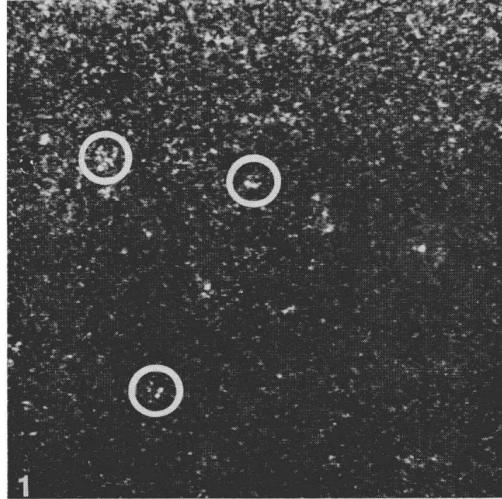


Figure 2.1: First image of single atoms. Uranium atoms on a thin carbon film. 210 nm full scale. Taken from [22].

2.1. Characterisation via Electron Microscopy and Spectroscopy

This is, of course, at odds with the frozen phonon model [35], in which Z-contrast is modelled as elastic scattering from atoms. In a pragmatic sense, both models can be used, as they produce similar results for image simulation using multi-slice simulation [34, 36].

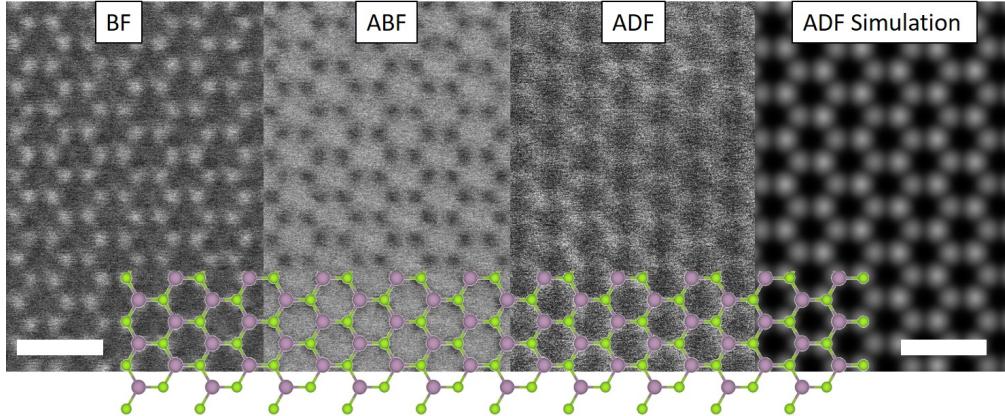


Figure 2.2: BF, ABF, ADF and ADF simulation of few-layer MoSe₂ imaged at 300 kV (UL Titan). Scale bars are 6.6 Å.

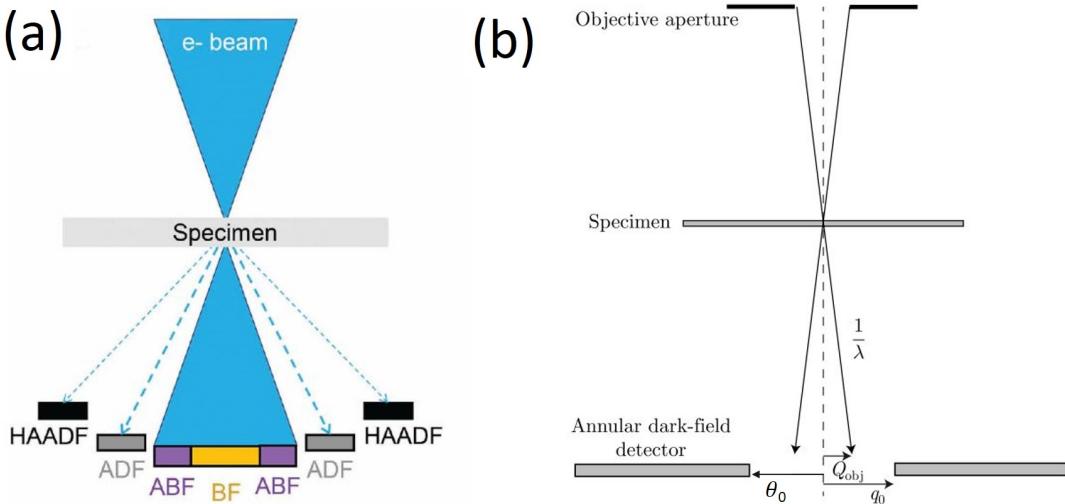


Figure 2.3: (a) Graphic of relevant detectors used to collect the electron beam signal transmitted through a sample in a STEM. These are bright field (BF), annular bright field (ABF), annular dark field (ADF) and high angle ADF (HAADF) Taken from [37]. (b) Basic angles of an ADF STEM detector. Q_{obj} is the reciprocal radius of the objective aperture. θ_0 and q_0 are the inner collection angle (rad) and inner collection vector (\AA^{-1}) of the ADF detector, respectively, and are equivalent. Taken from [29].

Modern Z-contrast images are therefore acquired at much higher angles than those used by Crewe *et al* [22], as well as being acquired with a rastering STEM beam [38]. In modern microscopes the inner detector geometries are, roughly, as follows: bright field (0-5 mrad), annular bright field (10-22 mrad), low to medium

angle ADF (40-80 mrad), and high-angle ADF (80-200 mrad) [39–42]. Each signal provides different information [39] and contrast, as shown in Figure 2.2. A graphic showing the radial positions of each detector in the microscope is shown in Figure 2.3(a). All can be acquired simultaneously (though the bright field (BF) detector can be removed to allow simultaneous EELS acquisition). The ADF detector angle ranges are given for each single-atom experiment mentioned in Table 2.1. The angles discussed below are shown graphically in Figure 2.3(b), which summarises the essential geometry of the ADF STEM detector [29].

By calculating the Z-contrast scaling exponent, n , one can predict the relative intensities of atoms in an ADF image. The elastic scattering cross-section, (σ), of a single isolated atom is given as:

$$\sigma(Z, q_0) = A(q_0) Z^{n(Z, q_0)}, \quad (2.1)$$

where q_0 is the inner collection vector of the ADF (\AA^{-1}), $A=1$ (Hydrogen), and n is the Z-contrast scaling exponent [29]. For unscreened Rutherford scattering $n = 2$. This value decreases for screened Rutherford scattering, and is usually between 1.4-1.8 experimentally for standard medium and high-angle ADF (MAADF/HAADF) STEM [10, 21, 43]. Once the electron scattering factor $f(Z, \theta)$ parameterisations [35, 44] have been calculated, one can calculate the scattering cross-sections for elements $Z>1$ via Equation 2.2:

$$\sigma(Z, q_0) = 2\pi \int_{\theta_0}^{\pi} |f(Z, \theta)|^2 \sin\theta d\theta, \quad (2.2)$$

where θ_0 is the inner ADF collection angle (rad) and θ is the ADF collection angle ($\theta_0 = q_0$). For high-angle scattering (e.g., HAADF detector ranges), the assumption of rotational symmetry of the detector response in Equation 2.2 is adequate [45, 46]. Treacy [29] corrected for the dependence on the objective aperture reciprocal radius, Q_{obj} , by averaging the scattering factors over the objective aperture:

$$f_{eff}(Z, q) = \frac{1}{\pi Q_{obj}^2} \int_0^{2\pi} \int_0^{Q_{obj}} f(Z, \sqrt{Q^2 + q^2 - 2qQ \cos\phi}) Q dQ d\phi, \quad (2.3)$$

where Q is the objective aperture reciprocal angle (\AA^{-1}). Rearranging Equation 2.2 and further correcting for the inelastic contribution - negligible at high angles as discussed above - the Z-contrast exponent is defined as

$$n(Z, q_0) = \frac{1}{\ln Z} \ln \left[\frac{\sigma(Z, q_0)}{\sigma(1, q_0)} \right] - \frac{1}{\ln Z} \ln \left[\frac{1 + \beta(1)}{1 + \beta(Z)/Z} \right], \quad (2.4)$$

where $\beta=1$ for low Z [29]. In subsequent reference to n , the inner ADF collection angle will be given in mrad, i.e., $n(Z, \theta)$

Equation 2.4 was used in this work to find the theoretical Z-contrast exponents for atoms in monolayer materials. A final correction must be made when calculating n experimentally with respective to an element other than Hydrogen (Treacy [29] shows an example from [21]). The above equations, other useful electron optics functions, as well as the electron scattering factor parameterisations from both Kirkland [35] and Lobato *et al* [44] can be found in the TEMUL toolkit GitHub repository [13] (TEMUL: Transmission Electron Microscopy University of Limerick).

The intensity of an atom's signal detected depends on the inner ADF collection angle, q_0 , as shown in Equation 2.4. The signal intensity also depends on the outer detector angle, though the contribution from 200 mrad to 2π is only $\sim 0.11\%$ [29]. Therefore, the annular range (outer-inner angles) of the ADF detector can have a large effect on the image magnitude and contrast [47], though for high-angle ADF annular ranges, small changes have little effect on image contrast. At lower annular ranges, if the annular range of the ADF detector is small (~ 10 mrad), a 1 mrad change in this range can change the resulting image contrast by as much as 10% [46]. However, a 1 mrad change in an annular range of >40 mrad affects contrast only slightly. The same decrease in inner angle as outer angle will increase overall the signal on the detector i.e., more signal comes from lower angles. Additionally, decreasing the inner ADF collection angle increases the intensity of lower Z elements. A practical example of the effect the inner ADF collection angle has on the resulting image contrast is shown in Figure 4.4 in Subsection 4.1 (Se Ion Implantation). Overall, an uncertainty of several mrad in the inner detector angle can cause a 10% error in peak intensity between simulations. Additionally, 5% error can arise from non-uniformity of the detector response. Therefore, it is recommended to analyse the detector response [46] and quantify the incident probe [48], especially for low-angle ADF and ABF, where small inner angle changes change the magnitude of the image signal significantly.

We will now discuss the extensive use of ADF STEM in atomic resolution imaging. Researchers have imaged single dopants, from B ($Z=5$) [49] to Au ($Z=79$) [50], in 2D materials, with many more each year. In graphene alone, single atoms of Si [7, 51–55], P [56], B/N [19, 20], Ge [57], Fe [58, 59], O, Ni, [49], Pt [49, 60],

2.1. Characterisation via Electron Microscopy and Spectroscopy

Pd, Ti [61], Co [60], Rh [62], In [60], Cr, Al and Au, [61], have been identified with ADF STEM imaging. Using medium to high-angle ADF, Krivanek *et al* imaged single substitutional C and O atoms in the lattice of monolayer hBN [21]. Using the B (Z=5) intensity as the internal image calibration, they reported a Z-contrast scaling exponent of $n(5,58) = 1.64$. The recorded MAADF intensities are displayed as a histogram in Figure 2.4(a), with the colour-coded graphic in Figure 2.4(b) showing the sorted elements at each atomic coordinate. MAADF detector angles increase the signal compared to HAADF, and also increase the signal of low Z elements like B, C, N, and O. The authors also utilised a rounded band-pass filter in reciprocal space i.e., a double Gaussian filter convoluted with the fast Fourier transform of the image. This filter both smoothens the pixel-to-pixel intensity variations and reduces the probe-tail of the atom-probe convolution to zero at the nearest neighbour atom coordinate [63]. This filtering technique is used throughout this thesis, and an enhanced version of it was developed and is described in Subsection 3.4.2(Image Filtering).

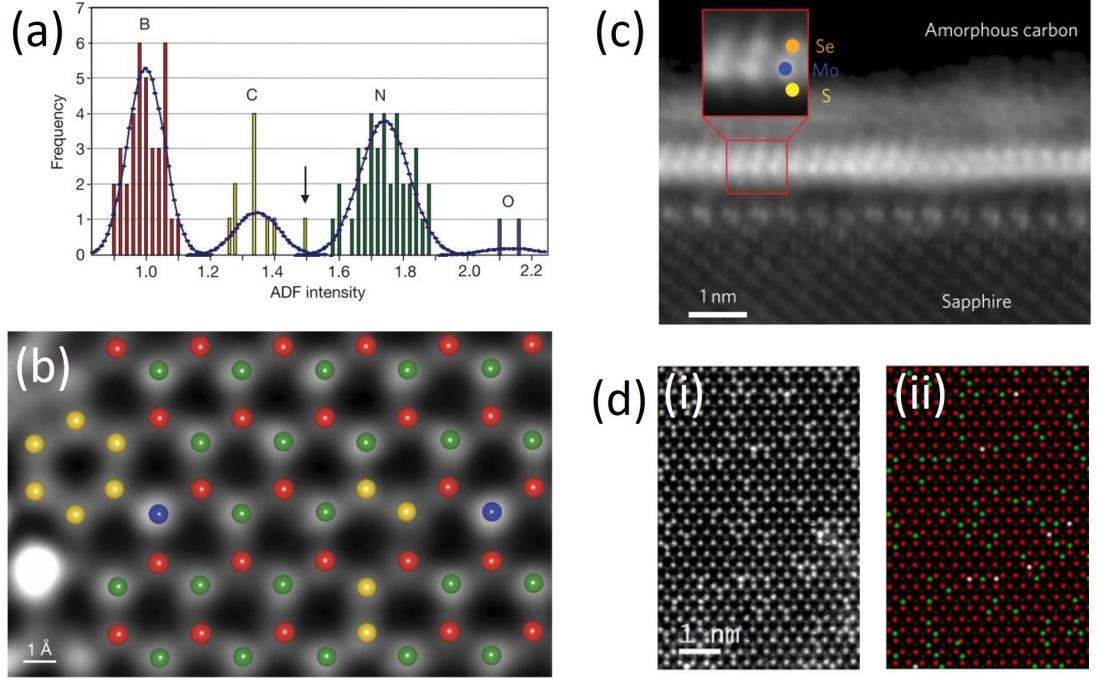


Figure 2.4: (a) Histogram of intensities recorded in a hBN monolayer. (b) Overlay of the sorted elements from (a) on the hBN monolayer MAADF STEM image. Red is Boron, yellow is Carbon, green is Nitrogen and blue is Oxygen. Taken from [21]. (c) Cross-sectional ADF STEM image of a Janus monolayer of MoSe₁S₁, taken from [64]. (d-i) ADF STEM image and (d-ii) identified elemental map of Se doped monolayer MoS₂ (Red is Mo, dark green is S₂, bright green is Se₁S₁, white is Se₂). Taken from [65].

The above filtering and element sorting methods have been used by several other studies [10, 19, 20, 43, 51, 65]. Susi *et al* analysed Si–C bond inversions

2.1. Characterisation via Electron Microscopy and Spectroscopy

in graphene with the commonly used combination of ADF and DFT [66]. The authors visualised the movement of Si and C atoms in the lattice, and calculated the energies involved over relevant time periods (~ 3 minutes). Many others have utilised DFT to model results garnered from ADF STEM [20, 67].

Table 2.1: Summary of analysis of single atoms via high-resolution (scanning) transmission electron microscopy in 2D and other materials.

Material	Impurity (element)	Experimental Technique	Simulation Technique	Analysis Tools	Ref
hBN	C, O	ADF (60 kV, 58-200 mrad)	-	Manual, DGF	[21]
Graphene	Si	ADF (60 kV, 86-190 mrad) CL Subscan EELS	DFT	DeConvHAADF [68], Power law	[52]
Graphene	Si	ADF (-) CL EELS	DFT	Manual [21]	[51]
Graphene	B, N	ADF (60-80 kV) (86-190 mrad) CL EELS	-	Manual [21]	[19]
Graphene	Si	ADF (55, 60kV) CL EELS	DFT	Manual, DGF	[54, 55]
Graphene	Si	M/HAADF (60kV 55-85 mrad/ 86-190 mrad)	DFT	-	[66]
Graphene	Ge, Pt	-	MD	-	[67]
Graphene	Fe	TEM	Multislice, DFT	-	[58, 59]
Graphene	Si	ADF (60 kV, 60-200 mrad)	-	Gauss Blur	[53]
Graphene	P	ADF (60 kV, 60-200 mrad)	-	Gauss Blur	[56]
Graphene	B, N	ADF (60 kV, 86-190 mrad) CL EELS	DFT	DGF [21]	[20]

(Continued)

2.1. Characterisation via Electron Microscopy and Spectroscopy

Table 2.1 *Continued.*

Material	Impurity (element)	Experimental Technique	Simulation Technique	Analysis Tools	Ref
Graphene	O, Ni, Pt	ADF (60 kV)	-	DeConvHAADF [68]	[49]
Graphene	Au, Ti, Al, Cr, Pd	ADF (60 kV)	-	-	[61]
Graphene	Si	ADF, EELS (60 kV)	DFT	Hyperspy[69], Power Law, PCA[70, 71]	[7, 72]
Graphene	Defect	TEM (80 kV)	Multislice, MD	QSTEM [73]	[74]
MoS ₂	Se	ADF (60 kV)	-	Manual [21] TEM SIM [35]	[10]
MoS ₂	Se	ADF (60 kV, 86-190 mrad)	DFT	In-house code [21] CASTEP [75]	[65]
NbSe ₂	Pt	TEM (80 kV) ADF (200 kV, 48-191 mrad)	Multislice, DFT	GMS, Strain++ JEMS [76], QSTEM [73]	[77]
MoS _x Se _{1-x}	CVD	ADF (60 kV) EELS (30 kV)	-	-	[78]
MoS ₂ /WS ₂	-	ADF [65]	-	-	[79]
MoSe ₂ /WSe ₂	-	ADF (80/200kV, 45-out mrad)	-	StackBuilder [80]	[81]
MoS ₂	Re, Au	ADF (60 kV) EDX (80 kV)	Multislice, DFT	MacTempasX [82]	[50]
MoS ₂	Cr, V	ADF (60, 80 kV) CL EELS	Multislice, DFT	JEMS [76] VASP [83]	[84]
MoS ₂	Co	ADF (60 kV) CL EELS	Multislice, DFT	JEMS [76] VASP [83]	[85]
MoS ₂	Co	ADF (60 kV, 85-190 mrad) CL EELS	-	PCA [86]	[87]

(Continued)

2.1. Characterisation via Electron Microscopy and Spectroscopy

Table 2.1 *Continued.*

Material	Impurity (element)	Experimental Technique	Simulation Technique	Analysis Tools	Ref
MoS ₂	Er	ADF (80 kV)	DFT	ESPRESSO [88]	[89]
MoS ₂	Nb, Re	ADF (100 kV, 80-out mrad)	DFT	VASP [83]	[90]
MoS ₂	Re	ADF (60 kV, 80-out mrad)	DFT	In-house Python [91]	[92]
MoS ₂	Pt	ADF (60 kV, 49.5-198 mrad)	Multislice, DFT	JEMS [76] VASP [83]	[93]
MoS ₂	Al	ADF (80 kV)	Multislice	QSTEM [73]	[94]
MoS ₂	Te	ADF (200 kV, 43–200 mrad)	DFT	VASP [83]	[95]
WS ₂	Se	ADF (100 kV, 70-200 mrad)	Multislice, DFT	QSTEM [73] VASP [83]	[96]
WS ₂	Nb	ADF (60 kV)	-	-	[97]
MoS ₂	Re	ADF (60 kV, 30-300 mrad)	Multislice, DFT	Prismatic [11, 12]	[8]
MoS ₂	Defects	ADF (60, 80 kV) (50-198 mrad) (39-156 mrad)	Multislice, DFT	JEMS [76] VASP [83]	[98]
MoS ₂	-	ADF (200 kV) (54–340 mrad)	Multislice	JEMS [35]	[99]
CNT	Cs, I	ADF (60 kV) (79-160 mrad)	Multislice	QSTEM [73]	[100]
Review	Review	TEM, ADF	DFT	-	[53]
Review	Review	TEM, ADF	DFT	-	[101]

Single dopant atoms imaged in TMDs are also shown in Table 2.1. Gong *et al* showed clearly that ADF STEM can differentiate between various chalcogen atomic columns by doping MoS₂ with Se at different percentages [65]. Alloying in TMDs via ion implantation was also verified at atomic resolution via ADF STEM by Bangert *et al* [10]. Replacing the single transition metal position has been completed for MoS₂ [8, 50, 89, 90, 92–94], though it should be noted that some single atom studies lack quantitative comparison with image simulations. Additionally, many of those that do compare to image simulation fail to make the comparison on an absolute scale relative to the intensity of the incident electron beam. Kim *et al* is a clear example of this, wherein the experimental image of the MoS₂ lattice is over-filtered and presents significant sample drift and astigmatism [94]. The experimental image is compared directly to a simulation via a line intensity profile, though the simulation clearly does not contain the same levels of blurring. Finally, the line profile traces across the transition metal columns only, failing to show any comparison between the chalcogen columns in each image. In [90], this lack of clarity in comparison is worse, as the image simulation itself is not shown, and again only the transition metal columns are compared [90].

Se and Nb have been identified in the lattice of WS₂ [96, 97]. Beyond single atom doping or alloying, successful creation of TMD Janus* monolayers [64, 96] and lateral (in-plane) heterostructures [78, 79, 81] have been demonstrated and imaged with ADF STEM. The properties of the above alloys and doped structures are described in further detail in subsection 2.2.2(Manipulating the Properties of 2D TMDs).

2D materials can be imaged while tilted away from the ideal imaging axis (orthogonal to electron beam). Initial studies used ADF STEM image simulations to show that by tilting MoS₂ and WS₂, the material thickness could be determined [99]. Recently, the impressive work of Tian *et al* has demonstrated that with precise control over hardware (tilting stage) and with reconstruction software techniques, one can truly extract the 3D material properties of a 2D material [8]. For 3D materials at atomic resolution, a deviation from the ideal imaging axis is almost immediately detrimental to the image quality of the individual atomic columns [102].

Quantification of single atoms and atomic columns has been shown by several groups. As shown in Figure 2.4(a-b) above for 2D materials, the intensity about every atom can be plotted on a histogram [21, 43]. The intensities similar to each other will fall as per the central limit theorem [103, 104], and the subsequent scaling via the Z-contrast scaling exponent will define roughly the position of each

*Janus refers to the Roman god of duality.

2.1. Characterisation via Electron Microscopy and Spectroscopy

element's intensity. At each element configuration's mean intensity, a Gaussian distribution is fit. With this method the uncertainty of each intensity belonging to an element configuration can be determined via the overlap between the Gaussian distributions. For imaging of all structures (2D and 3D), correctly filtered [105] image simulation must be used if accurate quantification is desired [48].

For 3D data, the quantification to determine the thickness of each atomic column can be done in several ways. In simple structures, wherein there is only one elemental species in the atomic columns, the thickness of each atomic column can be quantified via direct comparison with image simulation [45, 48, 106]. In these studies, Lebeau *et al* used the average intensity of the Au atoms within a radius of 0.04 nm from the fitted centre of each atom coordinate. The image was calibrated in an absolute intensity scale relative to the incident electron beam, and can therefore be directly compared to simulation after detector calibration [48]. Figure 2.5(a) shows the experimental and simulated atoms, and their sorting as a histogram based on absolute intensity.

Utilising Voronoi cells, Rosenauer *et al* determined the composition of thin film InGaN [107]. A Voronoi cell encompasses the pixels around an atom which are closest to that atom, though it is a general tool for geometric analysis [108, 109]. In this case, the average value within each atom's Voronoi cell was taken as the atom intensity. This method has also been utilised to understand composition and thickness of GaNAs [110], GaAsBi, GaInAs [111] and GaPAs [112, 113] structures from single ADF STEM images. Additionally, single atoms have been identified and tracked in bulk materials, such as Cs dopants in AlN thin films [106, 114].

H.E. *et al* used probe-position integrated cross-sections (PICS) (total, summed intensity within the cell) to by-pass the difficulties that come from determining the absolute intensity scale [116]. PICS - which are essentially calibrated Voronoi cells - were shown to be more reliable than peak (maximum) intensity in the case of homogeneous NPs, and to be robust to sample tilt, defocus and integration radius. In alloyed NPs, where the atomic columns themselves were still homogeneous, the PICS method also showed a clear decrease in intensity of the Pt columns due to the lower neighbour intensity Pd columns, and vice versa an increase in the Pd column intensities (see [116] for visualisation).

At around the same time, De Backer *et al* used the total integrated intensity within a radius around each atom to count atoms in Pb NPs and Au nano-rods [115, 117]. The experimental image, as shown in Figure 2.5(b), is compared with image simulation and a best-fit model is produced. They then developed a Gaussian mixture model to determine the likelihood of an atomic column intensity of this NP to be of a certain thickness. By varying the number of Gaussian 'components'

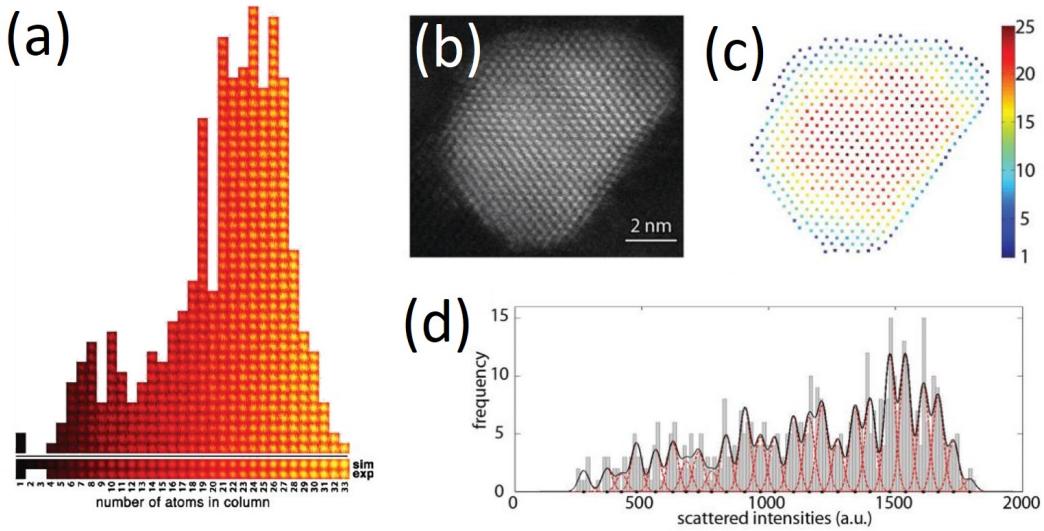


Figure 2.5: (a) Comparison between an experimental ADF STEM image and simulated intensities of different atomic column thicknesses in an Au foil and the resulting counts of each sorted into a histogram (y-axis is frequency of atom count). Taken from [45]. (b) Raw ADF STEM of a Pb NP. (c) Number of atoms in each atomic column of the Pb NP, defined by the Gaussian mixture model in (d). Taken from [115].

in the histogram shown in Figure 2.5(d), a local minimum number of Gaussians is used. Figure 2.5(c) shows the resulting NP structure, with the red and blue colours indicating many and few atoms in the atomic columns, respectively. Follow-on studies have used this method to analyse quantitatively Pt NPs [118], Ru nanostructures [119], and Pt wedges [120]. This Gaussian mixture model approach is implemented in StatSTEM [121] and now also in Atomap [4]. Since then, the groups of Van Aert and Nellist have developed improved methods for ADF STEM analysis of structure changes in NPs [122, 123].

EELS

Although ADF STEM is an accurate and reliable method for analysis of atomic resolution images, it can, in some cases, return ambiguous results [124]. A clear example of this was shown by Senga *et al* [100]. By encapsulating alternating Cs and I atoms in a double-walled CNT, they created a 1D ionic chain shown in Figure 2.6(a). ADF STEM images of the ionic chain revealed, via line intensity profiles, that the lower Z number I has a higher intensity than Cs, as shown in Figure 2.6(d,e). For single atoms, this should not be the case in Z-contrast (the simulations reflect Z-contrast correctly). In the experiment, non-equivalent atomic vibrations in each element causes the overlapping intensity of the nearest Cs and I neighbours to unequally increase and decrease the intensity, respectively. EELS spectrum images (SIs) were captured simultaneously, and correctly show the

2.1. Characterisation via Electron Microscopy and Spectroscopy

positions of the Cs and I atoms in the ionic chain, as shown in Figure 2.6(b,c). Indeed, EELS is also required because Z number does not scale altogether smoothly with scattering cross-section [125].

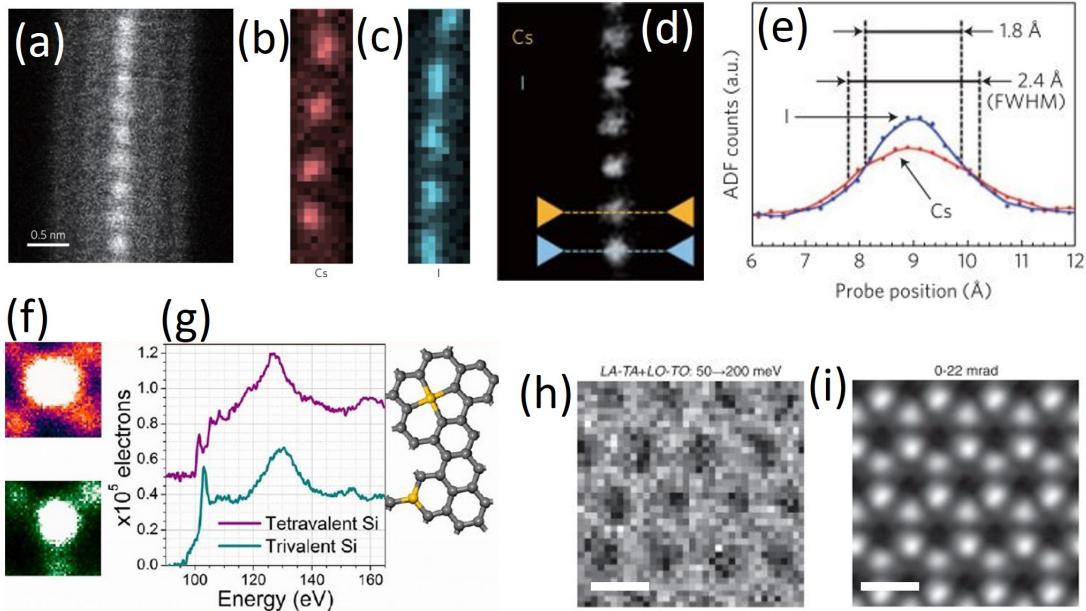


Figure 2.6: (a) Raw HAADF STEM image of a CsI ionic chain in a double walled CNT. EELS signal of the (b) Cs and (c) I $M_{4,5}$ edges. (d) HAADF image of six atoms in the ionic chain. Line profiles over the Cs and I atoms show in (e) that the I have higher ADF intensity. (a-e) Taken from [100]. (f) HAADF STEM images and (g) CL-EELS $L_{2,3}$ edges of tetravalent (top, red/purple) and trivalent (bottom, green) Si atom in a graphene lattice with their respective atomic models. (f, g) Taken from [52]. (h) Experimental and (i) simulated energy-loss signal from LA-TA + LO-TO phonons (50–200 meV) in a hBN lattice at atomic resolution. Scale bars 2.45 Å. (h, i) Taken from [33].

EELS is a multi-functional tool; core-loss (CL) EELS provides composition [100, 126–131] and bonding [52, 128, 132] information, low loss (LL) EELS provides plasmon [133], exciton [78] and band gap information [20], and recently vibrational EELS provided phonon information at atomic resolution [7, 33, 134]. Additionally, if the sample is thick enough for multiple scattering, EELS can also determine sample thickness, assuming the mean free path of the electron is known [135].

Chemical composition of individual atomic columns and atoms can be determined with CL-EELS. As the electron passes near an atom, it causes a energy level transition (ionisation) in the atom's electron cloud. The electron loses some energy due to this transition, and the amount of energy lost is dependent on which electron sub-shell the transition occurs in [125]. Therefore, by placing an energy-loss spectrometer along the central electron beam axis, the chemical composition of elements in a specimen may be found by looking at the relevant ionisation edges

2.1. Characterisation via Electron Microscopy and Spectroscopy

of the element. Changes to the shape (fine structure) of the ionisation edges can determine the bonding state of the atom [52].

Near the turn of the century, Varela *et al* demonstrated the ability of an aberration-corrected STEM to perform spectroscopy at atomic resolution [126]. By acquiring spectra at the atomic column positions of La, Ti and Ca in a LCTO perovskite structure, a La M_{4,5} edge signal could be detected over only the La atomic column. Muller *et al* showed that CL-EELS could be used to map the Ti and Mn 3d edge across an LSMO/STO interface [128]. Liu *et al* demonstrated single W atom identification via CL-EELS in stable monolayer TMD nanoribbons [131]. Single impurities in monolayer graphene introduced via low energy ion implantation were unambiguously identified via CL-EELS by Bangert *et al* [see Figure 2.15] [19]. More recently, single atoms of low Z number elements such as Li have been mapped by encapsulation in a CNT [129]. The work of Ramasse *et al* highlights the impressive capabilities of CL-EELS [52] with regards to bonding investigations. They used the relatively stable 2D structure of Si impurities in monolayer graphene to show that the fine edge structure in the energy-loss spectrum can differentiate between trivalent and tetravalent Si-C bonding at atomic resolution [Figure 2.6(f,g)].

Many different interactions occur to create the LL-EELS spectrum [136]. For plasmon energies at \sim 3-50 eV, the energy-loss is due to collective excitations of electrons in the lattice. In 2D materials, Si atoms in graphene have been shown to exhibit large quantum yield (QY) [137], while Au NPs have shown enhancement of photocurrent in few-layer MoS₂ FETs [138] [see subsection 2.2.2(Manipulating the Properties of 2D TMDs)]. At \sim 1-5 eV exciton transitions and band gap onsets (energy at which the density of states begins to fill) occur. An impressive study by Tizei *et al* reported band gap and exciton energy shifting across a lateral MoS₂-MoSe₂ monolayer heterostructure [78]. The MoS₂/MoSe₂ interface is shown in the ADF STEM image in Figure 2.7(a). LL and CL-EELS SIs were taken across

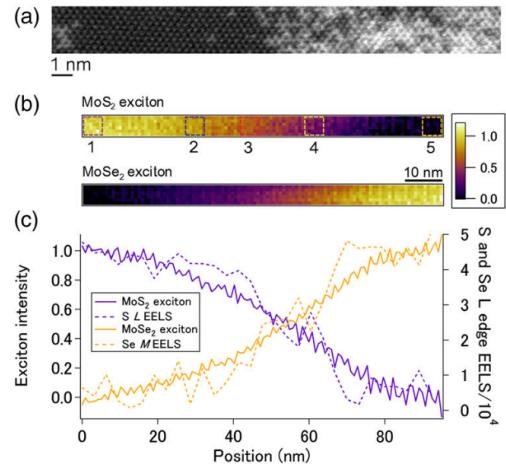


Figure 2.7: (a) HAADF STEM image of a CVD-grown MoS₂/MoSe₂ monolayer interface. (b) Exciton intensity across the interface. (c) Exciton and S L_{2,3} edge intensities across the interface. Taken from [78].

2.1. Characterisation via Electron Microscopy and Spectroscopy

this interface, wherein Figure 2.7(b) the intensity of the MoS₂ and MoSe₂ exciton peaks change. In Figure 2.7(c), the change in exciton and S L_{2,3} edge intensities are displayed across the interface. At the abrupt interface there is a diffuse, gradual core-loss chemical profile, as shown in Figure 2.7(c). However, there is not the same gradual change in the exciton peak energy [Figure 2.7(c)]. This is due to delocalisation effects, which are more pronounced at lower energy losses, and by the ~7 nm radius of excitons in MoS₂ [139]. Gradual energy transitions have been seen with PL in other alloying studies (not lateral heterostructures) [65].

Modern STEMs are now capable of performing EELS in the range of meV. This is referred to as vibrational or phonon EELS. Experiments by Hage *et al* have verified the nature of Z-contrast imaging (inelastic phonon scattering - TDS) [33, 34], and the effect of Si impurities on the acoustic and optical phonon modes in graphene [7]. Atomic resolution visualisation of the LA-TA + LO-TO phonon modes (50-200 meV) in few-layer hBN can be seen in Figure 2.6(i) [L: longitudinal, T: transverse, A: acoustic, and O: optical].

For materials with low EELS signals, such as 2D materials or so-called ‘weak’ CL edges, or if the spectrum has spurious spikes or is noisy, post-processing of EELS spectra is often required. A common tool for denoising of spectra and SIs is component analysis (eigenvalue decomposition), such as principal component analysis (PCA) [86, 140–142], independent component analysis (ICA) [143–145] and vertex component analysis (VCA) [146, 147]. PCA has been utilised by researchers and engineers for over a century and is still widely used today [148]. PCA reduces the dimensionality of data consisting of many interrelated variables, while retaining variation. The majority of the variation of the data is then represented by principal components which are orthogonal to each other. PCA is commonly used in geosciences as a method for unmixing the spectra obtained through remote sensing [149]. Bonnet *et al* tested the ability of PCA for EELS data with a simulation of Si/SiO₂ spectra [86]. Although the authors show the successful application of PCA on such a dataset, they stress that the principal components found via PCA are abstract, and have no physical meaning; they simply represent the data’s variance. Cueva *et al* discuss the common issues and artefacts arising from PCA analysis of EELS data [150]. One issue is the large difference in peak signal magnitude between LL and CL EELS spectra. To resolve this issue, a decaying power law fitted background subtraction [125, 151], or another of the examined methods [150], should first be used to extract the signal from the EELS data. The authors also examined weighted PCA (wPCA), and showed that it is unsuited to EELS-style data, though others have utilised it to good effect by employing Poissonian-weighted PCA [152, 153]. Overall, standard PCA is suitable

for improvement of signal-to-noise ratio (denoising) of EELS data [145, 150]. PCA was also shown to be a useful method for uncovering signals in noisy EELS data [154].

ICA, also known as blind source separation (BSS), assumes that the source of each signal in a dataset are independent, rather than just uncorrelated as in the case of PCA [144, 145]. ICA works by attempting to maximise the non-Gaussianity of the variables. PCA should be run on a dataset prior to ICA to insure the data is pre-whitened (covariance matrix is equal to the identity matrix), because ICA does not handle noise well, and because if the number of components is smaller than the SI dimensions, ICA algorithms can become inefficient [144]. Bonnet *et al* utilised ICA for the analysis of simulated and real Si/SiO₂ EELS data [145], while De La Peña [2] verified that by combining PCA with ICA, one can fully analyse an EELS SI, i.e., by extraction of chemical maps from EELS data. However, thick samples may cause issues due to non-linear multiple scattering [153].

PCA and ICA are explained in further detail and applied to an example EELS dataset in Subsection 3.4.8. In widely used electron microscopy software, PCA can be applied to EELS datasets through the (proprietary) HREM plugin for Gatan DM [70, 71]. Alternatively, open source toolsets can be used, such as scikit-learn [155], HyperSpy [69] or the Cornell Spectrum Imager (CSI) [150], among others. The FastICA algorithm is an efficient and robust version of ICA, and is implemented in the open source project Scikit-Learn [155].

Although EELS may seem an ideal analytical tool for high-resolution chemical and structural investigations, it requires a large number of electron-atom interactions to produce sufficient signal to be statistically significant above the inelastic background noise, i.e., EELS requires a relatively high electron dose/dwell time. As noted by Urban [156], the scattering cross-sections involved in inner-shell (CL) ionisation are relatively low, the resulting dwell times needed to gather sufficient signal will be relatively large. The dwell times involved are comparable to the time scale with which sample drift occurs. This has improved in recent years with the advent of better vacuums, sample holders and brighter electron sources. More electrons increase the probability of damage to the sample. This is not an issue for thick and/or high Z samples which are undamaged by the electron beam. However, for low-dimensional and beam sensitive materials, this is an ongoing problem which limits chemical mapping via EELS or ADF STEM. In the following section, sources of damage due to the electron beam are discussed, and the necessary trade-off between ADF STEM/EELS signal and sample damage is considered.

2.1.2 Electron-Matter Interactions

As a fast electron passes through a sample, the sample can become damaged through both elastic and inelastic scattering mechanisms [157–161]. Elastic scattering involves no loss of energy from the incident electron beam and can result in atomic displacement and for electron energy loss spectroscopy (EELS), the undesirable zero-loss peak. For inelastic scattering, the incident electron beam loses energy which causes damage through various mechanisms. However, these interactions can be used to map chemical and electronic properties in EELS. Multiple scattering of an electron may occur depending on the specimen composition and thickness. The manner with which each scattering event damages the sample are shown in Figure 2.8. In this section, the structural radiation damage due to elastic and inelastic scattering, and their combination, will be discussed. Following this, surface contamination and methods for reducing damage in relevant samples are examined.

Elastic Scattering

Elastic scattering in the electron microscope is the deflection of a fast electron by the nucleus of a specimen atom. Elastic scattering results in two related forms of specimen damage: atomic displacement and e-beam sputtering. Each depend on a threshold energy above which an atom can be displaced in the specimen in the case of atomic displacement or ejected fully from the specimen surface in the case of sputtering. The energy lost by a fast electron and transferred to an atomic nucleus in the specimen is

$$E = E_{max} \sin^2(\theta/2), \quad (2.5)$$

where E_{max} is the maximum kinetic energy defined by

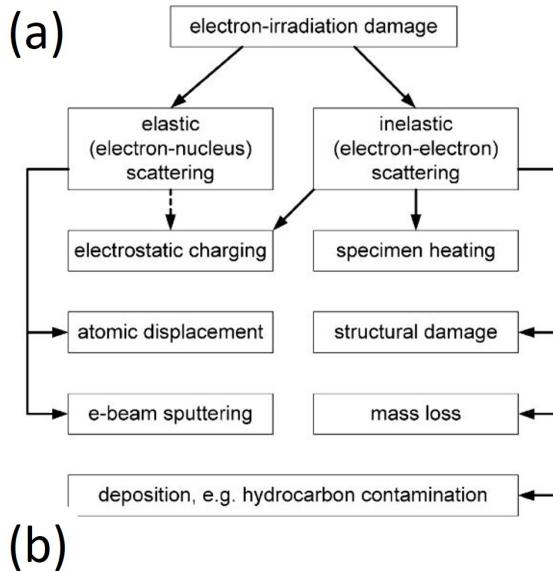
$$E_{max} \approx 2E_0 \frac{E_0 + 2m_0c^2}{Mc^2}, \quad (2.6)$$

where m_0 and M are the electron rest mass and atomic nucleus mass, respectively, c is the speed of light, and E_0 is the electron beam primary energy in eV [162]. At $\theta = 180^\circ$, the maximum energy is transferred (backscattering). Note that as the mass of the atomic nucleus increases, E_{max} decreases. If the energy lost by a fast electron and transferred to a lattice atom, E , is greater than the displacement threshold energy, E_d , of a lattice atom, then the atom can be permanently

displaced from its lattice position. In this case the primary electron beam energy is detrimental to the specimen. As elaborated upon by Egerton *et al*, the only method of avoiding atomic displacement in a specimen is to reduce the incident electron energy to below the threshold displacement energy, i.e., $E < E_d$ [161]. Heating the specimen cannot be used to mitigate damage, because it does not affect the electron-nucleus momentum transfer. Naturally, if the electron beam energy must be above the equivalent threshold displacement energy, reducing the electron beam current and exposure time is the next best solution.

Atoms near the surface of the specimen may be ejected completely via e-beam sputtering. Again, a threshold energy is required for this to occur, though it is usually less than the atomic displacement threshold displacement energy discussed above. At the surface, E_d becomes E_{sub} , where E_{sub} is the sublimation energy ($E_d = (5/3)E_{sub}$ for metallic solids [161]). Collisions from inside the sample to the outside can cause sputtering, though this is only relevant for energies well above the threshold displacement energy E_d [157, 163].

For 2D materials, such as monolayer MoS₂, atomic displacement and sputtering become almost the same, as the material is only made up of a surface. Several groups have studied via DFT the formation and movement of vacancies and atoms in monolayer graphene [164], hBN [165] and MoS₂ [166–168]. Figure 2.9(a) shows the DFT calculation set-up for an electron beam interacting with a monolayer TMD to sputter a chalcogen atom from the exit-beam side of the monolayer [166].



(b)

mechanisms of radiation damage in a TEM, together with typical values of characteristic dose D_e , cross section σ per atom (in barn= $\text{cm}^2 \times 10^{-24}$) and displacement energy E_d .

Mechanism	Specimen	$D_e(\text{C}/\text{cm}^2)$	σ (barn)	$E_d(\text{eV})$
Radiolysis	Organic	0.002–1	10^5 – 10^8	
Radiolysis	Inorganic	0.2– 10^6	0.1 – 10^6	
K-ionization	Any		10^2 – 10^5	
Bulk displacement	Conducting	10^3 – 10^4	10 – 100	10 – 50
Bulk diffusion	Conducting		10^2 – 10^4	0.5–1.5
Surface sputtering	Conducting		10^2 – 10^3	1–10
Surface diffusion	Conducting		$> 10^3$	< 1

Figure 2.8: (a) Classification of radiation damage in an electron microscope, separated into elastic and inelastic scattering. Taken from [160] (b) Table description given above the table. Blank entries are unknown or not relevant to the mechanism. Taken from [161].

2.1. Characterisation via Electron Microscopy and Spectroscopy

For TMDs, the chalcogen (S, Se, Te) displacement threshold energy E_d (usually written as T_d [165, 166, 168]) lie between 5 and 7 eV. These displacement threshold energy values and their vacancy formation energies, along with those for many other transition metals, are shown in Figure 2.9(b). For the exit-beam S sublayer in MoS₂, E_d is 6.9, 7.1 and 7.9 eV calculated with PBE-GGA [166], optB88-vdW and LDA [168] codes, respectively. PBE-GGA is the generalized gradient approximation of the Perdew-Burke-Ernzerhof exchange-correlation functional [169], optB88-vdW is a correction to the PBE-GGA functional [170, 171], and LDA is local density approximation [172]. Yoshimura *et al* explain that LDA often overestimates binding energies, while PBE-GGA underestimates binding energies. They then used the optB88-vdW correction to PBE-GGA, showing a middle ground of 7.1 eV.

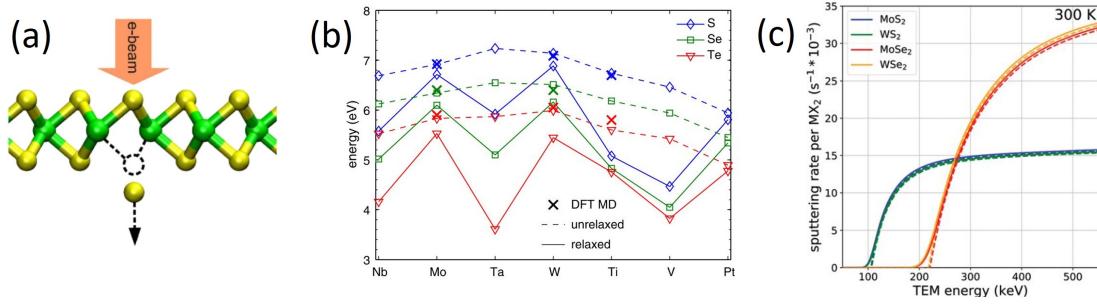


Figure 2.9: (a) Schematic for the DFT calculation of chalcogen sputtering from the bottom (exit-beam) side of a TMD monolayer. (b) DFT calculation of the displacement threshold energies E_d and vacancy formation energies. (a,b) taken from [166]. (c) Sputtering rates of the chalcogen atom per unit cell as a function of TEM primary beam energy for MoS₂, WS₂, MoSe₂, and WSe₂. Taken from [168].

Though the top S sublayer displacement threshold is initially much higher than the bottom (exit-beam) S sublayer, displacement of the bottom S reduces the top S to 8.1 eV (PBE [166]). This top S atom is likely to be displaced to the bottom S vacancy, and subsequently could be sputtered, creating a double S vacancy. An E_d of 6.9 eV corresponds to a primary electron beam energy of 90 keV. Figure 2.9(c) shows the sputtering rates per unit cell for monolayer MoS₂, WS₂, MoSe₂, and WSe₂ as a function of TEM primary beam energy. It shows that the energy required to sputter a chalcogen S and Se atom lies around 90 and 190 keV, respectively. Therefore, imaging below 90 keV is advised for these materials. Subbing the sublimation energy for MoS₂: $E_{sub} = 2.88$ [173] into Equation 2.6, Hansen *et al* calculated 66 keV as the threshold for S sputtering, and so recommend imaging at 60 keV [43, 174]. Though the DFT calculations specifically for electron-MoS₂ interactions [166, 168] are likely more accurate in this case than the stopping range of ions in matter (SRIM) calculations [173], the 60 keV recommendation

is a pragmatic one. This is because at edges and kink-sites, the displacement threshold energy is reduced, increasing the likelihood of sample damage [161, 174]. However, 60 keV should significantly decrease the sputtering damage inflicted upon the sample due to elastic electron-nucleus interactions. In recent years, with the advent of high-order aberration correction, 60 keV STEM imaging has achieved atomic resolution in ADF mode [5, 6, 21].

Inelastic Scattering

Figure 2.8(a) also shows the damage mechanisms due to inelastic scattering. As a fast electron interacts with a lattice atom electron, a large amount of momentum can be transferred, causing heating, structural damage, mass loss and also deposition of contamination. The majority of the energy in electron-electron collisions is converted to heat which at high doses can melt the sample [160, 175]. For 2D materials with low thermal conductivity, such as MoS₂ [176], this could be an issue, as there is also a limited (2D) radial heat flow.

Radiolysis (ionisation), which is damage to the specimen via inelastic scattering, is more pronounced in organic than inorganic materials. It seems radiolysis is detrimental at some threshold current density (dose), as emphasised by Egerton *et al* [160, 161]. Mass loss via radiolysis is often exponential with dose or irradiation time [161]. Figure 2.8(b) shows the usable range of dose (C/cm²) for radiolysis for both inorganic and organic materials.

Radiolysis can be limited by the electrical conductivity of the specimen [161]. In a metal, the abundance of extra electrons means ionisation is not an issue, because the time taken to replace a vacancy band hole after an excitation ($<10^{-15}$ s) is less than the time of an atomic vibration ($\sim 10^{-13}$ s) which could move the atom [158]. In semiconductors or insulators, there are less electrons available to fill holes in the valence band after an electron is excited. In this time an electron-hole pair is formed. If these are produced near the material surface, they can be ejected as secondary electrons. Within the material, they can exist as internal secondary electrons, which travel through the sample, creating further electron-hole pairs, and subsequently become thermalised within 10^{-12} s [175]. The remaining valence band holes may not be filled, and an atomic vibration may break bonds in the lattice, weakening the overall structure. Subsequent scattering may then displace these atoms. Cooling the sample can reduce radiolysis [158, 177]. In some cases, inelastic scattering of a fast electron from a lattice atom can excite an inner-shell electron. As noted by Egerton, this is much less likely (by a factor of about 100). However, one can imagine that at a very high dose rate (current density), inner-shell ionisation can damage the sample considerably.

2.1. Characterisation via Electron Microscopy and Spectroscopy

For MoS₂, the electron beam can induce local phase changes, from semiconducting (2H) to metallic (1T). It is expected that in the metallic 1T region, radiolysis effects are dampened [178]. This would be an interesting experiment to perform, and one which could simultaneously quantify such electron-matter interactions in two materials with one image.

For MoS₂ on graphite (high electrical conductivity), Hansen *et al* predict that radiolysis is expected to be negligible [43]. However, as the primary electron beam energy is reduced (60 keV here), the scattering cross-section is increased, increasing the probability of radiolysis [179]. For certain materials, then, a higher primary beam energy is beneficial. Hole drilling has been shown to be due to the current density (dose rate), not the total electron dose, in CaF₂ and Al₂O₃ [179]. The primary electron beam energy used for the materials was under the displacement threshold energy for elastic scattering. For TM oxides radiolysis occurs through the Knotek-Feibelman mechanism. This is the incident electron creating an inner-shell metal vacancy followed by inter-atomic Auger decay from the oxygen. This oxygen ion (positive or negative) is repelled by the metal ions and ejected into the vacuum, resulting in mass loss [159]. This process is known as electron-stimulated desorption (ESD).

For radiolysis in organic materials, e.g., polymers, the change in electronic configuration of a molecule due to excitation of the sample atom by a fast electron may not reform when de-excited [180]. This bond breakage can cause loss of H, O, and N to the surrounding. In poly(methyl methacrylate) (PMMA), for example, about 80% of the damage stems from secondary electrons in the polymer [181]. This is quite relevant to 2D TMDs, as many are synthesised or fabricated on polymers. H, O and N may substitute the chalcogen vacancies inherent in the TMD, or created via atomic displacement. The replacement of vacancies with H, C, N and O atoms has been calculated by Nguyen *et al* [77]. C, N and O were found to have positive dissociation energies and therefore provide stable replacements for Se in NbSe₂. Hydrocarbon contamination can also provide such replacement elements.

Minimising Radiation Damage

Egerton summarises well the checklist for whether elastic or inelastic scattering is the dominant damage mechanism in a sample [161]. Here, we briefly describe some methods for reducing damage in a sample while imaging with high energy electron beams. Overall, reducing the electron beam current and imaging time is the best method for reducing radiolysis, while imaging at a primary electron beam energy lower than the displacement threshold energy should reduce atomic displacement

and sputtering. Susi *et al* have succinctly reviewed these irradiation effects for 2D materials [182].

Although using a high Z element (such as W) to coat the exit-beam side of the sample can reduce sputtering damage, [160], it is not realistic for 2D materials; the W layer would be much brighter than the 2D material. However, researchers have used graphene monolayers as protection against sputtering from monolayer MoS₂ [183, 184]. Contrast-wise, the graphene contribution to the image signal can be filtered out via a simple Fourier filter. It also has been demonstrated that a carbonaceous shell around a CeO₂ nano-cube can reduce damage [185]. Though the damage is not quantified via atom counting, EELS is used to verify O vacancies and reductions in the nano-cube, similar to [186]. The authors found that damage to the CeO₂ nano-cube occurred above a threshold dose rate rather than a total electron dose, which implies that imaging below this threshold dose rate allows damage free imaging, regardless of imaging time. Below the threshold dose rate, recovery of the O vacancies occurs faster or as fast as vacancy production by e-beam damage.

Another consideration is the relative magnitude of scattering at different primary electron beam energies and how this affects image contrast. Electrons of lower energy are more strongly scattered. This implies that at lower primary electron beam energies a sample that is thin, e.g., low-D materials, will have sufficient image contrast. As mentioned above when considering radiolysis, this can also be detrimental, as there is more interaction between fast electron and specimen, increasing the probability of induced damage [182]. Additionally, for low-contrast biological specimens, low primary electron beam energies are sometimes preferred [161]. For ADF STEM, a lower inner collection angle produces more image contrast. Tian *et al* used a low inner ADF collection angle of 30 mrad to increase image contrast while reducing the electron current density [8]. It should be noted that this will also have the adverse effect of increasing the low Z contrast from hydrocarbon contamination. Additionally, the common ionisation events and their energies must be known before choosing the primary electron beam energy, as certain values may increase the probability of ionisation.

Heating the sample can reduce the gathering of hydrocarbon contamination as a result of radiolysis by desorbing the surface material but increases damage when imaging polymers [160]. Reducing the temperature of the specimen helps to minimise radiolysis in inorganic materials [177].

Finally, damage to the sample may be desirable, in the form of lithographic polymer or hydrocarbon polymerisation patterning [160, 187] or engineering nano-scale structures in 2D materials [101, 188]. With lower primary electron energies

[5, 189], better hardware, and better dose limiting acquisition and post-processing techniques [190], low Z samples can now be imaged at atomic resolution in three dimensions [8].

Contamination on 2D Materials

2D materials appear to be the ideal structure for imaging at high-resolution in an electron microscope. When suspended over vacuum, they are the thinnest materials available, and therefore single atoms and minute shifts in their position can be analysed quantitatively. Unfortunately, hydrocarbon or polymer contamination resulting from sample preparation or exposure to air can cause many contrast issues. Because of this, it is a frequent occurrence that only small regions of the 2D material are suitable for imaging at atomic resolution.

Hydrocarbons can diffuse along the specimen surface toward the electron beam, where they are then polymerised, making them difficult to remove and reducing image contrast. Egerton *et al* lists several methods for reduction of hydrocarbon contamination: heating the sample under a UV lamp in air or vacuum, plasma cleaning, electron beam shower, in-situ heating, and cooling of the specimen [160]. At atomic resolution, Lin *et al* showed that hydrocarbon contamination gathered near individual Re dopants in monolayer MoS₂. This verified that the dopants were defective sites, causing lower adsorption energy and subsequently providing anchor points for contamination, and overall effected the local physical and chemical properties of the material.

In mechanical exfoliation [19, 191], lithographic [192], and chemical vapour deposition (CVD) [193, 194] synthesis methods, polymers may be used as a protective or transfer layer, therefore requiring a solvent rinse which can introduce more contamination. Contamination on 2D material surfaces has been treated with varying degrees of success: heating with activated carbon [195], heating with radiolized water [196], laser heating [197, 198], heating after solvent cleaning [10, 199], in-situ TEM heating [200], heating in ultra-high vacuum, plasma treatment [201] and UV-ozone pre-treatment of polymer (PDMS) [202]. Surface contamination arising during or after sample preparation reduces image contrast which can adversely affect single atom imaging experiments. Rigorous sample cleaning is required for atomic resolution STEM and EELS experiments. For other experimental methods such as Raman spectroscopy and PL, this is not necessarily the case, though the effect of contamination on the experiment is difficult to discern. In Subsection 3.1(Sample Preparation) these considerations were taken into account, and appropriate cleaning was executed.

2.2 Two-Dimensional Materials and Transition Metal Dichalcogenides

Two-dimensional (2D) materials have substantially increased the interest in condensed matter systems due to their fascinating properties [203]. Atomically exfoliated molybdenum disulphide (MoS_2), a member of the transition metal dichalcogenide (TMD) family, is one such material. Through photoluminescence (PL), absorption and reflectivity measurements, Mak *et al* unambiguously discovered an indirect to direct band gap change from bulk to single layer MoS_2 [204]. Since then, several studies have shown that MoS_2 promises interesting electrical [205–209], optical [210–212], and mechanical properties [213, 214] which can be exploited for flexible (opto-)electronic devices. Choi *et al* [215] investigated multilayer MoS_2 and found photogeneration and photoresponse in wide spectrum ranges. Due to this, and MoS_2 multilayer band gap values, they concluded that MoS_2 could potentially be integrated into various optical sensors that require a broad range (UV to near-IR) of spectral responses.

Many of the above-mentioned papers use low spatial resolution spectroscopy for analysis of the properties of MoS_2 and other 2D materials. Though we will focus mostly on high-resolution studies in the following sections, and indeed throughout this work, it is important to note that both high-resolution and high throughput studies of 2D material doping are vital for the future use of these materials in industry. This review aims to give the reader a general overview of the breath of possible application of 2D TMDs in devices, as well as how manipulating the fine atomic structure can provide new properties and opportunities for the field. First, the structure and pristine properties of 2D TMDs, centred on the extensively studied MoS_2 , will be discussed in subsection 2.2.1(Properties of 2D TMDs). Then, subsection 2.2.2(Manipulating the Properties of 2D TMDs) will describe in detail the common methods for changing and controlling the properties of 2D TMDs. Finally, ion implantation of 2D TMDs will be discussed in subsection 2.2.3(Ion Implantation of 2D Materials).

2.2.1 Properties of 2D TMDs

In the previous decade, two-dimensional (2D) transition metal dichalcogenides (TMDs) have become a central focus in 2D materials research. TMDs come in many forms and naturally have different properties. The unit cell has the general form MX_2 , where M is a transition metal and X is a chalcogen e.g., MoS_2 , WSe_2 , NbTe_2 [216–220]. Figure 2.10 shows a monolayer MoS_2 example in 3D, image (top)

and cross-sectional views. Both image (top) and cross-sectional views are used extensively in Chapter 4(Atomic Characterisation of Ion Implanted 2D TMDs) and Chapter 5(Atom Dynamics in 2D TMDs). Ideally, the image (top) view is orthogonal to the electron beam, i.e., a projection of this view will be seen during transmission electron microscopy (TEM) imaging. The cross-sectional view is useful for visualising the atoms present in each atomic column as imaged in the image (top) view. In the case of the cross-sectional view in Figure 2.10, the atomic columns are alternating Mo₁ and S₂ element configurations. In the 3D structure additional layers of TMDs are bonded to each other via Van der Waals forces [216, 218–222]. These layers can be removed via mechanical, chemical, or other means. Sample preparation of TMDs is described in the methodology in Subsection 3.1(Sample Preparation).

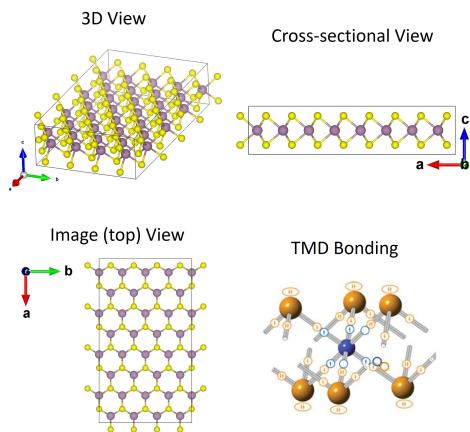


Figure 2.10: 3D, image (top) and cross-sectional views of the structure of the TMD MoS₂. The image (top) view is the imaging view, orthogonal to the electron beam. (TMD Bonding) The bonding of an octahedral TMD structure showing the distribution of valence electrons. Taken from [220].

TMDs have impressive physical properties. Using an atomic force microscope (AFM) Bertolazzi *et al* measured the stiffness and breaking strength of suspended monolayer MoS₂ to be 180 ± 60 N m⁻¹ and 15 ± 3 N m⁻¹ (23 GPa), respectively [213]. The Young's modulus of 270 ± 100 GPa is comparable to that of steel, though only a fourth of that of graphene [223, 224]. These experiments were carried out on samples of similar configuration to that of samples prepared for TEM i.e., suspended over a pre-fabricated hole. The above measurements were carried out for the in-plane strength etc. of monolayer MoS₂, which relates to the axis direction 'a' in

Figure 2.10 above. With such a Young's modulus, it is likely that MoS₂ and similarly strong TMDs will be utilised as reinforcing elements in composite materials. Because of the high breaking strain (6-11%), MoS₂ could also be used in flexible electronic applications [213, 225]. Additionally, introducing high strain into the TMD lattice can induce phonon shifts in MoS₂ [214], WS₂ [226, 227], WSe₂ [227], as well as MoS₂/WS₂ heterostructures [228]. These changes to the structure of pristine TMDs [229, 230] lend themselves to tunable flexible properties that can be incorporated into opto-electronic devices along with polymer substrates [214,

226].

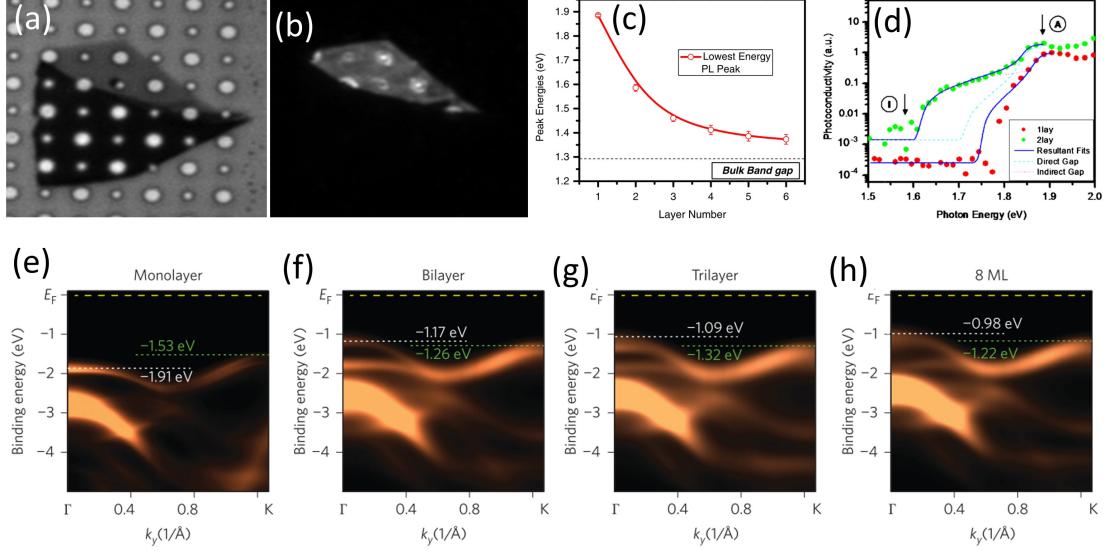


Figure 2.11: (a) Optical microscope image of monolayer and few-layer MoS₂. The large holes are 1.5 μm in diameter. (b) PL signal of the area in (a). (c) PL peak position of the relevant peaks as a function of the number of MoS₂ layers. (d) Photoconductivity as a function of photon energy for monolayer and bilayer MoS₂. (a-d) Taken from [204]. ARPES spectra of (e) monolayer, (f) bilayer, (g) trilayer and (h) eight-layer MoSe₂. (e-h) Taken from [231].

The band structures of several TMDs were calculated by Bromley *et al* [216], though the reason for differing n- and p-type doping of pristine MoS₂ on SiO₂ - a common substrate - was explored only recently [232]. Unlike graphene [233], TMDs have a non-zero band gap, and are therefore promising for electronic applications [234]. Additionally, the already high mobility in TMDs [235–237] can be increased via doping [94, 238] or decreased as a result of lattice defects [239]. Furthermore, band gaps in some TMDs such as MoS₂ and MoSe₂ change depending on their thickness [204, 231, 240, 241]. Mak *et al* verified this phenomenon by tracking the peak intensity of the photoluminescence (PL) response as well as the absorbance and reflectively of monolayer flakes to six-layer MoS₂ flakes [204]. Figure 2.11(a) and Figure 2.11(b) shows an optical microscope image and PL image of monolayer (top-region) and few-layer MoS₂ on a Si substrate with etched holes of varying sizes. The top-region visible in Figure 2.11(b) is the monolayer PL signal, with the suspended MoS₂ over the substrate holes producing the highest PL signal and a quantum yield (QY) of 10^4 compared to bulk. The few-layer region is not visible due to its low PL intensity. Figure 2.11(c) shows the peak PL intensity of the A and I emission peaks for monolayer and multi-layer, respectively. The monolayer band gap is measured to be \sim 1.8–1.9 eV, decreasing in value as more layers are added, with a bulk value of \sim 1.23–1.29 eV [219, 220, 242–244]. In Figure 2.11(d)

the photoconductivity of mono and bilayer MoS₂ matches well with theoretical modelling for direct and indirect transition behaviours, respectively.

Zhang *et al* verified in epitaxial beam grown MoSe₂ the indirect to direct band gap transition from multilayer to monolayer via angle-resolved photoemission spectroscopy (ARPES), as shown in Figure 2.11(e-h) [231]. Figure 2.11(e) shows the direct band gap nature of monolayer MoSe₂, as the valence band energy closest to the Fermi energy is highest at the K-point at -1.53 eV. The indirect Γ -point lies at -1.91 eV. This is flipped in bilayer, trilayer and other multilayer samples, with the indirect Γ -point lying closest to the Fermi energy. The direct band gap in monolayer is due the infinite TMD inter-layer distance, i.e., there are no other layers. The z-orbitals at the Γ -point contribute less in this condition than the xy-orbitals at the K-point, resulting in a direct K-point transition. Further measurements verify the band gap of \sim 1.58 eV for monolayer MoSe₂ (1.55 eV in [241]), compared to 1.1 eV in its bulk form [245]. Therefore, the MoSe₂ monolayer band gap is \sim 0.22-0.35 eV smaller than monolayer MoS₂. Exploiting this difference in band gap energy is the basis of the Se ion implantation experiments carried out during this work in Subsection 4.1(Se Ion Implantation in MoS₂), and is described in further detail in subsection 2.2.3(Ion Implantation of 2D Materials).

Regarding device development, Tongay *et al* speculate that monolayer MoSe₂ is “relevant to device applications involving the solar spectrum, as it is near the optimal band gap value for single-junction solar cells and photoelectrochemical cells” [241]. For MoS₂, its strong monolayer PL signal identify it as a possible photostable marker or sensor in other PL studies on the nano-scale. Additionally, the controllable band gap energies for differing layer thicknesses in both MoS₂ and MoSe₂ enable applications of these materials in photovoltaics. Monolayer transistors have already been developed using MoS₂ [205]. Growth of lateral heterojunctions of TMDs opens further possibilities for device development, as shown in the impressive atomic resolution work of Huang *et al*, in which MoSe₂-WSe₂ structures were created via physical vapour transport (PVT) [81]. The properties of pristine materials discussed in this section is, of course, only part of the TMD and 2D material research sphere. Changing the properties of pristine TMDs is an exciting avenue of research employing many different methods: from standard foreign atom doping and vacancy production to heterojunctions and in-plane phase transitions. In the next section, manipulation of 2D materials will be discussed. Finally we will focus on the manipulation of 2Ds via ion implantation.

2.2.2 Manipulating the Properties of 2D TMDs

There are numerous methods for doping and manipulating the properties of TMDs and other 2D materials. Indeed, for certain material properties such as graphene's electrocatalytic effect, doping can provide tens of thousands of possible experiments [246]. As described in subsection 2.2.1(Properties of 2D TMDs) above, TMDs offer an exciting prospect with regards to doping. Their metal/chalcogen structure [220, 247], variable band gap energy and indirect to direct band gap properties [204, 231, 240, 241], provide numerous choices for manipulation of their opto-electronic properties. The doping and manipulation of materials positioned on a substrate, as in the case of many of the electronic and plasmonic studies discussed here, will likely have different effects compared to effects in suspended materials (e.g., in high-resolution electron microscopy experiments) [248].

This section discusses the current state of the art methods used to dope, change, control, and/or manipulate the properties of TMDs, with reference to other 2D materials. Phase transitions, out-of-plane and in-plane heterostructures, and doping via substitutional atoms, adatoms and surface contacts will be described. Manipulation via ion implantation will be discussed separately in the subsequent subsection 2.2.3(Ion Implantation of 2D Materials).

Defect-doping of TMDs has been explored by several groups, though it leads to weaker lattice bonding and mechanical properties [167, 249]. Defect-doping can originate from electron-beam drilling [53, 98, 168, 250], ion-implantation [250–253], inherent defects created during growth [254], or by ambient effects [98]. Defect states have been shown to produce single photon emission (SPE) in both WSe₂ through physical lattice bending between gold contacts [255–257] and in MoS₂ through He-ion irradiation [251]. However, these SPE studies included no high-resolution microscopy or spectroscopy, and it is therefore difficult to be certain of the origin of the SPE.

Scanning transmission and transmission electron microscopy (S/TEM) has been utilised to create and study defects and holes in 2D materials [53, 98, 258] at atomic resolution. Extensive hole creation in monolayer MoSe₂ under the electron beam can produce electronically stable MoSe nanowires [259]. The edge of smaller holes and vacancies were studied extensively by Wang *et al* [98]. They modelled the adatoms produced from a 0.9 nm diameter hole, as well as other single and multiple vacancy sites, and their formation energetics using DFT. Other theoretical studies showed the effect the electron beam can have on the phase of MoS₂ by creating single and double vacancy lines [178]. These studies, as well as graphene electron beam drilling experiments, are summarised in a comprehensive review by

Zhao and colleagues [188].

MoS_2 has been shown to exhibit a phase transition from its semiconducting 2H phase to a metallic 1T phase under a STEM beam [200, 221, 262, 263]. The authors verified this as a controllable method, which opens the door to formation of varying conductive domains in monolayer TMDs simply by exposing the material to radiation, and without needing further dopants. Sulphur vacancies can create additional bands between the conduction band minimum (CBM) and Fermi level in bilayer MoS_2 [264]. It was found using DFT that the S vacancy and charge transfer between the two monolayers can be used to control the magnetic and electronic properties of the MoS_2 bilayer.

Many 2D materials lend themselves naturally to stacking due to their out-of-plane Van der Waals bonding, allowing one to build different heterostructures [260, 265, 266], just like using lego [see Figure 2.12(a)]. The exciting aspect of this is the differing properties evident in each different heterostructure, from light emitting diodes (LEDs) [261] to PL quenching [267]. Figure 2.12(b, c) shows the atomic model and actual stacked structure of different 2D crystals to make up a functioning LED: $\text{Gr}/\text{MoS}_2/\text{Gr}/\text{hBN}$.

Jariwala *et al* discuss how 2D heterostructures go beyond just including 2D materials [268]. Adding quasi-0D, 1D or 3D structures to the surface of monolayers or 2D heterostructures can augment functionality and alternative properties not available within the 2D material family itself. The addition of quantum dots, such as Au NPs, are discussed in detail at the end of this section and introduced experimentally in Subsection 4.2(Au Ion Implantation on TMDs). 2D hexagonal-Boron Nitride (hBN) has been shown to act as a good insulator substrate for

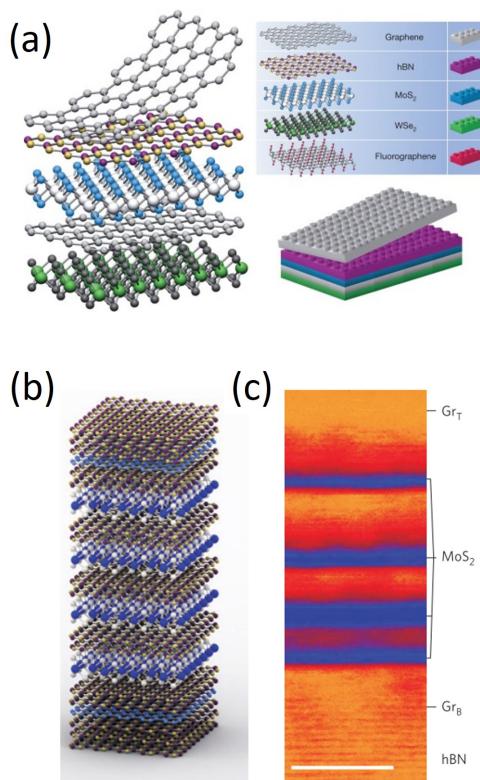


Figure 2.12: (a) Graphic depicting the various possible heterostructures available due to 2D material stacking [260]. (b) Atomic model of the structure imaged in (c), from which LED structures were built [261]. Gr_T and Gr_B are the top and bottom graphene layers, respectively. Scale bar is 5 nm.

graphene [269]. hBN has also been shown to act as a suitable substrate [270] and an n-type dopant for monolayer MoS₂ [271].

Creating lateral (in-plane) heterostructures is also a way of changing the pristine properties of TMDs. For example, Huang *et al* synthesised via PVT a MoSe₂-WSe₂ lateral monolayer heterostructure and verified their growth with atomic resolution STEM [81]. Monolayer junctions are now possible with these methods, allowing the researcher to create their property of choice within specific 2D domains e.g., different band gaps in different areas of the lateral heterostructure.

Depending on the doping percentage, MoS_xSe_{2-x} alloys have been shown to create tunable band gap structures [65, 78, 79, 272]. In their excellent study of MoS₂-MoSe₂ grown lateral heterostructures using high energy and spatial resolution electron energy loss spectroscopy (EELS), Tizei *et al* showed variable exciton energies as a function of distance from the interface [78]. Band gap and exciton energies of TMD structures relevant to this thesis are tabulated in Table 4.3 in Subsection 4.1.2(Low Loss EELS Characterisation). The MoS₂-MoSe₂ interface is shown in the ADF STEM image in Figure 2.13(a). From this interface EELS scans were taken over the surface and verify the movement of the low-loss EELS signal near the band gap onset. Control over the doping percentages of these structures, therefore, can lead to tunable electronic properties in single layer materials.

Utilising a lateral epitaxial growth method based on chemical vapour deposition (CVD), Duan *et al* created WS₂-WSe₂ and MoS₂-MoSe₂ lateral heterostructures [273]. At the same time, Gong *et al* [79] and Huang *et al* [81] created MoS₂/WS₂ and MoSe₂/WSe₂ lateral heterostructures, respectively. Figure 2.13(b) shows a WS₂/MoS₂ lateral monolayer heterostructure interface, with Figure 2.13(c) displaying the atomic model of the same area. The red arrow indicates a W atom in

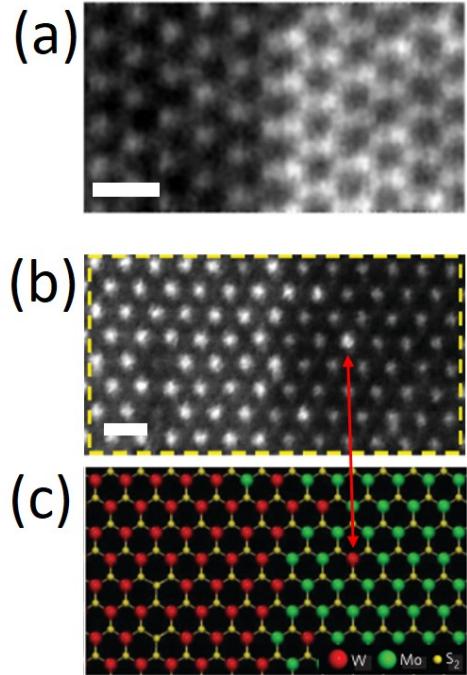


Figure 2.13: (a) ADF STEM image of MoS₂-MoSe₂ lateral monolayer heterostructure interface. Adapted from [78]. (b) ADF STEM image and (c) atomic model of WS₂/MoS₂ lateral monolayer heterostructure interface. Adapted from [79]. Scale bars are 5 Å.

Utilising a lateral epitaxial growth method based on chemical vapour deposition (CVD), Duan *et al* created WS₂-WSe₂ and MoS₂-MoSe₂ lateral heterostructures [273]. At the same time, Gong *et al* [79] and Huang *et al* [81] created MoS₂/WS₂ and MoSe₂/WSe₂ lateral heterostructures, respectively. Figure 2.13(b) shows a WS₂/MoS₂ lateral monolayer heterostructure interface, with Figure 2.13(c) displaying the atomic model of the same area. The red arrow indicates a W atom in

the MoS₂ lattice. It is therefore quite possible to create functional devices, from logic circuits to photodetectors, solely from single-layer materials, such as the above-mentioned TMDs. This is indeed beyond graphene, as pristine graphene properties are limited for electronic applications.

Measurement of CVD-grown Janus monolayers of selenized MoS₂ was reported by Lu *et al* [64], as shown in Figure 2.4. A Janus TMD monolayer is one in which the top and bottom chalcogen sub-layers are different elements e.g., in this case S-Mo-Se. The authors reported out-of-plane piezoelectricity and an optically active out-of-plane dipole in their novel structure, useful for study of light-matter where dipole orientation must be fixed. Such consistent control over the internal structure of a monolayer is astounding, and reveals a new form of 2D material.

Single substitutional atom dopants such as C [21, 77], N [77, 274], O [21, 77], Al [94], Si [51–54], V [84], Cr [84], Co [85, 87], Se [65, 275], Se (this work, ion implantation) [10], Nb [90, 97, 276, 277], Te [95], Er [89], W [238], Re [50, 90, 92, 276], Pt [77, 93], and Au [50] have been analysed experimentally by high-resolution electron microscopy, and are tabulated in Table 2.1 in subsection 2.1.1(Single Atom Microscopy and Spectroscopy). The introduction of foreign atoms into single layer MoS₂ is of great interest in terms of creating new properties, especially for optoelectronics. Monolayer TMDs can be substitutionally doped by replacing atoms of the chalcogen and/or transition metal sites. In the first results chapter below - Chapter 4(Atomic Characterisation of Ion Implanted 2D TMDs) - both were attempted: Se for introducing chalcogen doping and Au for doping of transition metal sites. Studies that carry out single atom substitution or predict material properties via first principles calculations are discussed in the following subsection. Adatoms, which are expected to be introduced when doping monolayers, are discussed in more detail in the subsequent subsection.

Se has been introduced into the MoS₂ lattice in various ways, as with the above-described lateral (in-plane) heterostructures. MoS_xSe_{2-x} monolayers have been grown via CVD at varying concentrations and verified with atomic resolution STEM [65, 275]. Both studies report gradual but distinct decreases in band gap energy from pure MoS₂ to Se doped MoS₂. Results from DFT calculations of band gaps vary, with the level of band gap doping at certain low concentrations lower or higher than found experimentally [176, 278]. By introducing via CVD a high percentage of Te (27.6%) into a MoS₂ monolayer, Yin *et al* reported an increased lattice constant and numerous line defects, though the overall 2H-MoS₂ structure is intact [95]. Strain in the lattice, and its effect on the band gap (measured via PL), was studied experimentally and with DFT. Vacancies assisted in releasing

2.2. Two-Dimensional Materials and Transition Metal Dichalcogenides

The equilibrium supercell parameters a with unit Å. The bond lengths $d_{\text{Mo-NM}}$ of Mo—NM bonds, $d_{\text{Mo-S1}}$ of Mo—S₁ bonds, $d_{\text{Mo-S2}}$ of Mo—S₂ bonds and $d_{\text{Mo-S3}}$ of Mo—S₃ bonds with unit Å. The h is the distance from impurity atom to the reference plane with unit Å. The calculated formation energies E_f under Mo-rich and S-rich conditions with unit eV. The calculated band gap values E_g with unit eV. The energies of Fermi level, VBM and CBM (E_F , E_{VBM} and E_{CBM}) with unit eV.

	a	$d_{\text{Mo-NM}}$	$d_{\text{Mo-S1}}$	$d_{\text{Mo-S2}}$	$d_{\text{Mo-S3}}$	h	$E_f^{\text{Mo-rich}}$	$E_f^{\text{S-rich}}$	E_g	E_F	E_{VBM}	E_{CBM}
Pristine	6.419	2.438	2.438			1.584	0.00	0.00	1.86	-4.41	-4.41	-2.55
H doped	6.392	2.044	2.430	2.419	2.431	0.901	1.17	2.42	1.93	-3.49	-4.58	-2.65
B doped	6.433	2.129	2.426	2.427	2.456	0.893	2.51	3.76	1.62	-4.61	-4.61	-2.99
C doped	6.418	2.037	2.446	2.435	2.480	0.806	2.36	3.61	1.79	-4.25	-4.25	-2.46
N doped	6.405	2.025	2.456	2.441	2.470	0.991	0.95	2.20	1.89	-4.39	-4.89	-3.00
O doped	6.393	2.091	2.444	2.427	2.445	1.056	-2.68	-1.42	1.79	-4.29	-4.29	-2.50
F doped	6.411	2.270	2.429	2.415	2.435	1.309	-1.78	-0.53	1.93	-3.25	-4.84	-2.91
Si doped	6.453	2.414	2.433	2.432	2.461	1.381	2.71	3.96	1.76	-4.29	-4.29	-2.53
P doped	6.427	2.457	2.440	2.443	2.435	1.601	0.36	1.61	1.76	-4.66	-4.66	-2.90
Cl doped	6.431	2.550	2.434	2.426	2.438	1.731	-0.13	1.12	1.86	-3.08	-4.73	-2.87
As doped	6.427	2.605	2.437	2.440	2.425	1.816	0.60	1.85	1.80	-4.47	-4.47	-2.67
Se doped	6.427	2.568	2.436	2.440	2.432	1.771	-1.20	0.05	1.84	-4.12	-4.12	-2.24
Br doped	6.437	2.695	2.433	2.429	2.431	1.958	0.37	1.62	1.86	-3.15	-4.76	-2.90
Te doped	6.438	2.767	2.435	2.441	2.428	2.050	-0.58	0.67	1.82	-3.95	-3.95	-2.13
I doped	6.444	2.892	2.435	2.433	2.427	2.165	0.80	2.05	1.86	-3.21	-4.76	-2.90

Table 2.2: Table taken from [278].

strain near line defects, while Te doped MoS₂ introduces much greater changes to the MoS₂ lattice compared to Se. Kim *et al* successfully substituted Mo in CVD-grown monolayer MoS₂ with Al atoms added via atomic layer deposition (ALD) from an Al₂O₃ thin film [94]. They report, in both vacuum and ambient conditions, a 30-fold mobility increase from 0.09 to 2.71 cm² V⁻¹ s⁻¹. They verified the substitutional Al at Mo sites using the QSTEM image simulation package [73], though they only show a line intensity profile over the transition metal sites. Indeed, it would be interesting to know whether the authors found Al in the chalcogen position due to the similar atomic numbers of Al (Z=13) and S (Z=16). The structure's ~660 nm PL peak increased and down-shifted, which the authors attribute to the Al incorporation and defect states.

Substitutional and adatom doping of MoS₂ has been shown at atomic resolution for both Re and Au by CVT [50]. Re was found to lower the doping energy for adjacent sites, which results in Re adatom segregation, implying that it may be difficult to create uniform Re doped MoS₂. Single Au atoms were found to be very mobile under the e-beam (likely adatoms). Both substitutional dopants were shown via DFT to introduce magnetism to the lattice. Magnetism has also been shown for V, Fe, Co, Cu [279], Mn [279, 280], B, N, and F [281] doped MoS₂ monolayers, as well as in the afore-mentioned S vacancies in bilayer MoS₂ [264]. Karthikeyan *et al* comprehensively calculated the possible magnetic properties of many elements in 2D TMDs [282]. For Re doping, EELS showed nearby gathering of hydrocarbon contamination due to substitutional Re sites acting as defective sites, lowering the adsorption energy and providing anchoring points for accumulating amorphous carbon.

First principles calculations, such as DFT, can be used to predict the local doping effects of single substitutional atoms, as mentioned above for magnetism. These studies can be utilised to further understand the band structure and bonding changes caused by individual dopants [282–284]. Of particular note is the impres-

2.2. Two-Dimensional Materials and Transition Metal Dichalcogenides

Parameters for the TM-CO systems: adsorption energy (E_{ad} in eV), charge gained by the adsorbed CO (Q_{CO} in e^-), charge gained by the embedded TM atoms (Q_{TM} in e^-), magnetic moment of the embedded TM atom (M_{TM} in μ_B), height of the embedded TM atom relative to the S plane (h in Å), average distance between the embedded TM atom and its neighboring Mo atoms (d_{TM-Mo} in Å), bond length of the adsorbed CO molecule (d_{C-O} in Å), the nearest distance between the adsorbed molecules and the TM atom (d_{CO-TM} in Å). For M (in μ_B), the values outside and inside the parenthesis are the magnetic moments of the adsorbed CO and the whole cell, respectively.

	E_{ad}	Q_{CO}	Q_{TM}	M	M_{TM}	h	d_{TM-Mo}	d_{C-O}	d_{CO-TM}
Au	0.91	-0.04	-0.07	0.11 (0.97)	0.02	0.62	2.87	1.15	2.02
Pt	1.38	-0.14	-0.12	0.00 (0.00)	0.00	0.52	2.79	1.16	1.92
Pd	0.96	-0.11	0.02	0.00 (0.00)	0.00	0.54	2.82	1.15	1.98
Ni	1.38	-0.22	0.25	0.00 (0.00)	0.00	0.36	2.73	1.16	1.79

Table 2.3: Table taken from [286].

sive breath of the work completed by Karthikeyan *et al* [282], in which they calculate via DFT the effect each transition metal has on the Mo-based TMD properties. Gillen *et al* performed DFT calculations on hexagonal Boron Nitride (hBN)/MoS₂ heterostructures [271]. They found that by introducing foreign atoms, indirect doping of the MoS₂ structure occurs. The introduction of intercalated sodium atoms between the layers increased the interlayer distance significantly. These changes have a large effect on the properties of MoS₂ and are of great interest for device construction.

The lattice parameters and electronic properties of pristine MoS₂ and non-metal dopants (H, B, C, N, O, F, Si, P, Cl, As, Se, Br, Te, I) in MoS₂ were calculated by Lu *et al* and other groups [176, 278, 285]. These values are shown in Table 2.2. They found a preferential substitution for S. For Se doped MoS₂, they found a deviation from pristine MoS₂ in all properties. However, the calculated band gap energy of Se-doped (1.84 eV) is quite similar to pristine MoS₂ (1.86 eV). The DFT calculation in each case was carried out with a 4x4 supercell, meaning a doping percentage of 3.13%. Comparing this minor band gap energy change to the 1.79 eV to 1.70 eV change found experimentally in 5.56% Se doped MoS₂ carried out in Subsection 4.1(Se Ion Implantation) and shown in Table 4.1, this experimental value lies closer to the 1.65 eV band gap calculated by Zhang *et al* for 2.7% and 5.4% Se-doped MoS₂ [176]. Calculated and experimentally found band gap energies are tabulated in Table 4.3 in Subsection 4.1.2(Low Loss EELS Characterisation).

Analysis of surface adatoms on 2D materials is also of great interest for improvement of gas sensor performance [286] and electronic doping [287]. Ma *et al* calculated the effect of introducing both substitutional transition metals (Au, Pt, Pd, Ni) into the chalcogen sublattice and surface CO and NO molecules on mono-

layer MoS₂ [286]. They found that the molecule adsorption caused charge transfer between molecule and lattice, inducing impurity states. The adsorption energies and other properties calculated for CO molecules on the metal doped lattice are shown in Table 2.3. Native adatoms and vacancies have been shown to suppress conductivity [288] as well as lead to n- and p-type doping [289]. Depending on the number of layers in the material, hydrogen adatoms can also dope the electronic structure of MoS₂ [287].

Doping can also occur due to materials or atoms contacting the surface of 2D materials. As 2D materials are defined in large part by their surface, these contact dopants can have a large effect. As mentioned, the results of dopants on suspended materials may be quite different to that on substrate supported materials [248]. This premise holds for the below-discussed surface contact doping.

An excellent and extensive review of TMD doping via contacts and interfaces can be found in Zhao *et al* [290]. The review presents the various methods currently used to manipulate TMDs solely with surface charge or interface doping. Surface charge doping via Au NPs has been demonstrated on MoS₂ flakes by Shi *et al* [291]. The Au NPs, deposited via a solution, were found to be placed selectively along the edges of chemically exfoliated (CE) and chemical vapour deposition (CVD) grown thin TMD layers. The decrease in relative intensity and peak frequency difference between phonon modes indicates n-type doping [292]. For the Au NPs on MoS₂, Raman spectroscopy showed the opposite, implying a p-type doping in the structure.

P-type doping was also found in CE MoS₂ via seeded particles or microwave anchoring of Au (or Ag) NPs by Sreeprasad *et al* [293]. Along with increased thermal conductivity, the authors found an upward shift in energies in Raman spectra to indicate p-type doping. They show also the preferential growth of Au along certain crystallographic directions and especially along the edges, where S atoms are unbonded. Therefore, metal NPs are promising for inherent contact doping of thin TMDs. Au and other transition metals, such as Ti, Pt and Ag, are often used as contacts in sensors and transistors [290, 294, 295]. They have also been used as antenna for optical-coupling [296], single photon emitters [257] for possible opto-electronic devices, and for increasing electrocatalytic activity [297]. Overall, the interesting contacting properties of metals - such as Au - with few-layer TMDs opens various opportunities for nano-scale contacting in electronic devices.

Lin *et al* conducted electrical measurements by creating a 4-5 layers thick MoS₂ FET [138]. They then decorated the sample with Au NPs with a diameter of

~15 nm and measured the response under light verifying an enhancement of the photocurrent. The enhancement is greatest at the plasmonic resonance of Au NPs at ~517 nm (~2.4 eV). The localized surface plasmon in the Au NPs enhances the local optical field near the NPs and thus the light absorption of the MoS₂ underneath. In a similar study, Li *et al* reported an absorption PL signal red-shift in Au NPs on monolayer MoS₂ as the NP concentration is decreased [298]. Additionally, by varying the Au NP concentration, they have control over the MoS₂ exciton binding energy.

Other than direct doping to tailor the electronic structure, exciting optoelectronic developments with various rods, dimers, and cavities are discussed by Baranov *et al* [299]. Plasmonic NPs have been shown to modify the photoluminescence (PL) spectra of thin TMDs in several studies [257, 300–302]. MoO₃ nano-clusters have been shown to counteract the damaging effect of Na to the capacity of MoSe₂ thin layers - and therefore life-span - of sodium ion capacitors (SICs) [303]. However, the cavities and particles used in many of these experiments are large (50-500 nm) compared to the single to few-layer nature of TMDs.

After having described methods to change the properties of 2D TMD monolayers and multi-layers, we will now discuss the advantages and disadvantages of introducing dopants into 2D materials via low energy ion implantation, the results of which will be presented and discussed in Chapter 4(Atomic Characterisation of Ion Implanted 2D TMDs).

2.2.3 Ion Implantation of 2D Materials

In today's semiconductor industry, electronic doping of materials is commonly achieved by introducing foreign atoms (dopants). Dopants are added in various ways. For Si, a staple of the semiconductor industry, elements in group 13, such as B and Ga, and elements in group 15, such as P and As, are often used as dopants. Similarly for carbon, its neighbouring elements in group 13 and 15 can be used for doping. For decades, ion implantation has been a standard method in the semiconductor industry for introducing foreign atoms into device materials [304] at nano-metre length scales [305]. Mass-separated ion beam deposition (MSIBD) systems, such as the ADONIS in Göttingen [306], have been used to successfully implant monolayer 2D materials [10, 19]. In this section, we discuss the implantation of foreign atoms into monolayer materials for the purpose of electronic doping, vacancy creation, or introducing novel material properties.

Ion implantation of graphene has been studied extensively compared to other 2D materials, as shown in Table 2.4 and discussed in [253]. Manipulation of monolayers requires ion implantation at energies considerably lower than used in the standard semiconductor industry, as modelled for graphene by Ahlgren *et al* [307]. They found the most efficient doping energy to be 40-50 eV for B and N, respectively, while most vacancies were found to occur at higher energies (>100 eV), as shown in Figure 2.14. Figure 2.14 shows the relative probability of introducing substitutional B or N atoms as compared to other defects. As we will discuss later, defects can often be desired, either as dopants or quantum wells, or even as sites for further ion implantation or chemical doping. The authors also calculated the probability of creating surface adatoms. They found that at lower energies (<30 eV), adatoms form readily on the surface of graphene monolayers.

A list of implantations into 2D and nanotube materials can be seen in Table Table 2.4. As mentioned above for graphene, theoretical and computer simulations [307, 309, 312] guided initial experimental work surrounding successful ion implantation into graphene [19, 56, 57, 308, 310, 311, 313, 314]. Optimum energies

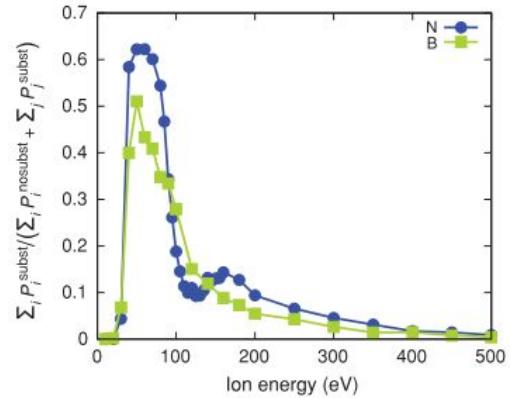


Figure 2.14: The relative probability of introducing substitutional B or N atoms as compared to other defects studied by Ahlgren *et al*. Figure taken from [307].

2.2. Two-Dimensional Materials and Transition Metal Dichalcogenides

Table 2.4: Summary of ion implantation into 2D and low-D materials.

Material	Dopant (element)	Energy (eV)	Fluence (1/cm ²)	Experiment/ Simulation	Ref
Graphene	B, N	25, 100, 200	(2 +- 1) x 10 ¹⁴	Exp	[19, 20]
Graphene	B, N	25	(2 +- 1) x 10 ¹⁴	Sim	[19]
Graphene	B, N	25	3 X 10 ¹⁴	Exp	[308]
Graphene	B, N	10-4000	7.5 x10 ¹⁵	Sim	[307]
Graphene	Ge	15-25	1 x 10 ¹⁴	Exp	[57]
Graphene	Ge	20-42	-	Sim	[57]
Graphene	Ge, Pt	10-3000	-	Sim	[67]
Graphene	He, Ar, Xe	30-1x10 ⁶	-	Sim	[309]
Graphene	B, N, F	20, 35	1x10 ¹⁴ -6x10 ¹⁵	Exp	[310]
Graphene	N	0-3x10 ⁴	2x10 ¹² -1x10 ¹⁴	Exp	[311]
Graphene	He, Ne, Ar, Kr, Xe	0-10 ⁶	-	Sim	[312]
Graphene	P	30	(4+-1)x10 ¹⁴	Exp	[56]
Graphene	Pt, Co, In	100-400	-	Exp, Sim	[313]
Graphene/Ir	Ar	100-1000	2x10 ¹³ - 1x10 ¹⁴	Exp	[314]
Graphene/Pt	Ar	140	-	Exp	[315]
Graphene/SiC	Ar	140	-	Exp	[316]
Graphene	Xe	131	-	Exp, Sim	[317]
Graphene/Ir(111)	Xe	0.5-3 x10 ³	1.57x10 ¹³	Exp, Sim	[318]
MoS ₂	Se	10	3x10 ¹⁵	Exp	[10]
MoS ₂ /hBN/ SiO ₂ /Si	He	3x10 ⁴	1x10 ¹²	Exp, Sim	[251]
WS ₂	Se	<10	-	Exp, Sim	[96]
SWCNT	B, N	0-300	-	Sim	[319, 320]
SWCNT	B, N	0-1x10 ⁴	-	Sim	[321]
SW, MWCNT	Ar	0.1-2x10 ⁴	-	Sim	[322]
Review	Review	-	-	Exp, Sim	[250]
Review	Review	-	-	Exp, Sim	[253]

for substitutional implantation, and avoidance of defect production, depend on the element and charge of the source and target [307, 309], as well as the ion's angle of incidence [67]. The work into graphene implantation was guided by initial single and multi-wall carbon nanotube (SWCNT, MWCNT) simulations [319–322],

In Ge simulations, 20–22 eV was found to be the ideal implantation energy for graphene [57]. Experimentally, it is difficult to estimate the optimum implantation energy due to sample preparation issues [19, 57] and electron-beam-sample interaction [19]. Initial polymer contamination and subsequent ambient hydrocarbon contamination, originating from either storage, the atmosphere and the ion implantation process can cover the surface of the 2D material, even after extensive sample cleaning. The ion-implantation process can introduce contaminants, which is sometimes considered as the major or even only source of contamination [253], which likely is not the case. Indeed, simulations of ion implantation regarding contamination-covered 2Ds could be highly beneficial in future.

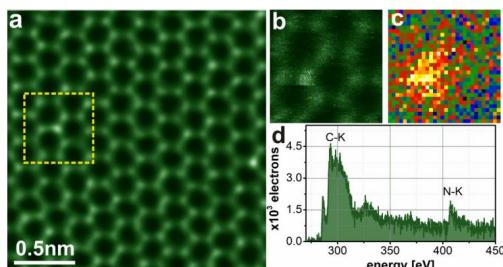


Figure 2.15: (a) Atomic resolution false colour ADF STEM image of N implanted monolayer graphene. The region in the yellow rectangle is magnified in (b). (c) EELS spectrum image of the region in (b). (d) EELS signal near the N implant. Figure taken from pre-print [19].

graphene, and the defect complexity can be controlled by the irradiation energy of Ar ions [314]. In separate studies, Ar ions with a source energy of 140 eV were used to understand point defects and divacancies in graphene on Pt(111) and SiC(0001), respectively [315, 316].

To study 2D materials free from substrate interference, suspended samples must be implanted. Experimentally, the first low energy ion implantation of monolayer graphene was reported by Bangert *et al* [19]. B and N were implanted at 25, 100 and 200 eV. Both B and N implanted graphene samples were imaged at atomic resolution via high-angle annular dark field (HAADF) scanning transmission electron microscopy (STEM). Additionally, the chemical signatures of B, C and N were

Generally, ion implantation into 2D materials can be split into two sections; 2Ds lying on a substrate and 2Ds suspended over vacuum [248]. Although this section will focus almost entirely on 2D materials suspended over vacuum, as this is the format we will be studying, it is certainly important to understand the effect a substrate can have on the material of interest, and how it effects the material's properties. The study of graphene on Ir(111) by Ahlgren *et al* showed more complex defect structures than that of suspended

identified by core-loss electron energy loss spectroscopy (EELS). Figure 2.15(a) shows a single substitutional N atom in monolayer graphene in the dashed yellow rectangle. This region is magnified in Figure 2.15(b). An EELS map of the N-signal intensity of this region is presented in Figure 2.15(c), with Figure 2.15(d) showing the EELS spectrum image signal for C and N around the substitutional N atom.

Not only was atomic resolution microscopy and spectroscopy used to identify single atom dopants introduced via low energy ion implantation, it was also used in a later follow-up study to show successful doping of graphene [20]. The authors compared the low-loss EELS signal with density functional theory (DFT) calculations to show that the low-energy ion implantation opened graphene's band gap, and therefore verified the p-type and n-type doping near substitutional B and N dopants, respectively. This interesting result opens the door to further manipulation of monolayer materials via single atom doping. If the electronic band gap of a zero band gap semiconductor such as graphene can be changed and verified on the atomic scale, then a plethora of opportunities will be opened by further low energy ion implantation experiments.

In further suspended graphene experiments, similarly low implantation energies were used [308, 310]. Xu *et al* implanted graphene with B, N and F at 20-35 eV [310]. It is important to note the successful implantation and unambiguous identification of B and N in graphene depended largely on finding atomically clean patches on the graphene surface, which is also the case for non-implanted samples. Additionally, consistent and reliable spatial resolution of scanning TEM at low energies (\sim 60 kV) to enable experimental proof was reached around this time [5, 6].

For both ion and electron beam interactions, a key parameter is the displacement threshold energy, which is the minimum energy required to eject a native atom from the lattice without immediate recombination into the vacancy [described in subsection 2.1.2(Electron-Matter Interactions)]. The implantation energy required must be sufficient to eject the native lattice atom without creating irreversible damage to the lattice. Ahlgren *et al* calculated the displacement of a single carbon atom in a graphene monolayer to be 22.59 [67], agreeing well with the 22.2 eV - 23 eV range of values calculated for carbon displacement in CNTs [165, 322, 323]. Values are similar for hexagonal Boron Nitride (hBN) monolayers, i.e., 19.36 and 23.06 eV for B and N, respectively. For Mo-based TMDs, the displacement threshold energies for the bottom layers of S, Se, and Te are 6.9, 6.4, and 5.9 eV, respectively [166]. For MoS₂, this relates to an electron beam energy of \sim 90 keV. It is worth noting that the vacancy formation energy and the threshold

displacement energy are comparable for MoS₂ and WS₂ [166], though not for Nb, Ta, Ti, and Pt-based TMDs. Liu *et al* calculated the vacancy formation energy for S to be 6.1 eV, which translates to an electron-beam energy of \sim 80 keV [324]. Additionally, near the edge of these materials, the displacement threshold energy is reduced [43, 166]. Therefore, a primary electron-beam energy of 60 keV is required for analysis of MoS₂ and implanted MoS₂ structures [43, 161, 183].

Some researchers have utilised a secondary implantation step, carried out at slightly lower energies, which is used to fill any vacancies created by an initial pre-implantation treatment [307, 311]. Wang *et al* instead utilised a laser (with equivalent energy of \sim 100 eV) to create these initial vacancies [313]. If these methods could be performed in constant vacuum, they would certainly increase the reproducibility of the technique for 2D materials. Klein *et al* used a helium ion beam to create optically active defects states in MoS₂ [251]. It is worth noting that the work was carried out on a substrate, with the MoS₂ monolayer covered by the insulator hBN.

Ion implantation of elements into TMDs has not been explored as extensively as in graphene and nanotube (NT) structures. As discussed in detail in Subsection 4.1(Se Ion Implantation), Bangert *et al* successfully implanted Se into monolayer MoS₂, creating a TMD alloy [19]. However, TMDs have been doped using other methods, which are discussed in more detail in subsection 2.2.2(Manipulating the Properties of 2D TMDs). In general, all of these methods aim to control the properties of 2D materials with the ion or electron beam for device fabrication, moving toward higher spatial resolution. Ahlgren *et al* estimate that with precise control over ion energies and by utilising focusing ion-beams, novel materials could be designed with a resolution better than 5 nm [307].

Plasma immersion ion implantation has been utilised to achieve p-type doped few-layer MoS₂ with P [325]. The method is described as “controllable and area selective”, and the resultant doped material was characterised via secondary ion mass spectrometry (SIMS), atomic force microscopy (AFM), Raman Spectroscopy, x-ray photoelectron spectroscopy (XPS), and electrical transistor measurements. The electrical characteristics were measured as a function of implant energy. Additionally, stopping range of ions in matter (SRIM) and DFT calculations were used to investigate the ion implantation procedure and density of states of p-type doped bulk MoS₂, respectively. The authors fabricated back-gated transistors via electron beam lithography on 300 nm Si/SiO₂ substrates. The MoS₂ crystals (bulk) were exfoliated and subsequently patterned. SIMS confirmed the presence of phosphorus in the top few layers of MoS₂. The P source was a PH₃/He plasma, to which the transistor was exposed after e-beam lithography. However, this im-

plantation method relies on a Si substrate and results in the loss of 3–4 of the top MoS₂ layers, as confirmed by AFM. This is not desirable for monolayer materials. Additionally, Raman spectroscopy showed the creation of defects in the top layers of the device. What is noteworthy is the presence of the [002] peak in XRD readings, which the authors took as verification that crystallinity remains in the MoS₂ crystal even up to the 2 keV energies used. XPS readings display a “core level shift of 0.67 eV towards lower binding energy for both molybdenum (Mo) and sulphur (S)”. This agrees well with the E_F shift towards the valence band, indicating the afore-mentioned p-type doping.

Changing the properties and understanding the physics of ion implantation into 2D materials is a growing field [10, 96, 253, 326]. Schwestka *et al* have developed an ion-beam spectrometer specifically for the analysis of 2D materials [317]. They verified its use through experiments on monolayer graphene by implanting Xe³⁰⁺ ions. Recently, Lin *et al* created novel Janus monolayers by utilising ion implantation energies of <10 eV [96]. Janus monolayers are discussed in subsection 2.2.2(Manipulating the Properties of 2D TMDs) above. Low energy ion implantation is used for the manipulation of monolayer MoS₂ in this work in Chapter 4(Atomic Characterisation of Ion Implanted 2D TMDs), and the resulting images are analysed in detail in Subsection 5.2(Atom Dynamics in Implanted TMDs).

2.3 Ferroelectric Materials and Domain Walls

Ferroic domain walls (DWs) are a relatively new field within 2D materials research. DWs can possess different properties such as conductivity, magnetism, and chemistry from the domains they separate. Additionally, they can be created, destroyed and moved by an external applied stimulus. As a 2D entity they require the spatial resolution of aberration corrected STEM as discussed below in subsection 2.3.1. A proper ferroelectric is one in which the primary order parameter, ferroelectricity in this case, drives the spontaneous creation of domains and DWs. Improper ferroelectrics are described by two order parameters. The first is the primary order parameter, which drives the non-electric order such as elastic or magnetic distortion. The second order parameter is the electric polarisation, which is induced by the primary order parameter and is a secondary effect [327]. For example, improper ferroelectricity can arise from geometric distortion of the lattice, such as in Boracites [14] or due to magnetism in Manganites, such as YMnO₃ [328].

In the case of Boracites, formation of a 180° charged wall is energetically favourable, as the strain at this type of DW is energetically favourable. This is not the case for a pure ferroelectric as in LiNbO₃ or PbTiO₃, as the DW type with the highest energy state would be electrically charged. Therefore, one must map out the strain (unit cell displacement) as well as the dipole polarisation to fully understand polarisation in Boracite domains. In the following short subsection, we discuss how atomic resolution electron microscopy has been used to understand polarities in ferroelectrics. For more information on domain formation in ferroic materials, see subsection A.1.1 in the appendices.

2.3.1 Ferroelectric Domains and Domain Walls at Atomic Resolution

Since the advent of microscopy techniques with reliable atomic resolution [329–331], polarisation mapping of individual atomic columns has become possible. Not only can atomic resolution ADF STEM easily identify chemical information through Z-contrast [332], it has allowed researchers to probe the sub-atomic shifts of dipoles, domains, and phases in ferroelectric and multiferroic materials [333–335].

Jia *et al* were the first to image a DW in a ferroelectric material (PbZrTiO₃) via TEM at 300 kV [336]. Figure 2.16(a) shows a DW separating two domains. The mapping of the PbO columns (yellow), Zr/Ti columns (red) and O columns (blue) inset on either side of the DW clearly show opposing polarisation. The

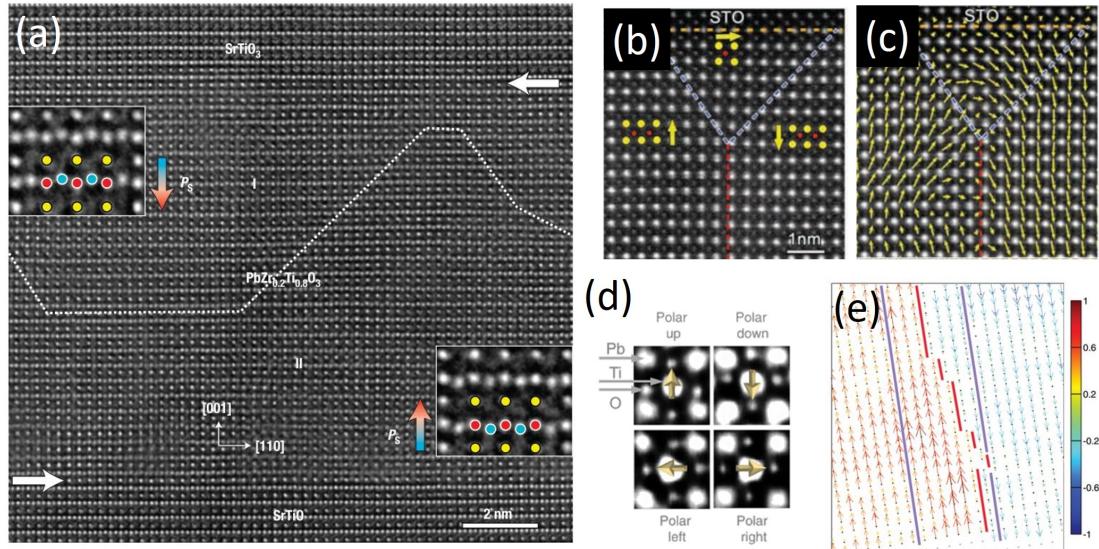


Figure 2.16: (a) TEM image of Polarisation of Ti/Zr (red) and O (blue) atomic columns from PbO (yellow) across a DW in PbZrTiO₃. Taken from [336]. ADF STEM image showing (b) the overall polarisation and (c) the individual Ti atomic column polarisation of the upper region of a vortex-like DW structure in PbTiO₃/SrTiO₃ multilayers (taken from [333]). (d) Method of polarisation mapping in PbTiO₃-like structures (taken from [335]). (e) Polarisation of Nb atomic columns across a DW in LiNbO₃ (taken from [337]).

Zr/Ti and O atomic column shifts and polarisation were mapped by the vertical distance deviation from the central point of the cubic perovskite structure with Matlab.

The first atomic resolution polarisation mapping of a vortex structure was reported in PbTiO₃/SrTiO₃ multilayers using ADF STEM [333]. Figure 2.16(b) and Figure 2.16(c) show the overall polarisation and individual atom polarisation in the upper region of a triple DW junction in PbTiO₃ (the vortex in this case is two three-fold DW junctions). The mapping of the polarisation was completed by using the Pb columns as a rigid structure, and measuring the movement of the Ti columns from their ideal position. This method is shown graphically in Figure 2.16(d), which is taken from Du *et al*, wherein the authors measured and simulated vortices and other structures in decreasingly thick layers of PbTiO₃/SrTiO₃ [335]. Vortex-antivortex structures have also been mapped in the same multilayer structure by Yadav *et al* [338].

Mapping the polarisation of atomic columns without reference to some fixed, stable lattice is difficult compared to a situation with a stable sublattice, as above in the above Pb structures. Gonnissen *et al* measured the deviation of atomic columns of Nb across an LiNbO₃ DW [337]. LiNbO₃ is an exciting ferroelectric material due to its electrical tunability [15, 339]. To measure the picometre deviations, the authors fitted the atomic column rows across the DW via linear regression. On

either side of the DW, 15 Nb atoms were used to fit two separate lines. The slopes of each of these two lines were restricted to be the same to keep the slope consistent for each atomic row. The y-intercept was not restricted. Presumably, only 15 Nb atoms on either side of the DW were used to show the largest atomic column shift, and therefore polarisation, of each Nb atomic column across the DW. The average width of the DW was calculated to be 174 ± 33 pm via a parametric model [337, 340]. Figure 2.16(e) shows the resulting polarisation mapping of the Nb atomic columns near the DW.

For various Manganite ferroic materials, polarisation near a DW has been quantified via ADF STEM in recent years [334, 341–343]. For example, topological defects in ErZrMnO_3 have been revealed via pattern changes across DWs of ADF STEM images [334], while new phase transitions can be resolved with the help of DFT [343]. These, and the many other recent investigations of (multi)ferroic materials, show clearly that ADF STEM is an ideal tool for analysis of such structures [344].

Recently, a quick method of fast Fourier transform (FFT) mapping of domains and DWs has been applied by Moore *et al* [345, 346]. This method involves choosing the relevant diffraction peaks in an image's FFT. By repeating this for different diffraction peaks related to different domains, one can map, even at low magnification, domains and DWs in ferroelectric materials.

Certain links below do not display directly due to **Redactions** in certain sections.

In Chapter 6(Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls), 2D DWs in several materials are analysed via atomic resolution ADF STEM: PbTiO_3 DW junctions (subsection 6.2.1, **Redacted**).

For semi-automated analysis of polarisation in atomic resolution images of ferroelectric (and other) materials, we have developed an open-source module `polarisation.py` as part of the TEMUL toolkit [13]. How to use each function is described in detail in Subsection 6.1(Tools for Visualisation of Subatomic Shifts), and implemented for the various above-described ferroelectric materials in Subsection 6.2(2D Domain walls). The source code is available on GitHub [13] and in Subsection A.6.

Chapter 3

Methodology

In this section, 2D material sample preparation, low energy ion implantation, electron microscopy, simulation and spectroscopy, and signal and data analysis are discussed at length. Justification for the chosen methods are outlined, and reasoning based on the above literature review is given.

For clarification, the author of this thesis built and operated the sample preparation setup used for exfoliation and stamping of 2D materials discussed in this chapter in Subsection 3.1 below. For sample preparation of ferroelectric materials examined in Subsection 6.2(2D Domain Walls), see Chapter (Publications & Collaborations). Ion implantation was carried out by collaborators, as outlined in Subsection 3.2(Low Energy Ion Implantation). The author aligned and operated the Titan Themis TEM in UL to its highest specifications, including the alignment of the image and probe aberration correctors. This microscope was used for microscopy and spectroscopy of pristine, as well as Au and Se implanted TMDs. The images and data of Se implanted MoS₂ shown in this thesis were taken by Prof. Quentin Ramasse using the SuperSTEM 3 microscope in Daresbury, U.K. [see Chapter (Publications & Collaborations)]. The areas of interest were directed by the author in most cases, and optimum imaging conditions were discussed beforehand. The author simulated the images shown in the thesis, with the exception of the Au NP simulations found in subsection 4.2.2(Comparison of Au Implants with Image Simulation). The variation of atomic resolution image simulation parameters were guided by previous simulations carried out by E. Courtney. Finally, at the time of initial thesis submission, the author wrote all of the code available for use in the TEMUL toolkit, and any adaptations of other works are stated and cited.

3.1 2D Material Sample Preparation

There are numerous, well-established methods for creating a monolayer 2D material sample. These methods can be broken into two sections, mechanical and chemical. Micro-mechanical exfoliation of single crystals using scotch tape [349] is the most popular mechanical method, though exotic mechanical methods such as atomic force microscopy (AFM) peeling [350] are also available. In the following subsections, various micro-mechanical methods for the preparation of few and monolayer 2D materials are discussed. First, an overview of the mechanical exfoliation technique is given. Following this, polymer and deterministic (stamping) preparation techniques are described. Finally, the sample preparation process carried out during this project, which combines the above-mentioned methods and techniques, is outlined.

Mechanical Exfoliation

Micro-mechanical exfoliation of single crystals produces high quality and clean pristine samples [191], and was therefore an obvious choice for this project, wherein high-quality, small-area samples are valued. However, mechanical exfoliation is not suitable for industrial-scale operations. Any combination of 2D material synthesis with high throughput ion implantation would require reproducible chemical synthesis. Chemical synthesis methods include chemical vapour deposition (CVD) [193, 351–354], shear-exfoliation of graphite in liquid (SEG) [355–358], epitaxial growth on an insulator - such as silicon carbide [359], intercalation [360], as well as many others [361–364]. The quality of some CVD-grown 2D materials, such as graphene, has in very recent years started to match those of mechanical exfoliation of single crystals, and we would recommend attempting experiments on both for future endeavours.

Preparation of single layer and few-layer TMDs is non-trivial, and is especially difficult for suspended transfer onto carbon quantifoil transmission electron microscope (TEM) grids. The process is as follows:

1. Exfoliation of a high-quality bulk single crystal (e.g., graphite, MoS₂, hBN) using clean adhesive tape [349]. The basic process is also shown in Figure 3.1.
2. Pressing the adhesive tape together several times to further exfoliate the material to achieve few-layers.
3. If the substrate is not as delicate as a standard TEM grid, e.g., Si/SiO₂, then adhesive tape can be pressed at this stage onto the substrate. If the

3.1. 2D Material Sample Preparation

adhesion between the flake and the substrate is greater than the adhesion between the Van der Waals bonding in the 2D material flakes, the flakes will be transferred to the substrate.

4. If the substrate is delicate, as in the case of thin TEM grids, the 2D material flakes on the adhesive tape must undergo some intermediate transfer to a polymer, as outlined below in Subsection 3.1(Polymer Transfer Methods).
5. Optical microscopy is used to identify a suitably thin flake.
6. Final transfer to the TEM grid can be completed by simply placing the polymer on the TEM grid by ‘fishing’ the polymer from a liquid, as in the case of CVD-grown flakes, or by deterministic transfer, as carried out in this project [see Subsection 3.1(Deterministic Transfer Methods)].
7. Optical microscopy is then used to confirm successful flake exfoliation and transfer.
8. The sample must then be cleaned, as described in Subsection 3.1.1(Cleaning of Prepared Samples).

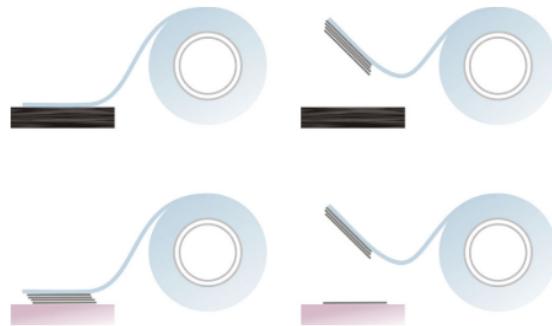


Figure 3.1: The micro-mechanical exfoliation technique for producing graphene and other 2D materials. Top row: Adhesive tape is used to exfoliate the several layers of graphite placed on a photoresist. Bottom left: The tape with graphitic flakes is then pressed against the glass substrate. Bottom right: Flakes remain on the substrate, producing high-quality graphene flakes ideal for electronic measurements [349].

Polymer Transfer Methods

Accurately and precisely positioning 2D material flakes onto these TEM grids without destroying both flake and grid requires flexible, gentle polymers. There are currently three dominant chemistry-dependant methods for 2D material transfer between various substrates, each shown in Figure 3.2. The current methods are the

3.1. 2D Material Sample Preparation

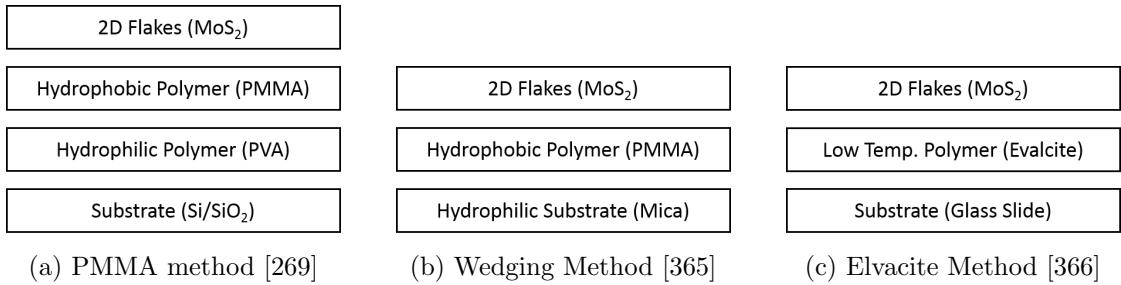


Figure 3.2: Three sample preparation methods used for the creation of 2D nanostructures on various substrates. Combined with the deterministic stamping method in Figure 3.3, these methods form the basis on the sample preparation used during this work.

PMMA method [269], the wedging method [365], and the elvacite method [366]. Each of these methods require the use of spin-coated polymers on a Si/SiO_2 (or similar) substrate.

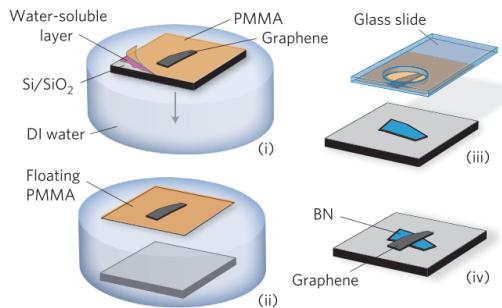
A cross-section of the PMMA sample transfer method can be seen in Figure 3.2(a). The 2D flake is stacked on two spin-coated polymer layers: a hydrophobic polymer (poly methyl methacrylate [PMMA]) and a hydrophobic polymer (polyvinyl alcohol [PVA]). These are spin coated on a Si/SiO_2 substrate. This method is based on the use of the hydrophilic polymer as a sacrificial layer. After the 2D flake is placed on the top polymer, using some mechanical exfoliation for example, the entire system is placed in a water bath. The hydrophilic polymer layer dissolves, resulting in a floating 2D flake/PMMA stack. This stack is retrieved with a glass or metal slide and then transferred to the desired substrate [269]. Figure 3.3(a) shows graphically the steps in this process, where the water-soluble layer is PVA.

The wedging method is similar to the PMMA method, as seen in Figure 3.2(b). The difference, however, is that the substrate is hydrophilic. A hydrophobic polymer layer is spin-coated onto this hydrophilic substrate. Water is again utilised as the active component and leaves the 2D flake/PMMA stack floating above the substrate in a water bath [365].

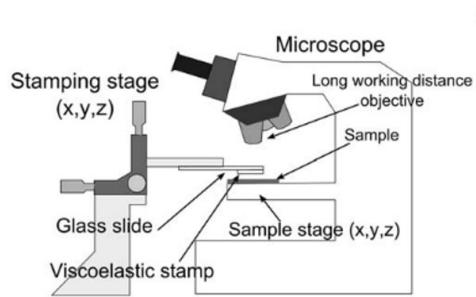
In the elvacite method in Figure 3.2(c), a glass slide is spin-coated with a low glass transition temperature polymer, elvacite (incorrectly written as evalcite in several publications). Flakes are transferred via the scotch tape method onto this polymer. The glass slide is then mounted to a micro-manipulator and the receiving substrate is heated to 75-100 °C. The polymer will melt upon contact and will adhere to the receiving substrate [366]. The polymer must then be dissolved in solvent as with the above-described wedging and PMMA methods.

Deterministic Transfer Methods

Deterministic stamping of 2D material flakes, as demonstrated by Castellanos-Gomez *et al*, can, in some cases, remove the spin-coating necessary for the above mentioned chemistry reliant methods [367], though it still uses a polymer. It is a simple, non-chemistry reliant and efficient method, with a yield of near 100% for transfer onto atomically flat areas and 30-40% for large area heterostructures. More importantly, for high-resolution studies, they also show successful transfer onto prefabricated electrodes (with tunnels), as well as SiN and holey-carbon TEM grids (though the latter was not reproducible).



(a) PVA-PMMA transfer method [269]



(b) Stamping [367]

Figure 3.3: (a)(i) Dissolving of a hydrophilic polymer in a water bath. (ii) Floating of a hydrophobic polymer/2D flake stack, which is ‘fished’ out with a glass slide. (iii) Glass slide/hydrophobic polymer/2D flake stack inverted and positioned over a rigid substrate/2D flake stack. The glass slide stack is then positioned and stamped on the rigid substrate, creating a heterostructure. (b) Deterministic stamping setup under an optical microscope.

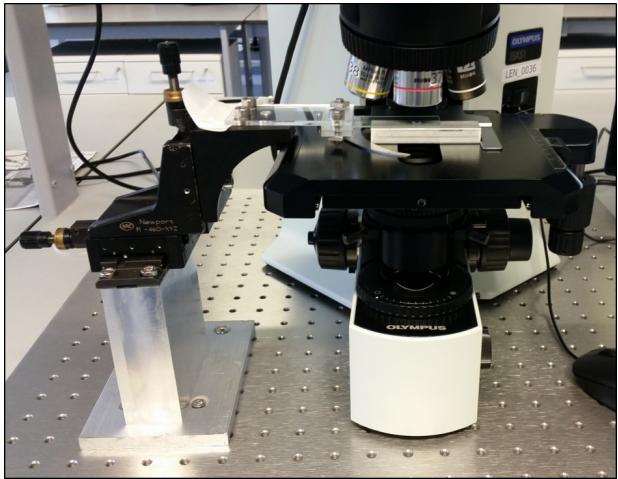
This transfer method uses a glass slide/PDMS polymer/2D flake heterostructure mounted onto a cantilever (glass or metal slide), similar to Figure 3.3(a)(iii). Figure 3.3(b) shows how the cantilever is mounted to an optical microscope, which is then lowered toward the acceptor substrate. As described below in subsection 3.1.1(Experimental 2D Stamping Setup), this experimental set-up was built during this work. This method is clean and reproducible. The authors successfully transferred few-layer graphene onto hexagonal boron nitride (hBN), which was placed onto a Si/SiO₂ substrate. The supplementary video of the transfer shows how tangible the viscoelastic stamping is; the liquid polymer is easy to track as it is lowered to and raised from the acceptor substrate (see video in [367, 368]).

As mentioned above, during this work we could not reproduce the polydimethylsiloxane (PDMS) polymer transfer onto carbon-quantifoil TEM grids. A less adhesive polymer was instead utilised (PMMA), and a combination of the stamping

3.1. 2D Material Sample Preparation



(a) Solidworks Render



(b) Laboratory setup

Figure 3.4: (a) A render of the Solidworks design for the deterministic transfer of 2D flakes onto TEM grids and other substrates. (b) Picture of the laboratory set-up with the transfer stage fixed beside the microscope.

and polymer methods was used throughout this project. Optical identification of 2D flake layer numbers is a vital step in the process of reliable sample preparation [369–371], as described in the next section.

3.1.1 Experimental 2D Stamping Setup

For TEM experiments, the 2D material specimen must be placed on a thin substrate that fits modern TEM sample holders. Standard TEM sample holders require a 3 mm diameter rigid substrate grid. For monolayer samples, areas with no substrate are required for atomic resolution imaging. In other words, the sample being viewed must be suspended over vacuum [372]. It is also possible to use graphene as a support for thin 2D material imaging and to remove its contrast with a Fourier filter [183, 184]. This method also has the benefit of protecting the sample against electron-beam induced radiation damage, as described in Subsection 2.1.2(Minimising Radiation Damage), though it is not a simple process (triple monolayer heterostructure).

In initial experiments, rigid Cu grids with a patterned carbon Quantifoil [373] membrane were used [373]. These Cu grids contaminated quickly, however, so Au Quantifoil grids were used instead. These were found to be cleaner over a longer period. However, it should be noted that as more samples were prepared, gradual improvements were made to the preparation techniques, which could be the reason for reduced contamination on the Au grids.

As discussed in Subsection 3.1, deterministic transfer of 2D materials onto

3.1. 2D Material Sample Preparation

various substrates has gained traction in the preparation of such samples. The accuracy of the easy-to-follow method depends largely on the quality of the 2D flake, the substrate cleanliness, and trial and error. Each transfer set-up is unique, but with practice can produce clean and reliable samples for study in the electron microscope. The design used throughout this project is shown in Fig 3.4. Fig 3.4(a) displays a render of the assembly created in the computer aided design program Solidworks. This design was built and implemented, and is shown in 3.4(b). The set-up is positioned under an Olympus BX51 optical microscope and is connected to a computer for digital display of images. This has the added benefits of video recording and removing the user from interacting with the set-up unnecessarily. Similar transfer apparatuses can now be bought complete from suppliers such as HQ Graphene [374].

The design shown in Fig 3.4 has some specific advantages and disadvantages. In terms of advantages, the stamping XYZ stage and microscope xyz stage are accurate to several microns, allowing precise movements in x, y and z directions. Any type of cantilever can be attached to the stamping stage meaning that the set-up is quite flexible and could be utilised for many different purposes, not just 2D flake transfer. The large microscope stage allows for many types of samples and substrates to be manipulated and used in the set-up. Having two XYZ stages increases the flexibility of the transfer method, as sometimes the microscope stage was used in the final stamping step, rather than the external XYZ stage. The stamping process is shown for an example substrate in Figure A.1 and Figure A.2 in Subsection A.2.

The main disadvantages of this set-up are with the cantilever-type tip (a glass or metal slide), the mechanical nature of the stamping XYZ stage, and with the microscope itself. First, the Perspex clamps are not machined to the same accuracy as the metal parts, such as the base and metal slide. As the slide is lowered it may not be perfectly level with the substrate. This limits the accuracy of the entire system. A simple fix would be to machine these parts from high quality aluminium. The workaround used was to adjust the slide's position and tilt with paper. The stamping XYZ stage must be handled to be used, therefore causing vibrations during stamping. Having an electronic, piezo-driven stage would remove these issues. The microscope lenses are fixed in this set-up. This increases the amount of movement done by the sample or substrate during transfer. Ideally, it would be the camera that would change height to line up with the stamping procedure, rather than vice versa. The microscope stage can move precisely, and has a fixed size. Having an external microscope/camera set-up and separating the microscope from the microscope stage entirely would give much more flexibility than what is

3.1. 2D Material Sample Preparation

allowed currently.

The sample transfer technique for creating thin samples of 2D materials was as follows:

1. Exfoliate single crystal with blue elastomer tape or PDMS. Qualitatively, yield of monolayers for elastomer tape is higher than PDMS. However, due to EU regulations, we were unable to purchase the elastomer in the latter half of the study.
2. Spin-coat a Si/SiO₂ wafer with PVA and PMMA. The spin-coating holder designed is shown in Figure A.2(a) in Subsection A.2.
3. The exfoliated crystal flakes were transferred to the Si/SiO₂ wafer.
4. An Olympus BX51 microscope with long focal length objective lenses (5x, 10x, 20x, 50x, 100x) was used to locate and identify monolayers and thin layers of 2D materials on the Si/SiO₂/PVA/PMMA substrate. Images of the location of suitable flakes were saved.
5. Using a forceps or knife, trace a circle around a suitable flake. The circle should be ~6-10 mm in diameter, at least larger than the metal o-ring.
6. Place deionised water droplets where the polymer has been cut. The water will dissolve the hydrophilic PVA, while floating the PMMA/flake stack.
7. Gently lower the Si/SiO₂/PVA/PMMA/flake stack into a water bath as close to horizontal as possible. The PMMA/flake stack will float onto the surface of the water. Remove the Si/SiO₂/PVA/PMMA substrate and leave to dry.
8. Dip a metal o-ring, with a inner diameter larger than a 3 mm TEM grid into a bath of PMMA. This helps to adhere the PMMA/flake to the o-ring.
9. Place the o-ring on a piece of rigid pre-holed double-sided thermal tape. This is shown in Figure A.2(b) in Subsection A.2.
10. With the o-ring, ‘fish’ the PMMA/flake stack from the water bath, taking care to align the centre of the o-ring with the centre of the floating PMMA layer. A successful example is shown in Figure A.2(b) in Subsection A.2 (the PMMA on the o-ring is difficult to make out).
11. Place the o-ring/tape/PMMA/flake stack on a glass slide, or a metal slide with a hole. This is shown in Figure A.2(c) in Subsection A.2.

3.1. 2D Material Sample Preparation

12. Adhere a TEM grid to a heavy substrate, such as a metal block, with double sided tape or otherwise, being careful to only touch the edges of the grid to the tape. This is shown in Figure A.1(c) in Subsection A.2 for a piece of Si wafer.
13. Place the block/TEM grid on the microscope stage.
Place the o-ring/tape/PMMA/flake on the stamping XYZ stage.
14. After bringing the TEM grid into focus with the microscope stage, and aligning the flake with a suitable hole in the TEM grid, lower the stamping XYZ stage until it comes in contact with the TEM grid. In some cases it helps to flood the TEM grid with water, making the PMMA layer lie wholly on the block/TEM grid and thus remain there. This can then be dissolved with basic solvents.
15. Raise the stamping XYZ stage slowly. If the transfer has been successful, the 2D flake will be positioned over a hole on the TEM grid.
16. Clean the sample for use in the TEM as described in Subsection 3.1.1(Cleaning of Prepared Samples) below.

The PDMS stamping method (see Figure A.1 in Subsection A.2) was used primarily in this work for transfer onto Si and SiN wafers, membranes, and in-situ microscope chips. The PMMA stamping method is used as an alternative to the above PDMS stamping method (see Figure A.2(b,c) in Subsection A.2). It is a more difficult procedure due to the weaker, sometimes thinner polymer, and because the Cu/Au TEM grid substrate is less rigid than the SiO₂ or SiN acceptor substrates used in PDMS transfer, it is therefore easier to destroy. The PDMS stamping method could not be used for standard TEM grids as it destroyed the carbon membrane.

Cleaning of Prepared Samples

In general, all 2D samples were cleaned initially with a 15 minute acetone bath, 15 IPA bath and 15 minute DI water bath. Prior to imaging, hydrocarbon contamination was minimised by baking the sample at 210 °C in a vacuum oven at 1 mbar (Titan Limerick) or at 150°C at a similar pressure (SuperSTEM Daresbury). For ideal imaging conditions, samples were left in the TEM overnight to stabilise the usual drift issues. This also seemed to stabilise the surface hydrocarbon contamination, stopping it from gathering near the beam and polymerising, as described in Subsection 2.1.2(Contamination on 2D Materials). The very clean column and improved vacuum of the SuperSTEM microscopes had a noticeable effect on the cleanliness of the samples during imaging.

3.2 Low Energy Ion Implantation

As described in subsection 2.2.3(Ion Implantation of 2D Materials), low energy (10-100 eV) ion implantation is an effective and scalable method for doping and manipulating two-dimensional (2D) materials. Implantation energies of the order of keV have been shown to remove several layers of material [325], and is therefore unsuitable for the monolayer materials examined in this project. The implanter utilised throughout this project is the ADONIS at the University of Göttingen, and was operated by H. Hofsäss, J. Amani and M. Auge. The ADONIS is a mass-separated ion beam deposition (MSIBD) system [306]. The basics of the system are described below in subsection 3.2.1, and the design of the TEM grid holders is described in subsection 3.2.2.

The ADONIS was used for both the Se and Au implantation presented in Subsection 4.1(Se Ion Implantation) and Subsection 4.2(Au Ion Implantation), respectively. Additionally, TEM samples of TMDs were implanted samples with Cr and Mn, though the samples themselves were either damaged or too contaminated for quantitative STEM analysis. This research is ongoing. The elements available for implantation by the ADONIS implanter are presented in Figure 3.5.

3.2.1 Ion Implanter Setup

With the ultra-high vacuum (UHV) deceleration chamber with sample stage, one can achieve implantation energies of as low as 10 eV. This chamber is displayed in Figure 3.6 along with the entire implanter schematic. Ions are extracted from the chosen source (see available sources in Figure 3.5 above) and accelerated toward the 90° mass separation magnet. The calibrated desired isotope is selected via the strength of these lenses and an aperture. Further focusing of the beam occurs prior to entering the UHV sample chamber. Deceleration of the focused ions occurs in this chamber via the defocusing lenses positioned around the sample (substrate) as shown in Figure 3.6(b). The total ion dose (fluence) is set via the measured current on the current integrator in the UHV chamber in Figure 3.6(a).

Figure 3.5: A list of the elements available for implantation using the ADONIS ion implanter. Sources available at the time of writing are highlighted.

3.2. Low Energy Ion Implantation

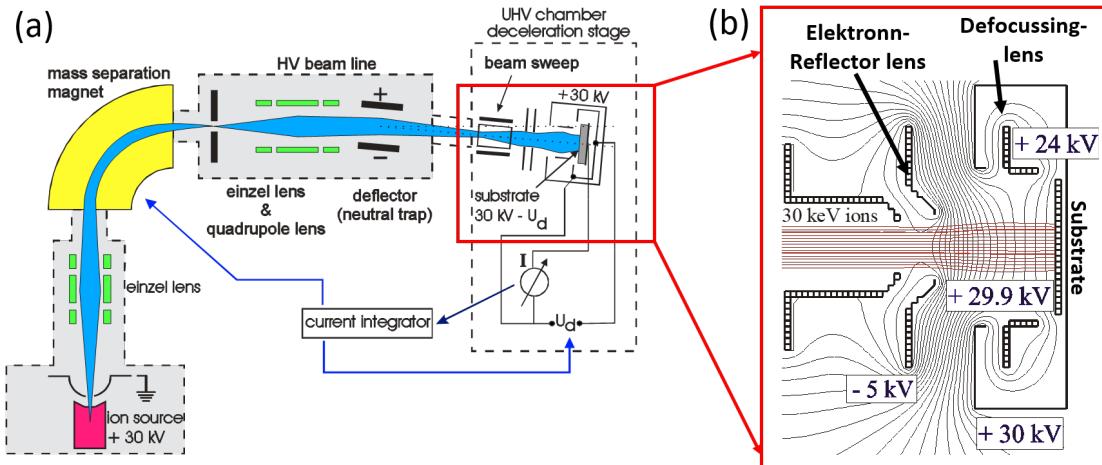


Figure 3.6: (a) Schematic overview of the ion implantation device used throughout this project for the ion implantation of 2D materials. The special deceleration stage is highlighted with a red box and its schematic is presented in (b). The TEM grid substrate is placed on the substrate position and the defocusing lenses slow the incoming ions from 30 keV to several tens of eV. Original images courtesy of Prof. Hans Hofssäss.

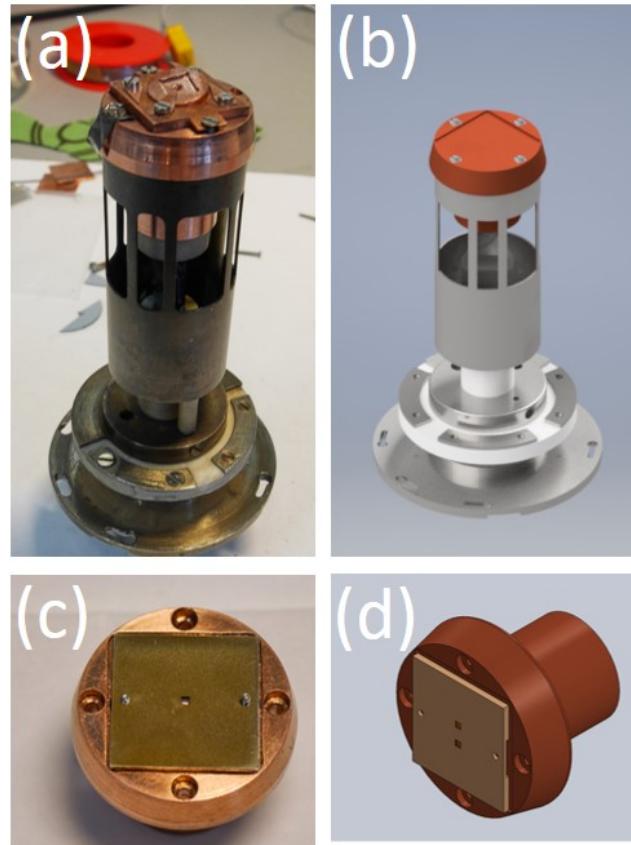


Figure 3.7: (a) Picture and (b) Solidworks render of the ion implanter sample holder and tip. (c) Picture and (d) Solidworks view of the ion implanter tip.

3.2. Low Energy Ion Implantation

The simulation software used during the project is the Monto Carlo method [375]. All predictive implantation simulations were completed by H. Hofsäss, J. Amani and M. Auge.

3.2.2 Ion Implanter Sample Holder

A picture and Solidworks render of the sample holder and tip in the ADONIS ion implanter are presented in Figure 3.7(a,c) and Figure 3.7(b,d), respectively. The TEM grids are positioned on a sample holder, which is in turn positioned on the central square at the top of Figure 3.7(a). The sample holder itself is contacted to the current integrator for ion beam current measurement during the experiment. In Figure 3.7(c) and Figure 3.7(d) a TEM grid sample holder is placed in the ion implanter tip. It is fastened in place by simple screws. The TEM grids are placed facing toward the ion beam in the central hole (c) or holes (d). The standard sample holder can fit four standard 3mm TEM grids.

It is worth noting here that nearing the end of the project, a Dens Solutions-compatible in-situ (heating & biasing) chip holder was designed and built for use with the ADONIS ion implanter sample holder. The original design and final design are shown in Figure A.3 in Subsection A.2.

3.3 Experimental and Simulation Methods

3.3.1 Acquiring Images and Spectra

During this work, transmission electron microscopy was used in scanning mode (STEM). The detector used was either a medium-angle or a high-angle annular dark field detector (MAADF/HAADF). As stated in Subsection 2.1.1(ADF STEM) in the previous chapter, ADF STEM is a widely used tool for imaging and quantification of crystalline structures at atomic resolution. In this section, the operating parameters of both STEMs used during this project will be outlined, and the pros and cons of each discussed. The FEI (now Thermofischer) Titan Themis 60-300 TEM in the University of Limerick (UL Titan) and the Nion UltraSTEM 100MC 40-100 ‘HERMES’ in SuperSTEM, Daresbury (SuperSTEM 3) were used during the project, though the latter was operated by the SuperSTEM staff. Some early experiments were also carried out using the FEI Titan 80-300 ‘ChemiSTEM’ in the Ernst-Ruska Centre, Forschungszentrum, Jülich, though these images do not appear in this manuscript*.

The UL Titan microscope can be operated in TEM or STEM mode, which can allow the user to quickly search for a sample area in TEM before changing to STEM for atomic resolution imaging. It is equipped with an image and probe corrector for high-order aberration correction of the TEM and STEM modes, respectively. It has a primary electron beam energy range of 60-300 keV, though the most commonly used are 60, 80 and 300 keV. Because it is not a dedicated STEM microscope, imaging 2D materials at atomic resolution in ADF mode at 60 keV is a challenge. This is due to low signal to noise, more limited resolution (slower electrons), and general alignment issues (the microscope was not consistently used at 60 kV). Increasing the electron current improved the signal on the ADF detector, but damaged the sample. Despite these issues, imaging of 2D materials was carried out at atomic resolution at 60 keV. As discussed in subsection 2.1.2(Electron-Matter Interactions), 60 keV is below the displacement threshold energy for S atoms in MoS₂. Single atom microscopy was also performed in the UL Titan at 80 and 300 kV. Though baking and cleaning of the TEM column was carried out prior to imaging, contamination was still sometimes an issue due to sample contamination. The HAADF detector collection angle range used for imaging in the UL Titan was 103-200 mrad, though lower inner angles were sometimes used to increase signal.

For consistent and reliable single atom microscopy, the SuperSTEM 3 microscope is viewed as one of the best in the world. Previous experiments of single atom

*It is worth noting that for first two years of the project the UL Titan microscope was not effectively operational, requiring the use of other microscopes.

3.3. Experimental and Simulation Methods

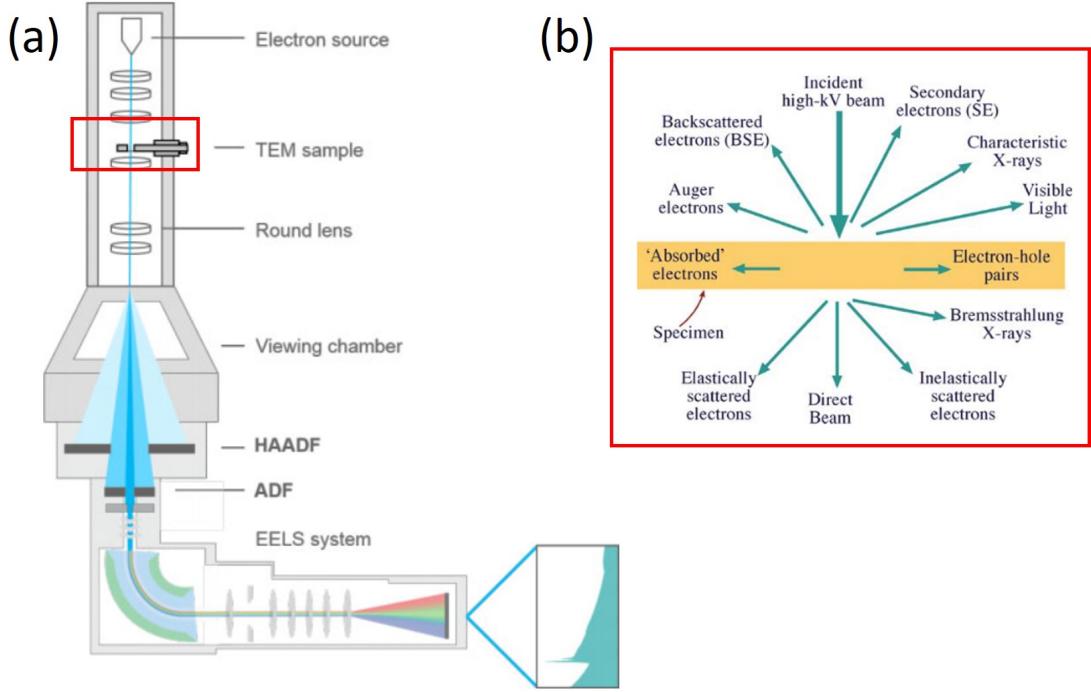


Figure 3.8: (a) Schematic of a scanning transmission electron microscope (STEM), showing the ADF detectors and EELS spectrometers. Adapted from [376]. The signals resulting from fast (incident high-kV) electrons interacting with a sample in a STEM. Taken from [28].

investigations into ion implanted 2D materials were carried out in the UltraSTEM 100 (SuperSTEM 2) microscope [19]. The SuperSTEM 3 is a dedicated STEM and therefore never leaves the STEM imaging mode (unlike the TEM/STEM UL Titan). It has a better vacuum ($\sim 10^{-8}$ mbar compared to the Titan's $\sim 10^{-7}$ mbar) and does not host in-situ experiments routinely, which reduces contamination. Additionally, the Nion STEM sample holder can hold 4 samples at once, unlike the 1 sample for the UL Titan holder. Therefore multiple samples can be viewed without disturbing the microscope, further increasing the stability and reducing sample drift. This also has the added benefit of having the probe correction calibration sample as one of the 4 samples. For the UL Titan, this calibration sample must be inserted separately, increasing the above-mentioned contamination of the column and sample drift. The SuperSTEM 3 sample holder (with the samples) is left to bake in vacuum for ~ 12 hours prior to insertion into the microscope, where it is then left overnight to reduce sample drift. The heavy cold-field emission electron source of the SuperSTEM 3 microscope is placed at the bottom of the microscope, further stabilising the set-up against vibrations (UL Titan source positions at the top). The HAADF detector collection range used during experiments was 83-185 mrad. MAADF images were simultaneously acquired at a 60-80 mrad range. Fi-

3.3. Experimental and Simulation Methods

nally, the low primary electron energy of 60 keV was used to reduce sputtering in monolayers. At 60 keV, the SuperSTEM 3 can reach a probe size of $\sim 1 \text{ \AA}$.

Both the UL Titan and SuperSTEM 3 microscopes were used for spectroscopic investigations (EELS, EDX). The UL Titan was operated at low primary energies (60-80 keV) for EELS and EDX investigations to increase the effective scattering cross-section and resulting signal. Both core-loss (CL) and low-loss (LL) EELS of ion implanted monolayers were carried out. When operated in monochromated mode, the zero-loss peak (ZLP) of the UL Titan generally managed a full width at half maximum (FWHM) of 160 meV, though it can be pushed further by optimisation [377]. SuperSTEM 3 can currently reach a FWHM of 15 meV, allowing for resolution of energy-loss from phonon scattering [33, 134]. Both microscopes have the ability to simultaneously acquire LL, CL spectrum images and ADF images. As discussed in Subsection 2.1.1(EELS), the high dwell time required for sufficient EELS signal limits its use and damages 2D materials. Examples of EELS spectrum images and this damage can be seen in subsection 4.1.2(Spectroscopy of Se).

A generalised set-up for simultaneous imaging and spectroscopy in STEM mode is shown in Figure 3.8(a) [376]. The electron beam is extracted from a source. This source can be thermionic, such as a tungsten filament or LaB₆ crystal, a field emission (FE) tungsten tip, or a cold-FE tungsten tip. The UL Titan and SuperSTEM 3 use a FE and cold-FE tip, respectively. In STEM mode, the objective lens focuses the electron beam to a single point on the sample, and raster scans this point across the sample. The interaction between the electron beam and sample creates a plethora of signal with which one can analyse a sample, some of which are shown in Figure 3.8(b) [28]. The characteristic X-rays emitted from near the surface of the sample allow one to map the chemical composition of the sample via an EDX detector. Additional detectors for collecting secondary and Auger electrons, as well as cathodoluminescence detectors, can be fit to the microscope. Secondary electrons have recently been used to image at atomic resolution [378, 379].

Electrons transmitted through the sample can be divided into elastically and inelastically scattered electrons, described in subsection 2.1.2(Electron-Matter Interactions). Electrons scattered at high angles are collected in the ADF (HAADF) detectors shown in Figure 3.8(a). This thermal diffuse scattering (TDS) forms the basis of the majority of images presented in this thesis. For more information on the formation of ADF images, see Subsection 2.1.1(ADF STEM). Electrons scattered at low angles are detected on a bright field (BF) detector. If an EELS signal is desired, this BF detector is retracted, and in some cases difference ADF detectors

3.3. Experimental and Simulation Methods

must be used. The electrons are separated by energy with a magnetic prism and are detected by a spectrometer. The majority of the electrons do not lose energy, and therefore lie at or near the zero-loss peak (ZLP). Electrons which lose energy spread out along the dispersion plane. Electrons which lose energy due to some characteristic loss process appear as peaks in the spectrum. More information on the different signals obtained via EELS can be read in Subsection 2.1.1(EELS).

3.3.2 Image Simulation and DFT Calculations

Image simulation was used to verify the contrast of ADF STEM images acquired throughout this project. The multislice approximation to the full quantum mechanical model of electron-matter interaction was used via the programs Dr. Probe [380] and Prismatic [11, 12]. In the multislice approximation, the specimen (model) is treated as a series of scattering planes (slices) of atomic potential [36, 381]. The scattered potential of each plane is projected onto the following plane until the electron wave transmits fully through the sample. The planes do not refer to any physical property of the sample, and there must be sufficient planes to represent the atomic potential of all atoms in the specimen. For 2D materials such as monolayer MoS₂, the number of planes was set between 3 and 6, depending on the structure.

The frozen phonon approximation to the thermal motion of atoms in a specimen was used for simulations [382]. The frozen phonon approximation uses correlated phonon modes, i.e., movement of the lattice. It is a popular and accurate method, and can be used in most situations [383]. Generally, 30 or more frozen phonons were set for each simulation [384]. To simulate large images (512 - 2048 pix²), Prismatic and its Python implementation of PyPrismatic was used. Prismatic's Prism algorithm allows for a speed increase of four orders of magnitude with negligible loss of accuracy [11]. Integrating PyPrismatic with the Python functions designed throughout the project allowed for extensive development of the algorithms described in subsection 5.1.1(Refinement of Atomic Resolution Images via Image Simulation). The interpolation factor of the Prism algorithm allowed this to be realistic on a standard work computer, allowing 1k x 1k to be calculated in a matter of seconds, rather than hours, without losing critical information. For hundreds of large images, the speed increase from the prism algorithm is the reason why the Model Refiner tool can be used iteratively, as described in subsection 5.1.2.

For simulations of monolayer MoS₂, contamination was not directly accounted for by including approximations of hydrocarbon positions on the surface of the monolayer. Instead, regions that appear generally brighter, which indicates hy-

3.3. Experimental and Simulation Methods

drocarbon contamination, appear with a single S atom above the Mo atom. This is to give a broad impression of contamination in a region, and is not an attempt at quantification of surface contamination. Though this would be an interesting investigation [385], it is outside the scope of this project.

The density functional theory (DFT) calculations used to understand the loss or gain in lattice energy due to dopant atoms and vacancies in subsection 4.1.1 (Microscopy of Se) and Subsection 5.2(Atom Dynamics) were simulated by M. Ghorbani-Asl and A.V. Krasheninnikov.

3.4 Signal and Data Analysis

As the size and complexity of transmission electron microscopy (TEM) data increases, methods for processing such data must improve commensurately. Indeed, the information produced by the microscope is only useful if it is processed and displayed in a meaningful manner. Well documented and wide ranging tools are necessary, and have unfortunately been lacking in the field of TEM. This is likely due to the proprietary and monopolised nature of analysis suites in the past, i.e., Gatan's Digital Micrograph. Now, more analysis tools are following the open-source mantra, wherein anyone can view the source code and contribute to their development. The development of Python-based open-source tools like HyperSpy and Atomap allow researchers to develop tools for their specific needs.

In this section, we discuss the methods used in this work for image analysis, atomic position analysis, nanoparticle (NP) identification, model visualisation, multidimensional data analysis, and atomic movement analysis. Finally, each analysis tool and Python package used and developed throughout the project are briefly described, as well as the interface for using such packages.

Python code in this manuscript appears with a grey background, e.g.,
`here_is_a_function` in the module `example_module.py`

3.4.1 Element Quantification in HR ADF STEM Images

Grouping and Quantifying Atom Intensities

To quantify the number of atoms in each atomic column in each sublattice detected in an atomic resolution STEM image, one must first create* a method for grouping the columns into elemental configurations based on statistical distribution. This has been implemented by several groups for single and multiple element species in atomic columns [45, 107, 115], as well as for different element configurations in 2D materials [9, 10, 21, 43]. For more information on these methods, see Sub-section 2.1.1(ADF STEM). We have implemented these methods as reproducible and easy-to-use Python functions. These can be found in the TEMUL toolkit on GitHub [13] (TEMUL: Transmission Electron Microscopy University of Limerick).

Additionally, fast image simulation is employed to refine the model created from experimental images for both 2D and 3D structures which we believe can rejuvenate the future direction for reliable and reproducible elemental quantification. This development is described in detail in subsection 5.1.2(Model Refiner (MR) Tool).

* "...the universe" - Carl Sagan

3.4. Signal and Data Analysis

Below, methods utilised during this project for the analysis of both single elements in long atomic columns and multiple element configurations in 2D materials are described.

Single Element Species in Atomic Columns:

Examples: Simple 3D structures, NPs

1. Plot the intensities of the atomic columns as a histogram. The background intensities are discussed in Subsection 3.4.3(Background Intensity).
2. Fit a series of Gaussian distributions over the histogram using a Gaussian mixture model if required [115].
3. Specify the thickness of the sample. Assign the highest intensity to this thickness, working down through the lower intensity Gaussian fits.
4. Create an xyz, element coordinate information file (.cif, .xyz, .vesta etc.) model for display subsection 3.4.6 and DFT calculations subsection 3.3.2.

The primary question for the above walk-through is to query whether a series of added Gaussian distributions describes correctly the actual distribution of intensities due to atom column thicknesses. If it does, one must assume that each subsequent Gaussian provides a linear increase in the number of atoms in an atomic column (starting from the lowest value). In the simplest case, wherein groups of atomic columns' intensities are clearly separated, one can fit a Gaussian distribution about each grouped atom intensity distribution [117]. In a more realistic case, wherein the atomic column intensities overlap, one must use the addition of multiple Gaussian distributions which provides the most accurate overall fit to the intensity values of the atoms' intensity distribution [115, 120]. By using a Gaussian mixture model, one can not only set the likely number of atoms in each atomic column, but also define the fitting error from the histogram data under each Gaussian. The intensity value at which each Gaussian distribution overlaps, indicates the uncertainty of element assignment of any values outside the overlap.

Various Element Species in Single Sublattice:

Examples: Doped 2D materials, Complex 3D structures

1. Plot the intensities of the atomic columns as a histogram. An example is shown in Figure 3.9.
2. Fit Gaussian distributions over the histogram dependant on two standard elements e.g., Mo and S₂ for the transition metal dichalcogenide sublattice

3.4. Signal and Data Analysis

of MoS₂. In Figure 3.9, the large yellow and blue Gaussian distributions are plotted over the S₂ and Mo intensities, respectively. The remaining, smaller Gaussian fittings are found in this case to be Se and SeS element configurations.

3. Using these distributions to define the elements in each xy position, create an element coordinate information file (.cif, .xyz, .vesta etc.) model for display subsection 3.4.6 and DFT calculations subsection 3.3.2.

Several atomic column configurations exist in doped 2D materials and in more complex 3D structures. Therefore, the distributions of atom intensities does not represent a linear increase in the number of atoms of one element. Here the focus remains on 2D material atomic configurations. In each atomic column there may be variations in elemental configuration due to the doping, eg., Se+S found in Figure 3.9. A reasonable model can be created based on an internal Z-contrast scaling factor n . n is found experimentally as the ratio of two elemental configurations, which is then used to scale the expected intensity value for all other elemental configurations, based on their atomic numbers (fully described in the next section). Theoretically, it is found by calculating the scattering cross-sections of individual atoms via Equation 2.4 with the objective aperture correction in Equation 2.3 [29]. Further corrections, described in Subsection 2.1.1(ADF STEM) must also be applied. Though the scaling factor n also depends on Z itself, it is (currently) approximated to be the same throughout the image for all values of Z [29] .

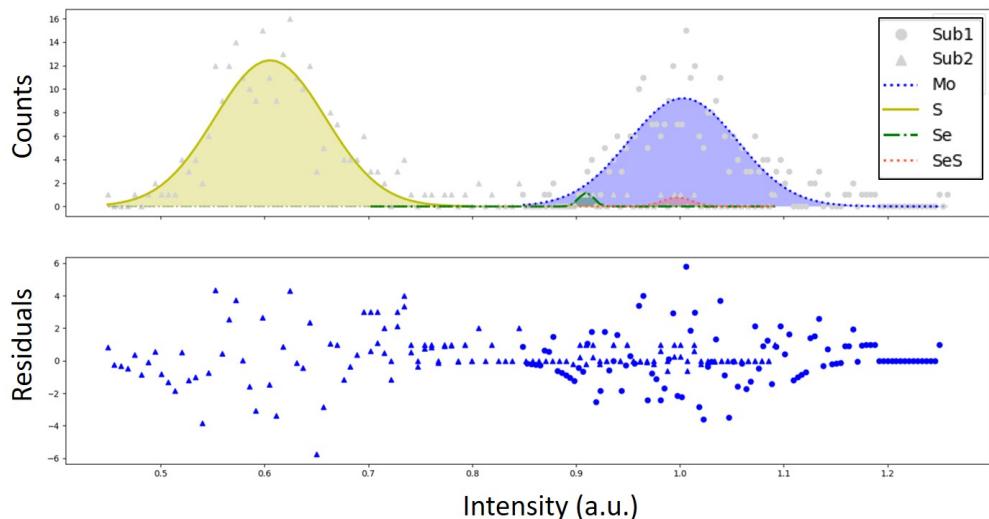


Figure 3.9: Gaussian fitting of the ADF Z-contrast intensities from the transition metal (circle) and chalcogen (triangle) sublattices from a Se ion implanted MoS₂ monolayer. The residuals from the Gaussian fitting are shown below. Overlap between the Gaussian curves can be used to estimate uncertainty of element configuration assignment.

Ideally, one could use a Gaussian fitting of all elemental configurations to calculate the error and uncertainty for each configuration. For example, the residual error of the fitting is shown in Figure 3.9. However, it is often the case that only a handful of instances of certain element configurations occur in each image. A method to overcome this lack of statistics is to combine the intensity values of a series of images normalised about some standard, e.g., a single stable atom or by absolute intensity.

Sort Atom Position Intensities into Element Configurations

This section describes the approach taken during this work to analyse intensities and assign element configurations. To correctly assign an atomic column intensity in a 2D material to one or more elements, the method employed by Krivanek *et al* [21] was used. The ratio of intensities method below was also used to verify the monolayer nature of different samples by comparison with image simulation. This method is implemented here using the data structure of the Atomap Python program. The relevant code can be found in the TEMUL toolkit [13], and will be added to the Atomap package after thesis submission.

Initially, the intensities of all atom positions in the image are calculated (see subsection 3.4.3 for atom position finding methods). This is done with either the total, maximum, mean, median, etc., value of the intensity around an atom defined by a given radius or Voronoi cell [see Subsection 2.1.1(ADF STEM)]. The distribution of these intensities is displayed with a histogram (see Figure 3.9) and the scaling ratio R is computed with

$$R = \frac{I_N}{I_D} = \frac{(x_N Z_N)^n}{(x_D Z_D)^n} \quad (3.1)$$

where R is the ratio of the numerator sublattice intensity to the denominator sublattice intensity. I_N and I_D represent the intensity value of the most common numerator and denominator element configurations, respectively (e.g., Mo, S₂ in MoS₂). x_N and x_D are the count of atoms in a standard column of the numerator and denominator, respectively. Z_N and Z_D are the atomic number of the numerator and denominator elements, respectively. Rearranging to find the Z-contrast scaling exponent, n , we find

$$n = \frac{\ln(\frac{x_D}{x_N}) + \ln(R)}{\ln(Z_N) - \ln(Z_D)}. \quad (3.2)$$

3.4. Signal and Data Analysis

This Z-contrast exponent n is used to scale the boundaries between the aforementioned elemental configurations. Similar intensities will naturally be grouped. Grouped intensities with substantial numbers of counts can be fit with a Gaussian distribution, defining the residual error and uncertainty of the group. Prior knowledge is required to correctly assign the element configurations; the most common elemental configuration must be known for each sublattice to correctly scale the intensities for other element configuration groups, such as Se_1 , Se_2 etc. The relationship between R and n is shown for the four most common TMDs in Figure A.4 in Subsection A.2.

Equations 3.1 and 3.2 are both calculated in the function `scaling_z_contrast` in the `model_creation` module of the TEMUL toolkit [13]. The scaling exponent n is then used to scale each element configuration with the function `find_middle_and_edge_intensities`, which returns the central (middle) and boundary (limit) intensities for each element configurations. The atom's theoretical z coordinate is then set by `assign_z_height`.

3.4.2 Methods for Analysing Images

To reproducibly analyse image data taken on the microscope, one must utilise many analysis tools. These range from spatial alignment to filtering, from basic image normalisation to advanced image simulation. Much of the methods described in this section are accessed through Python packages, and where reasonable, relevant code segments are included.

Image Alignment

For scanning TEM data, images can contain distortions due to movement in the fast and slow scan directions. The fast scan direction is the direction in which the probe moves, left to right. The slow scan direction is each step of the probe from the top to bottom of the image, orthogonal to the standard left-to-right probe direction. These scan directions produce different scan distortions, which are discussed below. During this project, the misalignment due to sample drift over the slow scan direction was corrected by rigid image registration.

As succinctly described by Ophus *et al* [386], slow scan distortions are routinely corrected during experiments, such as piezo jogging of the sample. Sample drift over time results from the slow scan direction. For an image series (movie), if the scan time of each image frame is low enough to exclude severe sample drift within that frame, the main alignment required for the slow scan direction is rigid image registration. Rigid image registration is simply the alignment of an image with one or more images in the image series by some xy shift. Therefore, it is the movement of each image by some static vector.

This is demonstrated in Figure 3.10, in which the highlighted orange circle in each image is the central pixel, for reference. The raw pixel data for two example images of intensity surrounding an atom, Image 1 and Image 2, are shown in Figure 3.10(a) and Figure 3.10(b) respectively. Image 2 is misaligned by some xy shift due to sample drift. In our simple example, this drift relative to Image 1 is (1, 1). Therefore, the pixel values in Image 2 must be shifted (-1, -1) to be correctly aligned with Image 1. The result of this shift on each image is shown in Figure 3.10(c) and Figure 3.10(d) for Image 1 (unchanged) and Image 2 (shifted), respectively. The NaN (Not a Number) values in Figure 3.10(d) and Figure 3.10(e) are lost information due to the shift. These edge pixels should be removed for final analysis. Alternatively, wraparound of pixel values can be used, assuming a perfect material structure. The xy components of the shift vector are generally saved, for reproducibility.

3.4. Signal and Data Analysis

Throughout the project several image registration software packages were tested. The Rigid Registration Python package [387] and Scikit-image's register translation [388], have been utilised for Python based analysis for files accessible with HyperSpy [69]. The in-built Velox image alignment software is best for .emd data files produced in the UL Titan, which can be optimised for repeating structures. Other options are the alignment tools available in Gatan Digital Micrograph, ImageJ, SmartAlign, and elsewhere.

The primary movement of the electron beam can result in non-linear image distortions, as discussed in [386]. The authors created an algorithm which can correct for these non-linear drift distortions with only two images, in which the fast scan directions are orthogonal. In other words, acquire an image, then rotate the scan direction by 90°. As the electron beam is moved from the end of one scan in the fast direction to the start of a new line, the new position may not be entirely accurate. This can be due to electron beam alignment or jump discontinuities in the beam or sample. Assuming the slow scan direction axis is consistent and accurate, the algorithm takes the slow scan direction axis as the correct fast scan direction for the second image. This removes the lottery of whether the beam jump from the end of one line to the start of the next is accurate in the fast scan direction. For full details, see [386]. Importantly, the authors provide the MATLAB code with sample data for testing the algorithm. This code was used to correct drift distortions in images near a domain wall (DW) in LiNbO₃ in ??, though none are displayed in this manuscript.

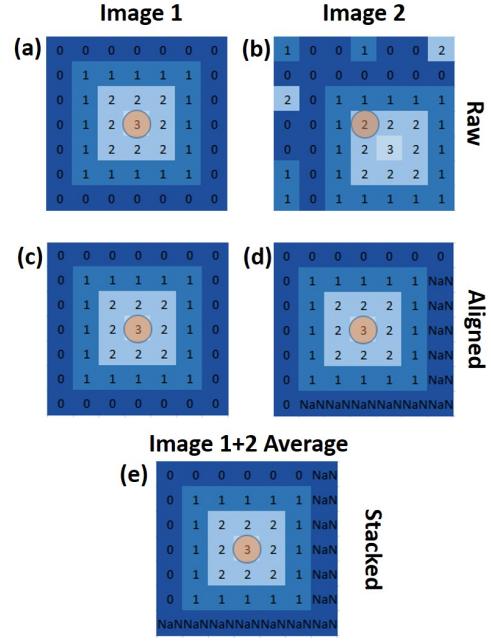


Figure 3.10: Image alignment of two images each represented by a 7x7 array of pixels. The central pixel in each image is marked by an orange circle for reference. (a-b) Initially, the image shift between image 1 and image 2 is calculated (often by cross correlation between the discrete Fourier transforms. (c-d) The calculated image shift is applied to image 2 to produce (d). Note the not a number (NaN) values in the bottom and right extremes of image 2. (e) The images are stacked by taking the mean of the pixel values. The NaN values represent lost information due to the alignment process.

Assuming the slow scan direction axis is consistent and accurate, the algorithm takes the slow scan direction axis as the correct fast scan direction for the second image. This removes the lottery of whether the beam jump from the end of one line to the start of the next is accurate in the fast scan direction. For full details, see [386]. Importantly, the authors provide the MATLAB code with sample data for testing the algorithm. This code was used to correct drift distortions in images near a domain wall (DW) in LiNbO₃ in ??, though none are displayed in this manuscript.

Image Calibration

For image data analysed throughout this project, basic length and intensity calibration was often required. For calibrations of length, one uses the sampling of the pixels. Pixel sampling is the measured length per number of pixels in the measured length as defined in Equation 3.3,

$$\text{Pixel Sampling} = \frac{\text{Image Size (physical units)}}{\text{Image Size (pixels)}} \quad (3.3)$$

Usually, this is given in nm/pix. The physical size of the image is often unknown, or incorrectly calibrated by the microscope's software. In these cases, theoretically calculated reference values, such as those calculated using DFT, should be used. In this case, the number of pixels spanning a unit cell can be used as the Image Size (pixels) in Equation 3.3. Knowing the pixel sampling allows analysis to be completed with pixel distances, which makes direct comparison during analysis easier. Final vector values, for example, can then be scaled by the pixel sampling when plotting or displaying the data. Sampling is used directly in many of the functions utilising Atomap's sublattice data structure, e.g., `get_vector_magnitudes`, `plot_polarisation_vectors`.

For intensity calibration, one can use internal or external intensity scaling. Internal scaling is scaling the intensity relative to a given intensity value in the image. This value can be the intensity value of a consistent atomic column in an image, such as the Mo atoms in single layer of MoS₂. This method is used extensively in Chapter 4(Atomic Characterisation of Ion Implanted 2D TMDs) and Chapter 5(Atom Dynamics in 2D TMDs). External intensity scaling is using the absolute fractional intensity each pixel receives during imaging, and can be quantitatively compared with image simulation [45, 48, 106]. Because the absolute scaling has been implemented in the Atomap Python package, it can easily be integrated into the TEMUL toolkit. This method was not utilised during this thesis, though the detector response data were acquired during imaging.

The `calibrate_intensity_distance_with_sublattice_roi` function of the `signal_processing` module of the TEMUL toolkit allows the user to calibrate the intensity and spatial dimensions of an image. It creates an Atomap sublattice from the atoms in the image, defined by the parameters `cropping_area` from which to calibrate the intensity and the sampling. To output a spatially calibrated image from this function, the `reference_image` must be set to a spatially calibrated image, or image sampling given.

Image Convolution/Filtering

Image convolution can be applied in the real or reciprocal space. In general, it is combining of two or more signals to produce a convoluted signal, which in our case is the filtered image. Real space convolution can be carried out with the `gaussian_filter` function from `scipy`, which will blur the input image with a Gaussian kernel. Reciprocal space image convolution can be applied in a similarly simple manner. For crystalline materials, masking the fast Fourier transform (FFT) diffraction spots in image data can be used to directly investigate material properties, as shown in ??(CuCl Boracite) and by [346]. Low-, high- and band-pass filters can also be applied to enhance desired signals.

Krivanek *et al* convoluted a band-pass (double-Gaussian) filter with the FFT of monolayer hBN image data. This filter has the effect of suppressing the background noise to zero and enhancing the contrast of the atoms. We have created easy to use Python functions for both FFT masking `get_masked_ifft` and FFT filtering with an optimised band-pass filter `double_gaussian_fft_filter`, which are available on GitHub [13]. A `compare_two_image_and_create_filtered_image` function is also available. It filters an image with a Gaussian kernel to best match experimental images with simulation. All above-mentioned functions are in the `signal_processing` module of the TEMUL toolkit.

For FFT masking, the mask coordinates are first chosen with `choose_mask_coordinates`. The parameter `norm` allows the user to set the image intensity to a logarithmic scaling. The masking coordinates are then input to `get_masked_ifft` which output the inverse FFT of the masked FFT. It has various parameters similar to the masking tools in Gatan Digital Micrograph (DM), such as `mask_radius`, whether the input image is in real or reciprocal `space`, whether to keep the masked area or its inverse with `keep_masked_area`, and whether to plot the masked FFT with `plot_masked_fft`. The functions were utilised in [346].

The `double_gaussian_fft_filter` function utilises the filtering method described by Krivanek *et al* [21]. Additionally, it calculates the ideal amplitude of the narrow negative Gaussian to reduce the lowest intensity in the image to zero and retain as much information as possible. The required parameters are the diameters of the inner (`d_inner`) and outer (`d_outer`) Gaussians in 1/distance units, the `real_space_sampling`, `units`, and the negative Gaussian increment `delta`. Smaller delta values increase accuracy but reduce speed.

The `compare_two_image_and_create_filtered_image` function compares a simulation to an experimental image and measures the structural similarity index (SSM) and the mean square error (MSE). It will then increase the stan-

3.4. Signal and Data Analysis

dard deviation of the Gaussian until a best match is calculated, outputting the filtered simulation. The Gaussian increment is set with `delta_image_filter`, `cropping_area` sets the region and `separation` sets the pixel separation of the simulation from which to calibrate the intensity scale. This function is used within the Model Refiner (MR) tool when refining the structure of the atom model with the experiment. This process is described in more detail in subsection 5.1.2. The next step of development for this function is to include Lorentzian kernel blurring [105], as well as a Gaussian-Lorentzian mix.

Measuring Error between Images

Both the mean squared error (MSE) [1, 389] and the structural similarity index (SSM) [390, 391] algorithms were used here to measure errors and overall similarity between pairs of images. The input images can be any type of experimental or simulated image of identical x-y shape. By filtering the simulated image and measuring the error and similarity, the best match between experimental and simulated atomic resolution images was found. This method is repeated iteratively to refine the 3D models created after element quantification, as in Chapter 5(Atom Dynamics in 2D TMDs).

The MSE is the difference between each pixel of two images, and therefore directly measures the error between images, as shown in Code Segment 3.1. If the pixels have an identical value, $MSE = 0$, as the error between the images is 0. If the images are in no way similar (all pixel values different), $MSE=1$. MSE is calculated for two dimensions as

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2, \quad (3.4)$$

where m and n are the x and y dimensions of the image, respectively (the images must have the same dimensions). I and K are the first and second images, respectively. i and j are the i^{th} and j^{th} pixel coordinates, respectively. Equation 3.4 is implemented in the `mse` function in Code Segment 3.1 below and in the `signal_processing` module of the TEMUL toolkit [13].

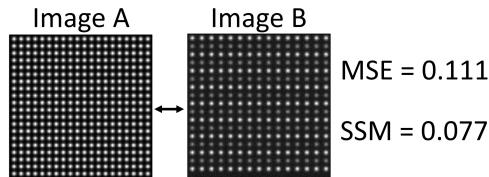


Figure 3.11: Mean Squared Error (MSE) and Structural Similarity Index (SSM) measured between two example images. These methods can be used to compare any two images, such as an experimental STEM image and its simulation.

MSE is a quick and easy image comparison method, but is limited when the images are spatially shifted. This can be overcome by cross correlating the images to align them spatially as described above in Subsection 3.4.2(Image Alignment). The most realistic way to overcoming any shortcomings of MSE is to use a second method that will measure the overall similarity in structure between two images. The structural similarity index (SSM) is a good complementary technique. SSM=1 for identical images and a small value close to 0 for uncorrelated images. It measures the similarity of several averaged windows taken over the same regions in each image, and therefore is less affected by image shifts. For two dimensions, the SSM is given by:

$$SSM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (3.5)$$

where x and y are the first and second images, respectively. μ , σ^2 , $\sigma_{x,y}$, and c are the mean, variance, covariance and division stabilizer, respectively [390]. SSM is implemented in the Scikit-Image package as the `structural_similarity` function.

```

1 import numpy as np
2 # Call this function to calculate the error between imageA and
3 # imageB
4 def mse(imageA, imageB):
5     err = np.sum(
6         (imageA.astype("float") - imageB.astype("float")) ** 2)
7     err /= float(imageA.shape[0] * imageA.shape[1])
8     return err

```

Code Segment 3.1: Mean Squared Error implementation from [1] in the `signal_processing` module.

3.4.3 Methods for Atom Position Analysis

To analyse the atomic column positions (atom positions) in atomic resolution scanning transmission electron microscopy (STEM) images, a plethora of analysis methods were applied. These are described in detail in the below subsections, beginning with feature location in Subsection 3.4.3(Initial Atom Position Finding), correct feature fitting in Subsection 3.4.3(Position Refinement), deciding whether to refine the position in Subsection 3.4.3(Toggle Refinement), and finally background subtraction methods in Subsection 3.4.3(Background Intensity Subtraction). Grouping and quantifying atom intensities is discussed in detail in subsection 3.4.1. We will briefly describe these analysis steps in terms of finding, refining, and quantifying with Python tools.

One can use the Python program Atomap to access the above-described methods, such as feature identification via pixel separation and sub-pixel position refinement. Tutorials are available at atomap.org [4]. The general steps taken to find all atom positions with sub-pixel accuracy is as follows. First, one uses the `get_feature_separation` function to view a range of pixel separations. After choosing the most suitable pixel separation, one creates a list of atom positions with `get_atom_positions`. One can then use this list of atom positions to create a `Sublattice` object. The atom positions' locations can then be refined with sub-pixel accuracy [4]. Finally, several sublattices can be combined into an `Atom Lattice` object, which can be stored for later use.

Initial Atom Position Finding

In images with repeating structures, such as atomic resolution scanning transmission electron microscopy (STEM) images, an effective method for feature location is pixel separation. A centre of a bright region of pixels is identified as a feature. The next feature of interest is constrained to being at least the inputted number of pixels away from all other features. This is the basic principle of feature location with pixel separation. This method is ideal for periodic structures, and is therefore widely used in STEM post processing [4, 392].

In some images, finding the initial atom positions is made difficult by a lack of brightness difference between atomic columns of different element configurations, e.g., in some Boracite structures, the Cu and CuCl columns can be of similar brightness. In other words, there is no clear first sublattice in some materials. Two methods of dealing with these issues were implemented during this work: inverse FFT filtering and deleting undesired atom planes from along different zone axes. The FFT filtering method uses the FFT functions described in Subsection 3.4.2(Image

Convolution/Filtering) above. Each periodic structure will naturally have diffraction spots in an FFT, and filtering out all other diffraction spots will reveal the desired first sublattice atom positions. Deleting atom planes from a repeating pattern can remove undesired atom positions in an image and can be done with the `delete_atom_planes_from_sublattice` function. It was utilised for images of the ferroelectric material PbGeO, and is described in Subsection 6.1(Tools for Visualisation of Subatomic Shifts).

Atom Position Refinement

Several position refinement algorithms were used throughout the project to correctly position atomic coordinates. The most commonly used algorithm is the two dimensional (2D) Gaussian refinement. Essentially, the algorithm calculates a best fit 2D Gaussian to the intensity values surrounding an atom. The centre of mass method is faster, but generally considered less accurate than the 2D Gaussian method. Each step in the Atomap implementation of 2D Gaussian and centre of mass refinement is given here.

Atomap 2D Gaussian Refinement:

1. Create a mask around each atom position's data (pixel values), limited by a given radius around the atom position.
 - (a) Hyperspy's `Gaussian2D`
 - (b) Atomap's `_atom_to_gaussian_component`
2. Create a 2D Gaussian distribution using the coordinate, standard deviation, and rotation of each atom position.
 - (a) Numpy's `ogrid`
 - (b) Atomap's `_make_circular_mask`
 - (c) Atomap's `_make_mask_from_positions`
3. Create a HyperSpy model, containing the above masked data as a 2D image and the above 2D Gaussians. Also store the masked image.
 - (a) Atomap's `_make_model_from_atom_list`
4. Using the fit method on the above HyperSpy model, fit the 2D Gaussians to the image data. In other words, fit the model Gaussians to the initially located atom position data. If successful, this fitting will return the fitted

3.4. Signal and Data Analysis

values of atom position coordinates, amplitude, standard deviation and rotation.

- (a) Hyperspy's `model.fit`
 - (b) Atomap's `_fit_atom_positions_with_gaussian_model`
5. If every fitting is successful, the atom position's attributes, such as coordinates, standard deviation etc., will be changed. If the Gaussian coordinates are outside the masked data radius, or if the standard deviation x/y ratio is too large, a centre of mass refinement is used instead (see below).
- (a) Atomap's `fit_atom_positions_gaussian`
 - (b) Atomap's `refine_position_using_2d_gaussian`
6. If called from a sublattice object, the above steps will be applied and stored for each atom position in the sublattice. The newly fitted atom positions will appear if plotted with `sublattice.plot()`
- (a) Atomap's `refine_atom_position_using_2d_gaussian`

Atomap Center of Mass Refinement:

1. Calculate the centre of mass of the data (pixel values) around an atom. This is simply finding the location with the highest average values, as compared to other locations.
 - (a) Numpy's `ogrid`
 - (b) Scipy's `center_of_mass`
 - (c) Atomap's `_calculate_center_of_mass`
2. Create a mask around the atom, limited by a given radius. Store the location of the centre of mass of this masked data as the new atom coordinate.
 - (a) Atomap's `get_center_position_com`
 - (b) Atomap's `refine_position_using_center_of_mass`
3. If called from a sublattice object, the above steps will be applied and stored for each atom position in the sublattice. The newly fitted atom positions will appear if plotted with `sublattice.plot()`
 - (a) Atomap's `refine_atom_positions_using_center_of_mass`

Toggle Refinement

The `toggle_atom_refine_position_automatically` function was designed to complement Atomap's manual `toggle_atom_refine_position_with_gui` picker tool function. The developed automatic function excludes from refinement those atom intensities which do not fulfill the chosen criteria. In Atomap terms it sets the atom's `refine_position` attribute to False, stopping its refinement with either method described above. The minimum and maximum allowed intensity of the atom can be set via the `min_cut_off_percent` and `max_cut_off_percent` parameters. The allowed intensities can be set between or outside these limits with `range_type='internal'` or `range_type='external'`, respectively. It was used to stop coordinate refinement of low atomic number elements, such as sulphur, when it was likely those atoms were close to very bright atoms, and to stop refinement of vacancy positions.

Background Intensity Subtraction

To analyse the intensity of atoms and atomic columns, one often needs to know and subtract the background intensity. This can be the average or local background intensity. A background subtraction tool is already included in Atomap. However, for our needs, several background intensity subtraction tools were designed such as `get_cell_image` in the `signal_processing.py` module, and `remove_local_background` and `remove_average_background` in the `intensity_tools.py` module, all of which can be found in the TEMUL toolkit [13].

Here variations for calculating the local background intensity around atom positions are discussed. These can generally be simplified to the following: Find the atom position, find the lowest intensity value a certain radius from that atom. This lowest intensity can be subtracted from the atom intensity to create a background subtracted image, not affected by local fluctuations in background intensity. Here the lowest background intensity in a Voronoi cell method (`get_cell_image`) is shown.

The method is shown with a basic and actual graphical example in Figure 3.12. The nearest neighbour pixels from each atom are calculated. To do this, first the distance from an atom to each pixel in the image is calculated. Four example atom positions are seen in the top row of Figure 3.12 as blue circles. The pixels closest to each blue atom position are marked by the thicker borderlines. In Figure 3.12(a), the first (top-left) atom position has a peak intensity of 10, the nearest neighbour pixels are defined as the 7x7 array surrounding it. A Boolean array is created,

3.4. Signal and Data Analysis

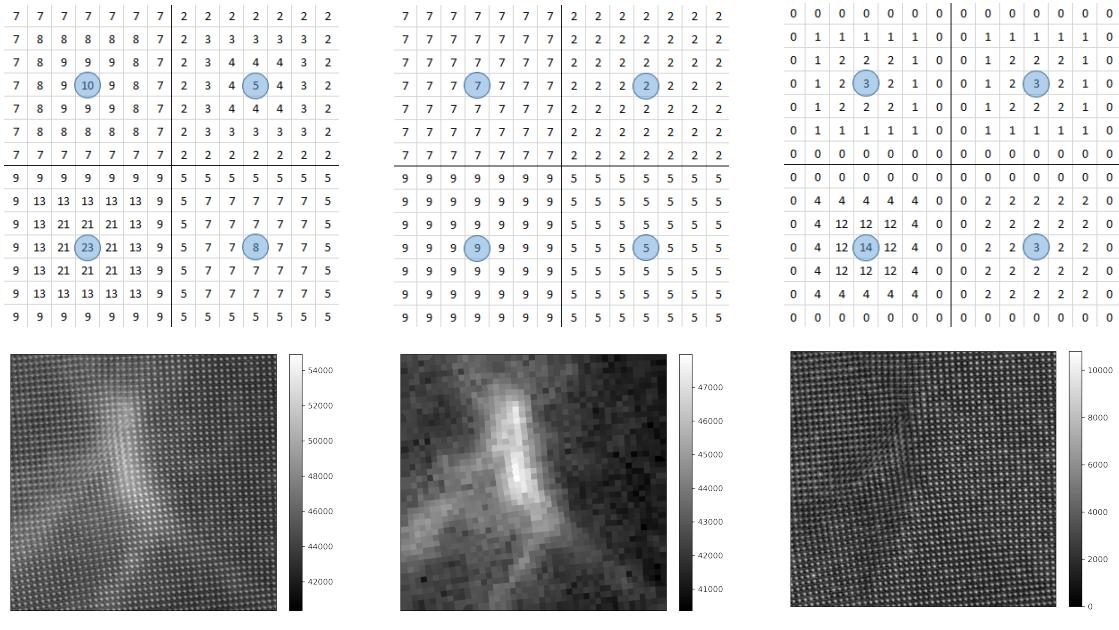


Figure 3.12: Images of the (a) raw, (b) local background, and (c) subtraction of raw - background image data for each Pb atom in PbTiO_3 . The top and bottom rows show a basic example with pixel intensity values and an experimental example, respectively. This data is analysed further in Figure 6.14.

in which these nearest pixels are set to True, with all other pixels set to False. Second, the smallest intensity value in this True region is found. For this atom in Figure 3.12, the lowest intensity value is 7. This value is multiplied by the Boolean array, and repeated across the image. This outputs Figure 3.12(b). This results in a minimum intensity array, in which the nearest pixels to the atom are the lowest intensity calculated around that atom (7 for the top-left atom position). This is the minimum intensity Voronoi (or ‘cell’) image. Finally, this minimum intensity array is subtracted from the raw data, which results in locally background subtracted pixel values around the atom in Figure 3.12(c). In the case of the top-left atom position, the peak intensity is now $10-7=3$. Looping over all atoms in the image, a locally background subtracted image is calculated.

The experimental data in the bottom row of Figure 3.12 is an atomic resolution image of PbTiO_3 [subsection 6.2.1(PTO Domain Wall Junctions)], and mirrors the background subtraction process described above. The reduction operation used on the array during this process is the minimum intensity (`numpy.min`). However, any operation can be used, such as summing (used in Atomap’s `integrate`), or max, mean, median etc. This function can be applied to any atomic resolution image.

The `remove_local_background` function removes the local minimum intensity

from 1 or more nearest neighbour atom positions of a separate, background sublattice. The `remove_average_background` function calculates the average of the minimum intensity of a separate, background sublattice. When these functions are used in within `get_sublattice_intensity`, the returned intensity list is changed accordingly.

3.4.4 Line Intensity Profile

Line intensity profiles are a quick and tangible way to view and compare intensities in atomic resolution images. The function `compare_images_line_profile_one_image` was designed to plot and compare two line profiles from one image. An example of its use is shown in Figure 6.13 in subsection 6.2.1(PTO Domain Wall Junctions).

The function `compare_images_line_profile_two_images` was designed to plot and compare two line profiles taken from two images (which must be of the same size). An example of its use is shown in Figure 6.14 in subsection 6.2.1(PTO Domain Wall Junctions). These functions are in `signal_plotting.py` of the TEMUL toolkit [13]. The `reduce_func` parameter allows the user to choose a reduction function, such as minimum, maximum, mean etc., and therefore change the resulting line profile. This capability was designed and implemented in scikit-image (skimage) [393, 394].

3.4.5 Nanoparticle Identification

To analyse the properties of nanoparticles (NPs) on various two dimensional (2D) substrates, we utilised the program Ilastik [395, 396]. Ilastik is an easy-to-use and install interactive image classification, segmentation and analysis toolkit. It uses pixel classification, object classification, counting and tracking, as well tracking with machine learning. Simply dragging and drawing with the computer mouse allows one to choose the foreground (NPs) and background (2D/carbon). The program provides a step by step guide through thresholding and batch image analysis for quick analysis of multiple images.

Figure 3.13 shows the Ilastik program window for the ‘Pixel Classification + Object Classification’ method. One loads the data using ‘Input Data’, and chooses the size and type of features expected with ‘Feature Selection’. ‘Training’ and ‘Thresholding’ allows the user to draw and mark the regions of interest mentioned above. ‘Object Feature Selection’ allows the user to choose what features of the segmented and trained image to export, such as feature diameter, pixel area, circumference. Then, the various labels for the data, such as ‘dimer NP’, ‘cluster’, ‘background’, are chosen within the ‘Object Classification’ step. ‘Object Informa-

tion Export' allows the user to choose the format with which to export the images and classified data. 'Batch Processing' is used to apply the same steps to any other images. Larger-than-RAM (very large) datasets can be handled in the 'Blockwise Object Classification' step.

3.4.6 Model Visualisation and Data Formats

The free-to-use Vesta program [397] was employed to visualise atomic models created from atomic resolution STEM data. This tool can view crystallographic information files (.cif) as well as many other formats (.vesta, .xyz). Additionally, one can output high quality renders of the models for presentation and reports Figure 3.14. The second program used to visualise atomic models was the Atomic Simulation Environment (ASE) Python package [398], as shown in Figure 3.14. Easily viewing models using Python is vital for checking the progress of the element refinement described in subsection 5.1.1(Refinement of Atomic Resolution Images via Image Simulation), as it can quickly find the source of a simulation error, such as a mis-assigned element. ASE can also open many file formats. The ASE's main use, however, is molecular dynamics (MD) and density functional theory (DFT) calculations.

Throughout the project, various file formats were required by different programs. For example, Prismatic's input format is .xyz, similar to that used originally by Kirkland in the simulation program TEMSIM [399]. We have designed several easy-to-use functions to change between the file formats used during the project. The functions `create_dataframe_for_cif` and `write_cif_from_dataframe` combined will produce a .cif file from a list of sublattice object (Atomap).

`create_dataframe_for_xyz` will create a .xyz file from a list of sublattices that can be opened in Prismatic. `convert_vesta_xyz_to_prismatic_xyz` will change a Vesta style .xyz file format into a Prismatic .xyz file. The functions were used throughout the project and can be found on the TEMUL toolkit GitHub [13].

The function `plot_atom_energies` visualises the assigned energies of atom positions in a list of sublattices. For example, if an Mo vacancy occurs in monolayer MoS₂, the energy of the system rises above its pristine energy state. This can be quantified and stored using the equations and functions described in this methodology section with the DFT calculated formation energies for vacancies shown in Figure 5.13. The 'energy map' plotted with `plot_atom_energies` represents these energy values as either a contour map or filled contour map.

3.4. Signal and Data Analysis

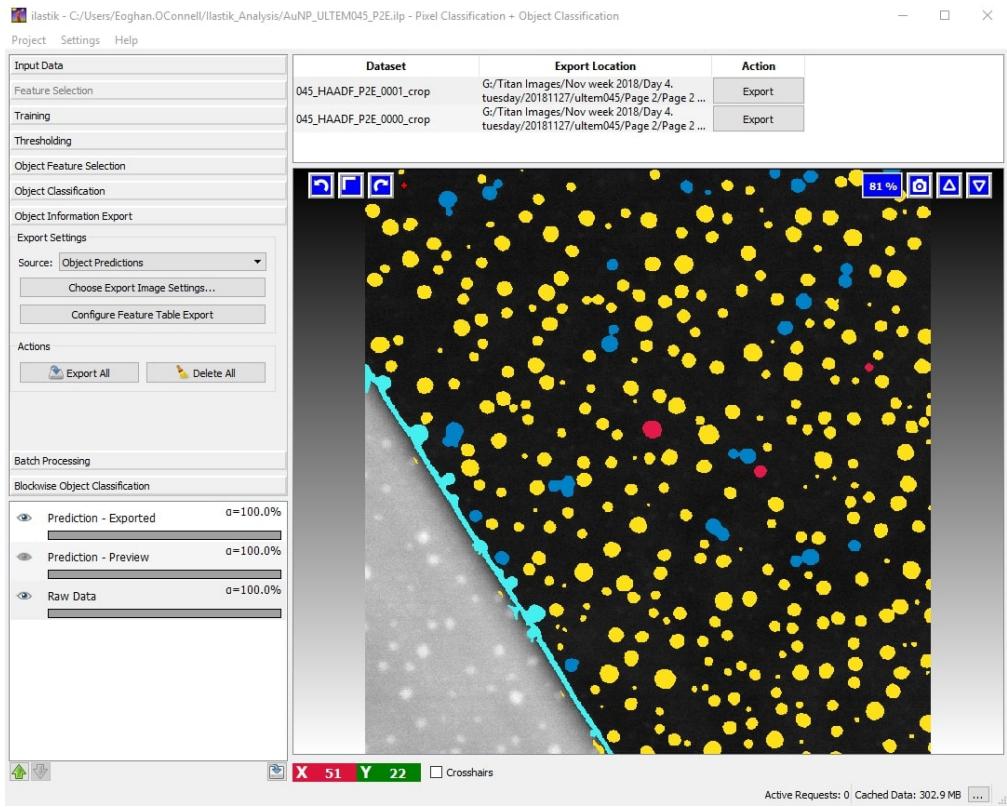


Figure 3.13: Pixel Classification + Object Classification step-by-step method used in the program Ilastik to analyse NPs on 2D materials.

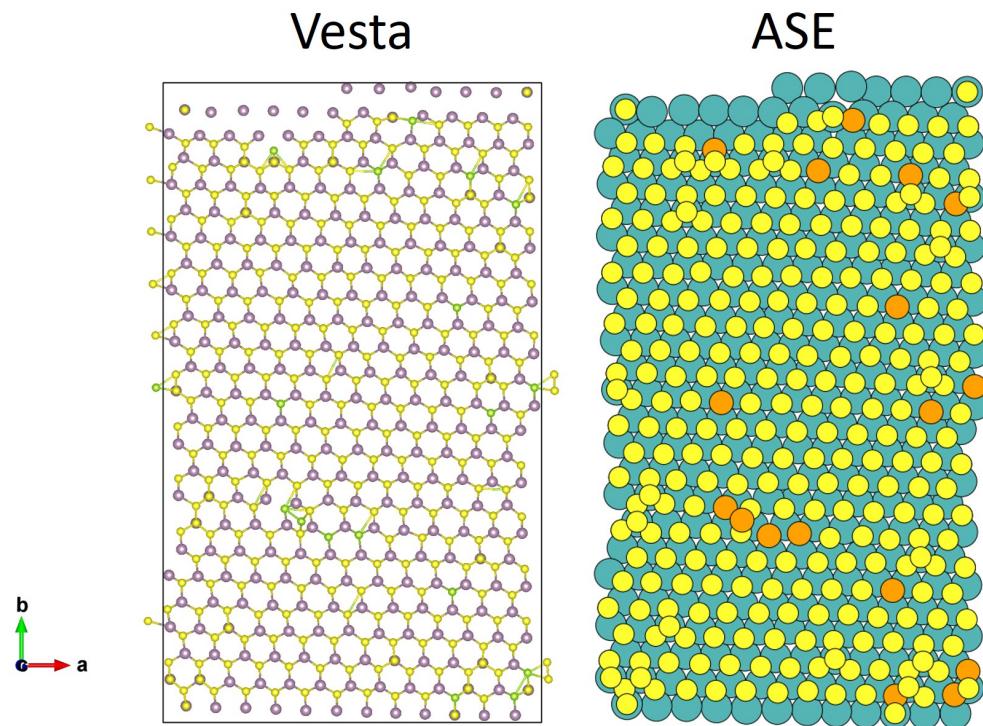


Figure 3.14: An atomic model created from HR STEM data visualised in Vesta (left) and ASE (right).

3.4.7 Methods for Analysing Atomic Movement and Polarisation

For atomic resolution structure investigations during this project, several methods were developed to aid automated analysis of domains, domain walls (DWs), and atom column movement. The methodology behind each development is given in the following sections, described in detail in Subsection 6.1(Tools for Visualisation of Subatomic Shifts) and implemented in Chapter 6(Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls).

Polarisation & Atomic Movement

The movement of atoms and atomic columns is measured as compared to some ‘ideal’ reference position. The ideal position is defined as those set by either a theoretical unit cell, a reference sublattice whose atomic positions move due to random thermal motion, or a curve fit to symmetry in the lattice. The ‘actual’ atom positions are defined as described in Subsection 3.4.3(Initial Atom Position Finding) and Subsection 3.4.3(Atom Position Refinement).

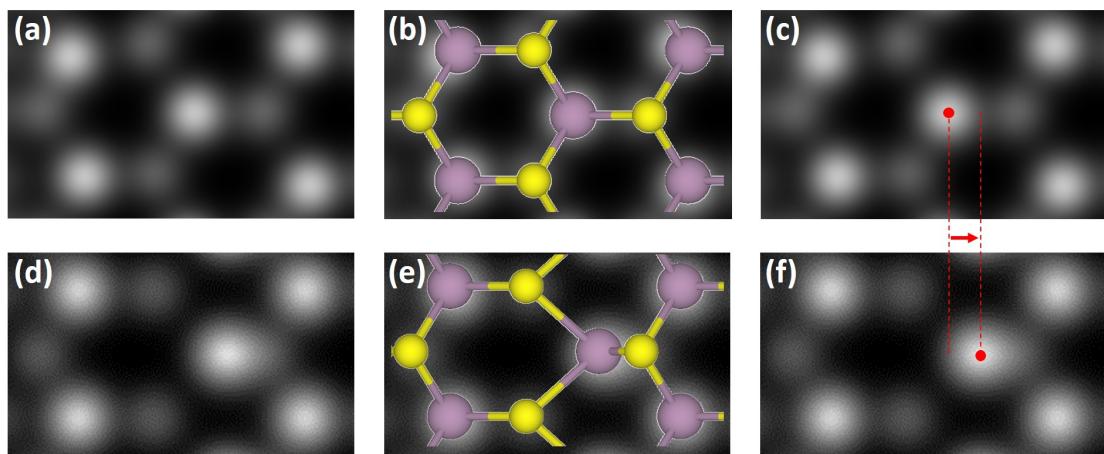


Figure 3.15: ADF STEM simulation of ideal (a) and perturbed (b) atomic columns in an MoS_2 monolayer. (b) and (e) show the simulated input model overlaid on the simulation. (c) and (f) show the same atomic position in each model marked as a red dot. The measured difference between their positions is the atomic movement, represented as the vector arrow.

The ideal unit cell dimensions for an ADF STEM simulation of an MoS_2 monolayer are displayed in Figure 3.15(a), with an exaggerated Mo atom shift in Figure 3.15(d). The ball and stick models (rendered in Vesta) are displayed in Figure 3.15(b) and Figure 3.15(e), and show the extent of the exaggerated atom shift. The atom shift distance is shown by the dashed lines and vector arrow in Figure 3.15(c) and Figure 3.15(f).

For the ferroelectric materials investigated during this project, measuring the polarisation of atomic columns is vital. For certain materials it is taken as the opposite of the atomic shift. An example of polarisation is shown in Figure 2.16(d) in Subsection 2.3(Ferroelectric Materials and Domain Walls). Structures that are more complex require annular bright field (ABF) imaging to resolve the movement of lower Z elements, such as oxygen. Within such structures, polarisation is a combination of several atomic position movements and charges. For a 3D structure, polarisation is defined as:

$$P = P_0 + n \frac{ea}{V}, \quad (3.6)$$

where P is the overall polarisation, P_0 is multivalued and depends on the unit cell definition, n is an integer, e is the electric charge, a is the lattice vector and V is the unit cell volume [400]. For strained ferroelectric materials, such as improper Boracites or PTO DW junctions, the shearing or tetragonality of the surrounding lattice effects the measurement of polarisation and should therefore be taken into account appropriately.

Strain

The fractional change in an object's length under tensile stress is the tensile strain, defined by:

$$\varepsilon = \frac{\Delta l}{l}, \quad (3.7)$$

where l is the original length of the object and Δl is the change in original length. In Chapter 6(Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls), the strain is used to characterise a domain wall (DW) junction in PbTiO_3 , and is implemented in the `get_strain_map` function of the `polarisation.py` module. This function calculates the distance between each atom and its neighbouring atom planes. Therefore, it calculates l in Equation 3.7 above. Changes to the expected l are measured (Δl) and displayed as a fraction or percentage of l , i.e., the strain ε . The function parameters and uses are described in detail in subsection 6.1.7(Material Property Maps: Strain, Angle of Zone and Lattice Spacing Ratio).

Atomic Position Deviation from Zone Axes

The method designed to calculate and visualise atomic position deviation from structural translation symmetries (zone axes) are discussed in the following subsections. The full function (`atom_deviations_from_straight_line_fit`) can be seen in the TEMUL toolkit [13]. This method is based on the polarisation mapping by Gonnissen *et al* [337], though there are some minor differences. The method quantifies the movement of atoms from certain zone axes to display the polarisation near a DW.

Initially, the atom position coordinates are found with Atomap, as described in subsection 3.4.3(Methods for Atom Position Analysis). An example structure with these atom positions overlaid is shown in Figure 3.16(a). The main translation symmetries are constructed and visualised with the `construct_zone_axes` and `plot_planes` functions, respectively. A single zone axis direction is plotted in Figure 3.16(b), with the atom planes 0-3 labelled. In Figure 3.16(c), the atoms in atom plane 3 are highlighted and labelled Atom 3.0 - 3.4. At this point, one has access to the atom coordinate values along each atom plane in each zone axis of the sublattice. Therefore, linear regression can be applied to the atom coordinates in each atom plane.

The exaggerated graphic in Figure 3.17 is used to describe how the above atom planes can aid the understanding of polarisation in structures in which atoms deviate from their own atom plane. Here, the atoms (black dots) move up and down in the left and right domains, respectively. As described above for Figure 3.16, a straight line (green) is fit to these atomic coordinates along a zone axes direction. The perpendicular vectors from each atomic coordinate to this straight line fit are calculated. Once the original atomic coordinates and the vector components

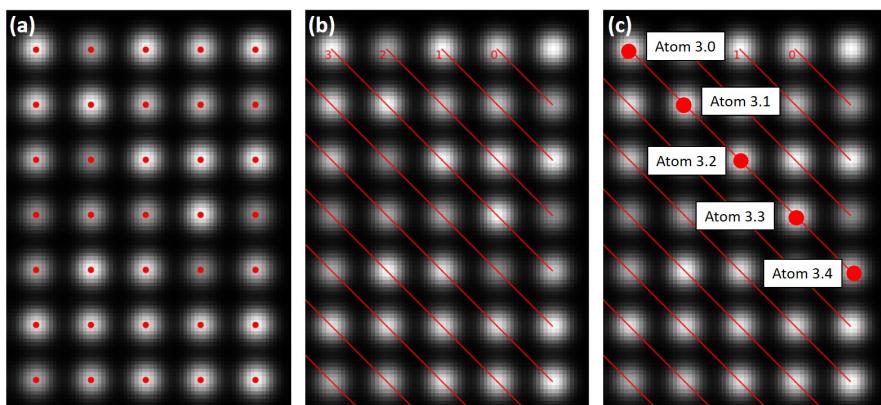


Figure 3.16: (a) Atom positions (red) overlaid on a test dataset. (b) Atom planes from an example zone axis constructed from the atom positions. The 0-3 atom planes are labelled. (c) Atom positions highlighted along an atom plane, labelled 3.0-3.4.

are known, the `plot_polarisation_vectors` function can be used to visualise the resulting vectors. In Figure 3.17 these vectors represent the opposite of the atomic shift of each atom. Depending on the material, this can be the ferroelectric polarisation. For example, for DWs in LiNbO₃ examined in ??, the polarisation is defined as the reverse of the atom shift.

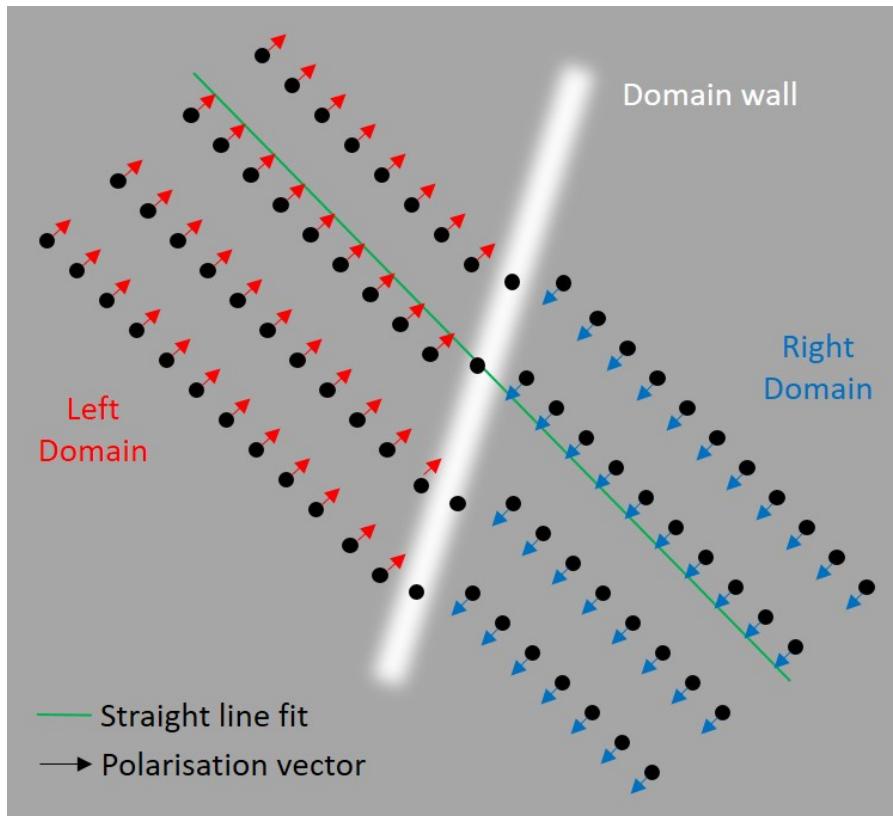


Figure 3.17: A graphical example of polarisation across a single domain wall in an atomic lattice. The domain wall is represented by the white rectangular band. A best straight line fit along a zone axis is shown by the green line. In the left domain the atom positions are shifted down from this straight line fit, and are therefore polarised in the upward direction. Vice versa for the right domain.

This process is calculated within the `atom_deviation_from_straight_line_fit` function, and plotted with `plot_polarisation_vectors`. Every zone axis can be easily calculated and plotted with `plot_atom_deviation_from_all_zone_axes`. Using several lines of example code in Code Segment 3.2 below, one can use and visualise the results of the functions. Application of these functions to ferroelectric materials can be seen in Subsection 6.1(Tools for Visualisation of Subatomic Shifts).

```

1 # Import the necessary packages and functions
2 import atomap.api as am
3 from temul.polarisation import
    atom_deviation_from_straight_line_fit

```

```
4
5 # Create a sublattice
6 atom_lattice = am.dummy_data.get_polarization_film_atom_lattice()
7 sublatticeA = atom_lattice.sublattice_list[0]
8 sublatticeA.find_nearest_neighbors()
9 sublatticeA.refine_atom_positions_using_center_of_mass()
10 sublatticeA.construct_zone_axes()
11
12 # find the atom deviation from the first zone axis
13 x,y,u,v = atom_deviation_from_straight_line_fit(sublatticeA,
14         save=None)
15
16 # find the atom deviations from every zone axis!
17 plot_atom_deviation_from_all_zone_axes(sublatticeA,
18         plot_style='vector', save=None)
```

Code Segment 3.2: Example code for analysing the atomic deviation from a straight line fit (see Figure 3.17) in the first zone axis and the vector addition of several zone axes.

Multiple Atom Position Deviations

In certain cases it is desirable to combine atom position deviations from multiple zone axes. This is completed by simple vector addition. The only difficulty encountered in this approach is that for each zone axis the atomic positions are either not listed in the same order, or the same atom position coordinates are not found in each zone axis. The latter part is visualised in the previous subsection in Figure 3.16(c), wherein the top right atom has not been included in the zone axis. This issue is overcome by the function `combine_atom_deviations_from_zone_axes`, which calls `atom_deviation_from_straight_line_fit`. The outputted arrays are combined by linking the same atom position coordinates from each zone axis, and subsequently adding their respective atom position vectors. Atoms on the edges of images, especially corners, may only have one atom deviation vector, and therefore should not be included when forming conclusions about the properties of the material.

Figure 3.18(a) presents a simplified example of this vector addition. The atom deviation, as described above, is calculated for each zone axis (zones 1-4 shown here). Below each atom deviation model, example components (u , v) of the atom shifts are shown. The addition of the atom deviation from each zone results in the ‘Vector Addition’, wherein a new ‘ideal’ atom position is defined. Repeating this process for all atoms in the sublattice, the polarisation of each atom from its position in an approximated ideal version of the material’s sublattice is produced. An example is shown in Figure 3.18(b), the magnitudes displayed in a contour

3.4. Signal and Data Analysis

map. There are several other methods for displaying this and all vector arrow data with `plot_polarisation_vectors`, as described in Subsection 6.1(Tools for Visualisation of Subatomic Shifts).

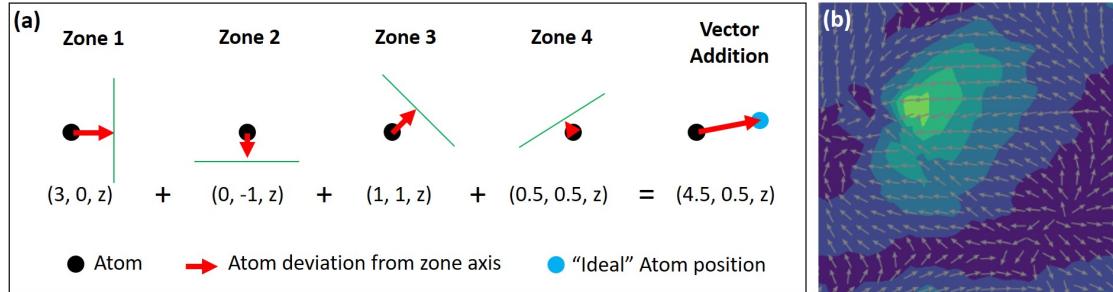


Figure 3.18: (a) Atom deviation from several zone axes, each labelled with example (u, v) components. These components are added to form ‘Vector Addition’: the final combined atom deviation from the ‘ideal’ position. (b) An experimental example of the process of vector addition from (a). The vectors shown as unit vectors, with the magnitude displayed in a contour map, to properly present the turning of the structure.

3.4.8 Multidimensional Data Analysis

As described in Subsection 2.1.1(EELS) and Subsection 3.3(Experimental and Simulation Methods), EELS can be used to detect elements in thin samples to great effect. In this section, the analysis process for EELS datasets is described.

For core-loss (CL) EELS datasets, the inelastic background noise must be removed to accurately analyse ionisation edges of lattice atoms. This is generally done by fitting a decaying power law immediately prior to the relevant edge onset [125, 151]. The EELS signal is then integrated over 30-50 eV starting at the element edge onset to create chemical maps. For component analysis, PCA denoising must be applied to the signal before ICA can be utilised. The methods used to garner information from low signal-to-noise EELS data via component analysis are described below in Subsection 3.4.8, as well as in the Literature Review in Subsection 2.1.1(EELS).

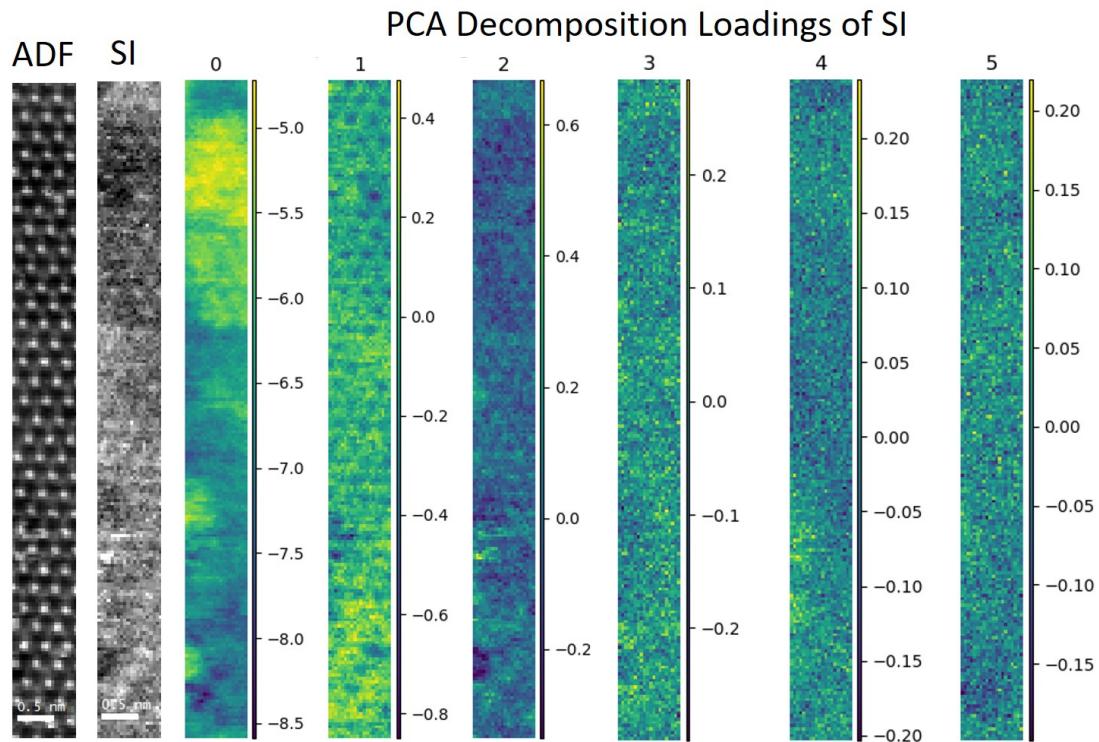
Brief Component Analysis Walkthrough

The visual process of PCA decomposition and denoising, and subsequent FastICA decomposition, is presented in two figures: Figure 3.19(PCA) and Figure 3.20(ICA). For details on the component analysis methods themselves see Subsection 2.1.1(EELS) or the excellent book written by the developers of FastICA [401]. First, the raw spectrum image (SI) data is decomposed with PCA into its component loadings in Figure 3.19(a) and factors in Figure 3.19(b). Component loadings 1 and 2 have some structure relating to the ADF image shown in Figure 3.19(a). However, there are no obvious peaks around the possible Se implant sites. Component 0 is likely the inverse of the background carbon signal [150], while components 3, 4, and 5 do not have clear structure. Figure 3.19(b) displays the relative intensity of the energy loss for each component, with the numbered lines corresponding to the respective map numbers in Figure 3.19(a). The scree plot in (b)(i) inset describes at a glance the amount of information (variance) contained in each component. The red triangles indicate the five components chosen to construct the denoised SI presented in Figure 3.20 which was then used for FastICA. Inset in (b)(ii) is the 40-100 eV energy range of the PCA factors.

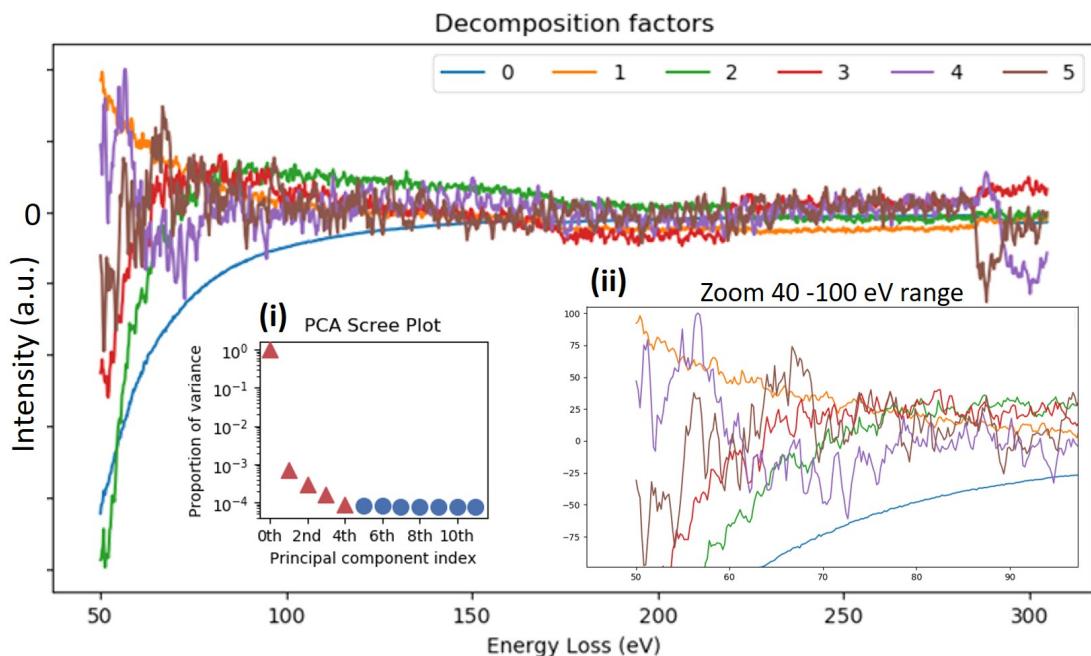
The FastICA algorithm is then applied to the PCA denoised SI shown in Figure 3.20 to produce five FastICA component loadings 0-4. Component 0 is likely the signal due to hydrocarbon contamination which you can see is the inverse of the above PCA component 0. Component 1 has more localised signal in areas shown in more detail in Figure 4.26 in subsection 4.1.2(Spectroscopy of Se). Component 2 seems to be a mix of both Mo and S maps, as it shows a lack in intensity at the

3.4. Signal and Data Analysis

centre of the MoS₂ hexagon. Alternatively, it may be the inverse of the Mo signal. Components 3 and 4 are likely background noise.



(a) PCA decomposition loadings (maps). Respective factors shown below.



(b) PCA decomposition factors (signals). Respective loadings shown above.

Figure 3.19: The first 6 PCA decomposition (a) loadings (maps) and (b) factors (signals) of the spectrum image (SI) in (a). Inset (b)(i) is a PCA scree plot, also known as a variance plot, showing the first 5 components (triangles) used to produce the final PCA denoised SI. Inset (ii) is a zoom of the 40-100 eV energy range of the decomposition factors.

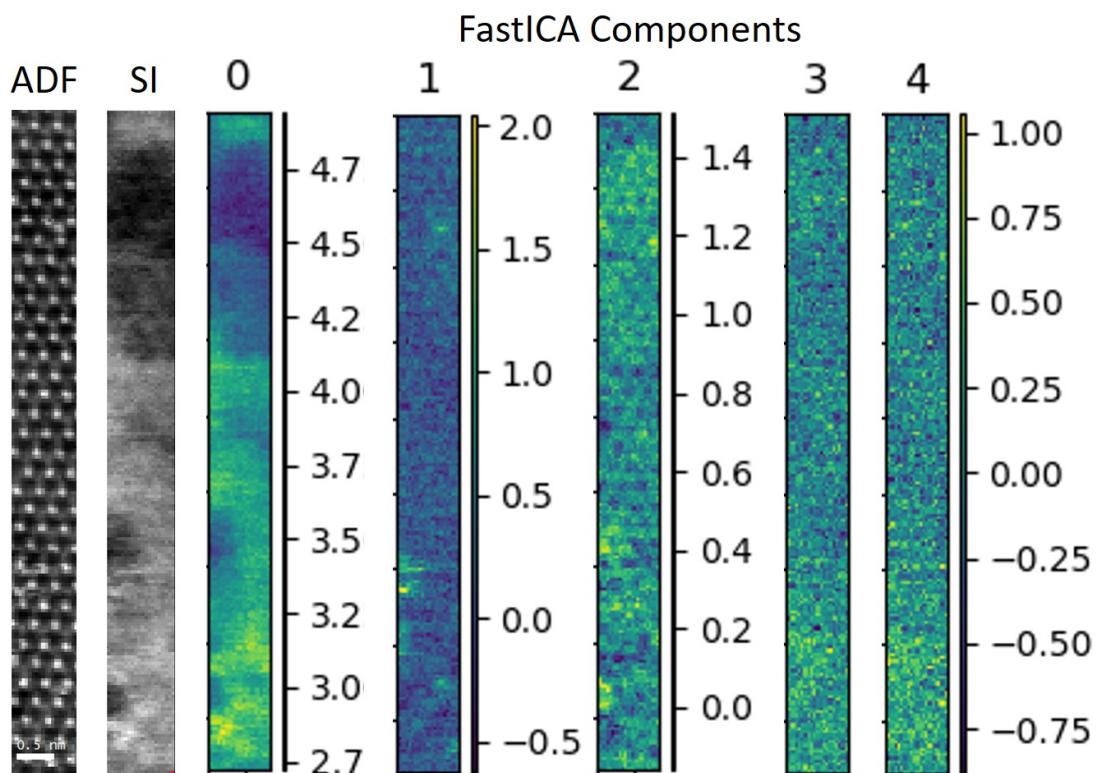


Figure 3.20: Experimentally acquired ADF image and PCA denoised spectrum image (SI) of a region of monolayer Se implanted MoS₂ split into its five FastICA components from the PCA components in Figure 3.19.

3.4.9 Analysis Tools

In the electron microscopy (EM) sphere of data and signal analysis, there are many proprietary and open-source tools available. Gatan Microscopy Suite (Gatan), Velox (FEI/Thermofischer Scientific), SmartAlign (HREM Research), Python, ImageJ and Ilastik, to name a few, have been utilised during this project to varying extents. In this section we briefly describe both the general uses and specific use cases of each analysis tool, as well as its pros and cons. During this project, we used the tools for the analyses indicated in Table 3.1 for their respective uses. All programs and packages used for analysis are listed in Chapter (List of Programs).

Table 3.1: Each main toolkit and program used throughout the project, with their respective use cases and frequency of use.

Toolkit/Program	Used to Analyse	Frequency of Use
Python	Line Profiles, Image Comparisons, Atom Intensity & Tracking Analysis, EELS & PCA/ICA Analysis, File Conversion, Data Manipulation	Daily
GMS	EELS Analysis, Line Profile, X-ray spike removal, Image Comparison	Weekly
ImageJ/Fiji	Line Profiles, Image Comparison, File Conversion	Monthly
Velox	Image Alignment, EDX Analysis	Monthly
Ilastik	Particle Analysis, Intensity Segmentation	Bi-Monthly
SmartAlign	Image Alignment	Once

Due to the near omnipresence of Gatan hardware in most high-resolution electron microscopes, the GMS tool is widely used. It continues to be a reliable and wide ranging toolkit; one can create images and script advanced macros, and with external scripts available online [402, 403], the toolkit is quite extensible. One can quantify multidimensional (e.g., EELS) datasets and align tomographic data in an intuitive manner. Overall, GMS was vital throughout the project. With regard to its downsides as a toolkit, GMS requires an often prohibitively expensive licence. It can feel clunky and is glitchy during use on the microscope. Its analysis routines, while probably reliable, often don't describe the exact method implemented during calculation. This is the case for newer methods especially, such as

its 4DSTEM tools [404]. Its ‘.dm’ file format is not open source and therefore one must either lose metadata when converting to other formats, or use an open-source toolkit such as Python’s HyperSpy and hope it can open the GMS file. Its graphical interface, although quick and easy to use, is dated and not set-up to output publication-ready figures. Additionally, one can script in the native GMS scripting code, though it has very little documentation and therefore isn’t very user friendly. This is changing with modern iterations of GMS, as a Python scripting tool has been introduced.

The FEI/Thermofischer Scientific software Velox has a cleaner interface than GMS, and is less prone to bugs during microscope imaging. Imaging with multiple detectors is intuitive, as is EDX acquisition and analysis. The image registration tool is easy-to-use, and the program offers a registration specifically for periodic (atomic resolution) images. However, as it is younger than GMS, it is much more limited in its scope; only the above-mentioned methods are available. Like GMS it is proprietary, and licences are expensive. Additionally, the ‘.emd’ format is not open-source (though the Velox developers are helpful and active in the open-source community). Overall, Velox is an intuitive, but unfortunately expensive, program.

The programming language Python was used daily throughout this project. Overall, Python is well documented, easy to understand and infinitely extensible, and many excellent open-source scientific packages are available. The most commonly used Python packages were HyperSpy (Subsection 3.4.10), Atomap (Subsection 3.4.10), Numpy, Scipy, Matplotlib, Pandas, Scikit-Image, Scikit-Learn (Subsection 3.4.10), Atomic Simulation Environment (Subsection 3.4.10), Rigid registration (Subsection 3.4.10) and PyPrismatic (Subsection 3.4.10), described in their respective sections. All of these packages are open-source and anyone can contribute to their development. As an example, we have developed tools for the Atomap and Sci-kit Image (skimage) packages during this project e.g., [394, 405], among others. Below the pros and cons of HyperSpy and Atomap are discussed.

These packages have been utilised intensely in the previous decade. With regard to EM-specific packages, HyperSpy is the figurative door opener for many researchers, as it allows one to open (most) GMS ‘.dm’ and Velox ‘.emd’ file formats, among others. HyperSpy is built on the fundamental Numpy Python package and is designed for intuitive multidimensional manipulation. One can visualise EELS data in EFTEM-like movie slice, carry out image registration, and quantify chemical (EELS etc.) data, to name but a few. The plotting toolkit used is the Matplotlib package, which allows users to plot with publication-ready quality and infinite customisation. Hyperspy, like all Python toolkits mentioned, is free to use and open-source. Hyperspy’s user and developer-base is ever growing, with a Gitter

3.4. Signal and Data Analysis

chatroom available to anyone with questions or issues. Regarding its downsides, it requires coding skills, though as stressed above, Python is a beginner-friendly programming language. Even so, it can be daunting to start with these programs. HyperSpy has a graphical user interface (GUI) similar to GMS, though it is not maintained at the speed of general code development. The usability is generally limited to that developed by users requiring said usability and because it is an open-source, free project, there are no paid maintainers, only devoted ones.

Atomap was used extensively throughout this project, and new tools were developed for it [405], available to other users. Atomap is built on the data structures of Hyperspy, and can be used for the analysis of atomic resolution images by identifying bright spots in periodic images. As described in subsection 3.4.3(Atom Position Analysis), one can easily refine the atom positions with sub-pixel accuracy, and analyse strain, polarisation, and atom intensity, to name a few [4]. However, like Hyperspy, it requires coding skills and, of course, on zone atomic resolution data for reliable results. Additionally, Atomap has no GUI. We believe aspects of it will be developed further in the coming years, such as atom intensity and quantification analysis tools which are currently under development [406].

Ilastik is a tool for the segmentation of multidimensional image data. It is intuitive and has an excellent GUI. Its use is described in subsection 3.4.5. It was employed during this project for the analysis of Au nanoparticles on 2D transition metal dichalcogenides (TMDs) as shown in Subsection 4.2(sec:Au Ion Implantation on TMDs). ImageJ is a lightweight and extensible image analysis toolkit. In general, ImageJ was used to open certain image formats such as ‘.tiff’, and to plot line intensity profiles. Like GMS, the scripting language used by ImageJ is not well documented. However, it is much more widely used, especially by the biological EM sphere [407].

3.4.10 Python and its EM Packages

Python is an open-source, easy-to-read programming language designed for scientific analysis. Basic scripting makes it easy to learn for beginners, and is extensible for advanced users through its object-oriented structure and comprehensive documentation in packages such as numeric Python (NumPy), scientific Python (SciPy), and the Matlab-like plotting package Matplotlib. Additionally, Jupyter Notebooks, a tool for sharing scientific analysis workflows easily through a browser [408], are now widespread in the scientific community. Perhaps the best example is the use of Jupyter Notebooks to share the Nobel Prize winning work of Abbott *et al*, wherein they display the characteristic ‘chirp’ of a detected gravitational wave for all to see and hear [409, 410].

In the field of electron microscopy, de la Peña *et al* have developed Hyperspy, a package for multidimensional data analysis [2, 69]. Packages for various aspects of electron microscopy analysis, from atom intensity analysis (Atomap [4]) to fast image simulation (PyPrismatic [11, 12]) have also been developed. Savitzky *et al* developed an image alignment routine for low signal cryo-EM data (Rigid Registration [387]), while Larsen *et al* created the Atomic Simulation Environment (ASE) for creating atomic models and calculating energetics of resulting systems (ASE [398]). Linking these packages together is trivial within a Python IDE (Integrated Development Environment) or Jupyter Notebook [9]. Below, a brief overview of a common (and generic) Python IDE is given, followed by brief descriptions of prominent Python packages used throughout this project. Then, a summary of the functions in the TEMUL toolkit developed during this thesis is given.

A short example of Python code linking HyperSpy and Atomap is shown in Figure 3.21(Python code). Python code is written and stored in a .py script (a text file) in the editor (solid blue box) in the top left of Figure 3.21(a). One can choose to execute (run) a single line of code, a block of code, or the entire script. This provides excellent trial-and-error testing. The output is shown in the iPython console labelled by the dashed red box in Figure 3.21(a). Any variables, such as lists or arrays, are displayed in the dotted green box in Figure 3.21(a). This run, evaluate, print, loop (REPL) routine promotes quick inspection of processed data. In this script, the necessary packages (modules) are imported, such as HyperSpy and Atomap, and a plot of an example perovskite structure containing vacancies in Figure 3.21(b) is created.

3.4. Signal and Data Analysis

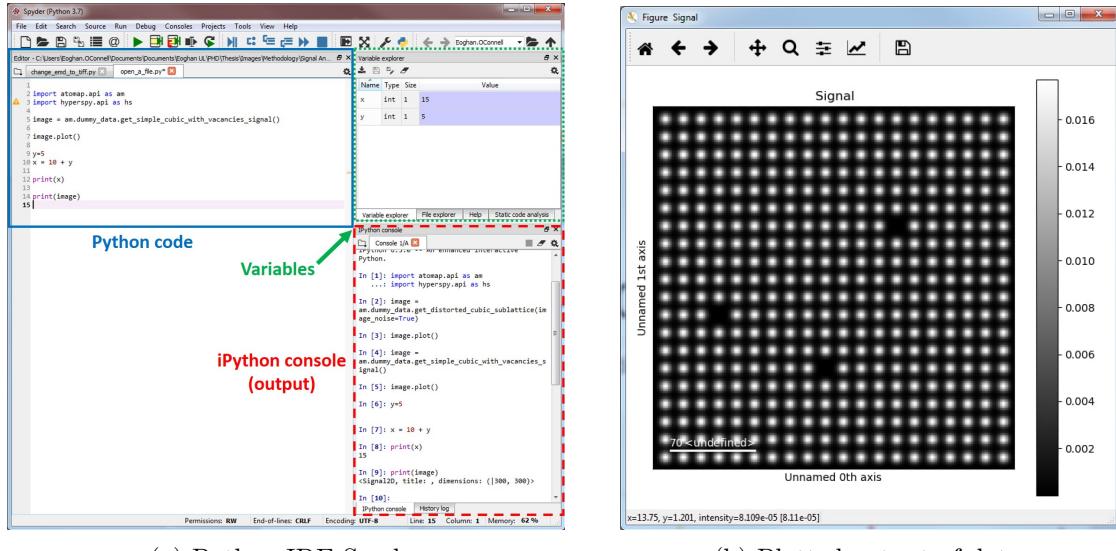


Figure 3.21: (a) Python IDE Spyder, with Python code (solid blue), variable explorer (dotted green), and iPython output console (dashed blue). The ‘image.plot()’ of the dummy data is output in (b).

HyperSpy

Hyperspy, initially called EELSLab, was developed to analyse multi-dimensional data by Francisco de la Peña during his doctoral studies [2, 69] and has been expanded since by many researchers (see hyperspy.org). For this work, HyperSpy was used to open proprietary software, such as Gatan’s .dm3 and FEI’s .emd file formats. Additionally, it was also utilised extensively for image and EELS calibration, filtering, plotting, and denoising.

Atomap

Atomap, developed by Nord *et al.*, is a tool for finding and analysing bright objects in an image [4]. It is built upon the data structure of HyperSpy and has been invaluable during this study. It was designed for the analysis of atomic columns in high-resolution TEM images. It provides an object-oriented structure, with atom positions making up sublattices, which in turn make up atom lattices. For an example structure, such as monolayer MoS₂, an atom position would represent a S or Mo column. All Mo atoms make up the Mo sublattice. The overall atom lattice is made up of a Mo sublattice and a S sublattice. The vast majority of original code developed during this project is centered around Atomap and the manipulation of the program’s structure. This code is given in Subsection 3.4(Signal and Data Analysis), Subsection 5.1(Open Source Development of Automated Atomic Resolution Image Series Analysis Methods) and in Subsection 6.1(Tools for Visu-

alisation of Subatomic Shifts) and can be found on the TEMUL toolkit GitHub [13].

Atomic Simulation Environment (ASE)

The Atomic Simulation Environment (ASE) is a package designed for the generalisation of various molecular dynamic (MD) and density functional theory (DFT) codes into the Python language [398]. One can use many MD and DFT codes directly through Python with ASE, such as Gromacs, Vasp, GPaw and Castep. It also acts as a kind of ‘babel’ program which can read from and write to many file formats. It was mostly used for visualisation of the models created from experimental data.

Image Simulation

Prismatic, and its Python implementation PyPrismatic, is a fast image simulation software developed by Ophus and Pryor [11, 12]. Although it includes a standard multislice algorithm for image simulation, the PRISM algorithm allows 1x1k images to be simulated in a number of minutes, rather than hours. We implemented fast simulation of experimentally defined models to refine said models iteratively, as described in Chapter 5(Atom Dynamics in 2D TMDs).

Image Registration

The Rigid Registration package developed by [387] was utilised to align and track the misalignment of movies. The package has clear, beginner friendly Jupyter Notebooks available for any new user. Hyperspy’s image alignment tool `align2D` uses the scikit-image package for estimating the misalignment between images [388]. A multidimensional alignment is also available with `align2D` by using `reference='stat'` [411]. Scikit-image itself can also be used to align images via the `register_translation` function.

Other Python Packages

Base Python functions, as well as core scientific tools, such as NumPy, SciPy, Pandas, Scikit-Image/Learn and Matplotlib were used extensively throughout this project. Other lesser known packages such as periodictable, PyCifRW etc., were also utilised.

TEMUL Toolkit

The TEMUL (Transmission Electron Microscopy University of Limerick) toolkit is a set of python functions and classes for analysis, both manual and automatic, of atomic resolution scanning transmission electron microscopy (STEM) data. The functions in the TEMUL toolkit are built on the HyperSpy and Atomap Python packages, as well as NumPy, SciPy, and Scikit-Image. Here we give a brief overview of the modules (folders) currently available on the TEMUL toolkit GitHub [13]. Functions, methods, classes and modules are presented with **grey backgrounds**.

The `signal_processing.py` module contains many functions for filtering and calibrating images. For example, the functions `double_gaussian_fft_filter` and `compare_two_image_and_create_filtered_image` are used during this thesis to filter images automatically based on the diffraction spots and image contrast, respectively. `get_cell_image` allows the user to create a Voronoi cell image with any intensity aggregation function, and can therefore be utilised for background intensity removal or atomic column intensity integration. The `signal_plotting.py` module includes line profile comparison tools and the `plot_atom_energies` function which produces the energy maps shown in subsection 4.1.1(Microscopy of Se implanted MoS₂). `intensity_tools` and `element_tools` build basic functions used in the high-level functions in `model_creation`, which are streamlined with the Atomap `Sublattice` class data structure. `model_creation.py` contains the `scaling_z_contrast`, intensity sorting and z-coordinate functions for the creation of .cif and .xyz files. The image (`image_difference_intensity`) and position (`image_difference_position`) intensity refinement algorithms described in subsection 5.1.1(Refinement of Atomic Resolution Images via Image Simulation) are also found in the `model_creation.py` module.

The `model_refiner.py` module contains the `Model Refiner` class, described in subsection 5.1.2(Model Refiner Tool), and can be used to automatically refine fast image simulations via the PyPrismatic simulation software. Additionally, it tracks and visualises the quantitative changes made to the image simulation and hence the outputted atomic model. The above simulations are handled by the `simulations.py` module. The PyPrismatic image simulation package is simplified down to several basic functions which include all parameters from the package. One such example is the `simulate_and_filter_and_calibrate_with_prismatic` function, which uses the developed automatic image comparison between experiment and simulation (`compare_two_image_and_create_filtered_image`) to create an auto-filtered and calibrated image simulation. The `Model Refiner` is used in this thesis to analyse dynamics in, and damage to, monolayer MoS₂ in Subsec-

tion 5.2(Atom Dynamics in Implanted TMDs with the MR Tool).

The `polarisation.py` module contains the functions described in Subsection 6.1(Tools for Visualisation of Subatomic Shifts). When the atomic positions are known and refined (with Atomap, for example), functions such as

`plot_polarisation_vectors` and `get_average_polarisation_in_regions` allow the user to effortlessly visualise atomic column shifts and polarisation in atomic resolution images. Strain, lattice spacings and other material properties are accessible via basic functions. This module was used to full effect for the analysis of ferroelectric materials in Subsection 6.2(2D Domain Walls in Ferroelectric Materials).

The functions and classes of the TEMUL toolkit are utilised throughout the thesis for both manual and automatic analysis of atomic resolution STEM data. Many more functions are available in the TEMUL toolkit than above-described and further developments will be added in future.

Chapter 4

Results - Atomic Characterisation of Ion Implanted 2D TMDs

Manipulating the structure and properties of monolayer and few-layer TMDs is currently of great interest. As described in subsection 2.2.2(Manipulating the Properties of 2D TMDs), the possible devices arising from such investigations range from single photon emitters to gas sensors. One such way of changing the properties of 2D TMDs is through low energy ion implantation [see subsection 2.2.3(Ion Implantation of 2D Materials)]. In the past, this has been verified with monolayer graphene, wherein the ion implantation of B and N locally opened the band gap energy, doping the material [19, 20]. As shown for monolayer graphene, the understanding of property manipulation lies on the atomic scale. Therefore, for both Se and Au implanted 2D TMDs, atomic resolution scanning transmission electron microscopy (STEM) and electron energy-loss spectroscopy (EELS) have been employed to understand changes in material structure and properties. The ADONIS implanter in the University of Göttingen was used for implantation. The aim in this chapter is to verify the successful implantation of Se and Au ions into TMDs, and to investigate any change to the pristine material properties.

Se ion implantation is investigated in Subsection 4.1(Se Ion Implantation in MoS_2). First, atomic resolution STEM is used to identify and quantify substitutional and surface Se atoms in and on MoS_2 . Second, chemical and opto-electronic properties are investigated in subsection 4.1.2(Spectroscopy of Se). Here, the implanted ion's chemical signature is confirmed with core-loss (CL) EELS in Subsection 4.1.2. Additionally, changes to the opto-electronic properties of pristine monolayer and few-layer MoS_2 are studied with low-loss (LL) EELS in Subsection 4.1.2. Au ion implantation is investigated in Subsection 4.2. The distribution of Au nanoparticle (NP) formation is studied with STEM, while the shape and size of Au NPs are studied on few-layer MoS_2 and MoSe_2 using image simulation.

4.1 Se Ion Implantation in MoS₂

Figure 4.1(a) shows the cross-sectional and image (top) views of monolayer MoS₂. Single and few-layer MoS₂ was implanted with Selenium ions at 10 and 15 eV with a fluence of $3 \times 10^{15} \text{ cm}^{-2}$ [10]. The below results are taken from the 10 eV ion implantation. This process is shown graphically in Figure 4.1(b), with an example Se₁S₁ atomic column configuration resulting from the ion implantation. The electron beam and ion beam directions are shown in Figure 4.1(a) and Figure 4.1(b), respectively. Various configurations of elements resulted from the implantation process, many of which are shown in Figure 4.2. The models in Figure 4.2 are side (cross-sectional) views of the monolayer as in Figure 4.1(a). The ‘element configurations’ are the probable atomic columns present in the implanted monolayer material. They are investigated at atomic resolution in subsection 4.1.1(Microscopy of Se implanted MoS₂) and subsection 4.1.2(Spectroscopy of Se) with STEM and EELS, respectively.

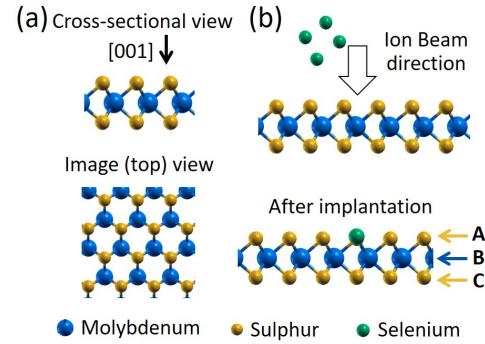


Figure 4.1: (a) Cross-sectional view and image (top) view of monolayer TMD. (b) Before and after ion implantation of a monolayer TMD. The top (**A**) and bottom (**C**) chalcogen sub-layers, and middle TM layer (**B**), are shown.

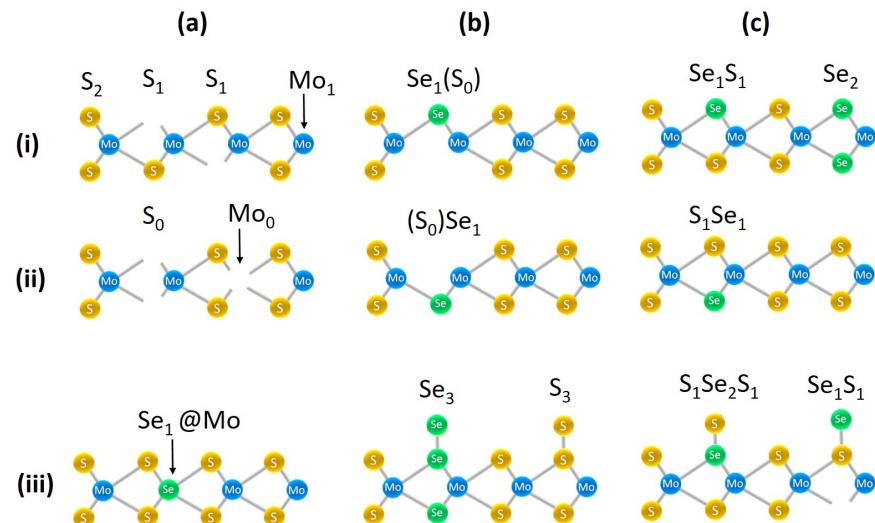


Figure 4.2: Cross-sectional view of various element configurations of the atomic columns in monolayer MoS₂ after low energy Se ion implantation. Not an exhaustive list.

4.1.1 Microscopy of Se implanted MoS₂

An overview of before and after 10 eV Se implantation into monolayer MoS₂ is shown in Figure 4.3. The images in Figure 4.3, and all images in this chapter unless otherwise stated, are high angle annular dark field (HAADF) scanning transmission electron microscopy (STEM) images acquired with the Nion SuperSTEM 3 microscope [see Subsection 3.3(Experimental and Simulation Methods) for details]. The regions in Figure 4.3(a) and Figure 4.3(b) are not the same, though they represent a standard overview before and after implantation. The pristine MoS₂ sample in Figure 4.3(a) has very few defects. After Se implantation, the lattice contains many defects, holes, substitutional and possible ad-atoms. Minimisation of electron beam effects, while retaining useful information is necessary, as described in subsection 2.1.2(Electron-Matter Interactions). The distinction between substitutional and ad-atoms is only possible with a series of images of an area. This is described in detail in section Subsection 5.2(Atom Dynamics in Implanted TMDs with the MR Tool). However, the very bright atoms over transition metal sites are likely candidates for surface adatoms, though they could be heavier substitutional metal atoms.

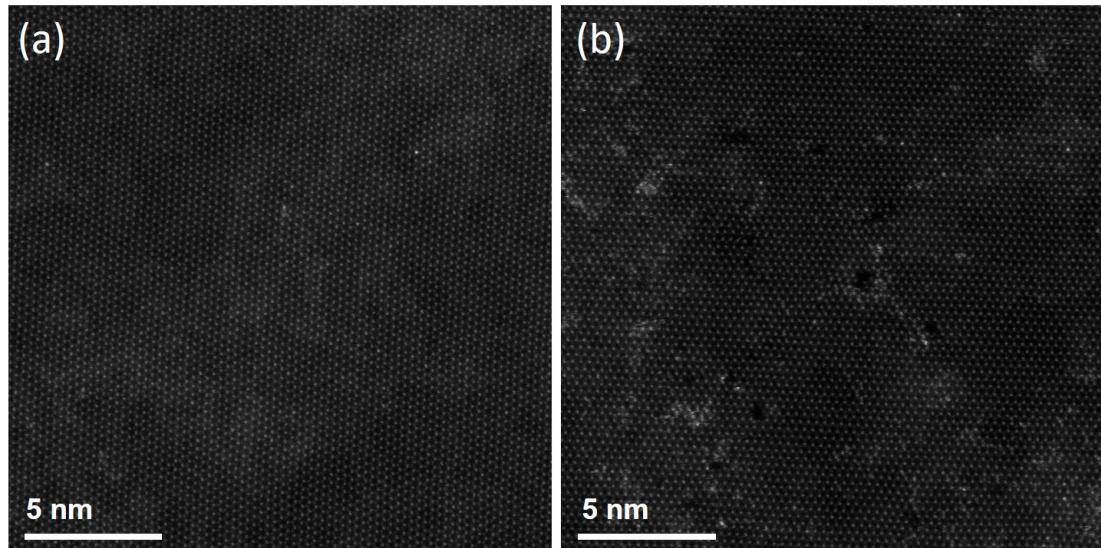


Figure 4.3: Raw HAADF STEM images of monolayer MoS₂ (a) before and (b) after 10 eV Se ion implantation.

In the following sections, the identification and quantification of elements in Se implanted monolayer MoS₂, as well as the effect hydrocarbon contamination may have on the quantification, will be discussed. Following this, individual cases of successful Se implantation as well as the formation of holes and vacancies are investigated. Finally, the overall statistics of automatically and manually analysed datasets will be summarised.

Element Identification in Implanted MoS₂

Initial images, such as Figure 4.3, were analysed both manually and automatically. The success of implantation in terms of the percentage of Se in the chalcogen (S) sublattice, is shown in Table 4.1. The percentage of S chalcogen sites inhabited by Se ions was measured as $5.566\% \pm 0.394\%$.

The intensity of each atomic column was sorted into element configurations, such as S₂, S₁Se₁, Mo₂, etc., using a similar approach to Krivanek *et al* [21]. Various element configurations are presented in Figure 4.2 above. Our approach is described in detail in subsection 3.4.1(Element Quantification in HR ADF STEM Images). In short, the intensities of the atom positions are plotted using a histogram, and compute the Z-contrast exponent. This Z-contrast exponent is used to scale the boundaries between the aforementioned elemental configurations, as described by equation 3.2 in subsection 3.4.1(Element Quantification in HR ADF STEM Images). To automatically analyse elements in 2D (and 3D) structures, an expansion of the Python Atomap package [4] was developed [13]. For Atomap's `Atom Position` and `Sublattice` classes, an `elements` and a `z_height` attribute was created. This allows one to utilise Atomap's logical structure for analysis of atomic column intensities. The 'TEMUL toolkit' developed during this project is still in development [412] (TEMUL: Transmission Electron Microscopy University of Limerick). The toolkit is used to its full effect in Subsection 5.2(Atom Dynamics in Implanted TMDs with the MR Tool).

The chemical composition of the assigned elements was verified via image simulation, as well as electron energy loss spectroscopy (EELS), as discussed in Sub-section 4.1.2(Single Atom EELS Characterisation). To verify the created models, the STEM image simulation packages Prismatic [11, 12] and Dr. Probe [380] were utilised. The simulation in Figure 4.4(a-Simulation) was completed using Prismatic's 'prism' algorithm within the Python package PyPrismatic. PyPrismatic is used in the following sections for comparison between experiment and simulation using the `compare_images_line_profile_two_images` function. Many aspects of the plot (see Figure 4.4) can be changed e.g., title, markers, integrating linewidth etc. This function is included in the TEMUL toolkit as a standalone function [13]. Quantitative matching between experiment and simulation was carried out for the data discussed in this chapter with the Model Refiner tool subsection 5.1.2(Model Refiner (MR) Tool). The MR tool can also be utilised to vary the sample and image simulation parameters, so to encompass many variations before comparison with experimental images. Automating this forward modelling process for parameters is discussed in Chapter 7(Conclusions & Future Directions). Furthermore,

4.1. Se Ion Implantation in MoS₂

comparison between experiment and simulation on an absolute scale relative to the intensity of the incident electron beam has not been carried out during this thesis due to easy access to consistent single Mo atoms in all images.

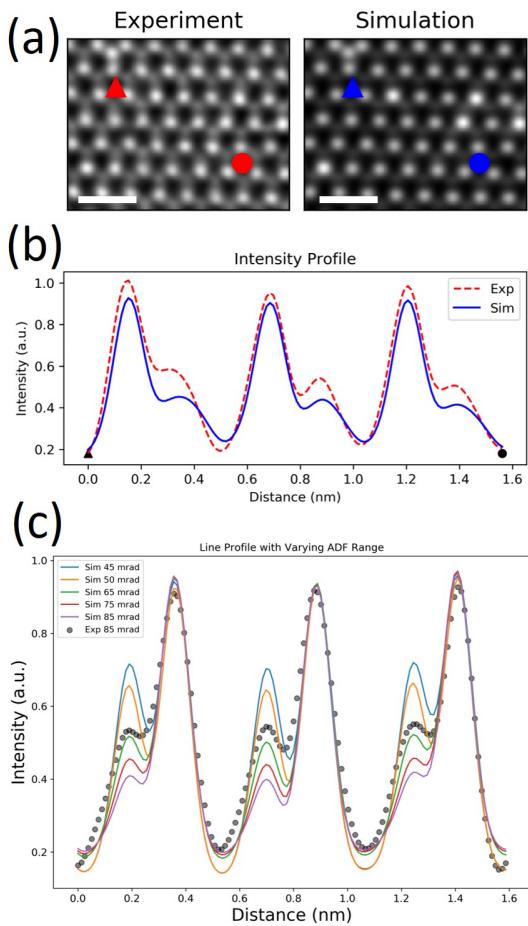


Figure 4.4: Cropped region of a double Gaussian filtered ADF STEM (a) experimental and (b) simulated image (scale bars are 6.3 Å). A line intensity profile is taken from the triangle marker to the circle marker in both and overlaid in (b). (c) Line profile of a similar region while varying the inner ADF limit (outer ADF limit = 185 mrad).

tallest peaks) match well, while the S₂ do not. This is partly due to the background experimental noise originating from electron beam broadening due to hydrocarbon and other surfactant contamination. This contamination has a detrimental effect on experimental images, especially those with low Z elements, such as sulphur. Lower Z elements atomic columns seem to be blurred and brightened dispropor-

In Figure 4.4, one can see the benefit of automatically producing these experiment to simulation comparisons simply by choosing the start point (triangle) and end point (circle) of the line intensity profile. We stress here that line intensity profiles are used only for basic comparison, and that for overall statistics and element quantification, the intensity was aggregated around each atom (sum, max, or mean intensity).

In Figure 4.4(a) and (b), as with all line intensity profile comparisons, the experimental and simulated images are displayed. The line profile is traced from the marked triangle to the marked circle, with an integrating linewidth of 5 pixels, and aggregated by the maximum value perpendicular to the line profile (see [394] for more details on intensity method). The line profiles for the experimental (red, dashed) and simulated (blue, solid) are automatically overlaid for comparison.

There are several caveats to this comparison, as with all comparisons between experiment and image simulation [385]. In the Figure 4.4(b) intensity profile, the bright Mo₁ positions (3

tionately in contaminated regions when compared to larger Z elements (see Figure 4.5).

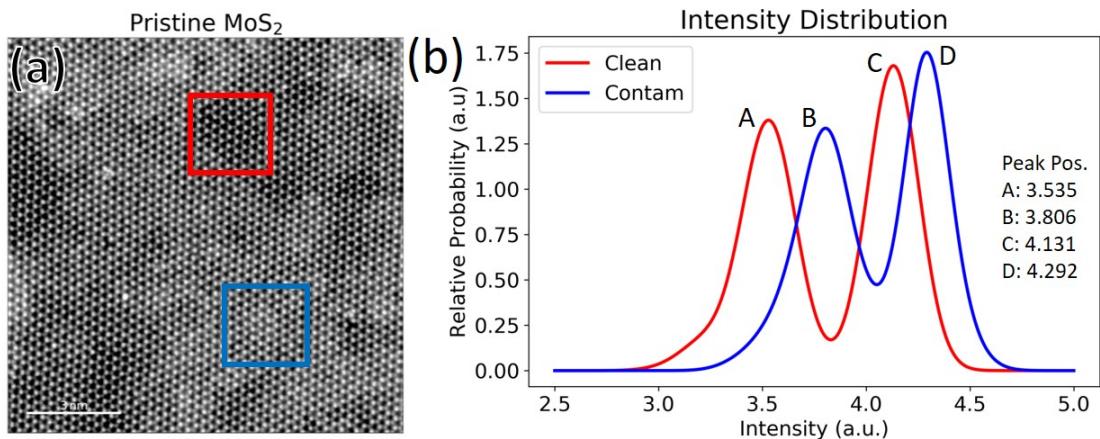


Figure 4.5: (a) A high-resolution ADF STEM image of pristine MoS₂. The red (top) and blue boxes represent a clean and contaminated region, respectively. (b) A PDF of the atom intensities from the regions in (a). The contaminated (contam) region is shifted higher due to the hydrocarbon contamination on the surface of the pristine MoS₂. The peak positions (Peak Pos.) due to the chalcogen atoms (A, B) and transition metal atoms (C, D) are labelled and numbered inset. (Scalebar is 3 nm)

This effect is clear when studying a standard pristine sample, such as pristine MoS₂, as seen in Figure 4.5. This sample was imaged as prepared prior to implantation and shows some interesting differences in intensity due to hydrocarbon contamination; brighter regions arise from an increase in contamination. In Figure 4.5(a), examples of clean and contaminated regions are highlighted in a red (top) and blue box, respectively. The probability density function (PDF) of the intensities from the atoms in these regions is plotted in Figure 4.5(b) [consistent colours between (a) and (b)]. The percentage change in intensity is larger for S than for Mo. This is due to the intensity contribution of the hydrocarbons on the MoS₂ surface. The PDF peak positions are labelled A-D and their values given. Peaks A and B are the chalcogen (S₂) intensities in the clean and contaminated regions, respectively. Similarly, the C and D peaks are the transition metal (Mo) intensities. Using the difference between these peaks, the disproportionate effect the contamination has on the chalcogen positions can be seen. The difference between the chalcogen (B-A) and transition metal (D-C) peaks are 0.271 and 0.161, respectively. Therefore, the difference between the chalcogen peaks is 1.7 times larger than the difference between the transition metal peaks. These differences are generally not taken into account in image simulation because hydrocarbons are rarely simulated on the surface.

With regard to the effect on the Z contrast, and subsequent assignment of

4.1. Se Ion Implantation in MoS₂

elements, the exponent n is 1.3 in the clean region and n is 1.1 in the contaminated region. A minimum intensity cut-off value of 2.75 was used in this example. These values imply that for smaller atomic number elements, such as sulphur, the effective intensity ratio between Mo and S₂ decreases when in the presence of hydrocarbon contamination. This has the effect of reducing the accuracy between experimental and simulated line profiles for chalcogen positions [see Figure 4.4(b)], and it appears that a lower annular dark field (ADF) inner collection angle should be used for simulation. Indeed, various ADF ranges are compared in Figure 4.4(c). Here, one can see that the intensity of the S₂ positions in the simulated ADF range of 85 mrad are well under the experimental values. This is likely partly due to the disproportionate effect hydrocarbon contamination has on the chalcogen intensities, as described above. The effect contamination has on an image series is shown in Subsection 5.2.3(Effect of Contamination on Movie Analysis).

For all simulations, the experimental ADF range given by the microscope was used. However, it is worth discussing the effect of changing the ADF range in greater detail. The experimental ADF range used was 85-185 mrad. However, it is possible that this value was not correctly calibrated during the experiment. Therefore, higher and lower inner ADF collection angles should be simulated, as presented in Figure 4.4(c). For the lower intensity S₂ element configuration, the 65 mrad detector displays a closer maximum intensity match to the experimental data than the 85 mrad detector. To get a more accurate view of the discrepancy, one could simulate with a lower Prismatic interpolation factor (iF), or with the multislice algorithm, or by using the Dr. Probe software. In all below comparisons between experiment and simulation, the ADF range is 85-185 mrad, iF is 64, and the number of frozen phonon configurations is 30. Additionally, the automatic Gaussian blurring method used to match simulation with experiment is not perfect, and only matches structure. A possible improvement could be the use of a Laplacian filter [105], as discussed in Subsection 7.2(Future Directions).

In the following sections, the elements in each atomic column will be compared with image simulation for two broad categories: Subsection 4.1.1(Regions affected by Se Ion Implantation), and Subsection 4.1.1(Vacancies and Hole Damage).

Regions affected by Se Ion Implantation

In this section, the effect of low energy Se ion implantation on pristine MoS₂ is discussed. Clear instances of single atom substitution due to implantation will be investigated, as discussed in [10]. Then, more complex structures, such as probable Se adatoms, are examined.

4.1. Se Ion Implantation in MoS₂

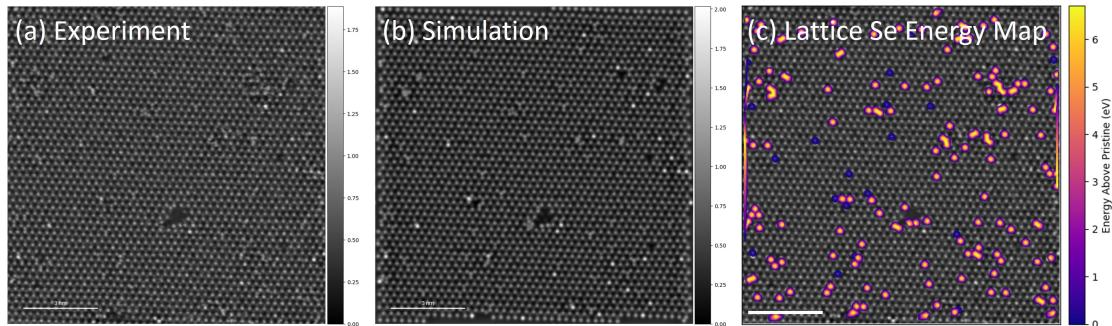


Figure 4.6: Annular dark field (a) experimental and (b) simulated image of Se implanted monolayer MoS₂. (c) Energy map of the Se implants found in the chalcogen sublattice by combining element quantification and DFT calculations. Se₁ and Se₁S₁ atomic column configurations appear as bright yellow (high energy) and blue (low energy), respectively. Scalebars are 3 nm.

The overview of a double Gaussian filtered (DGF) atomic resolution image is given in Figure 4.6(a) alongside the automatically produced best-match image simulation in (b). Many possible implants are visible at a glance. The `plot_atom_energies` function provides a quick overview of the implanted atoms found by the element quantification, and are shown in Figure 4.6(c) for the chalcogen sublattice. The atomic column configuration energy above a pristine value is calculated with density functional theory (DFT). Details on these calculations are shown in subsection 5.2.2(Analysis of Lattice Energetics). The calculated energies are then assigned to each atom in the image and plotted on a contour-map. Atomic column configurations which require a large change in energy above the pristine lattice energy appear bright, allowing for quick identification of anti-sites and vacancies in images. Within the energy map in Figure 4.6(c), Se₁ and Se₁S₁ atomic column configurations appear as bright yellow (high energy) and blue (low energy) areas, respectively. This is because the energy required to create Se₁ and Se₁S₁ configurations, for example, are ~ 6.7 eV and ~ 0.8 , respectively. The map provides a simple way to find successful Se implants which can be compared with the above-described `compare_images_line_profile_two_images`. Additionally, it can be used to quickly check whether the element quantification has missed any possible Se implant signals; the regions without the implants should be pristine and without large contrast changes.

Four standard examples of Se₁S₁ and Se₁ atom column configurations are presented in Figure 4.7, Figure 4.8, Figure 4.9 and Figure 4.10. The Se implant energy map, atom model, experimental image, simulated image and line intensity profile comparing experiment to simulation is shown in each of the figures. How the energy map is formed is described in subsection 3.4.6(Model Visualisation and Data Formats). Only the chalcogen sublattice Se atoms are displayed on each energy

4.1. Se Ion Implantation in MoS₂

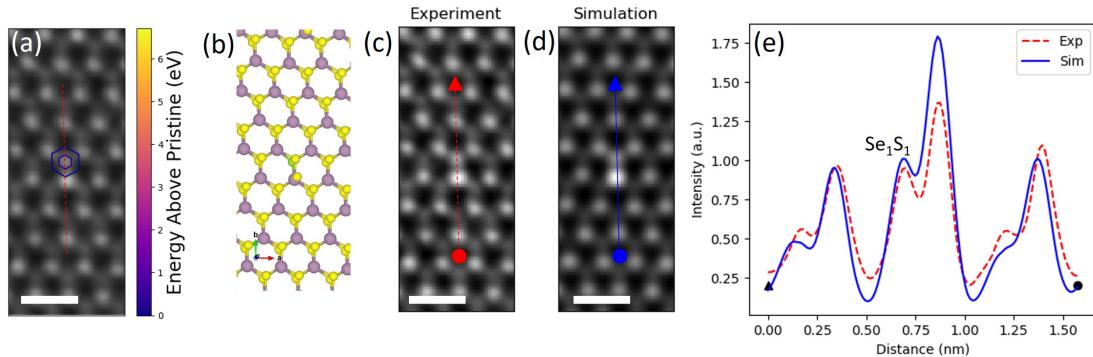


Figure 4.7: (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of a Se implant in the chalcogen sublattice. In (a) the Se implant is indicated by the blue rings corresponding to 0.8 eV. This is confirmed by the atom model in (b) showing the Se₁S₁ configuration. This higher chalcogen intensity is marked in the intensity line profile in (e). The large neighbour intensity is likely a Mo₁S₁ intensity, though it is incorrectly found as a Mo₁Se₁ configuration which would be corrected by using the Model Refiner tool. Scalebars are 5 Å.

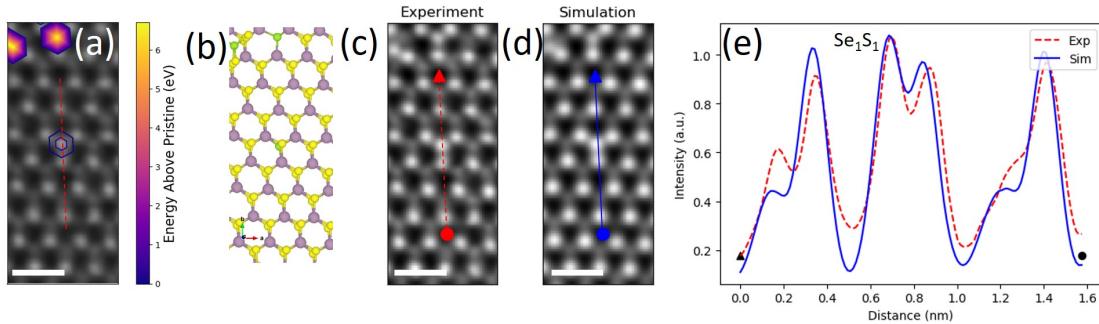


Figure 4.8: (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of a Se implant in the chalcogen sublattice. In (a) the Se implant is indicated by the blue rings corresponding to 0.8 eV (the bright yellow energies are examined in Figure 4.9). This is confirmed by the atom model in (b) showing the Se₁S₁ configuration. This higher chalcogen intensity is marked in the intensity line profile in (e). Scalebars are 5 Å.

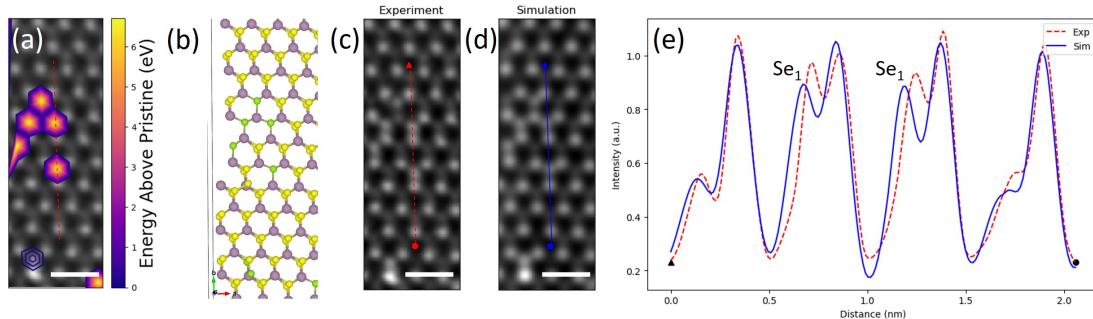


Figure 4.9: (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of several Se implants in the chalcogen sublattice. In (a) the Se implants are indicated by the bright yellow rings corresponding to 6.7 eV. This is confirmed by the atom model in (b) showing the Se₁ configuration. These higher chalcogen intensities are marked in the intensity line profile in (e). Scalebars are 5 Å.

4.1. Se Ion Implantation in MoS₂

map.

The Se₁S₁ atom column configuration has a formation energy of 0.8 eV and therefore is displayed with blue markers around the Se atom position. One such example is shown in Figure 4.7. In the centre of the energy map a chalcogen Se column is detected beside a bright transition metal site. In the element quantification these are initially assigned the element configuration of Se₁S₁ and Mo₁Se₁, respectively (b). By tracing a line intensity profile from the triangle to the circle in (c) and (d), one can compare quickly whether the assigned elements are correct in (e). The chalcogen position is a good match, whereas the transition metal site is not. If this image is run through the Model Refiner (MR) tool (see subsection 5.1.2), this incorrect intensity would be correct to a lower atomic number configuration such as Mo₁S₁.

Another line intensity profile example of Se₁S₁ is shown in the centre of Figure 4.8(a-d) marked by the expected blue rings. In the drawn line intensity profile in (e) the marked Se₁S₁ configuration matches very well with experiment. Additional high energy values can also be seen in the top left of Figure 4.8(a), which are shown in (b) as Se₁ atom column configurations. In this example the line profile is not traced over these Se₁ atoms. There is also a Se₁ configuration positioned in the centre of the MoS₂ hexagon which is the background sublattice. A stable atom intensity located here would suggest a surface adatom. These adatoms, among others, are discussed later in this section.

Several single Se₁ atom column configurations can be seen in Figure 4.9. The energy map in (a) detects six Se₁ columns and one Se₁S₁ column (bottom left).

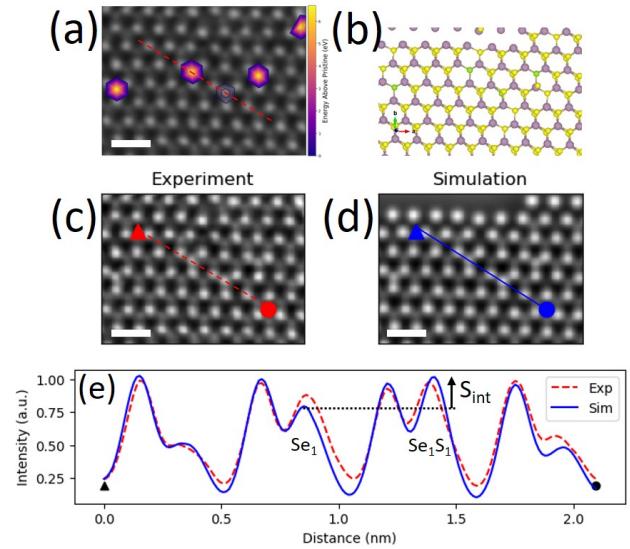


Figure 4.10: (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of a Se implant in the chalcogen sublattice. In (a) the Se implants are indicated by the yellow and blue rings corresponding to 6.7 and 0.8 eV. This is confirmed by the atom model in (b) showing the Se₁ and Se₁S₁ configurations. The Se intensities are marked in the intensity line profile in (e). The intensity due to a single sulphur atom is also shown (S_{int}). Scalebars are 5 Å.

4.1. Se Ion Implantation in MoS₂

The blurred Se₁ energy on the left of (a) is an artifact originating from its position near the edge of the image. Two of the Se₁ columns are included in the line intensity profile, as shown in the experiment and simulation in Figure 4.9(c-d). All intensities match well, though the positions do not. This is possibly due to inexact atom coordinate Gaussian refinement in this region.

The final example of successful Se implantation into monolayer MoS₂ is presented in Figure 4.10, wherein both Se₁S₁ and Se₁ are included in the line intensity profile together. It is worth reiterating that the energy of the system increases more by containing a Se₁ element configurations than Se₁S₁ because the former contains a vacancy in the atomic column. Four Se₁ columns and one Se₁S₁ column are assigned by the element quantification as seen in the energy map in Figure 4.10(a) and the atom model in Figure 4.10(b). The line profile is traced over the centre Se₁S₁ and Se₁ columns from left to right as shown in both the experimental and simulated images Figure 4.10(c-d). The simulated intensities match well with experiment in Figure 4.10(e). One can also quantify from this the probable intensity of a single sulphur (S_{int}) atom as labelled by the arrow in Figure 4.10(e).

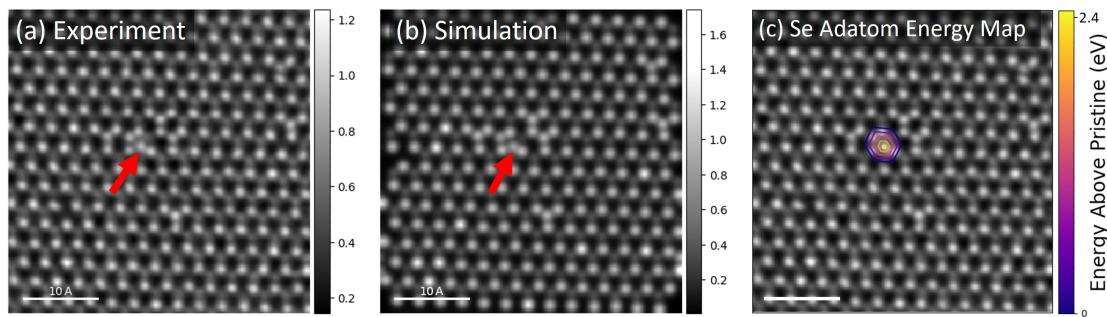


Figure 4.11: Annular dark field (a) experimental and (b) simulated image of Se implanted monolayer MoS₂. The red arrows mark the location of a possible Se surface adatom. (c) Energy map of the Se adatoms found in the background sublattice (centre of hexagon) by combining element quantification and DFT calculations. Scalebars are 10 Å. This cropped area is shown in context in Figure A.5 in Appendix Subsection A.3.

Possible surface adatoms are investigated in the lattice by looking at the intensities located in the centre of the MoS₂ hexagon (background sublattice). This sublattice is generally empty and therefore useful for local background subtraction [see Subsection 3.4.3(Background Intensity)]. However, in some instances there exists a resolved atom in this position. It is difficult to definitively state that these atoms are on the surface, but it is a likely scenario, as confirmed by the mobility of such atoms under the electron beam in movies [see Figure 5.17 in subsection 5.2.2(Analysis of Lattice Energetics)]. An overview of a possible surface adatom is shown in Figure 4.11. In Figure 4.11(a-b) the Se adatom is marked by red arrows, while in Figure 4.11(c) the Se₁ adatom is easily identified via the

blue and yellow rings. These correspond to ~ 2.38 eV above the pristine state with an additional unbound Se₁ atom present (see Figure 5.13(VIII) for details). The adatom's element configuration is quantified as before and verified with the MR tool.

The resulting atom model of the region around the adatom is shown in Figure 4.12(a). A diagonal line profile across both the experimental and simulated images in Figure 4.12(b,c) is drawn in Figure 4.12(d). The line intensity profile matches well between experiment and simulation, though the slight over-blurring of the best-match simulation causes the Se₁ position to seem offset. In Figure 4.12(e) a vertical (top-to-bottom) line profile of the same adatom is traced to show another angle. Again, the Se₁ atomic column configuration matches well with experiment.

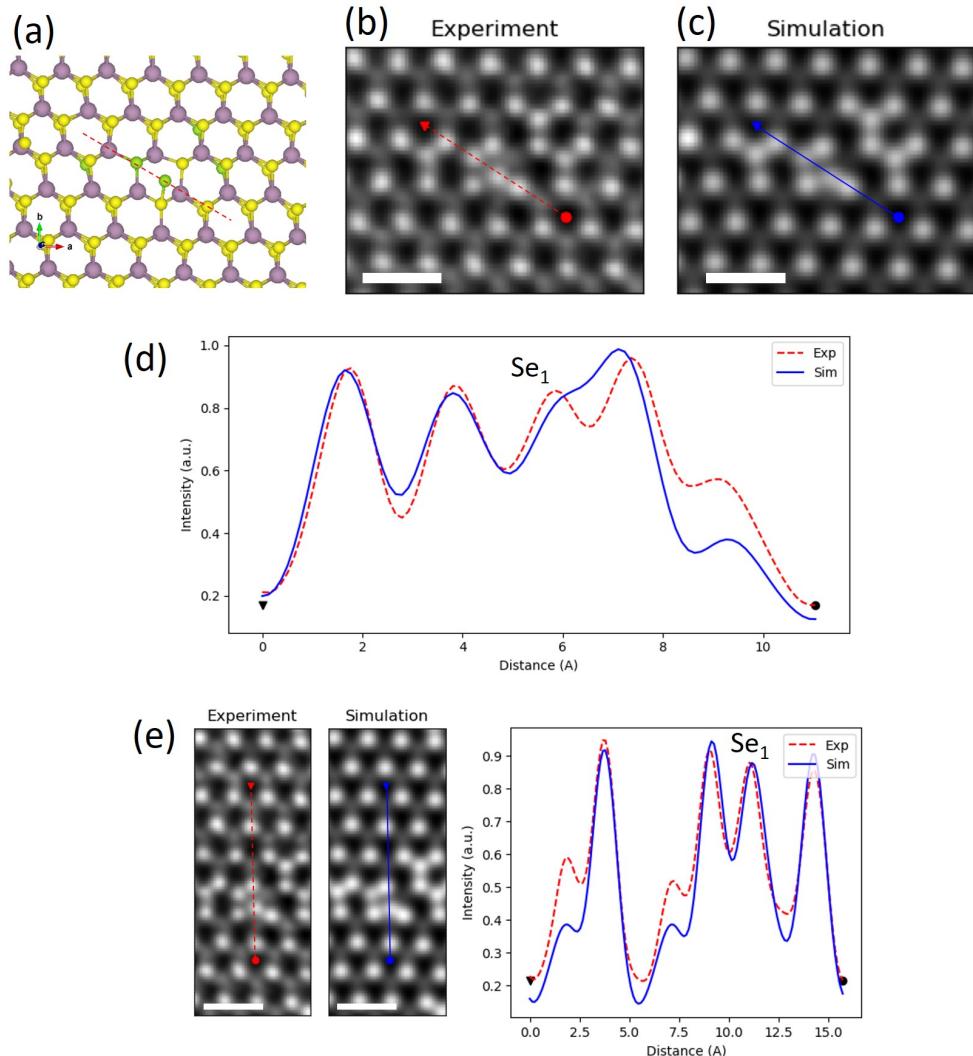


Figure 4.12: (a) Atom model, (b) experimental ADF image and (c) image simulation of a Se adatom in the background sublattice after ion implantation. The adatom found with the energy map and indicated in Figure 4.11(c) is confirmed by the atom model in (a) showing the Se_1 configuration in the centre of the hexagon. The Se adatom intensity is marked in the intensity line profile in (d). (e) Experimental image and Simulation of the same adatom as in (a-d), with the line profile tracing in the vertical direction. Scalebars are 5 Å.

Vacancies and Hole Damage

The effect low energy ion implantation has on the MoS₂ monolayer is investigated by looking at the number of vacancies and holes in the lattice. 10 eV Se ions create sulphur vacancies to subsequently fill the vacancy position. However, in some cases a vacancy or even a hole may be created by the incoming Se ion without being filled, essentially damaging the lattice. An overview of a Se implanted MoS₂ monolayer with a large hole is presented in Figure 4.13. Both the experiment in Figure 4.13(a) and automated best-match simulation in Figure 4.13(b) are shown, with the difference between each in Figure 4.13(c). The bright area in Figure 4.13(c) is the difference between Figure 4.13(a) and Figure 4.13(b) (Exp-Sim) and indicates some intensity in the hole. This intensity is most likely amorphous hydrocarbon contamination. The intensity values are below that of a single sulphur atom (see line profile in Figure 4.14 below). Due to several single (S₁) and double (S₀) sulphur vacancies, the energy map in Figure 4.13(d) highlights the hole as a high energy (yellow) region. S₁ vacancies are evident throughout the image (blue-purple). The vacancies solely due to implantation such as Se₁S₁ or Se₁ atomic column configurations are not considered in the energy map. Both imply a single sulphur vacancy due to implantation, but it is a desired implantation outcome and therefore not included in Figure 4.13(d). Energy maps including S₀ and S₁ and Se₁ vacancies, as well as S₀, S₁, Se₁, Se₁S₁ and Se₂ can be seen in Figure A.7 in Subsection A.3.

Two line intensity profile examples are traced across this hole in Figure 4.14 and Figure 4.15. In Figure 4.14(a) the chalcogen energy map of the hole shows high energies in regions with double sulphur vacancies (S₀). The atom model in Figure 4.14(b) shows some S atoms in the hole. It is likely these intensities are actually due to an amorphous hydrocarbon mix rather than S. However, the option was not given for the element quantification to assign hydrocarbon elements and therefore S is instead assigned. This approach was taken due to the difficulty of identifying, quantifying and reproducing the actual signal of hydrocarbon contamination.

How much of the hole the initial ion implantation creates is difficult to state, as imaging with the electron beam can easily create holes in monolayer materials [184]. The formation of such holes is discussed in more detail in Figure 5.16 in subsection 5.2.2(Analysis of Lattice Energetics). The line profile traced across the experiment Figure 4.14(a) and simulation Figure 4.14(b) is shown in (e), wherein the hole intensity is marked. The experimental intensity is higher than that of the simulation in this region, probably due to hydrocarbon contamination residing in

4.1. Se Ion Implantation in MoS₂

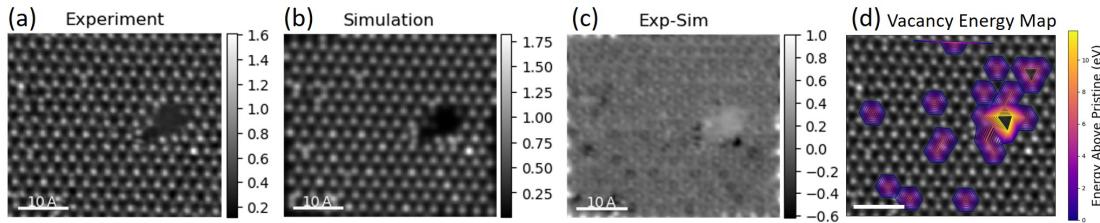


Figure 4.13: Annular dark field (a) experimental and (b) simulated image of Se implanted monolayer MoS₂. The dark contrast in each image indicates a hole. (c) The difference between the experiment (a) and simulation (b), wherein the hole becomes a bright region. (d) The energy map of the sulphur vacancies S₀ and S₁. Scalebars are 10 Å. This cropped area is shown in context in Figure A.6 in Appendix Subsection A.3.

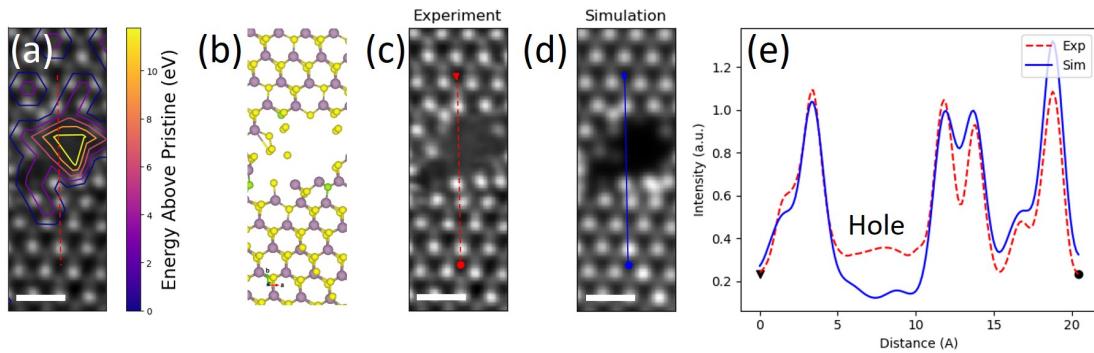


Figure 4.14: (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of hole in monolayer MoS₂. The yellow region in (a) indicates double sulphur vacancies (S₀ ~11.8 eV) while the blue region indicates single vacancies (S₁ ~5.9 eV). The experiment and simulation match well except in the hole itself. The background intensity in the experimental image is likely due to hydrocarbon contamination. Scalebars are 5 Å.

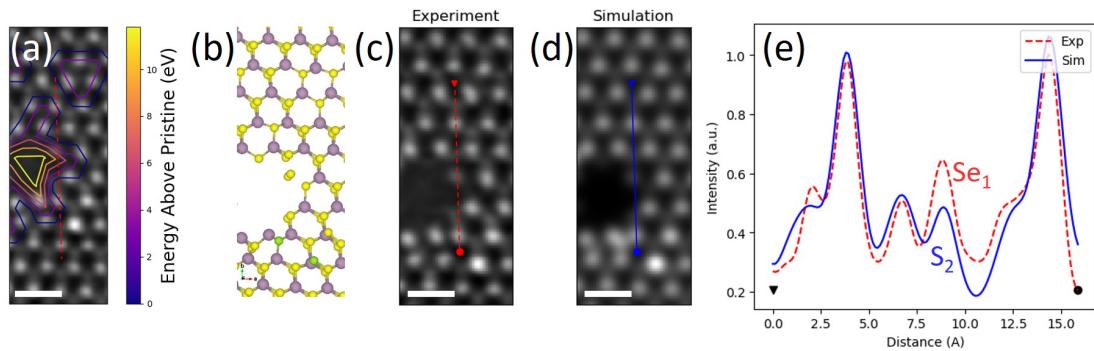


Figure 4.15: (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of hole in monolayer MoS₂. The yellow region in (a) indicates double sulphur vacancies (S₀ ~11.8 eV) while the blue region indicates single vacancies (S₁ ~5.9 eV). (e) The line profile comparison between experiment and simulated intensities. Scalebars are 5 Å.

the hole. This phenomenon has been investigated for image series in Figure 5.27 in Subsection 5.2.3(Effect of Contamination on Movie Analysis). The intensity value of ~ 0.35 in the hole is equal to an atomic number of $Z \approx 12$. It is likely, then, that this is made up of a combination of several carbon and hydrogen atoms in an amorphous layer, as expected on a TMD monolayer.

In the line intensity profile of Figure 4.14(e,) the pair of peaks to the right of the hole are likely single Mo atoms (Mo_1). This has been correctly assigned, even though the first Mo atom is not in its ideal transition metal position. This misplaced Mo atom is a likely candidate for ejection from its lattice position due to its lack of stable bonds at the edge of the hole. Alternatively, it may move to a more stable chalcogen dangling bond at another part of the hole (see Figure 5.16), or it may reside as a surface adatom (see Figure 5.17). The referenced figures are both in subsection 5.2.2(Analysis of Lattice Energetics).

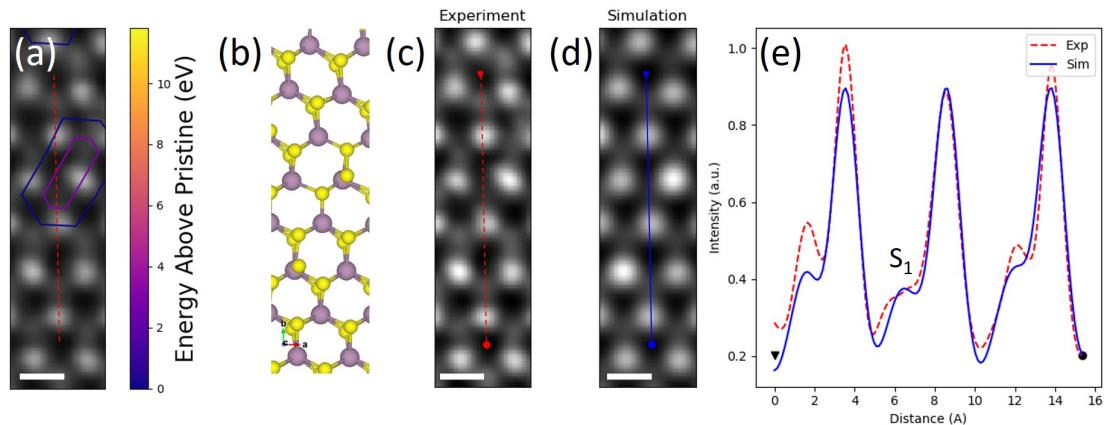


Figure 4.16: (a) Energy map, (b) atom model, (c) experimental ADF image and (d) image simulation of a single sulphur vacancy (S_1) in monolayer MoS₂. In (a) the S vacancy is indicated by the blue and purple rings corresponding to 5.9 eV. The line profile across the vacancy (e) demonstrates a good match between simulation and experiment. Scalebars are 2.5 Å.

In Figure 4.15 the right side of the hole is shown. The line intensity profiles in the experiment (c) and simulation (d) are traced over the hole and over what should be a transition metal position. Instead, the model detects a double sulphur column (S_2), as seen in (e). This underestimates the experimental intensity, which is likely a single Se atom rather than the S_2 . Correctly assigning intensities with automatic element quantification becomes difficult when the usually separate transition metal and chalcogen sublattices are no longer separated in terms of xy coordinates, as is the case near a hole. In other words, the program currently does not correctly handle these cases, though possible corrections based on calculated likelihoods are discussed in Subsection 7.2(Future Directions). In any case, trial and error simulations are vital when investigating holes in monolayer structures.

4.1. Se Ion Implantation in MoS₂

In Figure 4.16 a single sulphur vacancy is investigated. The vacancy energy map is shown in (a), with the purple and blue rings indicating two single sulphur vacancies ($S_1 \sim 5.9$ eV). Only one of these is included in the line profile (dashed red line). The model in (b) shows these vacancies. The line intensity profile traced across the experiment and simulation is shown in Figure 4.16(e). The S_1 atom position is a good match between experiment and simulation, though the vacancy position is not perfectly centred on the line profile. Interestingly, the nearest Mo atom intensity is lower for the experimental line profile than the other Mo atoms in the line profile (dashed red lines). This is partly due to the S_1 signal intensity not contributing as much as a S_2 signal would to the Mo atom intensity. It may also be partly due to the Mo atom itself moving more than the other Mo atoms during image acquisition, which blurs the atom, thus reducing its intensity.

Statistics Before and After Implantation - Autoanalysis

To collect large scale element counting and quantification automatic analysis is required. The program Atomap was used to find and refine the atomic column coordinates of monolayer MoS₂ before and after low energy Se ion implantation. Element quantification was completed using a method similar to [21], as described in subsection 3.4.1(Element Quantification in HR ADF STEM Images). Examples of regions analysed automatically before and after implantation are shown in Figure 4.17 and Figure 4.18, respectively. Analysis of the pristine lattice (Figure 4.17) is easier than the implanted lattice (Figure 4.18). There is less damage (vacancies, holes) in the pristine lattice prior to implantation. During implantation, the incoming Se ions eject S atoms from the chalcogen lattice and subsequently replace them.

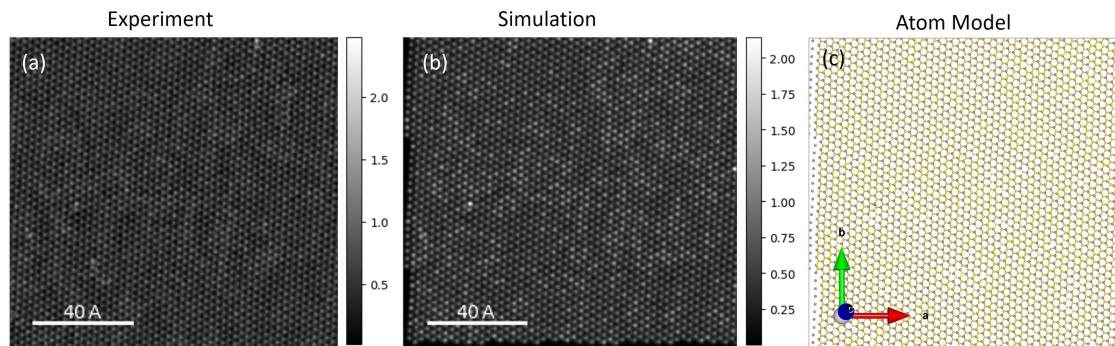


Figure 4.17: Double Gaussian Filtered ADF STEM (a) experimental and (b) simulated image of pristine monolayer MoS₂. (c) Atom Model created after element quantification and used to simulate the data in (b).

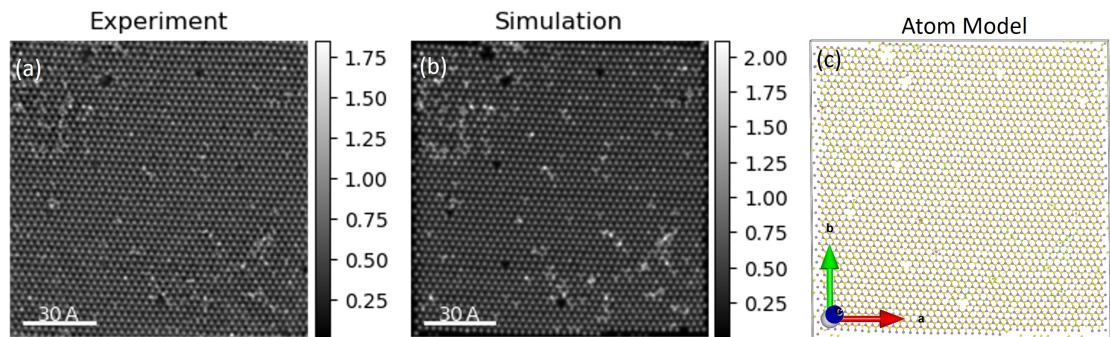


Figure 4.18: Double Gaussian Filtered ADF STEM (a) experimental and (b) simulated image of 10 eV Se ion implanted monolayer MoS₂. The created alloy can be approximated by the chemical formula MoS_{1.87}Se_{0.13}. (c) Atom Model created after element quantification and used to simulate the data in (b).

Though there are more vacancies after Se implantation, this is expected and in some cases desirable. The desirable vacancy formation is due to two S atom being

4.1. Se Ion Implantation in MoS₂

ejected from an atomic column (S_0) after several ion impacts, with only one Se atom filling a chalcogen column position. The resulting Se_1 column is counted as a vacancy, but the result of successful Se_1 implantation is desirable.

In Figure 4.17(a-b), the brighter transition metal columns are identified as Mo_1S_1 configurations, wherein the S atom is positioned as a surface adatom [see subsection 3.3.2(Image Simulation and DFT Calculations) for details]. These positions don't necessarily mean there is a single S atom above each Mo column. However, they give a clear picture as to where general regions of hydrocarbon contamination exist. This is discussed in more detail in Subsection 7.2(Future Directions). Overall, the pristine structure is relatively clean with no holes detected. In contrast to this, the Se implanted sample has several holes containing one or more Mo vacancies, and small patches of dense hydrocarbon contamination mixed with possible adatoms, as seen in the bottom right section of Figure 4.18(a-b).

The results of the number of chalcogen vacancies and number of substitutional Se atoms are given in Table 4.1. The total percentage of chalcogen atomic positions occupied by vacancies and substitutionally implanted Se atoms are given, as well as the chalcogen vacancies and Se atoms per square nm and per unit cell. 21188 chalcogen positions were analysed for data before ion implantation, which equates to $\sim 875 \text{ nm}^2$ (the edge of images are not taken into account). For the implanted samples, 16996 chalcogen positions were analysed, which constitutes $\sim 725 \text{ nm}^2$. In this context the chalcogen positions are the number of atomic coordinates i.e., two atomic coordinates in S_2 .

Table 4.1: Statistics of the chalcogen (Chal.) positions occupied by vacancies (Vac.) or substitutional (Subst.) Se and before and after 10 eV Se ion implantation for automatically analysed regions (example regions are shown in Figure 4.17 and Figure 4.18). The percentage, number per square nm, and number per unit cell are shown. The total area analysed was $\sim 875 \text{ nm}^2$ (21188 chalcogen positions) and $\sim 725 \text{ nm}^2$ (16996 chalcogen positions) before and after implantation, respectively.

	Chal. Vac (%)	Chal. Vac (nm^{-2})	Chal. Vac (per unit cell)	Subst. Se (%)	Subst. Se (nm^{-2})	Subst. Se (per unit cell)
Before Implant	1.756 ± 0.255	0.425 ± 0.061	0.042 ± 0.006			
After Implant	6.554 ± 0.582	1.537 ± 0.137	0.152 ± 0.014	5.566 ± 0.394	1.304 ± 0.092	0.129 ± 0.009

As seen in Table 4.1, the percentage of chalcogen vacancies increased by $\sim 5\%$ between the pristine and implanted sample. This roughly triples the number of chalcogen vacancies per square nm. More vacancies destabilise the lattice during imaging, increasingly the likelihood of further hole formation. Hole formation is discussed in more detail in subsection 5.2.2(Analysis of Lattice Energetics). There

4.1. Se Ion Implantation in MoS₂

is an increased probability of vacancy production and hole formation following electron beam irradiation of the material. The large percentage of $6.554\% \pm 0.582\%$ chalcogen vacancies is therefore likely skewed to a higher value due of the electron beam scan.

After 10 eV Se ion implantation, substitutional Se atoms make up $5.566\% \pm 0.394\%$ of the total chalcogen positions. In terms of area, this equates to 1.304 ± 0.092 Se atoms per square nm and 0.129 ± 0.009 per unit cell. Therefore, the chemical formula for this transition metal dichalcogenide (TMD) monolayer alloy is approximately $\text{MoS}_{1.87}\text{Se}_{0.13}$.

Statistics Before and After Implantation - Manual Analysis

Prior to the development of the automatic element quantification tools, both pristine MoS₂ and Se implanted MoS₂ were analysed manually by tracing line profiles across atoms. An example of each is shown in Figure 4.19, with clean and contaminated regions marked by dashed yellow boxes and red boxes, respectively. Hydrocarbon contamination corresponds to brighter regions over several nanometres. In Figure 4.19(a) the pristine sample has some clean patches surrounded by relatively contaminated areas. The Se implanted in Figure 4.19(b) looks cleaner than Figure 4.19(a), though the region within the dashed yellow line is the only truly large clean region. In the majority of the shown example images, the contamination in both Figure 4.19(a) and Figure 4.19(b) is not so detrimental that the atomic lattice cannot be resolved correctly.

Line intensity profiles were used to analyse the intensity of atoms and therefore quantify the elements in each atomic column. In Figure 4.20(a) one such line profile is displayed from point ‘A’ to point ‘B’. The resulting intensity profile inset shows an expected MoS₂ lattice with a bright intensity in a chalcogen position. This is assigned as a Se₁S₁ intensity by scaling the values of each atom by the experimental Z-contrast exponent Z^n , where Z is the atomic number of an atom and n is the scaling exponent. For the image in Figure 4.20(a) the Z-contrast exponent n is calculated to be $n = 1.6$, as shown in Figure 4.20(b). The average S, Se and Mo intensities are plotted with the $Z^{1.6}$ exponent in Figure 4.20(b). For more detail on the calculation of the Z-contrast exponent see subsection 3.4.1(Element Quantification in HR ADF STEM Images).

The elements were assigned separately for the clean and contaminated regions to check whether the contamination has any effect on the creation of vacancies and implantation of ions. In Subsection 5.2.3(Effect of Contamination on Movie Analysis), the contamination is shown to have a stabilising effect on a monolayer MoS₂ lattice. Therefore, one would expect the same in single frames.

For the manual analysis, the percentage of vacancies in chalcogen positions before and after implantation (averaged over clean and contaminated regions) was $4.06\% \pm 0.47\%$ and $3.53\% \pm 0.25\%$, respectively. In the clean regions this percentage is $5.54\% \pm 0.79\%$ and $4.61\% \pm 0.33\%$ before and after implantation, respectively. However, in the contaminated regions, this percentage is $2.58\% \pm 0.15\%$ and $2.45\% \pm 0.16\%$ before and after implantation, respectively. These values are summarised in Table 4.2 for a total analysed area of $\sim 45 \text{ nm}^2$. There is a clear decrease in the number of vacancies in the contaminated region, and a small decrease in the number of vacancies after implantation. Both results can be expected. The

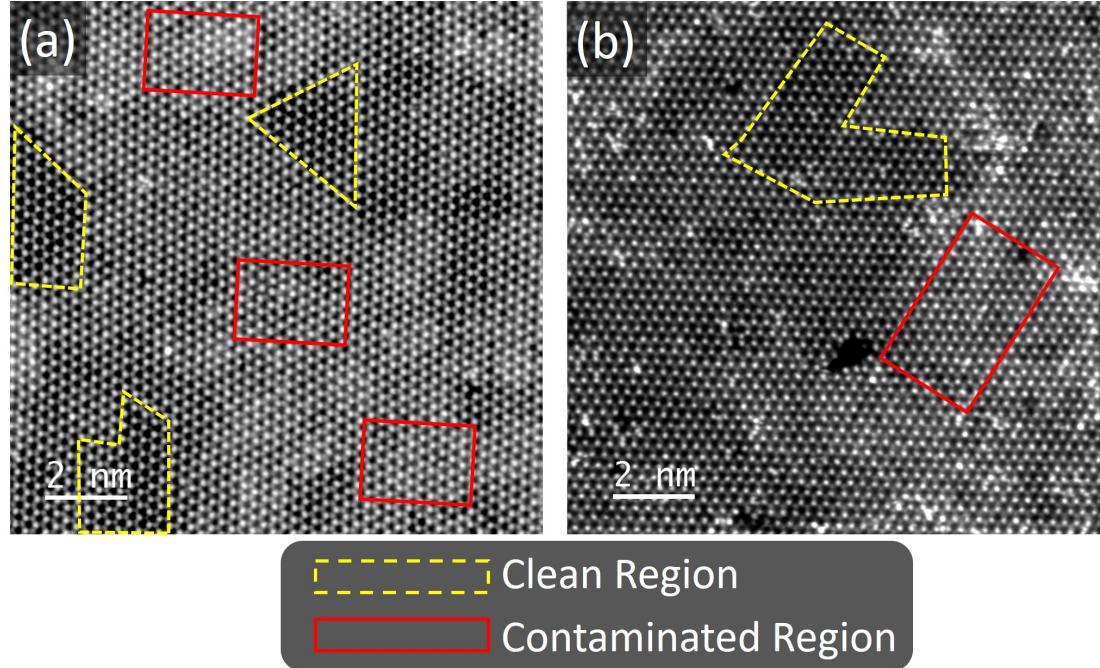


Figure 4.19: Examples of clean (dashed yellow) and contaminated (red) regions in monolayer MoS₂ analysed both before (a) and after (b) 10 eV Se ion implantation.

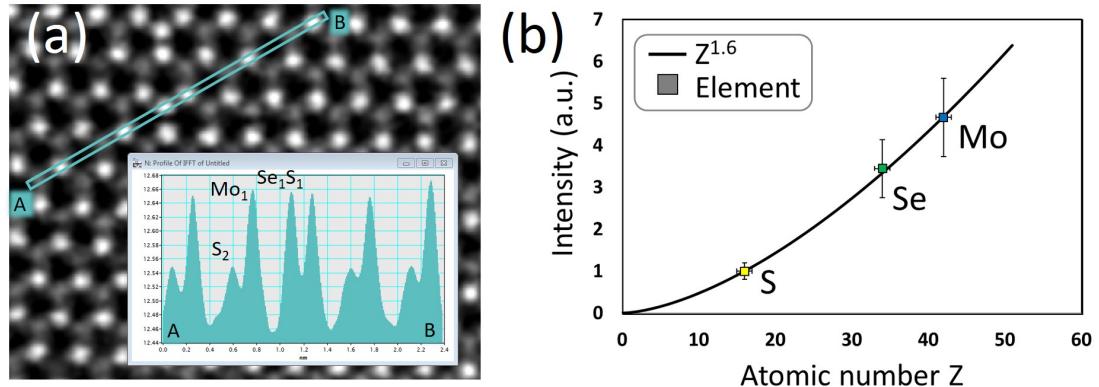


Figure 4.20: (a) Line trace from point A to point B over several atoms in a 10 eV Se implanted MoS₂ monolayer. The intensity profile from this line (averaged over 8 pixels) is shown inset. (b) A plot of HAADF intensity versus each element in the periodic table up to Z=52. The average single element intensities for S, Se and Mo from all line profiles taken from the image in (a) are overlaid, resulting in a $Z^{1.6}$ scaling relationship.

4.1. Se Ion Implantation in MoS₂

decrease in the percentage of chalcogen positions being vacancies in contaminated regions is due to the above-mentioned stabilising effect of the hydrocarbons on the surface acting as a screen to stop the electron or ion beams from affecting the surface, and also acting as a glue to stop any vacancies evolving into holes. The decrease in vacancies after implantation is expected because the Se ions are chalcogens, and will therefore fill the vacancy positions. Creating and subsequently filling vacancies is a technique which has been utilised by several groups [307, 311]. In the previous section the vacancies were measured to have increased after implantation, which is attributed to electron beam effects (many more atom positions analysed).

Table 4.2: Percentage of chalcogen positions occupied by vacancies before and after implantation for clean and contaminated regions. The average is also displayed. The percentage of chalcogen positions occupied by Se atoms due to 10 eV ion implantation, as well as the percentage of Se ions retained by the lattice from the ion beam are shown. The total area analysed manually was $\sim 45 \text{ nm}^2$.

Area	Chalcogen vacancies before implantation (%)	Chalcogen vacancies after implantation (%)	Substitutional Se implants (%)	Substitutional Se ion retention (%)
Clean (Manual)	5.54 ± 0.79	4.61 ± 0.33	3.29 ± 0.33	2.86 ± 0.29
Contaminated (Manual)	2.58 ± 0.15	2.45 ± 0.16	1.47 ± 0.16	1.26 ± 0.14
Average (Manual)	4.06 ± 0.47	3.53 ± 0.25	2.38 ± 0.25	2.06 ± 0.22

The number of substitutional Se implants as a percentage of the total number of chalcogen positions is also presented in Table 4.2. On average, they make up $2.38\% \pm 0.25\%$ of the chalcogen positions. More are present in the clean regions ($3.29\% \pm 0.33\%$) than in the contaminated regions ($1.47\% \pm 0.16\%$). The percentage of Se ions from the incoming ion beam retained in the lattice was, on average, $2.06\% \pm 0.22\%$. More Se ions are retained in the clean regions ($2.86\% \pm 0.29\%$) than in the contaminated regions ($1.26\% \pm 0.14\%$).

Using the automatic element quantification to understand the difference between clean and contaminated regions' elements is also possible, though it requires further development of the tools created during this project. Some ideas for development of this are described in Subsection 7.2(Future Directions). The manual analysis is useful for understanding this difference between clean and contaminated regions. However, manual analysis is not very accurate (using line profiles is not the same as integrating around an atom's radius in 2D), is slow, is not scalable, and can be user biased (e.g., which region is chosen). Therefore, the previous section, wherein automatic analysed of many images was completed, is considered more reliable.

4.1.2 Spectroscopy of Se implanted MoS₂

To unambiguously confirm the presence of Se atoms in monolayer MoS₂, electron energy loss spectroscopy was used [100]. As described in Subsection 2.1.1(EELS), it can be used to identify single dopant atoms via characteristic peaks arising from electron energy loss due to ionisation of inner-shell electrons (core-loss EELS). First, single atom characterisation of Se implanted MoS₂ monolayers via core-loss (CL) EELS is discussed. Then results of low-loss (LL) EELS data showing manipulation of the material's band structure are presented.

Single Atom EELS Characterisation

Electron energy loss spectroscopy (EELS) is a versatile tool for the analysis of the electronic and bonding structure of materials. For two-dimensional materials, it can identify single native and dopant atoms at high-resolution as discussed in subsection 2.1.1(EELS) [133]. In this section, the use of EELS for the identification of the core loss signal of Mo, S and Se atoms in monolayer Se implanted MoS₂ is discussed. As described in subsection 2.1.2(Electron-Matter Interactions) and subsection 5.2.3(Effect of Electron Beam), increasing the effective electron beam current on monolayer materials increases the likelihood of damage to the material. It is therefore worth noting now that the structure of the lattice before an EELS scan or spectrum image may not be the structure after the scan. Furthermore, as Se is a chalcogen and has a higher ejection probability than a transition metal such as Mo, it cannot be assumed that a Se atom will remain in place during scanning. During a scan, the atom may be displaced leaving blurred signals or scan distortions affecting both annular dark field images and EEL spectrum images. Therefore, it is difficult to be completely certain of the elements present in atomic columns. More datasets were acquired, but the most clear examples are shown in this section. Single-atom mapping of monolayers is challenging from a hardware and technical standpoint, and the unfavourable Se-M_{4,5} edge only makes definitive element identification more difficult.

An example overview high angle annular dark field (HAADF) scanning transmission electron microscopy (STEM) image of Se implanted MoS₂ is presented in Figure 4.21(a). A dark field image and spectrum image of the highlighted rectangular region were acquired simultaneously. The overview and spectrum image have a dwell time of 64.8 ms and 0.01 s, respectively. The spectrum image is acquired in a thin vertical shape to minimise the effect of the horizontal line scan on the data; long dwell times accentuate the negative effects of sample drift. The HAADF STEM image of this highlighted region is displayed in Figure 4.21(b).

4.1. Se Ion Implantation in MoS₂

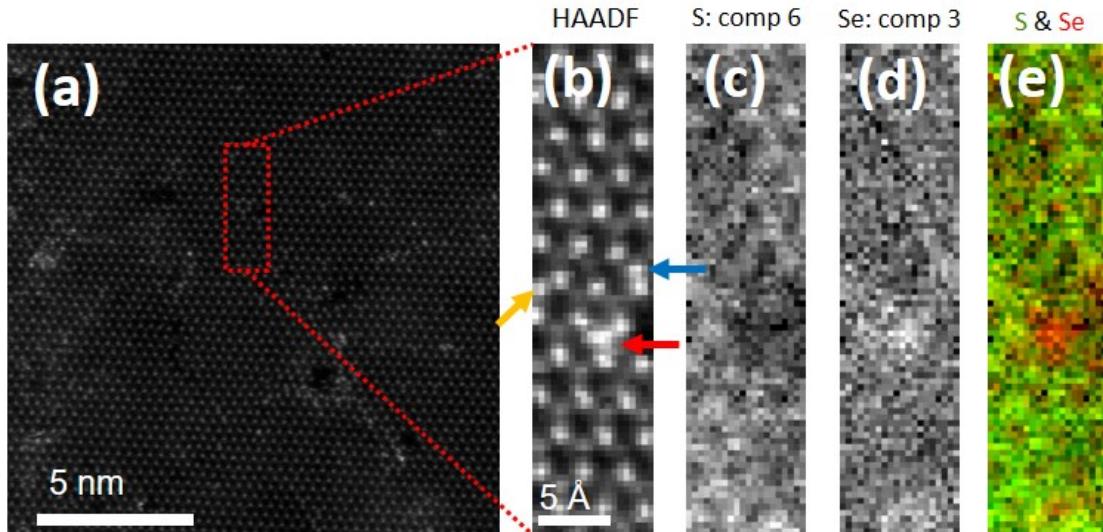


Figure 4.21: (a) HAADF STEM overview image of Se implanted monolayer MoS₂. The highlighted region is enlarged in (b), where a substitutional Se implant is marked by the red arrow. Other Se-like intensities are shown by the orange and blue arrows. (c-d) Component 6 and 3 of a PCA denoised EELS spectrum image of the region in (b) are assigned as sulphur and selenium signals, respectively. The S and Se signals are false coloured and overlaid in (e). The PCA factors are displayed in Figure 4.22.

A probable Se₁S₁ or Se₂ atomic column configuration is indicated with a red arrow. We speculate that this is a Se₂ atom configuration due to its high intensity and high signal after principal component analysis (PCA) denoising. After applying PCA to remove noise, the main principal components of the data were analysed. Figure 4.21(c-e) shows the result of this process for the spectrum image. The sixth (comp 6) and third (comp 3) components are displayed as the S and Se signals, respectively. These are false coloured and overlaid in Figure 4.21(e) to further display the enhancement of signal near the substitutional Se atom.

Figure 4.22 shows the PCA energy loss signals (factors) for the two maps (loadings) in Figure 4.21(c-d) as well as the fourth component's factor and loading. Selenium's M_{4,5} and sulphur's L_{2,3} edge onsets are indicated by the two vertical dashed lines throughout the figure. In component 3 in Figure 4.22(a), the signal rises above an intensity of 0 close to selenium's M_{4,5} edge (57 eV). In Figure 4.22(b) the signal is seemingly the inverse of (a). This is possible, as PCA components do not have any physical meaning, as described in subsection 2.1.1(EELS). In Figure 4.22(c), the signal rises above 0 intensity at \sim 160 eV, close to sulphur's L_{2,3} (165 eV) edge onset.

All above-mentioned configurations were verified via image simulation. To do this we utilised the Model Refiner (MR) tool developed during this thesis (subsection 5.1.2). The intensity refinement algorithm, described in detail in Sub-

section 5.1.1 was used for this and subsequent datasets. The image simulation verification is shown in Figure 4.23. The experimental image from Figure 4.21(b) above is analysed using the methodology set out in subsection 5.2.1 for a single image.

All element quantification and image simulation is handled with the MR tool, described in subsection 5.1.2. In short, the MR tool tracks the quantification of elements via comparison of image simulation with the original experimental image. After two intensity refinements, the simulation in Figure 4.23(b) is produced. The atomic model used for simulation is given as a plan view and side view in Figure 4.23(c) and Figure 4.23(d), respectively.

The substitutional Se implant detected with EELS above is also detected here in Figure 4.23, indicated by the red arrows in the experimental, simulation, and both atom models. The element quantification detects a Se₁S₁ configuration, rather than the above-hypothesised Se₂ configuration.

The program places the Se atom in the bottom sulphur sub-layer, facing away from the electron beam. This is the default location, and should not be taken as an assumption on the sub-layer position. Two other likely instances of Se₁S₁ are found by the program and marked in Figure 4.23 as blue and orange arrows. They certainly match the Se₁S₁ ADF intensity. This raises the question of why does the PCA not show these locations as clearly as the location pointed to by the red arrow. In the original figure - Figure 4.21(d) - the denoised chemical map signal does increase slightly at these other likely Se locations (orange

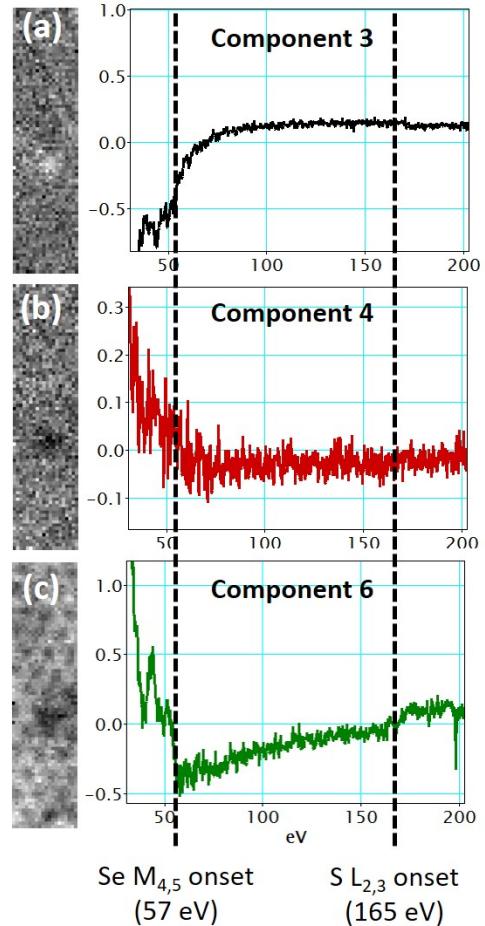


Figure 4.22: The 3rd, 4th and 6th components of the PCA denoised EELS spectrum image scanned over the highlighted region in Figure 4.21(a). The maps (left) are created from the signals (right) by integrating the signal over a 50 eV energy window after the onset of the relevant element edge. The 3rd and 6th components are likely the Se and S signals, respectively.

129

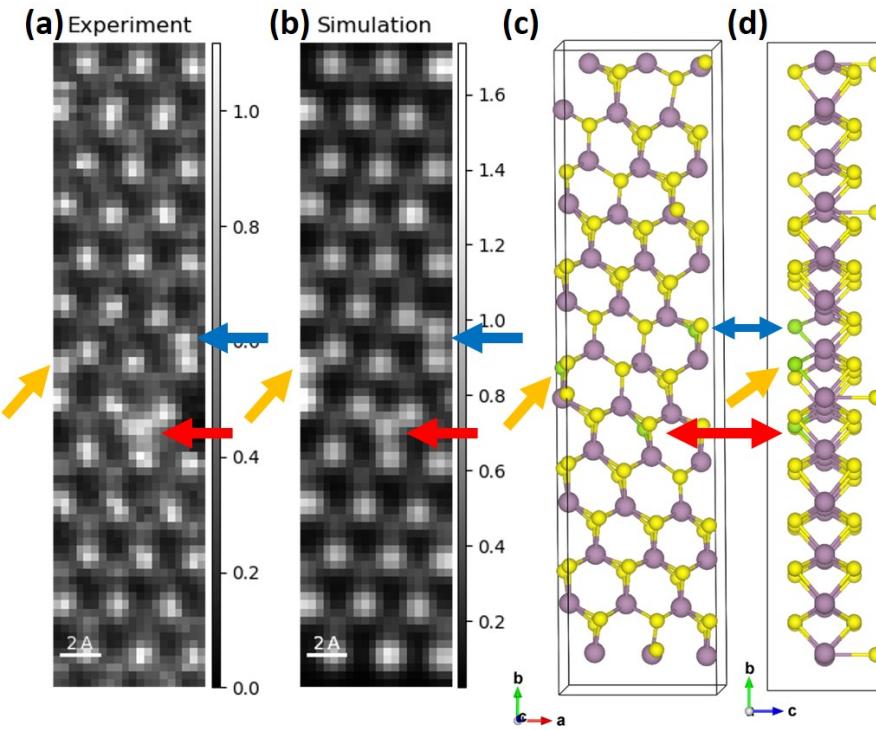


Figure 4.23: (a) Experimental HAADF STEM image of the EELS scan region in Figure 4.21 above. (b) Automatic and refined image simulation of the same region. (c) Refined atomic model used to produce the simulation in (b). (d) Cross-sectional view of the MoS₂ monolayer lattice showing the automatically found substitutional Se implants. The red, orange and blue arrows indicate locations of Se atoms, as detected by the refinement of the atomic model.

and blue arrows). It is possible that the very high signal Se position marked by the red arrow is in the bottom chalcogen sublayer, whereas the other Se locations are in the top layer (facing the beam). For ideal image focus, the electron beam is converged to a point on the top chalcogen layer. The electron beam then diverges, causing it to have a larger interaction cross-section with the bottom chalcogen layer. Therefore, the bottom chalcogen layer has a larger scattering cross-section and a higher signal. Overall, it is not conclusive through EELS that there exists a substitutional Se atom in this example.

Another explanation could be that the high EELS signal marked by the red arrow is not a substitutional Se atom, and instead is a metal adatom (not in the lattice). In the ADF experimental image there is clear scan distortion at this position, which is likely to occur when imaging a surface adatom. Additionally, the EELS signal would be high for such a stable adatom. A candidate would be Fe, which also has an M edge near 55 eV; the M_{2,3} edge at 54 eV. However, it is unlikely for an adatom to be stable long enough to survive a 0.01 s dwell time EELS scan.

4.1. Se Ion Implantation in MoS₂

Although the PCA method can help to denoise and indicate some structure from EELS data as shown above, more analysis is required. Independent component analysis (ICA) is a useful tool for this. It was utilised through the FastICA algorithm [401] with the HyperSpy Python program in this section to identify single dopants in monolayer MoS₂. The HyperSpy program allows one to easily analyse multidimensional datasets, such as these EELS spectrum images. A brief walkthrough with real data of the PCA/ICA process is shown in Subsection 3.4.8(Brief Component Analysis Walkthrough).

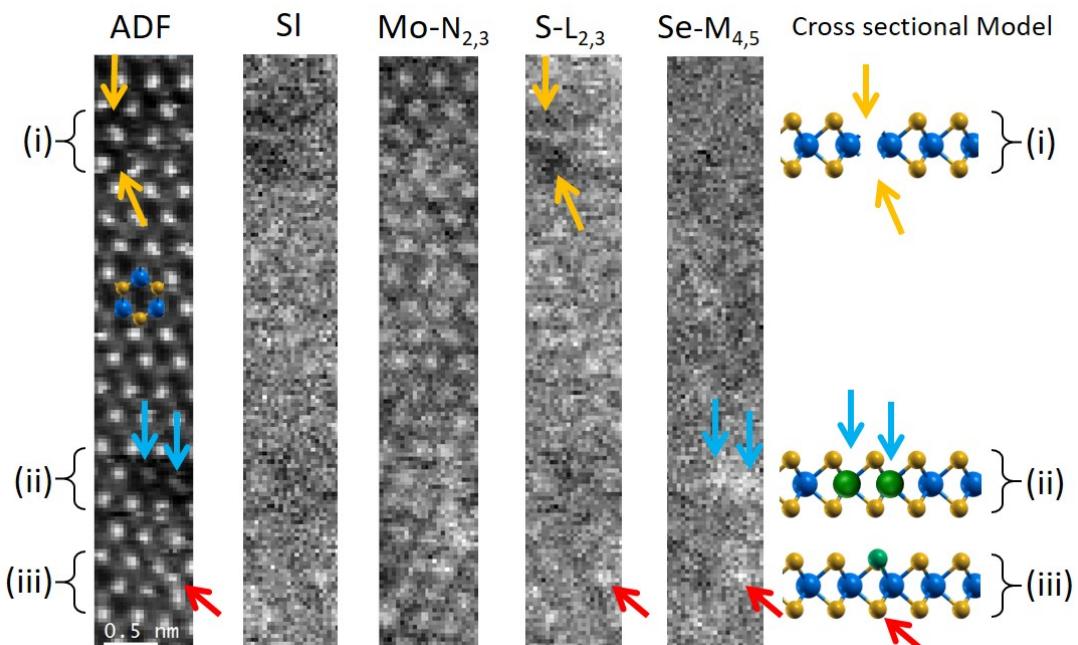


Figure 4.24: EELS spectrum image (SI) over the atoms in the ADF image. FastICA analysis of the SI produces three relevant component maps: Molybdenum N_{2,3}, Sulfur L_{2,3} and Selenium M_{4,5}. The cross-sectional models display a likely lattice configuration. The areas in (i), (ii) and (iii) are discussed in the main text; they display interesting results due to the electron/ion beam.

As in Figure 4.21 above, a thin vertical HAADF image and spectrum image (SI) are simultaneously scanned, as shown in Figure 4.24(a) and (b), respectively. The SI is then denoised with PCA and separated into its probable Mo, S and Se components with ICA. The right side of Figure 4.24 displays the likely atomic column configurations. These configurations are a cross-section of the monolayer.

In Figure 4.24, three areas of interest are inspected: (i) chalcogen vacancies, (ii) Se substitution in Mo vacancies, and (iii) Se substitution in the chalcogen sublattice. The plan-view of MoS₂ is overlaid in the centre of the ADF image. The dark areas in the ADF image and SI match, indicating vacancies in Figure 4.24(i) and Figure 4.24(ii). In Figure 4.24(i), orange arrows indicate the chalcogen vacancy

4.1. Se Ion Implantation in MoS₂

locations. It is likely that the high dwell time SI acquisition created these vacancies. The vacancies are reflected as a lack of energy loss signal in the sulphur L_{2,3} edge signal map. In the cross-sectional model on the right of Figure 4.24 shows a probable double chalcogen vacancy atom configuration.

In Figure 4.24(ii), the electron beam has likely damaged the material's structure, ejecting and filling two Mo lattice atoms, as well as ejecting several nearby S vacancies. These sites are marked by the blue arrows. The signal blurring in this area in the Mo-N_{2,3} component map further hints at Mo atom damage. The increase in signal in this area in the Se-M_{4,5} component map also backs up the characterisation of Se atoms present in the Mo positions due to the electron beam. The model (right) in area (ii) presents a configuration after scanning, though as stated above, it is likely several S atoms vacancies would also be present when several Mo atoms are ejected from the lattice due to the bonding of MoS₂, as described in subsection 2.1.2(Electron-Matter Interactions).

In Figure 4.24(iii) a substitutional atom is present, highlighted by the red arrow. In both the S-L_{2,3} and Se-M_{4,5} component maps, this atomic position has a high intensity. Therefore, it is probable that a single Se substitution is present, as displayed in the model (right) in Figure 4.24(iii). Whether this Se implant is situated in the top or bottom chalcogen sublayer (relative to the electron/ion beam) is difficult to determine, as the ADF intensity difference between each is masked by experimental noise [10]. The high intensity in the S-L_{2,3} map in this region is puzzling. It could be that the atomic column with Se is stable and therefore more S signal is detected, but this is not seen normally during EELS scans. Alternatively, and more likely, the decomposition via PCA and ICA is not perfectly representing the physical structure of the lattice.

The ADF image for this region is simulated in the same manner as described for the previous region using the model refiner (MR) tool (Figure 4.23). The three areas (i), (ii), and (iii) are again marked in Figure 4.25. Figure 4.25(i) is identified as a single S vacancy (S₁) by the orange arrow. It may be a double S vacancy (S₀), but due to the nearby Mo atoms adding intensity, it is detected as S₁. This is expected at such a low image sampling (very few pixels per atom). It is easy to see the atomic column configurations in Figure 4.25(c-d). In Figure 4.25(ii) the proposed Se in transition metal positions are identified (blue arrows), though their intensities do not match perfectly. This could be due to electron beam effects such as the long dwell time causing the atom to appear lower in intensity due to large movement. Another candidate could be other impurity atoms, such as Fe. Additionally, there are numerous nearby S vacancies detected by the program, as can be seen in Figure 4.25(c). In Figure 4.25(iii) the substitutional Se implant shown

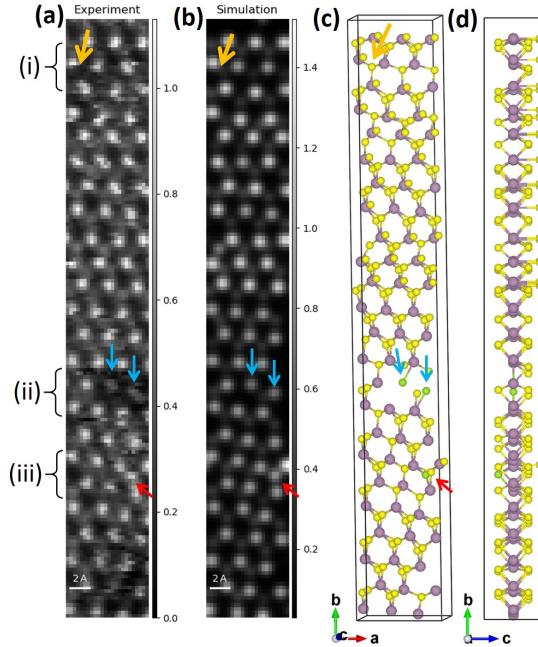


Figure 4.25: (a) Experimental HAADF STEM image of the EELS scan region in Figure 4.24 above. (b) Automatic and refined image simulation of the same region. (c) Refined atom model used to produce the simulation in (b). (d) Cross-sectional view of the atom model. Scalebars 2 Å.

by the EELS data is supported by element quantification which finds an atomic column configuration of Se₁S₁ (red arrow) [see Figure 4.25(c) and Figure 4.25(d)].

An interesting outcome of the automated MR tool image simulation is the detection of several intensities higher than the average Mo intensity in the top half of the ADF image in Figure 4.25(a). This is clear in the simulation (b) and in the cross-sectional atom model (d). These Mo₁S₁ configurations are not concrete and act as an indicator for contamination if detected in a region. It is more likely for there to be a mix of hydrocarbons lying on the lattice rather than just sulphur. Regardless, they show that the upper part of the image is more contaminated than the lower part. It would be an interesting study to model the most likely structure of a hydrocarbon on the surface of such a monolayer and to iteratively simulate the lattice to find a best match. This is possible with the MR tool described in subsection 5.1.2(Model Refiner (MR) Tool), but is unfortunately outside the scope of this thesis.

Figure 4.26 plots the original ADF image and component 1 from the FastICA analysis expanded upon in Figure 3.20 in Subsection 3.4.8(Brief Component Analysis Walkthrough). We assign this ICA component the Se label due to its increase in signal at probable Se atom intensities in the ADF image as done previously (Figure 4.22, Figure 4.24). The three labelled areas are (i) a single Se implant,

4.1. Se Ion Implantation in MoS₂

several Se implants, and (iii) a displaced Se atom after SI acquisition.

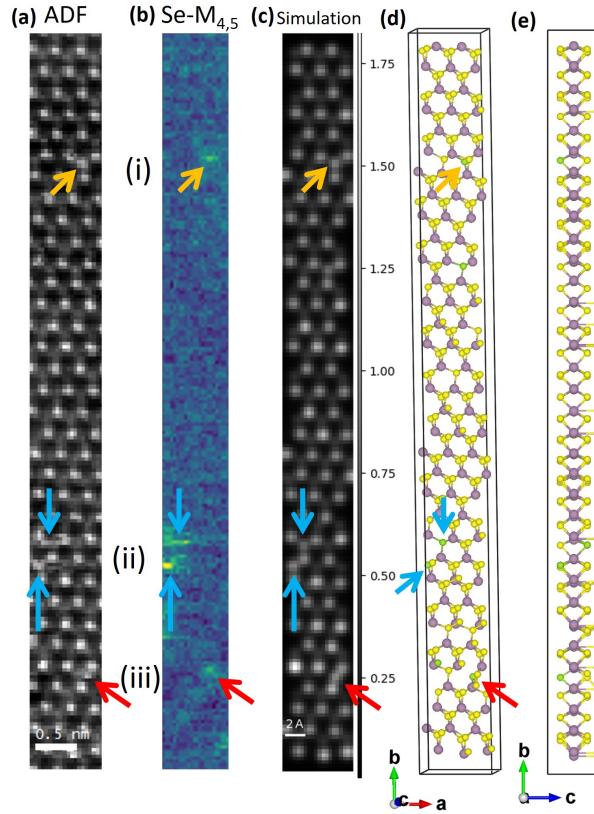


Figure 4.26: ADF and FastICA selenium M_{4,5} component map. Areas (i), (ii) and (iii) are described in the main text. (a) Experimental HAADF STEM image of the EELS scan region in Figure 3.19 above. (b) Automatic and refined image simulation of the same region. (c) Refined atomic model used to produce the simulation in (b). Scale bars are 2 Å.

In Figure 4.26(i), there is an increased intensity due to a substitutional Se implant sitting in the chalcogen lattice marked by the orange arrow. The corresponding rise in Se signal in the (b) Se-M_{4,5} component map is promising, and indicates a high probability of successful Se implantation. With the automated element quantification this position is identified as a Se₁S₁ atomic column configuration Figure 4.26(c-d). This is perhaps the clearest example of a successful substitutional Se implant identified by both EELS and image simulation.

In Figure 4.26(b, ii), there are two separate instances of a rise in Se signal marked by the blue arrows. In the ADF image, the downward pointing arrow points to a blurry but high chalcogen intensity indicating a Se atom intensity disrupted by the electron beam during the spectrum image scan. The simulation and atom models in Figure 4.26(c, d) show this position is identified as a single Se atom. It should be noted that the scan distortion makes it difficult to conclude this atomic column configuration as accurate. The upward pointing arrow marks

4.1. Se Ion Implantation in MoS₂

a stable high intensity chalcogen signal. This is likely a single or even double Se implant, as the component map signal intensity is highest at that location. The image simulation in Figure 4.26(c) and atom models in Figure 4.26(d-e) attribute this position's intensity as a Se₁S₁ column.

In Figure 4.26(a, iii), a blurred high chalcogen intensity indicates some movement of an atom during acquisition (red arrow), with the Figure 4.26(b) Se map showing a higher intensity at that position. It is likely that the substitutional Se atom was not very stable during scanning, as often observed during the high dwell time EELS scans. The element quantification and MR tool image simulation again identify a Se₁S₁ column in Figure 4.26(c-e).

Overall, the identification via EELS chemical mapping of Se implanted substitutionally in MoS₂ was successful. As stated in the introduction to this section, single-atom mapping in a monolayer with EELS is challenging from a hardware and technical standpoint, and that the unfavourable Se-M_{4,5} edge only makes definitive element identification more difficult. Atomic resolution image simulation was used to verify likely Se EELS signals.

Low Loss EELS Characterisation

Low-loss (LL) EELS was used to understand the effect of low energy Se ion implantation on the band structure of monolayer and few-layer MoS₂. Figure 4.27 presents the LL EELS signal from both pristine and Se implanted MoS₂. Figure 4.27(a) and Figure 4.27(b) show the annular dark field (ADF) and spectrum image (SI) for the pristine and implanted monolayer, respectively. For Figure 4.27(b) care was taken not to overlap the SI region of interest (red box) with the bi, tri, and quad-layer areas of the sample. Figure 4.27(c) displays the background subtracted LL signal. The zero loss peak (ZLP) background signal was removed by applying a power law. More information can be found in the appendices in subsection A.3.1(Low loss EELS). The shape of the implanted LL signal peaks (red line) in Figure 4.27(c) are not as sharp as the pristine shape (black line). This shape is similar to that shown by Tizei *et al* wherein the larger exciton radius in MoS₂ was shown to spread the LL signal in alloyed regions [78].

Table 4.3: LL EELS measurements of band structure properties of pristine and Se doped MoS₂. Numbers from literature are given under *Literature*. E_{BG}: optical band gap energy (onset). E_{exc,A}, E_{exc,B}, E_{exc,C₁,C₂}: A, B, C₁ and C₂ exciton peak energies, respectively. Uncertainty in experimental reading for pristine: ± 0.026 eV, implanted: ± 0.046 eV. *requires further fitting, higher uncertainty. [†]Bilayer implanted region not as accurate (smaller region). Literature values taken from [78, 234, 413, 414].

Material	E _{BG}	E _{exc,A}	E _{exc,B}	E _{exc,C₁,C₂}
<i>MoS₂ (this work)</i>				
Pristine Mono	1.79	1.94	2.07	2.94, 3.19
Pristine Bi	1.76	1.93	2.10	2.93, 3.22
Implanted Mono	1.70	1.95	2.12	3.10, 3.35
Implanted Bi [†]	1.76 [†]	1.95 [†]	2.14 [†]	3.06, 3.42 [†]
Implanted Tri	1.73	1.94	2.12	3.08, 3.49
Implanted Quad	1.76	1.94	2.14	3.08, 3.52
<i>Literature</i>				
MoS ₂ Mono	1.87	1.95	2.2	2.8, 3.25
MoS ₂ Bulk	1.23			
MoSe ₂ Mono	1.55	1.68	1.86	2.6, 3.0
MoSe ₂ Bulk	1.09			

Table 4.3 shows the band gap energy, A exciton, B exciton and C exciton values measured during this thesis for the 10 eV Se ion implantation experiment shown in

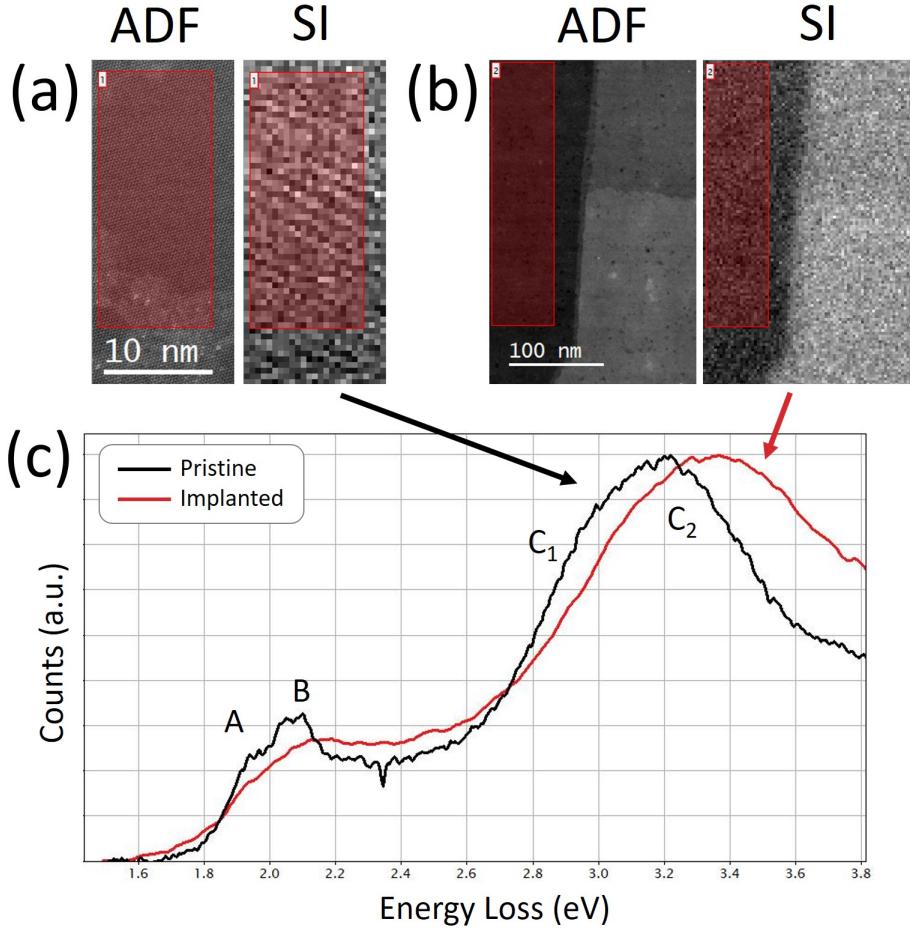


Figure 4.27: ADF and SI of an area of (a) pristine and (b) Se implanted monolayer MoS₂. (c) The pristine (black line) and implanted (red line) signals from the regions of interest (red boxes) in (a) and (b), respectively.

Subsection 4.1.1(Statistics Before and After Implantation). The optical band gap energy (E_{BG}) was measured from a straight line fit from the the signal onset below the first exciton peak along the rising edge of the first exciton peak. The various exciton peaks $E_{exc,A}$, $E_{exc,B}$, and E_{exc,C_1,C_2} were defined by their peak value [414]. This implanted sample is a MoS_{1.87}Se_{0.13} alloy, equating to 5.56% substitutional implantation. The pristine MoS₂ band gap energy (E_{BG}) was measured at 1.79 ± 0.026 eV, between the values measured and calculated in literature (1.75-1.89 eV) [78, 234, 413, 414]. In the implanted monolayer, E_{BG} was measured as 1.70 ± 0.046 eV, a change of ~ 0.09 eV. At the relatively low doping percentage of 5.56%, this is ~ 0.15 higher than the pristine MoSe₂ value of ~ 1.55 eV found in other studies [78, 234]. Zhang *et al* calculated via DFT a band gap energy of 1.65 eV for both 2.7% and 5.4% Se doped MoS₂. This value is close to our experimental value of 1.70 eV. This verifies that low energy Se ion implantation can be used to change the band structure of monolayer MoS₂, and to control the band gap energy

4.1. Se Ion Implantation in MoS₂

as carried out by other methods [65, 78]. Not only has the band gap energy been changed by Se implantation, but also the values of the exciton A, B, and C (split) exciton peaks have shifted. Indeed, the energy separation between the A and B exciton peaks increases in the implanted monolayer. It is likely that a desired energy transition can be achieved by doping at different Se percentages, as carried out in other studies [272].

4.2 Au Ion Implantation on TMDs

Low energy ion implantation of the transition metal site was attempted with Au ions at 10 and 25 eV. As shown in recent studies, Au was used to substitute the transition metal site after incorporation via chemical vapour transport (CVT) [50]. Here however, Au nanoparticles (NP) formed on the surface of both MoS₂ and MoSe₂ after 10 eV and 25 eV ¹⁹⁷Au⁺ ion implantation. The ion-beam fluence for all Au implantations was 1×10^{15} ions cm⁻². More details on the ADONIS implanter are found in Subsection 3.2(Low Energy Ion Implantation). The relatively low implantation energies of 10 and 25 eV resulted in the TMDs acting as a type of net in which the incoming Au ions were caught. Figure 4.28 shows both outcomes: the originally intended substitutional implantation in Figure 4.28(a) and the actual outcome of Au NPs in Figure 4.28(b). Single substitutional Au implants may be present in the lattice, but clear evidence was not found. The formation of Au NPs could occur in TMDs due to an initial substitutional Au atom acting as a nucleation point for the NP. Alternatively, the incoming Au ions may bond strongly with the S lattice atoms, thus resting on the surface and forming NPs.

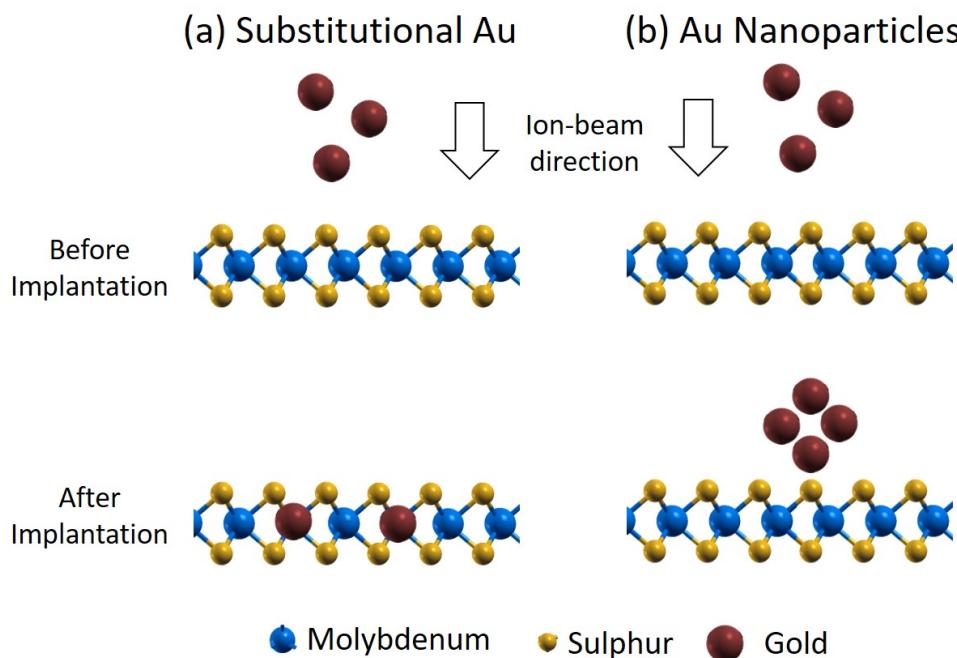


Figure 4.28: Cross-sectional graphic of ¹⁹⁷Au⁺ ion implantation forming (a) substitutional or (b) nanoparticles (NPs) on a monolayer TMD before and after implantation.

high-resolution scanning transmission electron microscopy (STEM) was used to analyse the effect of 10 eV and 25 eV Au implantation of few-layer transition metal dichalcogenides (TMDs). Imaging was carried out in the University of Limerick in

4.2. Au Ion Implantation on TMDs

a Titan Themis microscope at 80 keV, unless otherwise stated. The annular bright field (ABF) and high-angle annular dark field (HAADF) inner and outer detector angles were 13-22 mrad and 103-200 mrad, respectively.

The large atomic number of Au ($Z=79$) makes it trivial to identify Au NPs on either few-layer MoS₂, few-layer MoSe₂, or the carbon quantifoil substrate, due to the Z-contrast nature of ADF STEM imaging. The following sections will investigate the distribution of the Au NPs on implanted MoSe₂ and carbon substrates for 10 eV Au ion implantation. The Ilastik segmentation program [395, 396] was used to gather statistics, as described in subsection 3.4.5(Nanoparticle Identification). No discernible difference was found between the 10 eV and 25 eV implantations, as identical ion beam total doses were used. Following this, high-resolution image simulation is utilised to further understand the size, shape and orientation of individual Au NPs on the TMD surfaces.

4.2.1 Distribution of Au NPs on TMDs

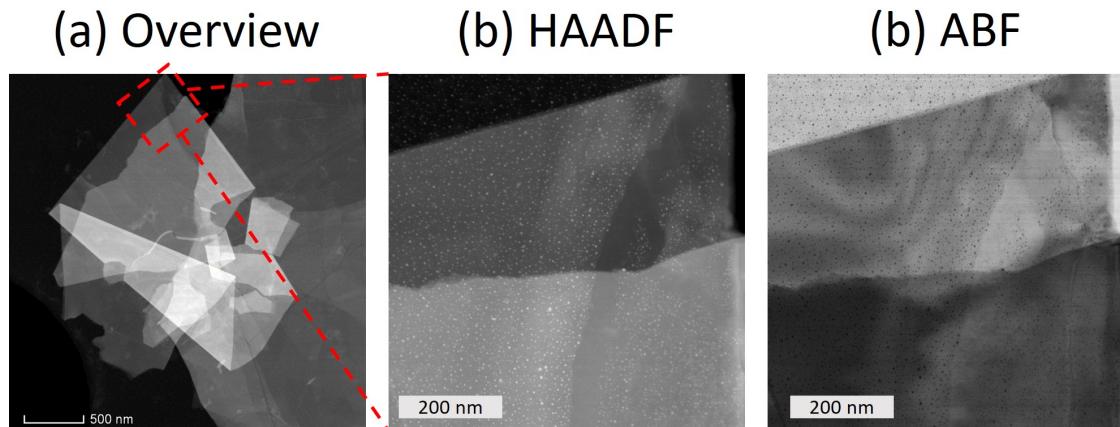


Figure 4.29: (a) Low magnification HAADF STEM image of 10 eV Au implanted MoSe₂ flakes. The dashed red rectangle is magnified in (b, c). (b) HAADF and (c) ABF image of Au implanted MoSe₂ flakes over the carbon quantifoil substrate (left) and suspended over vacuum (right).

10 eV Au implantation of 2D MoSe₂ flakes resulted in the formation of Au nanoparticles (NPs). This section investigates how the Au ion implantation effected different MoSe₂ flake thicknesses, as well as the carbon quantifoil substrate. An overview of different layer thicknesses of implanted MoSe₂ and the carbon substrate is presented in Figure 4.29. The low magnification high-angle annular dark field (HAADF) scanning transmission electron microscopy (STEM) overview image in Figure 4.29(a) shows a typical section of the mechanically exfoliated MoSe₂ samples. The sample preparation method is described in Subsection 3.1. The

4.2. Au Ion Implantation on TMDs

MoSe₂ flakes are often folded upon themselves as shown in Figure 4.29(a). The layer thicknesses in the overview range from 2/3 layers up to 9/10 layers in the brightest areas.

The area highlighted by the dashed red rectangle is magnified in Figure 4.29(b) and Figure 4.29(c) as a HAADF and an annular bright field (ABF) STEM image. In Figure 4.29(b) and Figure 4.29(c) the Au NPs appear as bright and dark circular spots on the image, respectively. Figure 4.29(b) is split into three bands of intensity: The dark top region is the carbon quantifoil substrate, the middle region is a thin area of MoSe₂ (~3-4 layers), and finally the lower region is a thicker area of MoSe₂ (~4-5 layers). The contrast is inversed in the ABF image in Figure 4.29(c), making the Au NPs easier to identify in the top carbon region. The right side of the MoSe₂ flakes are suspended over vacuum, and therefore have a different contrast from the rest of the adjoining flake. A thin region of vacuum is visible at the top-right area of both the HAADF and ABF images.

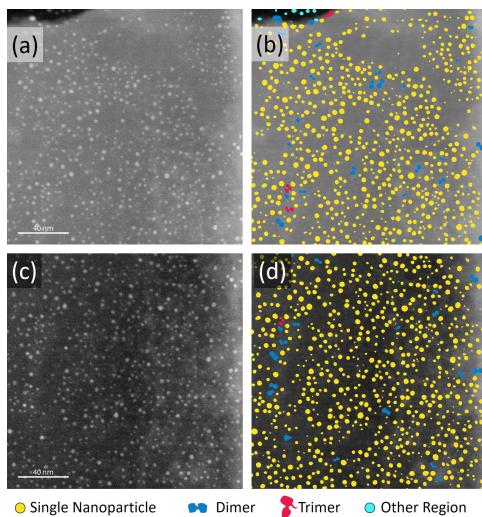


Figure 4.30: ADF STEM images (a,c) before and (b,d) Au NP identification with the Ilastik program. The different shapes identified were grouped into four main categories: Single Nanoparticle (yellow), Dimer (blue), Trimer (red), and NPs from regions not analysed in the current frame (turquoise).

the NPs and the substrate. For this reason, only single NPs (yellow) are included in the size distribution analyses below. Any NP in a region with a much darker or brighter background than the average background is treated as a different substrate and so is labelled as ‘other’ (turquoise). An example of this is seen in the top

The Ilastick segmentation program was utilised to analyse the distribution of the Au NPs on the MoSe₂ flakes and the carbon substrate. Two examples of the NP labelling is shown in Figure 4.30. The raw ADF STEM images in Figure 4.30(a,c) are segmented and each NP identified and grouped according to its morphology. The output of each is shown in Figure 4.30(b,d). In these images the material on which the NPs lie is ~4-5 layer MoSe₂ suspended over vacuum.

The most common morphology is single NPs, marked by the yellow circles in Figure 4.30(b,d). The dimer (blue) and trimer (red) labelled particles are either due to the low mag overview imaging carried out on the samples or due to the overlap in intensity between

4.2. Au Ion Implantation on TMDs

left area of Figure 4.30(b). The Ilastik analysis routine used during this project is described in subsection 3.4.5(Nanoparticle Identification).

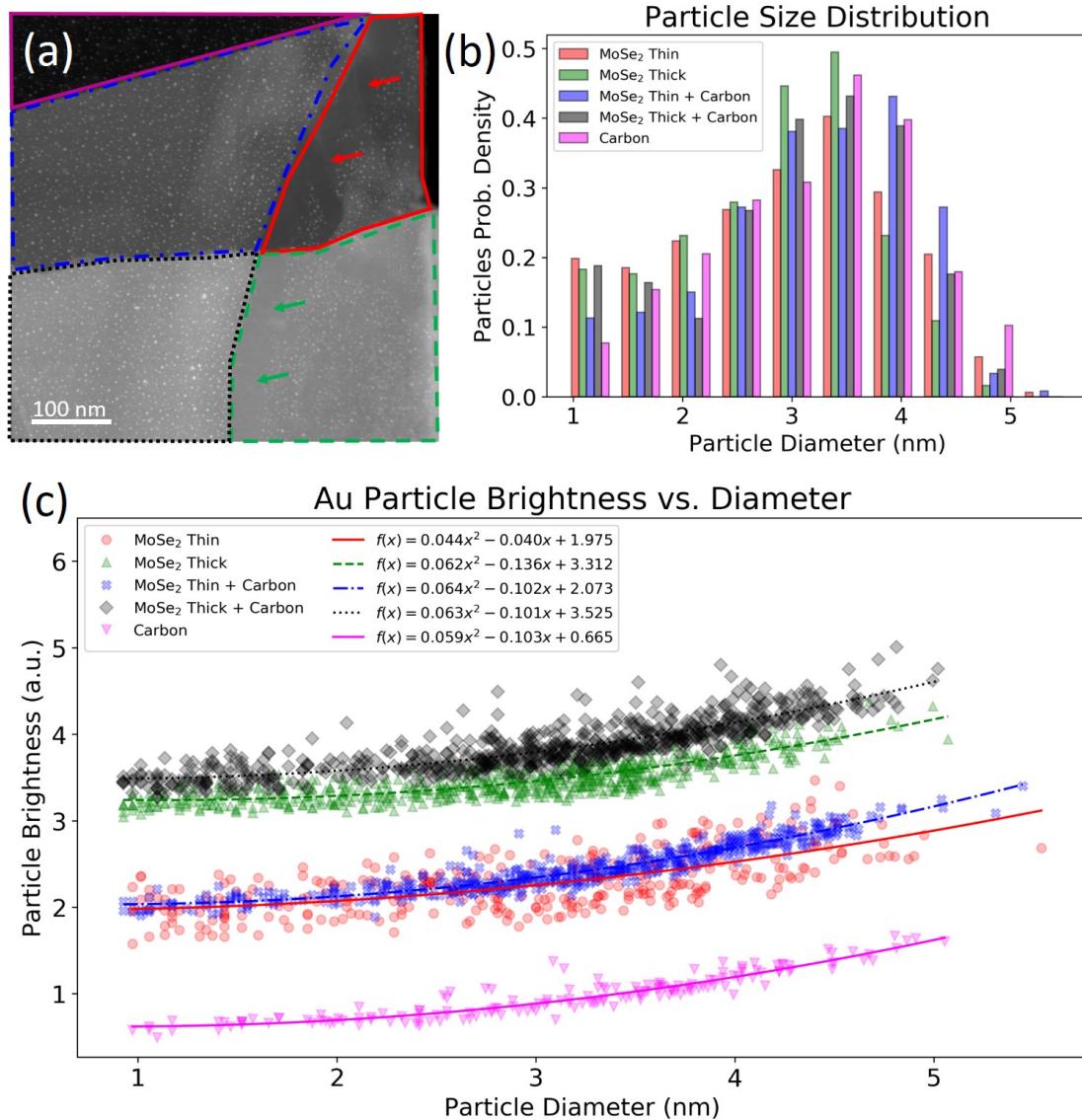


Figure 4.31: (a) Five regions of the ADF STEM image shown above in Figure 4.29. (b) Probability density and (c) ADF intensity brightness of the Au NPs diameter for each region in (a).

Figure 4.31(a) shows the HAADF STEM image from Figure 4.29(b) above split into its respective regions: thin MoSe₂ (solid red border), thick MoSe₂ (dashed green border), thin MoSe₂ on the carbon substrate (dash-dot blue border), thick MoSe₂ on the carbon substrate (dotted black border), and the carbon substrate (solid pink border). These colours and patterns are used consistently throughout Figure 4.31.

The arrows in the thin and thick MoSe₂ regions point to a lack of Au NPs in those areas. The suspended thin and thick MoSe₂ regions lie lower than their

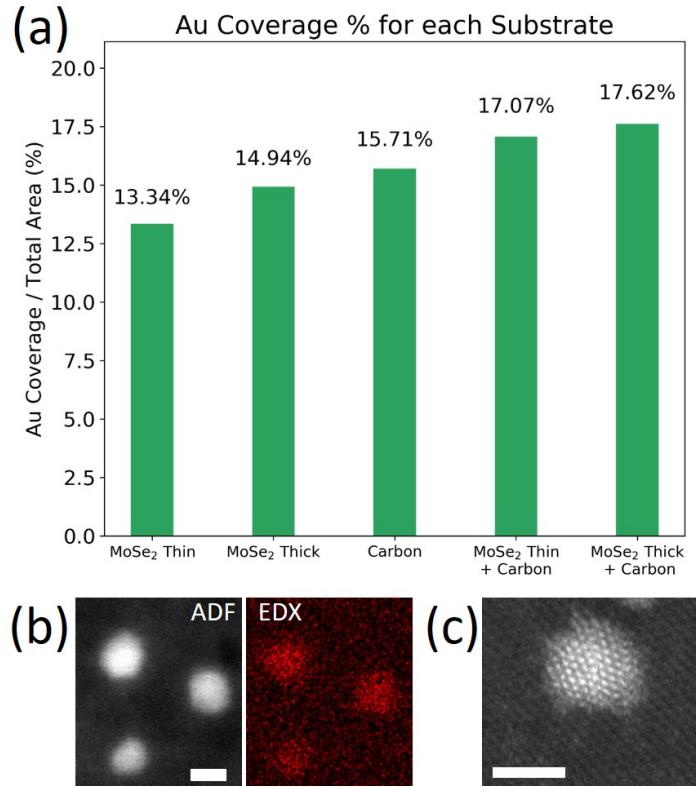


Figure 4.32: (a) Percentage coverage of Au NPs after 10 eV implantation into different thicknesses of MoSe₂ flakes. The five regions are presented in Figure 4.31. (b) Simultaneously acquired ADF image and EDX Au spectrum of three Au NPs on a thin (3-4 layer) MoSe₂ flake. (c) Atomic resolution ADF image taken at 300 kV of an Au NP on a thin (3-4 layer) MoSe₂ flake. Scale bars are 2 nm.

connected flakes on the substrate. Therefore, the lack of NPs is possibly due to screening/shadowing of the incoming ions during implantation. Figure 4.31(b) presents a histogram of the Au NPs for each of the five regions in Figure 4.31(a). For clarity, the y-axis is the probability density of the count of the NPs, rather than the raw number counted at each diameter. Overall, the trend is consistent between the regions, though the suspended MoSe₂ (red and green) are skewed to the left. This is partly due to surface contamination as, in some cases, the bright blobs of surface contamination is identified by the Ilastik program as a small Au NP. For context, the average NP diameter for all regions is 3 nm, with a median of 3.13 (see Table 4.4). Figure 4.31(c) shows the relationship between the NP's maximum HAADF intensity (brightness) and the particle diameter. The NP's intensity in each region rises gently as the NP diameter increases. The second-order polynomial equation for each region is shown in the legend. The trends verify that the carbon substrate has no major effect on the Au NP identification.

The density of Au from each region in Figure 4.31(a) above is shown in Figure 4.32(a). The Au density rises as the substrate thickness increases. However,

4.2. Au Ion Implantation on TMDs

Table 4.4: Mean, median and mode diameter (nm) and surface area coverage (%) of the Au NPs in each region shown in Figure 4.31(a).

Region of Figure 4.31(a)	d_{mean}	d_{median}	d_{mode}	$A_{surface}$
MoSe ₂ Thin	2.94	3.06	3.3	13.34
MoSe ₂ Thick	2.86	3.01	3.5	14.94
MoSe ₂ Thin + Carbon	2.86	3.01	3.5	17.07
MoSe ₂ Thick + Carbon	3.18	3.28	2.9	17.62
Carbon Substrate	3.17	3.3	3.3	15.71

this could be exaggerated in bright substrates by over-estimation of Au NP sizes; the surrounding bright pixels are more likely to be included in the NP than on darker substrates. Figure 4.32(b) shows an ADF image and EDX spectrum of three Au NPs on the thin MoSe₂ flake suspended over the vacuum. An Au NP of ~ 2 nm diameter at atomic resolution is displayed over the same region in Figure 4.32(c). The type of Au NP is discussed below in subsection 4.2.2(Comparison of Au Implants with simulation).

Table 4.4 displays the mean, median and mode diameter of the Au NPs formed via 10 eV ion implantation into MoSe₂. It also shows the Au surface area coverage presented above in Figure 4.32(a).

4.2.2 Comparison of Au Implants with Image Simulation

Image simulations were completed for Au₁₄₇ and Au₅₅ NP sizes with different facets in contact with the MoS₂ monolayer: square, triangular, and point orientations. Symmetrical face-centred cubic (FCC) “magic number” Au NPs are used in the simulations [415, 416]. Figure 4.33(b, c, d) show square-facet, triangular-facet, and point contacted Au NPs on monolayer MoS₂, respectively. There are, of course, many other possible orientations for Au NPs on TMD surfaces. These orientations were chosen are a starting reference. For reference, an experimental image of a Au NP of similar size to Au₁₄₇ and Au₅₅ NPs is shown in Figure 4.33(a, ii), while Figure 4.33(a, iii) shows the relative size of this NP with respect to other Au NPs on the monolayer MoS₂ surface. Figure 4.33(a, i) displays the results of three line intensity profiles traced across the Au NP in Figure 4.33(a-ii). Similar line profiles are also completed in Figure 4.33(b-d, i) for the simulated images of Au₁₄₇ and Au₅₅ NPs. These line intensity profiles were traced across their respective images along different facets, resulting in clear differences in the intensity levels and profiles between each contact orientation.

Figure 4.33(b, i) shows a steep sided intensity profile for both Au₅₅ and Au₁₄₇,

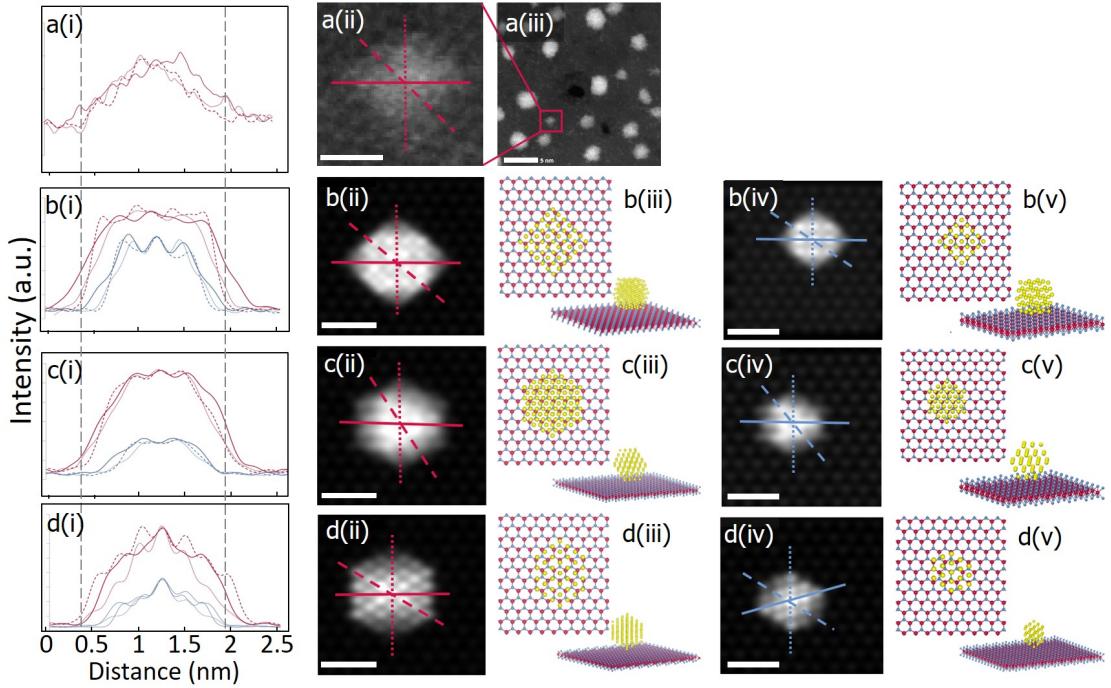


Figure 4.33: (a, i) Intensity profile of the experimental ADF STEM image in (a, ii). (a, iii) Low magnification overview of Au NPs on monolayer MoS₂. The red rectangle is magnified in (a, ii). (b-d, i) Intensity profiles of the simulated ADF STEM images in (b-d, ii) and (b-d, iv). (b-d, ii) Simulated images of the models in (b-d, iii). (b-d, iv) Simulated images of the models in (b-d, v). (b-d, iii) and (b-d, v) Plan-view (top-left) and 3D view of the Au NP on monolayer MoS₂. Scale bar of (a, iii) is 5 nm, all other scale bars are 1 nm.

implying a consistent thickness across the NP. The line profile of Figure 4.33(c, i) has a rounded profile when compared with Figure 4.33(b, i). This rounded shape, along with the higher intensity in Figure 4.33(c, i), hint towards the NP being thicker in the centre than at the edges. Comparing Figure 4.33(c, iii) and Figure 4.33(c, v), this pattern agrees with the crystal structures. If the triangular facet is contacted to the substrate, the maximum crystal width is larger than its contacting surface, resulting in overhanging sections not in contact with the substrate surface. This accounts for the less intense edges in Figure 4.33(c, ii) and Figure 4.33(c, iv). Figure 4.33(d, i) shows a steady increase in the intensity from the outer edge to centre of the point-contacted NP, indicating the NP has a smaller contact area with the substrate compared to Figure 4.33(b) and Figure 4.33(c). The sharp peak at the centre of the profile marks the thickest region of the crystal.

Au₁₄₇ in a point-contacted orientation [Figure 4.33(d, ii)], and triangular contact orientation [Figure 4.33(c, ii)] have ~1.55 and ~1.22 times the peak intensity of Au₁₄₇ in a square contact orientation [Figure 4.33(b, ii)]. This infers that for NPs of the same diameter range, lower intensity NPs are more likely to have a greater

4.2. Au Ion Implantation on TMDs

contacting surface with the substrate. In this example, the experimental Au NP in Figure 4.33(a) is best-matched with the point-contact oriented simulation of the Au_{147} NP shown in Figure 4.33(d, ii). Therefore, by combining the relative intensity with the line profile shape, one can deduce the contact orientation of Au NPs on thin TMDs.

4.3 Chapter Conclusions - Atomic Characterisation of Ion Implanted 2D TMDs

Low energy ion implantation into 2D TMDs has been shown for both Se (substitutional) and Au (nanoparticle formation). ADF STEM, image simulation and EELS were employed to verify the effects of ion implantation on the surface of monolayer and few-layer MoS₂ and MoSe₂. The varied results from 10 eV Se and Au ion implantation show how vital it is to investigate the atomic structure of such implanted materials, but also the wide range of possibilities low energy ion implantation can provide.

Se ion implantation has been verified to create substitutional anti-sites and vacancies in monolayer MoS₂, as was discussed in detail in subsection 4.1.1(Microscopy of Se implanted MoS₂). Image simulation verified the Z-contrast intensity arising from single substitutional Se atoms and vacancies. The energy required to create anti-sites and vacancies was calculated via DFT [work completed by collaborators, see Figure 5.13 in subsection 5.2.2(Analysis of Lattice Energetics)]. Single atom chemical analysis via core-loss EELS, coupled with image simulation, confirmed successful Se implantation in Subsection 4.1.2. In Subsection 4.1.2, low-loss EELS showed the change in band gap and exciton energies in monolayer and few-layer MoS₂ due to Se implantation. For the MoS_{1.87}Se_{0.13} alloy created (5.56% Se doping), the band gap changed by 0.09 eV. Overall, Se ion implantation into a monolayer TMD has been successfully demonstrated. It was used to manipulate the band structure of MoS₂, similar to other doping methods, such as CVT [50] and CVD [78].

Au ion implantation at low energies provides a method for depositing Au nanoparticles (NPs) on the surface of monolayer and few-layer material without causing lattice damage, as verified by ADF STEM in Subsection 4.2. It is expected by increasing the total ion dose (fluence), one can increase the size of the Au NPs, and vice versa. The most exciting possible use of Au implantation on 2D TMDs is for truly nano-scale contacts for various electronic devices. The Au NPs were confirmed to be conductive on the TMD monolayer surface via DFT calculations. By controlling the ion beam, one can envisage patterning of conductive, ultra-thin 1-5 nm Au contacts on the monolayer and few-layer material surface.

Chapter 5

Results - Atom Dynamics in 2D TMDs

In this chapter we discuss the development of methods for automated analysis of atomic resolution images, and the application of these methods to movies of Se implanted MoS₂. In Subsection 5.1, two algorithms based on intensity comparison between experiment and fast image simulation are explained. These algorithms were utilised for the single frame images analysed in Subsection 4.1(Se Ion Implantation). Then, the Model Refiner Python class (MR tool) is examined. The MR tool is used to track the automated analysis via the above-mentioned algorithms. In Subsection 5.2(Atom Dynamics in Implanted TMDs with the MR Tool), the MR tool is used to analyse vacancies, hole formation and adatoms in movies of Se implanted MoS₂, as well as the detrimental effect of the electron beam.

5.1 Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

Currently, automation of atomic resolution images of simple structures, such as SrTiO₃, can be completed with Atomap [4]. However, to achieve a truly one-click analysis of more complex images and movies, further development is required. The initial atom finding must become robust enough to take into account all variations in the lattice for simple and complex structures, and must be flexible enough to identify edge cases, such as adatoms and vacancies in a TMD monolayer. Automation of the refinement procedure for movies must be developed for holes (e.g., in monolayers). Tools are developed for these procedures, and all steps outlined below are run using a combination of the Python packages described in subsection 3.4.10 and original code developed during this thesis. All scripts, functions, classes, mod-

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

ules and contributions to open-source packages completed during this thesis are included in the TEMUL toolkit GitHub repository [13] (TEMUL: Transmission Electron Microscopy University of Limerick).

5.1.1 Refinement of Atomic Resolution Images via Image Simulation

Deviations between experimental and simulated image data are sometimes unavoidable [385]. However, large deviations can be used to automatically change the initial experimental atomic model to one which reduces the deviations. Overall, this will reduce error between the images and, when handled correctly, increases the accuracy of the atomic model. To implement this, two algorithms were developed: an intensity refinement algorithm (`image_difference_intensity` function) and a position refinement algorithm (`image_difference_position` function). In the following subsections, the methodology of each algorithm is outlined while completing an example implementation with test data (columns of Ti atoms). Furthermore, the `Model Refiner` object, within which the user accesses and runs the model refinement, data tracking, and data visualisation, is examined in subsection 5.1.2(Model Refiner (MR) Tool). In Subsection 5.1.2(Se implanted monolayer MoS₂), the algorithms are applied to real data. All code can be found on GitHub [13].

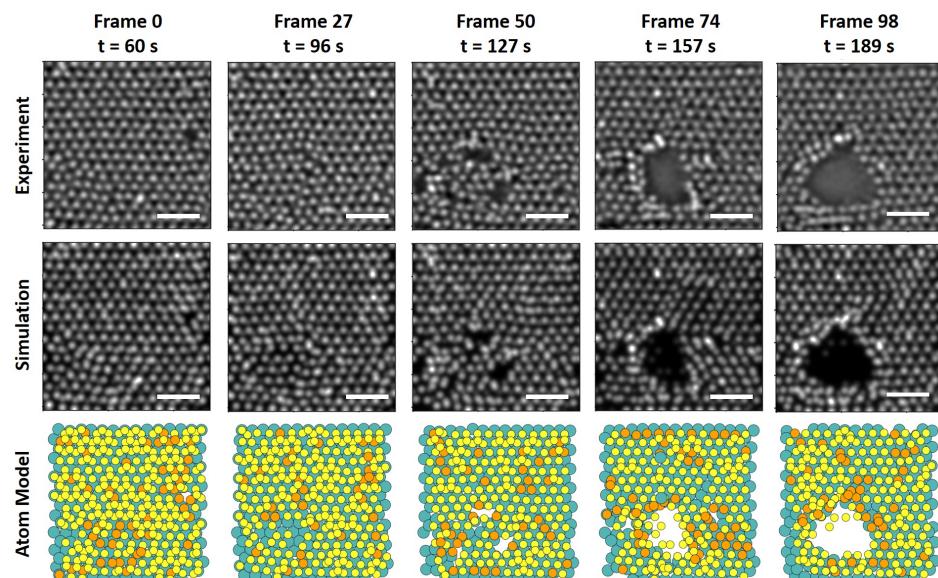


Figure 5.1: The DG filtered Experimental image, final Simulated image, and final Atom Model for five frames of an image series of Se implanted MoS₂, analysed using the algorithms and tools designed in this chapter. Scale bars are 1 nm.

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

To preview the capabilities of the designed refinement algorithms and Model Refiner tool, five frames (0, 27, 50, 74, 98) from a movie of Se implanted MoS₂ are displayed. The filtered experimental and simulated images, as well as the final atom model for each frame are shown. This image series was automatically analysed using the algorithms and tools developed in this chapter. The full list of steps in this process is given in subsection 5.2.1(Image Series Automated Analysis Steps). As explained in Subsection 3.4(Signal and Data Analysis), text with a grey background indicates a code function or class. In the following sections, an ‘element list’ refers to a list of atomic column element configurations. An element configuration example would be Se₁S₁: a Se atom above a S atom. An element list then is several of these: [Se₁S₁, Se₁, S₂ etc.]. These element configurations are shown graphically in Figure 4.2 in Subsection 4.1(Se Ion Implantation in MoS₂).

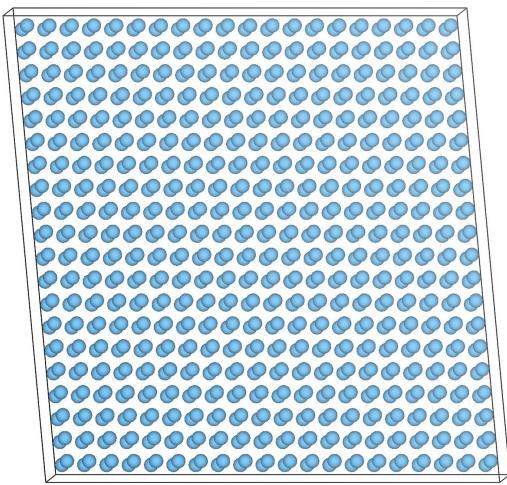
Intensity Refinement Algorithm

In this section, the process behind the intensity refinement algorithm used within the `image_difference_intensity` function is outlined and discussed.

1. Two images are compared - reference (experimental) and comparison (simulated) images in this case - using the `image_difference_intensity` function. The reference image will be part of an Atomap sublattice object, as seen in Figure 5.3(a). The simulation images for each refinement are plotted in Figure 5.3(b:1-3).
 - (a) These images should be spatially and intensity calibrated before refinement. The intensity calibration can be an internal reference - such as a consistent atomic column e.g., single Mo atoms in monolayer MoS₂, or calibrated using the ADF detector [see subsection 2.1.1(ADF STEM)].
 - (b) The difference between the images is found by subtracting the simulated image from the experimental image. For each example refinement in Code Segment 5.1, the difference images between the sublattice and each simulation image are shown in Figure 5.3(c:1-3).
2. A temporary difference sublattice is created containing:
 - (a) Image data from the difference image.
 - (b) Atom position data from the original sublattice. The atom positions are overlaid on the difference image data as shown in Figure 5.3(c:1-3).
3. The intensity of each atom position in the temporary sublattice is calculated with the `get_sublattice_intensity` function. The mean, median, max, min, or total (summed) techniques can be used.
4. In the histogram of these measured intensities, the mean (red dot) and first four standard deviations (green dots) of the histogram intensities are calculated, as seen in Figure 5.3(d:1-3)
5. The intensities outside a user specified number of standard deviations are placed in two arrays:
 - (a) One array for too-bright intensities resulting from the model underestimating the combined atomic number at that position. In other words, the simulation is too dark at that position.

- (b) One array for too-dark intensities resulting from the model overestimating the combined atomic number at that position. In other words, the simulation is too bright at that position.
6. The arrays are passed to `change_sublattice_atoms_via_intensity` for sorting.
- (a) For the too-bright intensities, the next largest element configuration in the sublattice's `element_list` will be assigned to the elements at the too-bright atom position in the sublattice.
 - (b) For the too-dark intensities, the next smallest element configuration in the sublattice's `element_list` will be assigned to the elements at the too-dark atom position in the sublattice.
7. If specified by the user, the temporary difference sublattice and a histogram of the intensity values will be plotted for later inspection, as it can show issues with intensity values. An example of each after each refinement are shown in Figure 5.3(c-d:1-3).

(a) Before Refinement



(b) After Refinement

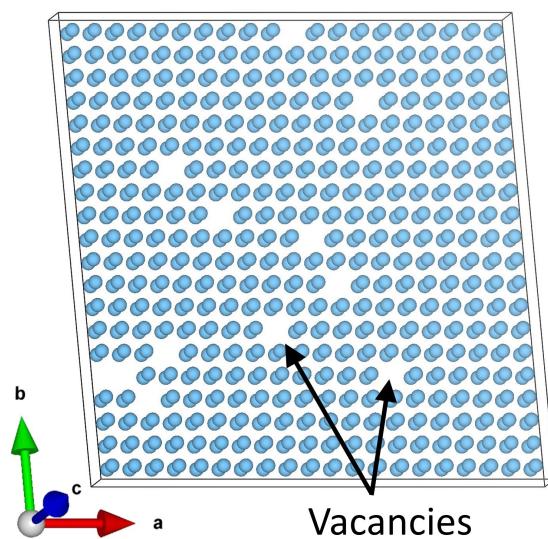


Figure 5.2: 3D render of a test atomic model (a) before and (b) after intensity refinement shown in Figure 5.3. The vacancies were found by the Model Refiner tool using the `image_difference_intensity` function.

The class object `Model_Refiner` is used to refine one or many sublattice objects, each with their own `element_list` (described in the introduction to this section). The `image_difference_intensity` function described above is accessed by

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

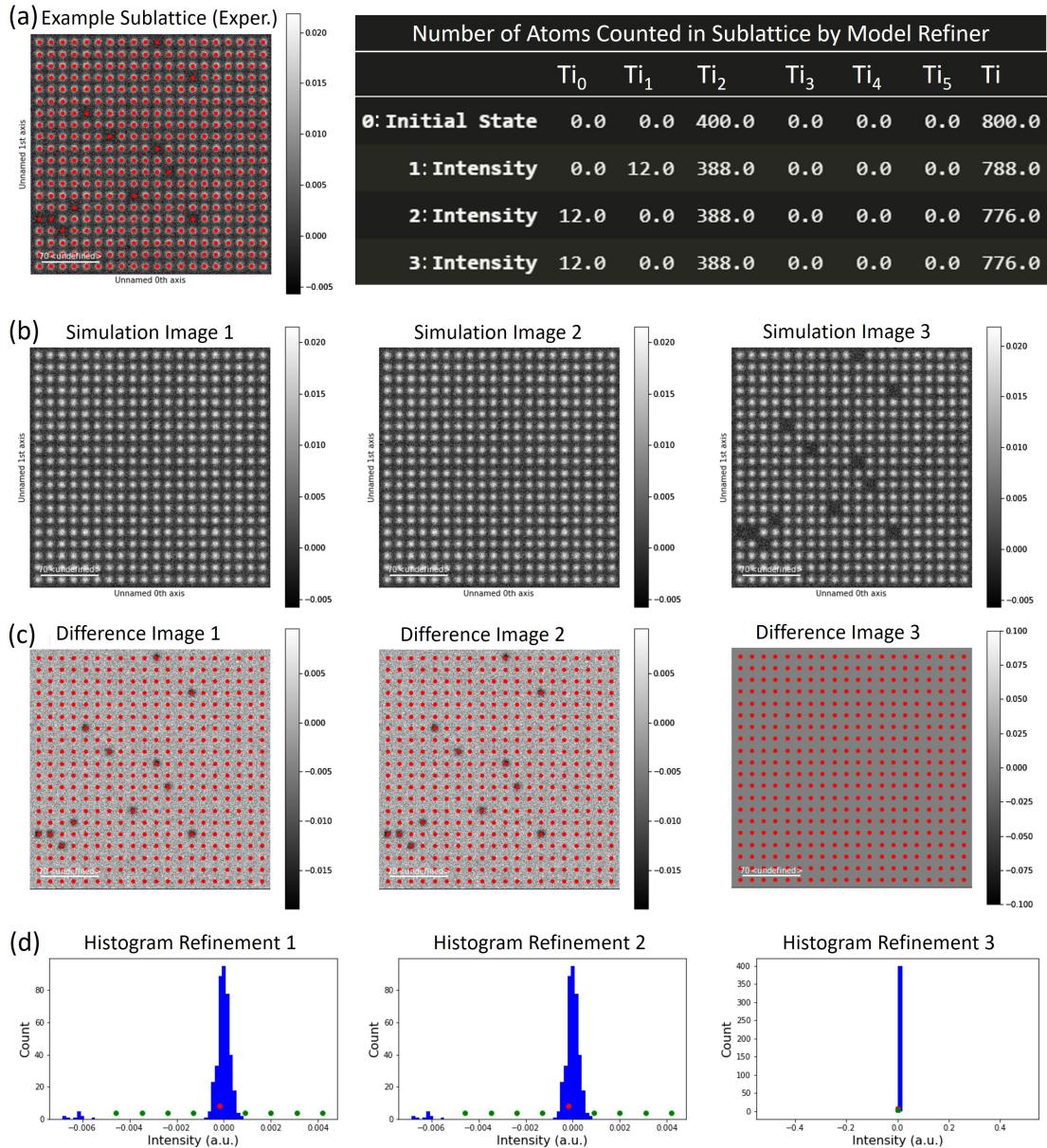


Figure 5.3: (a) Example cubic Ti sublattice (test experimental data) with several vacancies. All atom positions are labelled with the Ti₂ element configuration as seen in the '0: Initial State' row of the table 'Number of Atoms Counted in Sublattice by Model Refiner'. (b) Simulation images for each intensity refinement. (c) The resulting temporary difference sublattices for each intensity refinement. These are the sublattice image minus the simulation image. Simulation Image 3 can be thought of as a correct simulation of the test experimental data in (a). (d) The distribution of the atom position intensities of each difference image in (c). The mean (red dot) and first four standard deviations (green dots) are plotted on the histograms for reference.

calling the `model_refiner.image_difference_intensity_model_refiner` class method (a class method is a function). An example of this method is shown in Code Segment 5.1 and produces the images in Figure 5.3. In this example the atomic columns are each made up of two titanium atoms (Ti_2). One titanium atom is given as Ti_1 , zero as Ti_0 etc. There are 12 atomic columns with no Ti atoms i.e., Ti_0 vacancies. The `image_difference_intensity` function is used in this example to match an image simulation to the experimental image (Example Sublattice). The initial and final 3D atomic model of this basic example is shown in Figure 5.2.

Figure 5.3(a) shows an example test sublattice and represents an experimental image. Figure 5.3(b) shows three separate simulations of the example sublattice in Figure 5.3(a). Figure 5.3(b - Simulation Image 1) does not reflect the experimental sublattice in Figure 5.3(a), as it does not contain vacancies. The aim of the intensity refinement is to correct this. By iteratively comparing the example sublattice in Figure 5.3(a) with image simulations in Figure 5.3(b), the final correct simulation is created in Figure 5.3(b - Simulation Image 3). The Difference images in Figure 5.3(c) are the example sublattice image minus each simulation image. This shows the vacancies as dark regions in Figure 5.3(c - Difference Image 1) and Figure 5.3(c - Difference Image 2), as well as in the histogram in Figure 5.3(d). Finally, Figure 5.3(c - Difference Image 3), reflects the match between the Figure 5.3(a) and Figure 5.3(b - Simulation Image 3). This is an exaggerated example, as having perfect matching between experimental image and simulation image is unrealistic.

The log which tracks all of the Ti atomic columns is shown in the table in Figure 5.3(Number of Atoms Counted...). This is a table of the counts of each atomic column in the sublattice after each refinement. For example, at the beginning of the process (Initial State) there are 400 Ti_2 atomic columns detected, which equates to 800 single Ti atoms. By refining the sublattice with the intensity algorithm (1: Intensity, 2: Intensity etc.), the vacancy locations are reflected in the simulation images. The integers in the row names (0, 1, 2 etc.) show the order of the refinement, and the word (Initial State, Intensity etc.) show the refinement algorithm used at each step. This data log can be accessed via `get_element_count_as_dataframe`. Additionally, this data is visualised in Figure 5.4 by calling `plot_element_count_as_bar_chart` which shows the count of each element configuration after each refinement. The term ‘element configuration’ is explained graphically in Figure 4.2 in Subsection 4.1(Se Ion Implantation into MoS_2). The count of each element configuration is shown as a function of intensity refinement and element configuration in Figure 5.4(a) and Figure 5.4(b),

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

respectively.

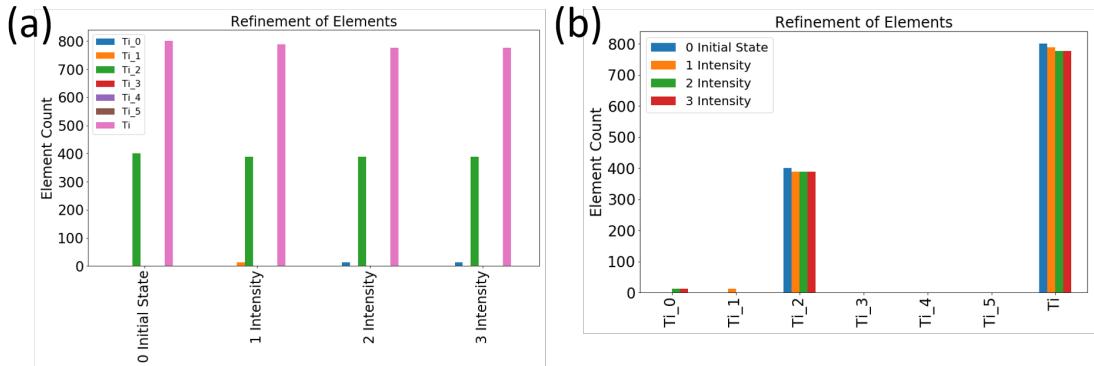


Figure 5.4: Change of the element configurations as a function of (a) the intensity refinement and (b) the element configurations. The simple method `refiner.plot_element_count_as_bar_chart` in Code Segment 5.1 will produce these plots.

```

1 # Import the necessary packages and functions
2 from temul.dummy_data import
3     get_model_refiner_with_vacancies_refined
4
5 # Create the example model_refiner object, which will perform 3
6     # intensity refinements automatically
7 refiner = get_model_refiner_with_vacancies_refined(
8     image_noise=True, test_element="Ti_2",
9     filename="Refine Example")
10
11 refiner.sublattice_list[0].plot()
12 Out: Figure 5.3(a)
13
14 refiner.comparison_image.plot()
15 Out: Figure 5.3(b)
16
17 # inspect the refinement history directly
18 refiner.element_count_history_list
19 Out: [Counter({'Ti_2': 400}),
20       Counter({'Ti_2': 388, 'Ti_1': 12}),
21       Counter({'Ti_2': 388, 'Ti_0': 12}),
22       Counter({'Ti_2': 388, 'Ti_0': 12})]
23
24 # inspect the refinement history with a pandas dataframe
25 refiner.combine_individual_and_element_counts_as_dataframe()
26 Out: Figure 5.3(a, right)
27
28 # notice the flip_colrows parameter

```

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

```
27 refiner.plot_element_count_as_bar_chart(  
28     element_configs=2, flip_colrows=False, fontsize=24)  
29 Out: Figure 5.4(a)  
30  
31 refiner.plot_element_count_as_bar_chart(  
32     element_configs=2, flip_colrows=True, fontsize=24)  
33 Out: Figure 5.4(b)
```

Code Segment 5.1: The element configurations of a dummy cubic structure with vacancies refined by comparison with an dummy simulation via the intensity refinement algorithm. The outputs (Out:) for each step can be seen in both Figure 5.3 and Figure 5.4.

Position Refinement Algorithm

In this section, the process behind the position refinement algorithm used within the `image_difference_position` function is outlined.

1. Compare two images - experimental and simulated in this case - using the `image_difference_position` function. The experimental image will be part of an Atomap sublattice object (Figure 5.5(a)). The simulation image for a test dataset is plotted in Figure 5.5(b).
 - (a) These images should be spatially and intensity calibrated before refinement. The intensity calibration can be an internal reference - such as a consistent atomic column e.g., single Mo atoms in monolayer MoS₂, or calibrated using the ADF detector [see subsection 2.1.1(ADF STEM)].
 - (b) The difference (diff) between the images is found by subtracting the simulated image from the experimental image, and vice versa for diff_inv. For the example position refinement in Code Segment 5.2, the difference images between the sublattice and the simulation image are shown in Figure 5.5(d).
2. Temporary difference sublattices are created containing:
 - (a) Image data from the difference images.
 - (b) Atom position data detected from the difference images. Bright regions in the difference images will be chosen as new atom positions. The new atom positions depend on the parameters chosen such as the number of new positions allowed (num_peaks), and the pixel separation from current sublattice positions (pixel_threshold).
3. The allowed new atom positions will be added to the sublattice without being assigned an element configuration. The elements can be assigned via the general techniques or via `image_difference_intensity`.
 - (a) The new atom positions can also be added to a new sublattice if desired using the `image_difference_position_new_sub` function.
4. If specified by the user, the newly refined sublattice and the difference images will be plotted for later inspection as shown in Figure 5.5(c-d).

The class object `Model_Refiner` is used to refine one or many sublattice objects, each with their own `element_list`. The `image_difference_position` function described above is accessed by calling the

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

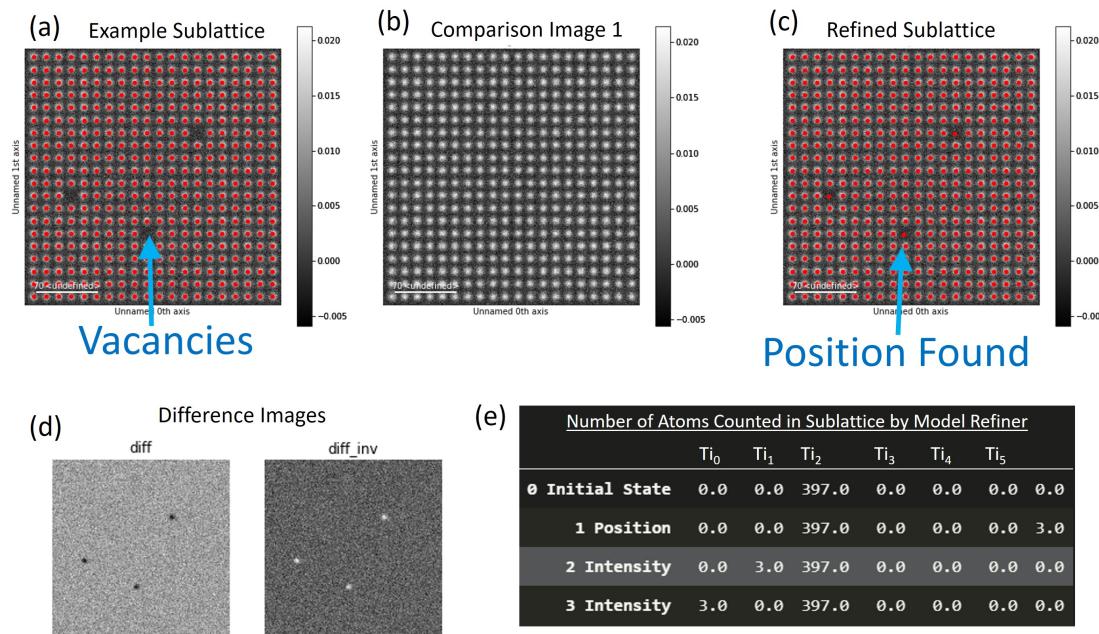


Figure 5.5: (a) Example of cubic Ti sublattice with three vacancies (one labelled with blue arrow). All atom positions are labelled with the Ti_2 element configuration as seen in the ‘0 Initial State’ row of the (e) Refinement History table. (b) Example simulation image. Results of each comparison between the example sublattice and simulation image are tabulated in (e). (c) The resulting sublattice after the position refinement (one vacancy position is labelled with blue arrow). (d) The difference (diff) image resulting from subtracting the (b) simulation image from the (a) example sublattice image, and vice versa for diff_inv. (e) Table of the element refinement after one position refinement and two intensity refinements.

`model_refiner.image_difference_position_model_refiner` class method (function). This function is used in the dummy data in Code Segment 5.2 to produce the images in Figure 5.5. Figure 5.5(a) is an example test sublattice made up of Ti_2 . There are three vacancies in the sublattice. By comparing it with a second image without vacancies (e.g., a simulation) in Figure 5.5(b), a sublattice with the vacancy positions can be created in Figure 5.5(c). This process works by taking the difference (diff) between Figure 5.5(a) and Figure 5.5(b), as shown in Figure 5.5(d, diff). Figure 5.5(d, diff_inv) is the inverse of the diff image, and is also used to find new positions.

The table in Figure 5.5(e) shows the initial state of the sublattice from Figure 5.5(a). It then shows the new atom positions found by the ‘1 Position’ refinement. These 3 new atom positions are detected under the ‘ Ti ’ column of the table. Two intensity refinements, described in the previous section, assign these vacancies correctly as Ti_0 element configurations.

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

```
1 # Import the necessary packages and functions
2 from temul.dummy_data import
3     get_model_refiner_one_sublattice_3_vacancies
4
5 # Position Refinement
6 refiner = get_model_refiner_one_sublattice_3_vacancies()
7
8 # sublattice and simulation image before refinement
9 refiner.sublattice_list[0].plot()
10 refiner.comparison_image.plot()
11 Out: Figure 5.5(a-b)
12
13 refiner.image_difference_position_model_refiner(
14     pixel_threshold=10, filename="Example")
15 Out: "3 new atoms found! Adding new atom positions."
16 Out: Figure 5.5(c-d)
17
18 # You can also see the sublattice after position refinement with
19 refiner.sublattice_list[0].plot()
20 Out: Figure 5.5(c)
21
22 # Combination of above Position with Intensity Refinements
23 refiner.image_difference_intensity_model_refiner()
24 refiner.image_difference_intensity_model_refiner()
25
26 refiner.get_element_count_as_dataframe()
27 Out: Figure 5.5(e)
28
29 refiner.plot_element_count_as_bar_chart(
30     element_configs=2, flip_colrows=True, fontsize=24)
31 Out: Figure 5.5(f)
```

Code Segment 5.2: The element configurations of a dummy cubic structure with vacancies refined by comparison with an dummy simulation via the position refinement algorithm. The outputs (Out:) for each step can be seen in Figure 5.5

5.1.2 Model Refiner (MR) Tool

In this section the Model Refiner (MR) tool is described. The MR tool is a Python class. In Python, a class is simply an object which has some attributes. A simple class example could be a door, with a colour attribute. The colours themselves can change, blue door or red door, for example, but the colour attribute must still be a colour. Well-designed classes make it easy for a user to access advanced object functionality. For example, the Atomap Sublattice class allows a user to refine thousands of atoms in an image with 2D Gaussians with one line of code.

Table 5.1: The attributes, explanation (Python type), and how to assign the attributes of the Model Refiner. Not all attributes are shown.

Attribute	Explanation (Type)	How to Assign
sublattice_list	sublattice objects to be refined, (Atomap Sublattice)	sublattice_and_elements_dict
element_list	element list used to refine the element configurations in each sublattice (List of Strings)	sublattice_and_elements_dict
comparison_image	image compared with each sublattice image, usually an image simulation (HyperSpy 2D signal)	comparison_image
name	name of the Model Refiner (String)	name
element_count_history_list	Count of the element configurations and elements after each refinement (List of Counters)	Auto-assigned
refinement_history	refinement methods used for each refinement, (List of Strings)	Auto-assigned

In the previous sections the class `Model Refiner` was utilised. After inputting a list of Atomap Sublattices, element lists, as well as an image for comparison (such as a simulation), the `Model Refiner` can refine the elements in each sublattice automatically. The main attributes are tabulated below in Table 5.1. There are many other attributes for the MR tool. These can be examined in the source code on GitHub [13]. The `Model Refiner` class is used for experimental data in Subsection 5.1.2(Using the Model Refiner on Experimental Data) below.

The name, brief explanation, and the action taken by each `Model Refiner` method can be seen in Table 5.2. In Python, a method is simply a function that a class can use. For example, to run the intensity refinement algorithm, one runs the following line: `model_refiner.image_difference_position_model_refiner()`. The functionality of the `Model Refiner` can be accessed easily after instantiation (creating a class instance), as shown in Code Segment 5.3 in Subsection 5.1.2 below. Once the `Model Refiner` has been created, its attributes and methods (functions) can be accessed and run.

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

Table 5.2: The methods, explanations of method, and action taken by each method, of the Model Refiner class. These methods (functions) can be called easily, as seen in Subsection 5.1.2(Using the Model Refiner on Experimental Data).

Method	Explanation	Action/Output
element_count	Counts the number of each element configuration in all sublattices in the sublattice.list.	Updates the .element_count attribute
update_element_count_history	Runs the element_count method and adds the results to the element_count_history_list	Updates element_count_history_list
compare_latest_element_counts	Checks if the previous two element_count_history_list entries are the same	Boolean
get_element_count_as_dataframe	Count the element configurations and display in a table.	Pandas Dataframe
get_individual_elements_as_dataframe	Similar to get_element_count_as_dataframe but with a count of the individual elements	Pandas Dataframe
combine_individual_and_element_counts_as_dataframe	Create pandas dataframes with both the element configurations and individual element counts.	Pandas Dataframe
plot_element_count_as_bar_chart	Configure a suitable bar chart from the above created pandas dataframes.	Pandas Bar Chart
image_difference_intensity_model_refiner	Utilise the image_difference_intensity function to refine the Sublattices.	Changes the element configurations inplace
image_difference_position_model_refiner	Utilise the image_difference_position function to refine the Sublattices.	Adds new atom positions
repeating_intensity_refinement	Utilise the image_difference_intensity function several times to refine the Sublattices.	Changes the element configurations inplace

Using the Model Refiner on Experimental Data

As described so far in this section, the **Model Refiner** can be used to automatically refine the element configurations in a list of Atomap Sublattices. To show how useful this can be for experimental data, one must combine the **Model Refiner** class with annular dark field scanning transmission electron microscopy (ADF STEM) simulations. Ideally the simulations should be fast and be easily integrated into a Python workflow. Both are available in the excellent PyPrismatic Python package. For the experimental data below, the following steps are used:

1. Filter and calibrate an experimental image [see Subsection 3.4(Signal and Data Analysis)].
2. Find the sublattices and assign the elements [see Subsection 3.4(Signal and Data Analysis)].
3. Create a **Model Refiner** object.
4. Iteratively refine the **Model Refiner** with fast image simulations.

To verify that they produce reliable and accurate results, the intensity and position algorithms and Model Refiner class (MR tool) were tested on known models, simulated with both Dr. Probe and Prismatic. First, a basic atomic model was constructed, such as a metal NP (Au, Cu) or a homogeneous FIB lamella-style structure. These models contained many different atomic column thicknesses. For

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

heterogeneous atomic columns, pristine 2D material models were used for testing. In all model cases, the MR tool and algorithms correctly assigned the number of atoms in each atomic column.

Depending on the type of material analysed, the initial element assignment will change. For example, for a nanoparticle (NP), the z-dimension bond length and assumed thickness, or size, of the structure is used to assign the number of atoms in the column. Doing this is rather simple with functions in the `model_creation.py` module of the TEMUL toolkit. These 3D structures can be created from an experimental or simulated image with the TEMUL toolkit, and the relevant functions will be added to Atomap in future. It is worth noting that methods from literature [121] have been implemented recently in Atomap. For 2D materials, the approach to element quantification is different, as described in subsection 3.4.1(Element Quantification in HR ADF STEM Images). Because Atomap is built on HyperSpy and in Python, we expect it to become prominent in the field of ADF STEM element quantification.

Se implanted monolayer MoS₂

To analyse Se ions implanted at low energy into monolayer MoS₂, the above described Model Refiner Class is utilised. After finding the atomic positions via Atomap, one can easily set up the information needed for the Model Refiner, such as image sampling and material thickness, as seen in Code Segment 5.3 below. The atom position finding and refining methods are described in subsection 3.4.3(Methods for Atom Position Analysis).

```
1 # Script Part 1
2
3 s = example_data.load_Se_implanted_MoS2_data()
4 real_sampling = s.axes_manager[-1].scale
5 image_size_z_nm = 1.2294 / 2
6
7 atom_lattice = am.load_atom_lattice_from_hdf5("Atom_Lattice_max.
8     hdf5")
9 atom_lattice.plot()
10 sub1 = atom_lattice.sublattice_list[0]
11 sub2 = atom_lattice.sublattice_list[1]
12 sub3 = atom_lattice.sublattice_list[2]
13
14 # Sublattice elements
15 element_list_sub1=["Mo_0", "Mo_1", "Mo_1.S_1", "Mo_1.Se_1", "Mo_2"
16     ]
17 element_list_sub2=["S_0", "S_1", "S_2", "Se_1", "Se_1.S_1", "Se_2"
18     ]
19 element_list_sub3=["H_0", "S_1", "Se_1", "Mo_1"]
20
21 # dictionary used to construct the Model Refiner Class
22 sub_dict = {sub1: element_list_sub1,
23             sub2: element_list_sub2,
24             sub3: element_list_sub3}
25
26 # Create the Model Refiner
27 refiner = Model_Refiner(sub_dict,
28                         sampling=real_sampling * 10,
29                         thickness=image_size_z_nm * 10,
30                         name="Se Implanted MoS2")
31
32 # Inspect the elements in the Model Refiner
33 refiner.get_element_count_as_dataframe()
34 refiner.plot_element_count_as_bar_chart(2)
```

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

```
33  
34 # Set the calibration area manually  
35 refiner.set_calibration_separation(11)  
36 refiner.set_calibration_area()
```

Code Segment 5.3: Se implanted MoS₂ Model Refiner example part 1. The Model Refiner object is constructed and labelled ‘refiner’. Three sublattices are found via Atomap. The elements present in the image are found and visualised in Figure 5.7(b). The calibration criteria is set as shown in Figure 5.7(c).

The Model Refiner tracks the accuracy of these simulations by comparison with the reference (experimental) image. The trend in Figure 5.6 mirrors that of the improvement of the simulations from left to right in Figure 5.8(b-d). The mean square error (MSE) is reduced and structural similarity index (SSM) increased. The code for automatic simulation and tracking of error and similarity is given in Code Segment 5.4. Both MSE and SSM are described in Subsection 3.4.2(Measuring Error between Images). The user can set a threshold similarity, which would define the point at which the similarity between the comparison image (simulation) and reference image (experimental) is suitably accurate, therefore stopping the refinement process.

As discussed in subsection 5.1.1 (Refinement of Atomic Resolution Images via Image Simulation), the Model Refiner can be used to easily tabulate and visualise the elements in the constructed Atomap sublattices [see Figure 5.7(b)]. If required, the tabulated data can be exported as a ‘.csv’ file for external analysis with widely available software such as Microsoft Excel. A calibration area can be specified if one wishes to auto-calibrate any simulation completed by the Model Refiner, as seen in the final two lines of Code Segment 5.3. In this case, the calibration area is set to the bottom right of the image in Figure 5.7(c), wherein the region of interest is selected by the user. For this material - monolayer MoS₂ - this region is free

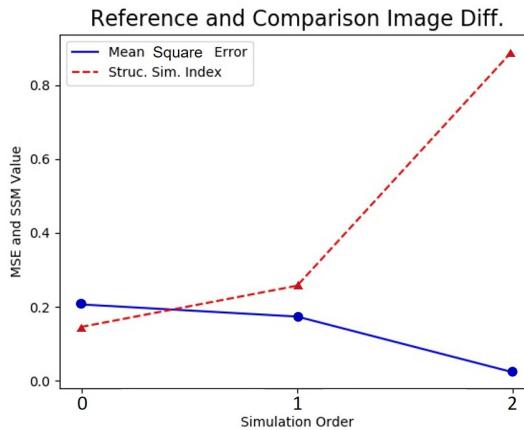


Figure 5.6: Measure of the mean square error (MSE) and structural similarity index (SSM) between the Model Refiner’s reference (experimental) image and comparison (simulated) image. These values are automatically tracked by the Model Refiner (see Code Segment 5.4).

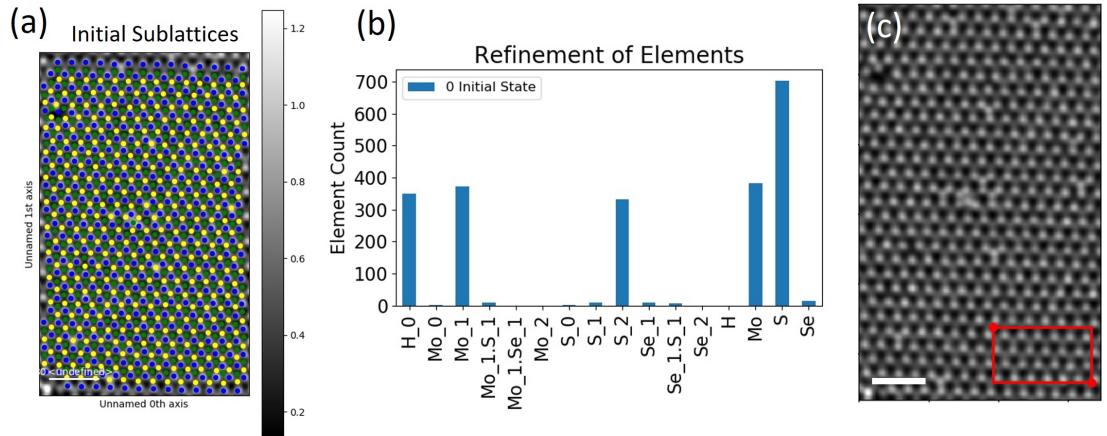


Figure 5.7: (a) Atom lattice, consisting of three sublattices (blue, yellow, green) of a region of Se implanted monolayer MoS₂. (b) The constructed Model Refiner count of the elements configurations in the three sublattices. (c) The area with which the Model Refiner will calibrate any simulations can be set by the user (highlighted by the red box). This region has no adatoms or dopants. Scale bar is 1 nm.

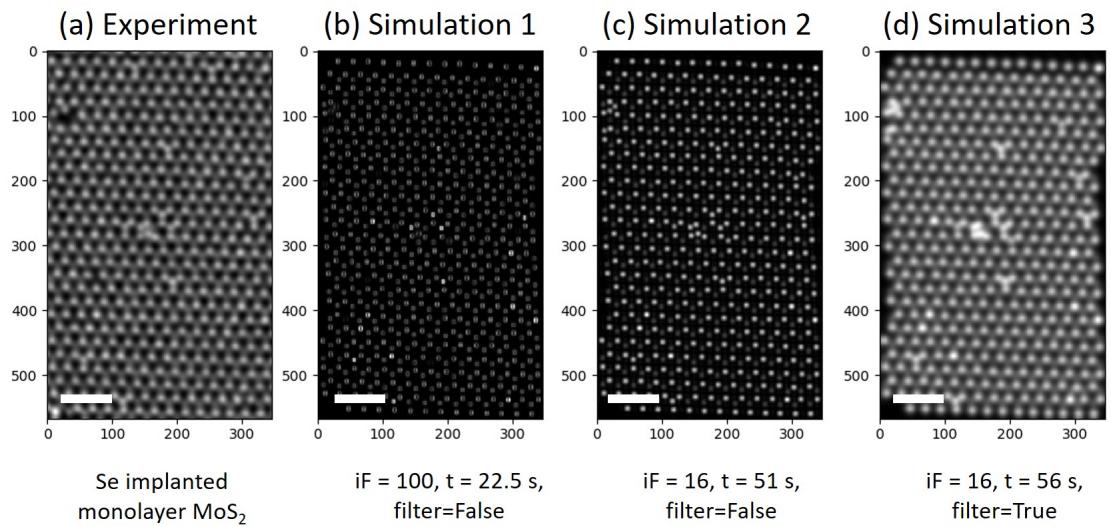


Figure 5.8: (a) Automatic double-Gaussian filtered HAADF STEM image of Se implanted MoS₂. Using the Model Refiner tool, this image is automatically calibrated and set as the Model Refiner's reference image. (b-d) Fast image simulation within the Model Refiner using the PyPrismatic program. The simulations are completed with an interpolationFactor (iF) of (b) 100, (c) 16 and (d) 16 with auto-filtering. The time taken for each simulation on a standard laptop (no GPU) is also noted under each image. Scale bars are 1 nm.

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

of adatoms and dopants. Therefore, the combined calibration area (red rectangle) and pixel separation (11 pixels) will use the single Molybdenum atoms to calibrate the simulated model.

The automatic double-Gaussian filtering in Figure 5.8(a) is described in Subsection 3.4.2(Image Convolution/Filtering). Several simulated models of this Model Refiner are shown in Figure 5.8. These are: 1. an uncalibrated, unfiltered simulation, 2. a calibrated, unfiltered simulation, and 3. a calibrated and auto-filtered simulation. By increasing the interpolationFactor (iF) parameter, one can increase the speed of the simulation, which will naturally decrease the accuracy. An interpolationFactor value of ~ 16 is suitably accurate in this case [11].

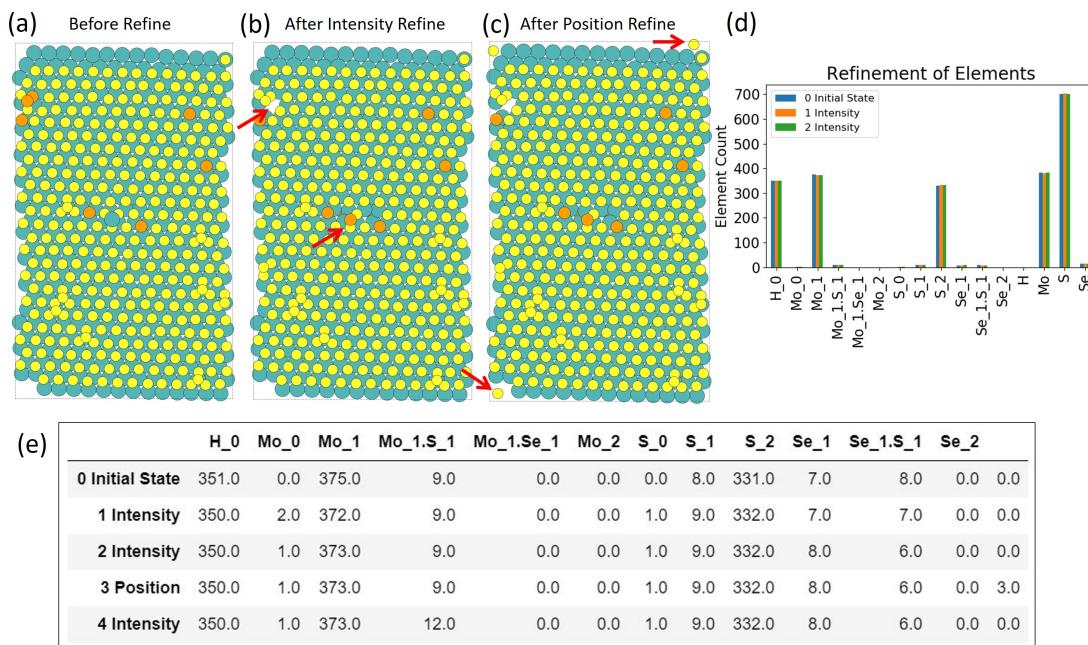


Figure 5.9: 3D .cif model visualised with the ASE package (a) before refinement, (b) after intensity refinement, and (c) after position refinement. The red arrows highlight regions where the refinements have changed the model's structure to better match the experimental image in Figure 5.8(a). Further intensity refinement is required to better match these new positions with the experiment. Visualisation of the elements in the Model Refiner initially and after each refinement as (d) a bar chart and (e) a table of the element counts.

The actual refinement of the Model Refiner is accessed through the `image_difference_intensity_model_refiner` and `image_difference_position_model_refiner` class methods (functions). The Model Refiner automatically tracks the change of elements after each refinement. After a new simulation of the auto-created element model, an intensity refinement is completed in Code Segment 5.4. Running `get_element_count_as_dataframe` allows the user to inspect the changes after each refinement, as in Figure 5.9(d). This

allows the Model Refiner to compare the newly assigned elements in the simulation to the experimental image and check for similarity. One can also view the ball-and-stick model in Figure 5.9(a-c) via the Atomic Simulation Environment (ASE) package. During the intensity refinement - Figure 5.9(b) - a hole is detected (top left red arrow). The central red arrow highlights some changing elements. Before refinement in Figure 5.9(a), this atom was assigned as a Mo_1 adatom. After refinement in Figure 5.9(b), it is reassigned to a Se adatom.

An example of the position refinement is shown in Code Segment 5.4, wherein the refiner finds three new atom positions. After using the intensity refinement, these are initially assigned to Mo_1S_1 intensities. Two of these new positions are highlighted with red arrows (top right, bottom left) in Figure 5.9(c). Subsequent simulation and intensity refinement will correctly set these positions to the intensity best matching the element (Mo_1 , not shown). Additionally, if a refinement (either intensity or position) is mistakenly run, one can revert to a previous version of the Model Refiner, shown in line 44 of Code Segment 5.4.

After the refinements logged in Figure 5.9, the final state of the Model Refiner is displayed in Figure 5.10. At first glance, the Figure 5.10(Simulation) is a good approximation of the Figure 5.10(Experiment), though there are still issues in this example that will be corrected by further intensity refinement. In some areas, the simulation in Figure 5.10 displays extra intensity at chalcogen sites, indicating an overestimation of the intensity at those locations. This is likely due to overlap with the nearest neighbour transition metal atoms. This can be corrected by reducing the radius around which the intensity of the atom is recorded. However, choosing a too-small radius can cause other issues, as not including the entire atom's radius (intensity) is unrealistic. Therefore, the best approach is to use atomic radii (e.g., covalent radii) taken from the Periodictable Python package and scaled to pixels by the `atomic_radii_in_pixels` function in the TEMUL toolkit. To view and use the above script, see the `example_scripts.py` module of the TEMUL toolkit on GitHub [13].

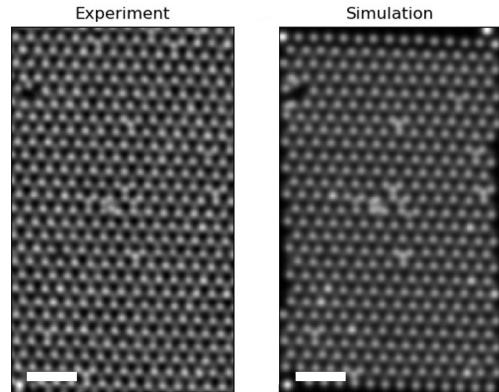


Figure 5.10: State of the Model Refiner's reference image (Experiment) and comparison image (Simulation) after the refinements logged in Figure 5.9. The too-bright atoms at the top-left and top-right of the Simulation will be corrected by further refinements. Scale bars are 1 nm.

5.1. Open Source Development of Automated Atomic Resolution Image Series Analysis Methods

```
1 # Script Part 2
2 # simulate the elements in the Model Refiner
3 refiner.create_simulation(sublattices="all",
4                           filter_image=True,
5                           calibrate_image=True,
6                           filename="",
7                           probeStep=None,
8                           interpolationFactor=16,
9                           percent_to_nn=None,
10                          mask_radius=2)
11
12 refiner.plot_reference_and_comparison_images("Experiment", "Simulation 3")
13
14 # View the tracked error between the experiment and simulation
15 # easily with
16 refiner.plot_error_between_comparison_and_reference_image()
17
18 # refine the model via intensity
19 for i in range(2):
20     refiner.image_difference_intensity_model_refiner()
21     refiner.create_simulation(sublattices="all",
22                               filter_image=True,
23                               calibrate_image=True,
24                               filename="",
25                               probeStep=None,
26                               interpolationFactor=16,
27                               percent_to_nn=None,
28                               mask_radius=2)
29
30 Out: Changing some atoms
31
32 # refine the model by finding new atoms
33 refiner.image_difference_position_model_refiner(
34     sublattices="all", pixel_threshold=20, num_peaks=10)
35 Out: 3 new atoms found! Adding new atom positions.
36
37 refiner.image_difference_intensity_model_refiner()
38 Out: Changing some atoms
39
40 # inspect the results
41 refiner.get_element_count_as_dataframe()
42 refiner.plot_element_count_as_bar_chart(2)
43
44 # Compare the experiment and simulation
45 refiner.plot_reference_and_comparison_images("Experiment", "Simulation")
46
47 # revert to the previous version if a mistake is made
```

```
44 | refiner = refiner.previous_refiner_instance
```

Code Segment 5.4: Se implanted MoS₂ Model Refiner example part 2. The Model Refiner can track the sublattice elements and simulate with PyPrismatic as shown in Figure 5.7 and Figure 5.8, respectively. Refinement of the elements is completed by using the image difference intensity and position functions. The user can quickly inspect the effect of the refinements with a bar chart or table shown in Figure 5.9(d, e). If a refinement is mistakenly called, one can revert to a previous version of the Model Refiner with the previous_refiner_instance method (Line 44).

5.2 Atom Dynamics in Implanted TMDs with the MR Tool

In the following sections, the methods developed in Subsection 5.1 above are used to analyse movies of Se implanted MoS₂ at atomic resolution. First, the full process is outlined, followed by testing of certain steps in subsection 5.2.1(Image Series Automated Analysis Steps). The MR tool, discussed in subsection 5.1.2 above, allows the user to quantitatively analyse the concentration and chemistry of the atoms, as well as the energetics of the lattice overall. This is utilised in subsection 5.2.2(Analysis of Lattice Energetics). Finally, in subsection 5.2.3(Effect of Electron Beam), we will attempt to quantify the detrimental threshold dose rate for Se implanted monolayer MoS₂ at 60 keV using the MR Tool.

5.2.1 Image Series Automated Analysis Steps

Many of the below steps are discussed in Subsection 3.4(Signal and Data Analysis). However, several steps warrant some deeper discussion here as they apply to the analysis of multiple aligned frames. This process was reduced to a one-click script for movie analysis.

- For all frames, remove x-ray spikes. GMS was used for this step, though implementing a median filter or similar in Python is a possibility.
- Align all movie frames. Save new aligned series as a stack and individually (incl. the edge data to keep the full frame size).
- Depending on the atom position method:
 - Find the atom positions using the averaged image stack (AV method).
 - Find the atom positions using the first image (PF method).
- For each frame in the movie do the following steps:
 1. Calibrate the image.
 2. Filter and calibrate the filtered image.
 3. Depending on the atom position method chosen, either:
 - Use the averaged image stack atom positions (AV). Refine the atom positions with Atomap, save the refined atom positions.
 - Use the saved refined atom positions from previous frame (PF). Refine the atom positions with Atomap, save the refined atom positions.

4. Assign elemental configurations using the method described in subsection 3.4.1((Element Quantification in HR ADF STEM Images) and functions in the `model_creation.py` module.
5. Assign z-height using the theoretical values (`model_creation.py` module).
6. Save the Atom Lattice object.
7. Create a Model Refiner (MR) object.
 - Refine with position refinement algorithm as described in Subsection 5.1.1(Position Refinement Algorithm).
 - Refine with intensity refinement algorithm as described in Subsection 5.1.1(Intensity Refinement Algorithm).
 - Save the simulation and experiment with metadata. Save the image comparison data and plot between experiment and simulation. Save a table of the MR Tool refinements with all elemental configuration data. Save the Atom Model.
8. Save the new Atom Lattice object.
9. Continue at step 1 for the next frame in the movie.

For step 3, two methods for obtaining each frame's atom positions were investigated: the averaged atom positions (AV) or the previous frame's atom positions (PF). The aligned and averaged image (AV) gives a clear 'look through' of the entire stack, and therefore gives a good initial assumption for each frame in the entire stack. For certain situations e.g., lattice holes, taking the previous frame's atom positions (PF) could be beneficial. However, if there are holes or lattice distortions in the initial image, this PF method could have a negative knock-on effect for all images in the series.

Below, several frames of movie 008 are used to verify which atom position method (PF or AV) returns more accurate results. It should be noted that the edge atoms and atom subtraction can cause issues with the PF atom position method. Therefore, slightly different atom position refinement methods must be used for the PF method i.e., no Mo atom subtraction, centre of mass refinement for S₂ atoms. Frames 0 (t=60s), 50 (t=127s), and 98 (t=190s) are plotted in Figure 5.11. During this image series, the Se implanted MoS₂ is damaged by the electron beam, creating a hole. It is a complex image series to analyse. The previous frame atom position (PF) method and averaged atom position (AV) method simulation and respective atom models are shown for each of these three frames on the left and right side of the Experiment, respectively. Along with this visual comparison between the PF and AV methods, the structural similarity index (SSM), mean square

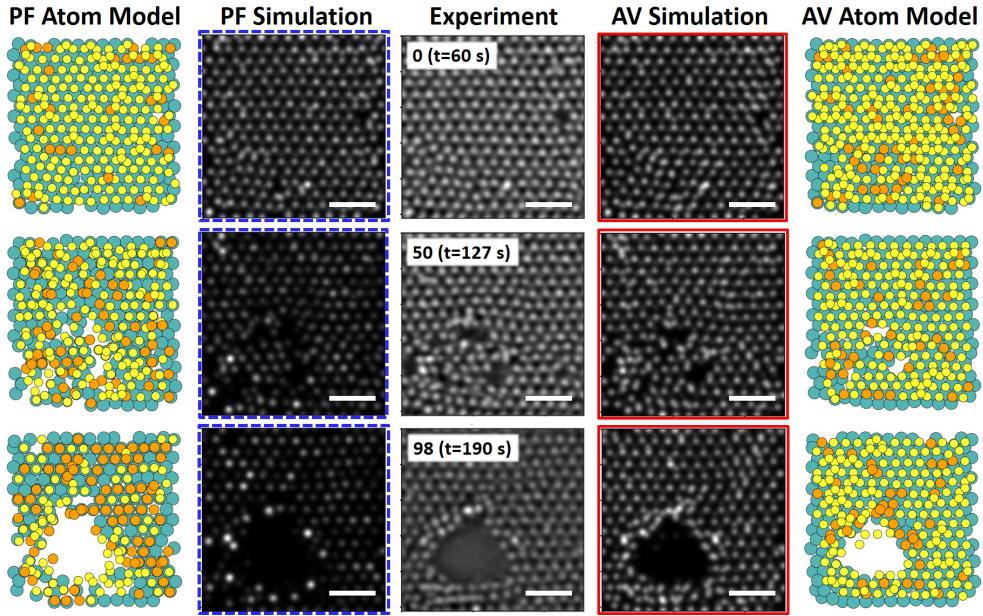


Figure 5.11: Frames 0 ($t=60\text{ s}$), 50 ($t=127\text{ s}$), and 98 ($t=190\text{ s}$) of an experimental movie. The Atom Model and Simulation for the PF and AV atom position methods are shown on the left and right of the experiment, respectively. Scale bars are 1 nm.

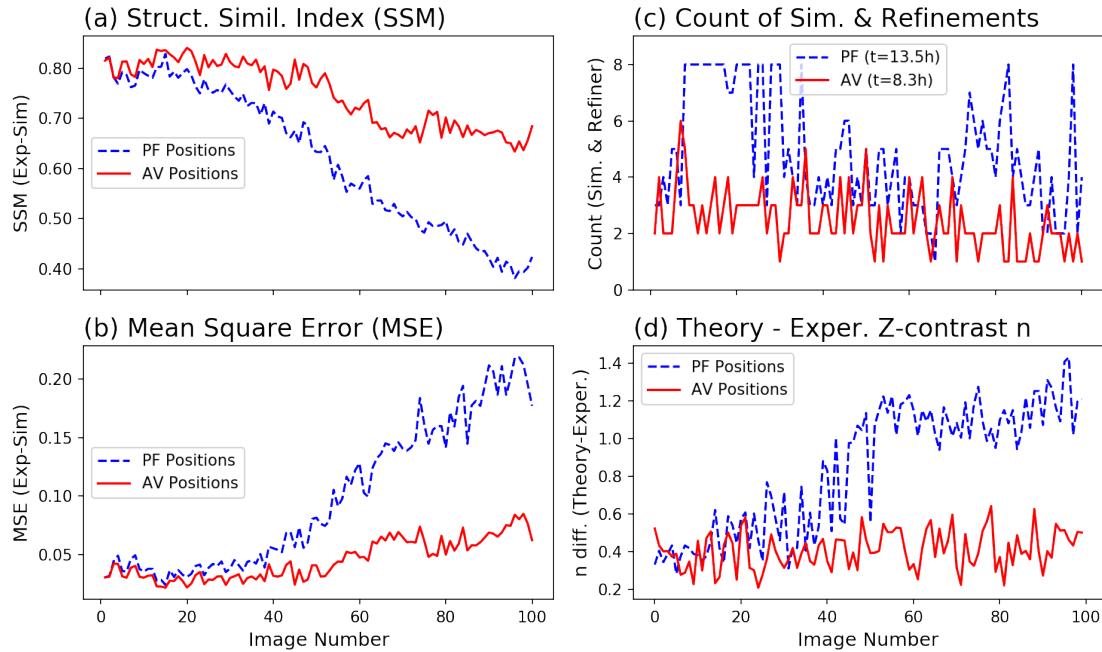


Figure 5.12: The (a) structural similarity index (SSM), (b) mean square error (MSE), (c) number of simulations and Model Refiner (MR) refinements, and (d) difference between the theoretical and experimental Z-contrast exponent n for the averaged image (AV) and previous frame (PF) atom position methods. (a-b) The SSM and MSE are calculated between the experimental frame and final simulation after the MR refinements. (c) Each MR refinement, and therefore simulations, were limited to 8 in this calculation. (d) The difference between the theoretical and experimental exponent (n_{diff}) is shown for each method. The experimental Z-contrast is calculated as described in Subsection 2.1(ADF STEM). The theoretical Z-contrast exponent is calculated as 1.86975.

error (MSE), number of simulations and Model Refiner refinements required have been calculated and plotted in Figure 5.12. Additionally, the difference between the theoretical and experimental Z-contrast exponent ($n_{diff} = n_{theory} - n_{exper}$) is plotted in Figure 5.12(d).

Initially, in Figure 5.11(frame 0) the PF and AV methods perform well, with a high SSM (~ 0.8) and low MSE (~ 0.05) between Experiment and Simulation. However, as the movie progresses, the PF method begins to exhibit issues, likely due to the above-mentioned possible negative knock-on effects of using the previous frame's atom positions for each subsequent frame. Between frame 50 and 98, a large hole forms in the monolayer lattice. In the PF Simulation and AV Simulations for Figure 5.11(frame 98), one can see that the AV method outperforms the PF method. The AV Simulation hole in Figure 5.11 matches well with the Experiment, whereas the PF Simulation's very bright hole edge atoms cause contrast issues which is not seen in the Experiment. Additionally, with the PF Atom Model of Figure 5.11(frame 98), it is clear that the number of Se atoms has increased by an unrealistic amount. In this respect, the AV method is consistent between frame 0 and frame 98. The differences are quantified in Figure 5.12(a-b), wherein the PF method's MSE quadruples from the initial frame to the final frame. In the same period, the AV method's MSE increases by a factor of only ~ 1.5 . Similar negative trends are seen in the SSM, wherein the PF and AV method's SSMs are reduced by 50% and 12%, respectively.

The number of simulations and Model refiner (MR) refinements for each frame is plotted in Figure 5.12(c). The PF method required more simulations and MR refinements than the AV method, naturally increasing the time taken for the automated program to complete; it took 13.5 hours and 8.3 hours for the PF and AV methods, respectively. An average of 4.86 and 2.47 simulations (and therefore MR refinements) were computed per frame. It should be noted here that just two threads were utilised for each pyprismatic simulation, and no GPU was used. Even so, it took only 8.3 hours to perform ~ 247 , 512x512 pix simulations with a probe step size of ~ 0.007 nm. The number of MR refinements was limited to 16 (8 position, 8 intensity) for this image series.

The difference between the theoretical and experimental Z-contrast exponent n is plotted in Figure 5.12(d). The theoretical value of the exponent, n_{theory} , for an inner detector angle of 85 mrad was calculated as 1.86975 when considering both Mo and S [29], as described in Subsection 2.1(ADF STEM). The difference between the theoretical and experimental value, n_{diff} , is consistent in the AV method (average=0.409). For the PF method (average=0.841), the value increases by a large amount from frame 20. Upon further inspection, this is due to the large

number of Se atoms supposedly detected by the PF method throughout the movie. This is consistent with the large number of Se atoms visible in the PF Atom Model in Figure 5.11(frame 98).

Overall, the AV method outperforms the PF method. Therefore, the AV method is utilised throughout the rest of this chapter. One alternative could be to use the AV method to find the initial atom positions, and subsequently use the PF method for each subsequent image. Ideally, the Model Refiner would simulate both AV and PF and simply choose the best comparison with the experimental image, or use a combination of both. This is discussed further in Subsection 7.2(Future Directions).

In step 7 above, the number of refinements and simulations completed with the Model Refiner (MR) tool must be limited for each frame. The above example was limited to 8 simulations, and is implemented within a loop as shown in Code Segment 5.5

```

1 # Loop and limit the Model Refiner
2 # Model Refiner instantiated as "refiner".
3
4 while not refiner.compare_latest_element_counts():
5
6     refiner.create_simulation(
7         sublattices='all', filter_image=True,
8         calibrate_image=True, interpolationFactor=8
9         filename='model_refiner_sim_{}'.format(i_filled))
10
11    refiner.image_difference_position_model_refiner(
12        chosen_sublattice=refiner.sublattice_list[1],
13        sublattices='all', pixel_threshold=pixel_threshold,
14        num_peaks=10, comparison_sublattice_list='auto')
15
16    refiner.image_difference_intensity_model_refiner()
17
18    if len(refiner.element_count_history_list) > 15:
19        break

```

Code Segment 5.5: Example loop for the Model Refiner (MR) tool for step 7 in the analysis steps outlined in subsection 5.2.1.

5.2.2 Analysis of Lattice Energetics, Ad-atoms and Vacancies with the MR Tool

To fully understand the effects of ion implantation and STEM imaging, series of images (movies) must be quantitatively analysed. Therefore, in this section movies of 2D material atoms moving under the electron beam are analysed both manually and automatically. The manual analysis provides a clear view of the individual movement of atoms and vacancies between image frames, while the automatic analysis via the Model Refiner (MR tool) will provide an understanding of the energetics and concentration of atoms over time.

In this section, the movement and energetics of single atoms, vacancies and the creation of holes under the electron beam are discussed. All images shown in this chapter were acquired using the Superstem 3 STEM. Even at the low primary electron energy of 60 keV, the electron beam has a large effect on the movement of atoms. The electron beam can damage the lattice by either displacing an atom to a new position or by altogether knocking (sputtering) the atom out of the lattice permanently. This can occur via elastic and inelastic scattering events, as described in subsection 2.1.2(Electron-Matter Interactions).

The calculated threshold knock-on energy for the smallest atom in MoS₂, S, is 6.9 eV [166]. Therefore, an electron beam under \sim 90 keV can be used for safe imaging of S atoms. Others have calculated the safe imaging energy as 66 keV, much closer to the 60 keV used in these experiments. As discussed in subsection 2.1.2(Electron-Matter Interactions), the 90 keV estimate is assumed to be more accurate, as it was calculated via density functional theory (DFT) [417]. In either case, the value is above 60 keV, which should minimise elastic sputtering of atoms by the electron beam. However, if lattice bonds are damaged and weakened via inelastic scattering (e.g., ionisation), sputtering may occur.

The energetics of individual vacancies, substitutional atoms and ad-atoms can be calculated with DFT. DFT was carried out by collaborators [see subsection 3.3.2 (Image Simulation and DFT Calculations)]. The lattice configurations (I-XI) calculated are shown graphically in Figure 5.13, alongside the energy required to produce the configuration. The change in energy ΔE is calculated by subtracting the energy of the system in B from A. For example, in configuration I, the energy E_A is 5.90 eV higher than that of E_B , resulting in $\Delta E = 5.90 \text{ eV}$. In other words, it requires 5.90 eV to produce a single S vacancy in monolayer MoS₂. Interestingly, to substitute a single Se atom for a S atom, the energy of the system must be raised by 5.90 eV, and subsequently reduced by 5.10 eV. These processes are examined in Figure 5.14 in detail.

5.2. Atom Dynamics in Implanted TMDs with the MR Tool

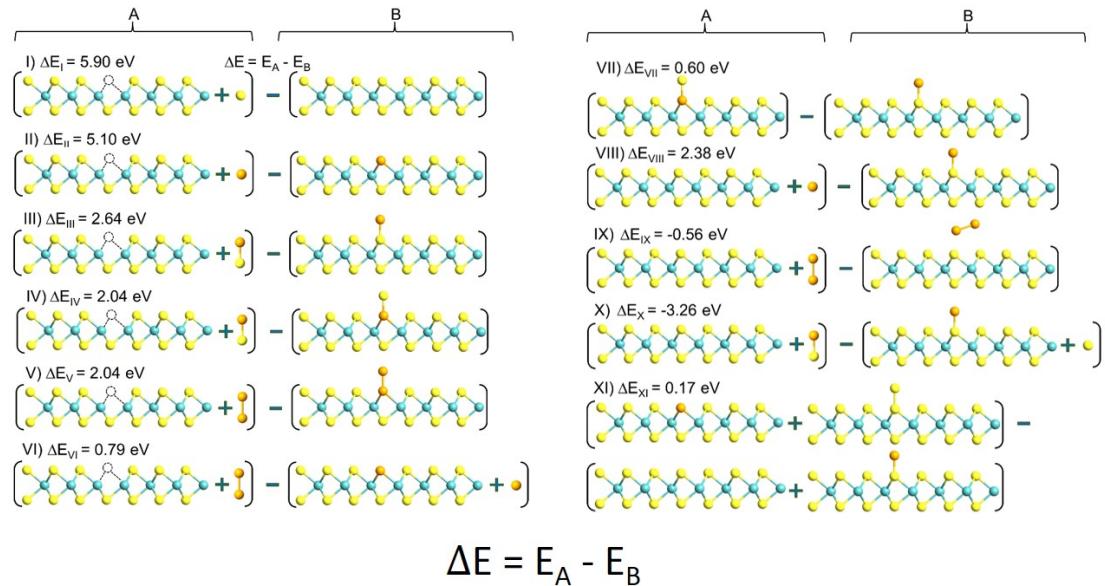


Figure 5.13: DFT calculations of formation energies of substitutional and ad-atom Se in monolayer MoS₂. The difference between the energy of the left ‘A’ columns and right ‘B’ columns is calculated by the formula $\Delta E = E_A - E_B$. The DFT calculations were carried out by Dr. M. Ghorbani-Asl and Dr. A.V. Krasheninnikov.

The series of images presented in Figure 5.14 is three consecutive frames from a movie of Se implanted MoS₂ imaged at 60 keV (SuperSTEM 3). For each frame in Figure 5.14(a) two atom positions of interest are highlighted with a blue (left) and red (right) arrow. From the first to last frame, one can see that the blue (left) atom intensity is initially low, increases, and finally decreases to the initial intensity; vice versa for the red (right) atom intensity. The change in intensity is likely due to the movement of a Se atom and a chalcogen vacancy. This Se-vacancy swap is analysed manually and automatically (with the MR tool) in Figure 5.14 and Figure 5.15, respectively.

These intensity changes are categorised as most likely due to either Path 1: a vacancy swap or Se/S swap, or Path 2: Se ad-atom movement. The two energy paths are displayed in Figure 5.14(b)(i-v), wherein the blue (left) and red (right) arrow labels are consistent with the arrow labels in Figure 5.14(a). In Figure 5.14(b), the atom models are cross-sectional; the electron beam direction is directed toward the top of the model. See Figure A.10 in Subsection A.4 for more details on the manual analysis.

For path 1 in Figure 5.14(b,i), the blue (left) and red (right) chalcogen atomic columns are S₂ and Se₁S₁, respectively. In step (ii), between frame 39 and frame 40, a S vacancy is created, which requires +5.9 eV. Following this, a Se atom moves into this vacancy (iii). This Se jump requires +5.1 eV, but immediately relaxes

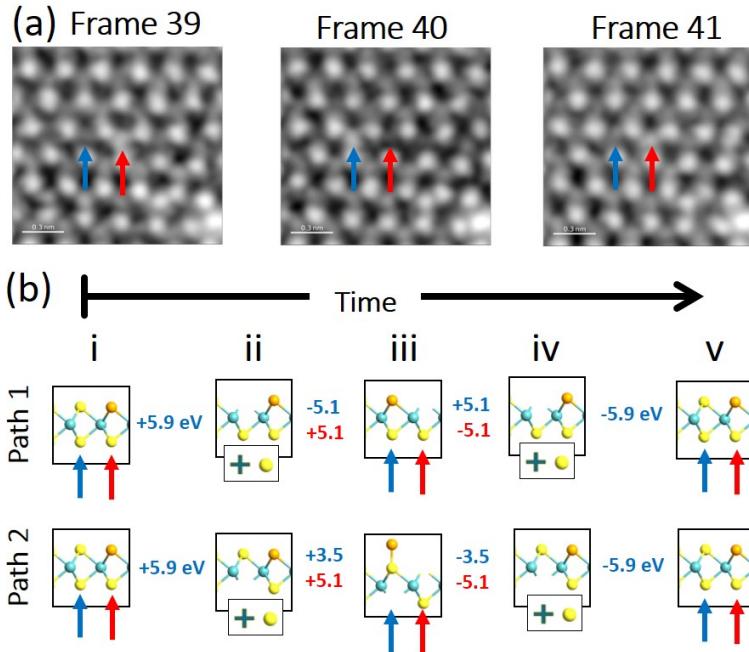


Figure 5.14: Manual analysis of a Se-vacancy swap in three consecutive frames of an Se implanted MoS₂ monolayer image series. (a) Double Gaussian filtered frames 39, 40 and 41 of the image series with the blue (left) and red (right) arrows indicating the atomic columns in which the Se-vacancy swap occurs. (b) Two possible energy paths for the Se-vacancy swap over the three frames. The energies - blue for left, red for right atomic columns - are taken from the DFT calculations in Figure 5.13 above. Scale bars are 3 Å.

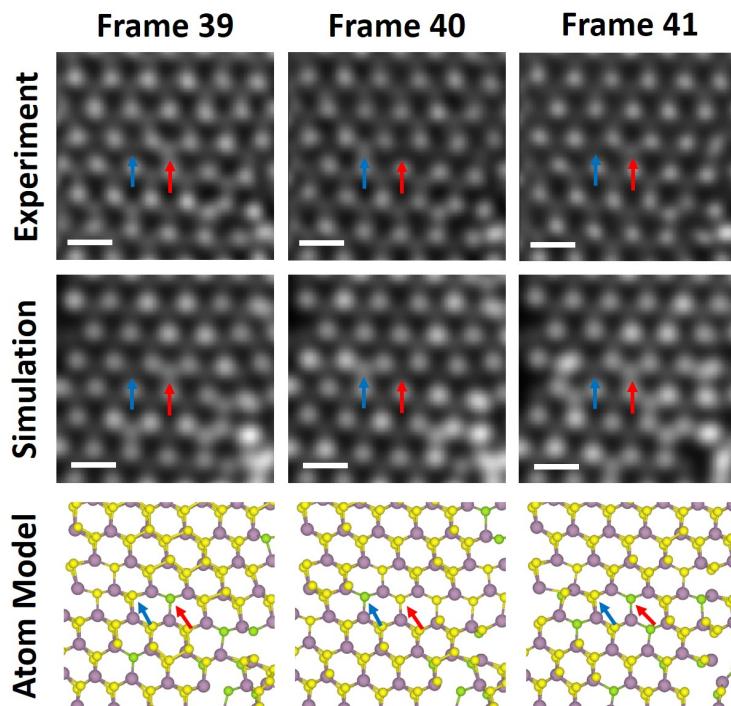


Figure 5.15: Model Refiner (MR) automatic analysis of the region from Figure 5.14 above. The final simulation and atom model produced by the MR algorithms are displayed under their respective experimental images; frames 39, 40 and 41. The blue (left) and red (right) arrow labelling is consistent with Figure 5.14 above. Scale bars are 3 Å.

with -5.1 eV into the chalcogen vacancy position. This movement is reversed in (iv), as is the original S vacancy in (v). Between frame 39 and 40, the total energy change is $\Delta E = 5.9 - 5.1 + 5.1 = +5.9\text{ eV}$. Between frame 40 and 41, this is reversed, leaving the total energy change in these two atomic positions from three frames as $\Delta E = 0\text{ eV}$.

Rather than the substitutional Se replacing the S vacancy, the possibility of a Se ad-atom is presented in Figure 5.14(b) path 2. As in path 1, the total energy change is $\Delta E = 0\text{ eV}$. To discuss which scenario is more likely is difficult, as these are not the only two energy paths available. However, it would seem more likely for path 1 to occur as it is a simple Se-vacancy swap with $\Delta E = \pm 0\text{ eV}$ between both (ii-iii) and (iii-iv). This can be compared to the larger change in energy $-\Delta E = \pm 8.6\text{ eV}$ required in path 2 (ii-iii). Additionally, the electron beam is less likely to knock a chalcogen (e.g., Se) atom out of the lattice if it is in the top layer [166]. Therefore, path 1 seems more likely if we assume the vacancy swap occurs in the top layer facing the electron beam.

Figure 5.15 shows the automated Model Refiner (MR) tool analysis of the same region as Figure 5.14 for the above frames 39-41. The filtered experimental and simulated images, as well as the atom model is shown. The blue (left) and red (right) arrows point to the same chalcogen positions as in Figure 5.14 above. In the relevant atomic columns the simulation matches well with the experiment. After several simulations (3-5 in each frame), the MR tool produces atomic models similar to that produced from the above manual analysis. The blue (left) chalcogen atomic column is assigned as S_2 , Se_1 and S_2 for frames 39, 40 and 41, respectively. The red (right) chalcogen atomic column is assigned as Se_1 , S_1 and Se_1 for frames 39, 40 and 41, respectively. In all three frames, the simulations find the moving substitutional Se atom not to be above a S atom, therefore disagreeing with the manual analysis of Se_1S_1 .

In Figure 5.15(Frame 40), the red arrow shows a single S atom in the top chalcogen sublayer. This is due to the design of the assignment function `assign_z_height`, and does not consider probabilities of whether the top or bottom layer is most likely to be occupied, as discussed by Komsa *et al* [166]. Figure 5.15 demonstrates the reliability of the MR tool for analysing a Se-vacancy swap in the chalcogen sublattice.

Figure 5.16 shows the formation of a hole in monolayer MoS₂ analysed with the Model Refiner tool. The hole itself is matched well between experiment and simulation for each frame. In this scenario, several adatoms and lattice atoms move to different sites in and near a hole in Se implanted MoS₂ monolayer.

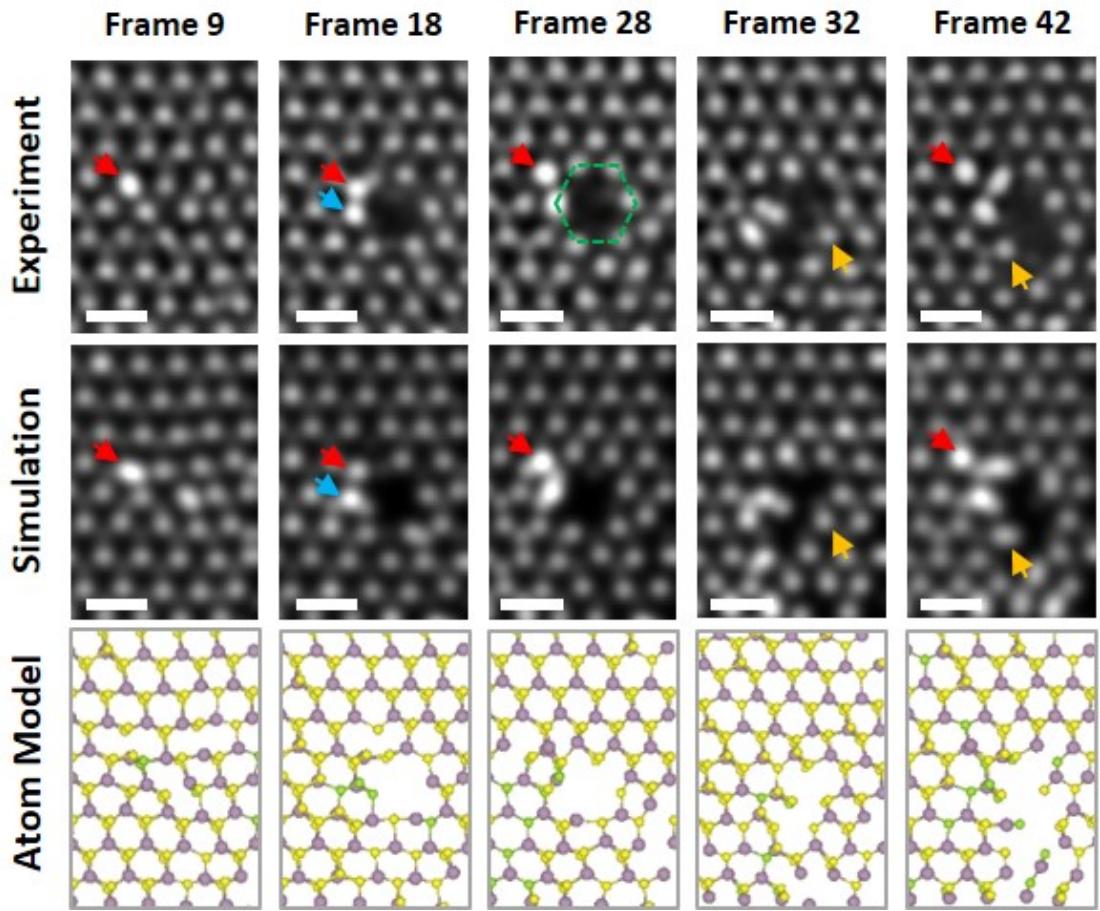


Figure 5.16: Crops of five experimental frames of an automatically analysed image series using the MR tool. The simulation and atom model of each frame crop are below the respective experimental frames. The red arrows in frames 9, 18, 28 and 42 highlight a possible adatom, as do the blue arrows in frame 18. The orange arrows in frames 32 and 42 point to possible movement of a lattice Mo atom. Each frame number is also the number of seconds after the scan begins. Scale bars are 5 Å.

The hole begins to form with several S vacancies, followed by several Mo vacancies. Before Figure 5.16(Frame 9), suspected ad-atoms move about the surface (not shown). One of these ad-atoms can be seen in Figure 5.16(frame 9) as a bright ellipse, labelled by a red arrow. As shown below in the Figure 5.16(frame 9) simulation and Figure 5.16(frame 9) atom model, this intensity, which one might assume is an adatom, is assigned by the MR tool as a Se_2 positioned beside a Mo_1S_1 configuration. This is not necessarily true, and shows an inherent limitation due to blurring as described later on.

In Figure 5.16(frame 18) the red arrow is pointing to the same adatom. The two bright atomic columns can be distinguished in this region by manually tracking the adatom's movement between Figure 5.16(frames 9 and 18) (not shown). It is now positioned over a chalcogen atomic column (red arrow). The simulation

does not reflect this position's intensity very well, as it has been assigned S_2 , with a neighbour Mo_1S_1 atom-adatom pair. One could speculate that the bright atomic column, marked by the blue arrow, is likely a Mo_2 configuration. This is because a Mo atom is ejected from the lattice near the centre of the frame. It is likely that this Mo atom was not ejected completely from the lattice. This is a common occurrence in monolayer MoS_2 holes, around which the ejected atoms gather. However, due to the nearby chalcogen columns assigned Se_1 atoms, this position is assigned Mo_1Se_1 , rather than the more probable Mo_2 configuration. A stray Se_1 lattice atom is assigned near the hole due to high nearby intensities, rather than the more likely S_1 or S_0 .

In Figure 5.16(frame 28), the red arrow returns to its original position, as in Figure 5.16(frame 9). As mentioned previously, it is assigned a Mo_2 value. The bright atomic column below is again assigned a Mo_1Se_1 value. The green hexagonal ring marked in Figure 5.16(frame 28) was stable for ~ 10 seconds, from Figure 5.16(frame 19 to 29) (not shown).

In Figure 5.16(frame 32), the red arrow atom intensity moves out of view (not shown). The lattice hole becomes larger and less stable after the green hexagonal ring pattern is destroyed in frame 29 (not shown). The orange arrow points to a Mo_1 lattice atom at the edge of the hole moving between Figure 5.16(frame 32) and Figure 5.16(frame 42). This Mo atom moves between frames due to the ejection of its neighbour S_1 and S_2 atoms. After they are ejected, the Mo atom moves to its new position possibly due to dangling S_1 or S_2 bonds on the left side of the hole. In Figure 5.16(frame 42), the red arrow points to the same Mo_2 atom-adatom pair as seen in Figure 5.16(frame 28), though this is difficult to say for certain due to the movement of atoms in and out of the region.

Some issues with element assignment are now discussed in relation to Figure 5.16. Figure 5.16(frame 9) shows a current limitation of the method for pure comparison between experiment and simulation in cases wherein there is a bright elliptical blurring i.e., wherein an atomic column is not well resolved. This bright blurring causes the Atomap coordinate refining - either with a 2D Gaussian fit or centre of mass [see subsection 3.4.3(Methods for Atom Position Analysis)] - to move the nearby atom positions toward this brightness, even when most of the intensity has been removed by the subtracting a negative 2D Gaussian [4]. It is likely that this bright intensity is a Mo adatom over a Mo lattice position. The reason this hasn't been 'correctly' identified as Mo_2 is that the chalcogen column has not been resolved due to blurring between two or more nearby atoms. This should be compared with Figure 5.16(frame 28), in which the red arrow is at the same location. In this case, the atomic column is circular (on zone), rather than

elliptical, and therefore does not blur any surrounding columns. Because of this, the column is identified as a Mo_2 atom-adatom pair.

In Figure 5.16(frame 18), neighbouring chalcogen positions were ‘pulled’ in by the very high intensity transition metal site, causing them to be assigned Se_1 . A consequence of this incorrect coordinate refinement is the mis-assignment of the very high intensity transition metal site to an intensity lower than Mo_2 i.e., Mo_1Se_1 . The same pulling effect occurs in Figure 5.16(frame 28). The simulation of Figure 5.16(frame 28) shows blurring effects at the two bright atomic columns indicated by the red and blue arrows. This is likely due to the nearby chalcogen columns again being ‘pulled’ in by the high intensity values during coordinate refinement. For Figure 5.16(frame 32), the frequent movement of atoms at the edge of the hole means the Model Refiner does not match very well with the blurred atom intensities, as described for Figure 5.16(frame 9 and 28) above.

After the initial Mo atom is ejected from the lattice, vacancies will increase unless the beam is stopped or the dose rate is reduced. In subsection 5.2.3(Effect of Electron Beam), we discuss the effects of the electron beam on Se implanted MoS_2 monolayers. The hole shown from Figure 5.16(frames 18-42) has some general intensity, which is likely a mixture of sulphur and the ubiquitous hydrocarbon contamination. In especially contaminated samples, one often sees a delay in hole formation due to the hydrocarbons acting as a protective barrier, similar to graphene [183, 184]. Subsection 5.2.3(Effect of Contamination on Movie Analysis) investigates this phenomena with regard to dose rate.

In general, the automated simulations match well with the experiment Figure 5.16. Clear deviations occur between experiment and simulation when there is either a blurring of the lattice, thereby reducing resolution, or where very bright atom intensities skew the atom coordinate refinement.

Figure 5.17 presents the movement of a possible adatom on the surface of the Se implanted MoS_2 lattice. This cropped region is taken from a different area of the same movie as Figure 5.16 above. The atomic column intensity was verified for several frames of the movie via direct, manual comparison between experiment and simulation. Its intensity was confirmed as a single Mo adatom. Knowing this, we can examine how well the Model Refiner (MR tool) deals with categorising the adatom’s intensity over various atomic columns (red arrows).

In Figure 5.17(frame 03), the adatom is located over a lattice Mo site, with some bright intensity in the nearby bottom right chalcogen site. At a glance, the experiment and simulation match well. The atom model correctly shows a Mo adatom over the lattice Mo (red arrow). The nearby bright chalcogen site

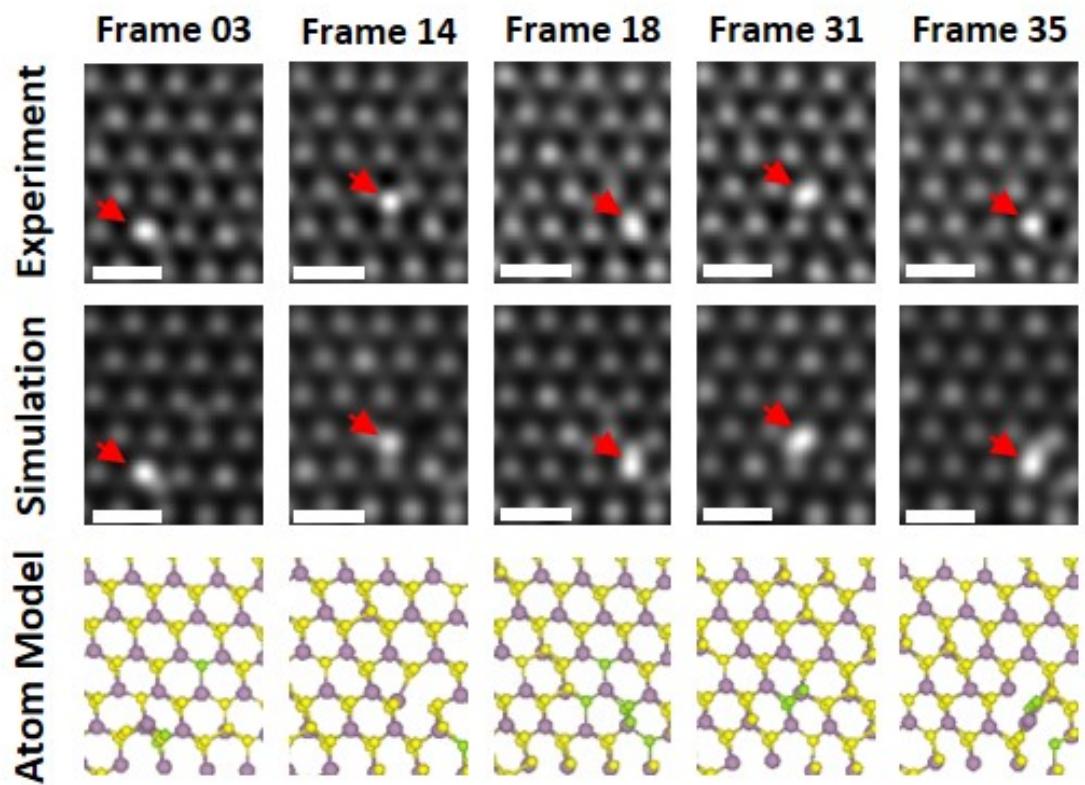


Figure 5.17: Crops of five experimental frames of an automatically analysed image series using the MR tool. The simulation and atom model of each frame crop are below the respective experimental frames. The red arrows in each frame highlight a possible Mo₁ adatom. Each frame number is also the number of seconds after the scan begins. Scale bars are 5 Å.

is assigned an Se₂S₁ intensity, likely due to the position registering an extra high intensity from the Mo₂ site, similar to the ‘pulling’ discussed for Figure 5.16 above.

Between Figure 5.17(frame 03) and Figure 5.17(frame 14), the adatom moves over a chalcogen atomic column. In this run of the MR, the option was not given for a Mo adatom to lie on a double S positions (S₂Mo₁). However, the program still performs well, attempting to match the S₂Mo₁ intensity (verified with manual simulations). To do this, it refines the nearby lattice Mo atom toward the bright adatom position. This can be seen in the Figure 5.17(frame 14) simulation, where the top right lattice Mo atom is seemingly missing, though the situation is clear in the atom model. Therefore, the MR correctly assigns an intensity of S₂Mo₁ to the adatom’s position, at the expense of counting an extra Mo atom. This does not effect the lattice Mo count, but there is no extra Mo adatom counted for this frame.

In frame 18 of Figure 5.17, the adatom is imaged between atomic column positions, creating a blurring that, as explained above for Figure 5.16, is difficult

for the MR program to deal with. It assigns an intensity of Se_2 and Mo_1Se_1 in the chalcogen and transition metal atomic columns, respectively. Additionally, the Se_2 column is ‘pulled’ toward the Mo_1Se_1 column due to the latter’s high intensity. This simulated intensity matches well with the experiment, though it does not assign the most simple option - a Mo adatom - to this position.

In Figure 5.17(frame 31), the adatom is again positioned between chalcogen and transitions metal columns. As found in Figure 5.17(frame 18), Se_2 and Mo_1Se_1 columns are identified. Perhaps an “Occam’s Razor” approach, in addition to a lattice and adatom tracker would be a method for correctly assigning these high, blurred intensity values. The Occam’s Razor heuristic refers to the simplest explanation often being the most likely one. In this case, it is physically simpler for there to be a Mo adatom, rather than Se_2 and Mo_1Se_1 columns appearing from a standard MoS_2 lattice between two subsequent frames. This would, of course, require the design of a lattice and adatom tracker, in which the program would assign probabilities to each element configuration based on the previous frames’ configurations. These methods are described in Subsection 7.2(Future Directions) and are, unfortunately, outside the time scope of this thesis. In addition to the above-mentioned lattice and adatom tracker, automatically measuring the stability of the adatom at different positions would be very interesting.

Finally, in Figure 5.17(frame 35), the adatom sits over a lattice Mo atom. This is reflected in the atom model with a Mo_2 element configuration. However, in the simulation there is some blurring toward the top right of the adatom position that is not present in the experiment. This is due to the coordinate refinement being ‘pulled’ toward the bright adatom position, as described previously.

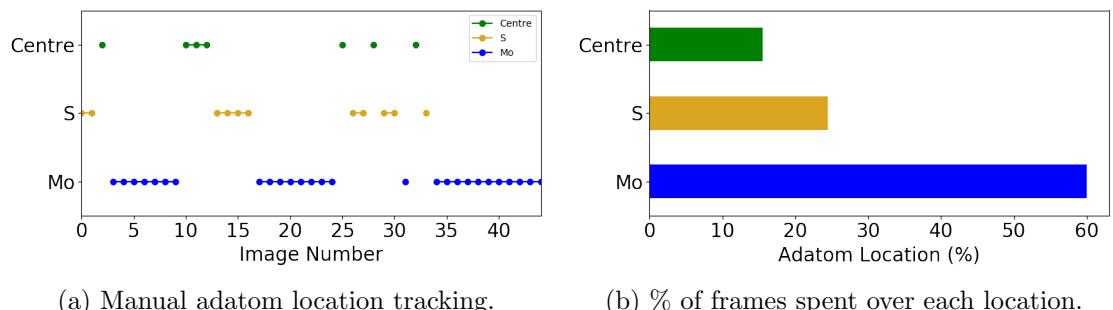


Figure 5.18: (a) Manual tracking of the location over which an adatom sits in Figure 5.17. (b) The percentage of the frames over which the adatom sits for each location. Centre refers to the centre of the MoS_2 hexagon. S: chalcogen column, Mo: transition metal column

This adatom location tracking has been completed manually for the above adatom, with the results plotted in Figure 5.18. The centre, S, and Mo locations refer to the centre of the MoS_2 hexagon (background), chalcogen, and transition

metal sites, respectively. The location over which the adatom sits is plotted in Figure 5.18(a). Between stable periods over the Mo location, the adatom moves between the centre and S locations several times. This pattern repeats twice in this short 45 frame movie. In Figure 5.18(b), the percentage of the frames over which the adatom is located is plotted. The Mo adatom is located 60% of the 45 frame movie over a transition metal (Mo) position. Therefore, it is likely more energetically favourable for this to occur, where the Mo adatom relaxes into the mirror positions of the lattice Mo atom.

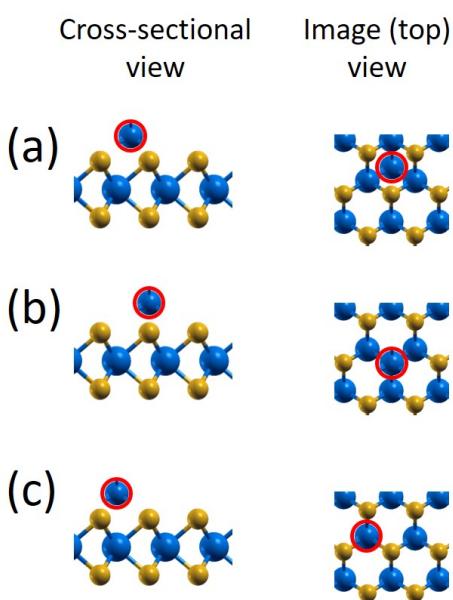


Figure 5.19: Cross-sectional and image (top) views of an Mo surface adatom (blue atom with red border) over the (a) centre of the hexagon, (b) chalcogen and (c) transition metal position.

The physical position of the Mo adatom analysed in Figure 5.18 on the surface of the monolayer is shown in Figure 5.19(a), Figure 5.19(b) and Figure 5.19(c) for the centre of the hexagon, chalcogen and transition metal positions, respectively. The adatom is shown as a blue atom with a red border, to distinguish it from the lattice Mo atoms. From a quick glance, it would seem that in the cross-sectional view, both Figure 5.19(a) and Figure 5.19(c) appear the best location for the adatom to reside; it is positioned close to three S lattice atoms. In Figure 5.19(b), the adatom is located over a S atomic column, and is therefore realistically only able to bond to this atom.

From the image (top) view, the Mo adatom positioned over the Mo lattice atoms in Figure 5.19(c) reflects the pristine lattice structure best. The adatom may spend 60% of the time at this position due to the lattice bonding configuration of TMDs. Unpaired covalent electrons from the three nearby S atoms may bond with the Mo adatom temporarily as the adatom moves around the monolayer surface. A diagram of the bonding structure of TMDs is shown in Figure 2.10 and in [220]. This may explain why the adatom sits in a position wherein it bonds with several lattice S atoms. However, it does not explain why it sits predominantly over the TM column and not equally over the centre of the hexagon position. This behaviour could be due to the mirroring of the lattice when positioned over the TM column [Figure 5.19(c)], therefore reflecting

the pristine lattice best as mentioned above. DFT calculations would be required to calculate the activation energies involved in such adatom behaviour.

5.2.3 Effect of Electron Beam on TMD Monolayers with the MR Tool

As seen in the previous section, the electron beam greatly effects the lattice of the Se implanted MoS₂. Holes, such as that shown in Figure 5.16 above, are often detrimental to the analysis of the sample and quantification of the elements. The following section will attempt to find the cause of these detrimental effects. Both the total cumulative electron beam dose and the rate at which this electron beam dose hits the sample (dose rate) are investigated as possible causes. First, the change in dose rate is examined by changing the electron beam pixel dwell time. Following this, low dose rate (monochromated) and high dose rate movies are examined, and a threshold dose rate is found. Finally, the effect of contamination on the production of lattice vacancies is discussed. The model refiner (MR) tool was used to analyse all movies in this section.

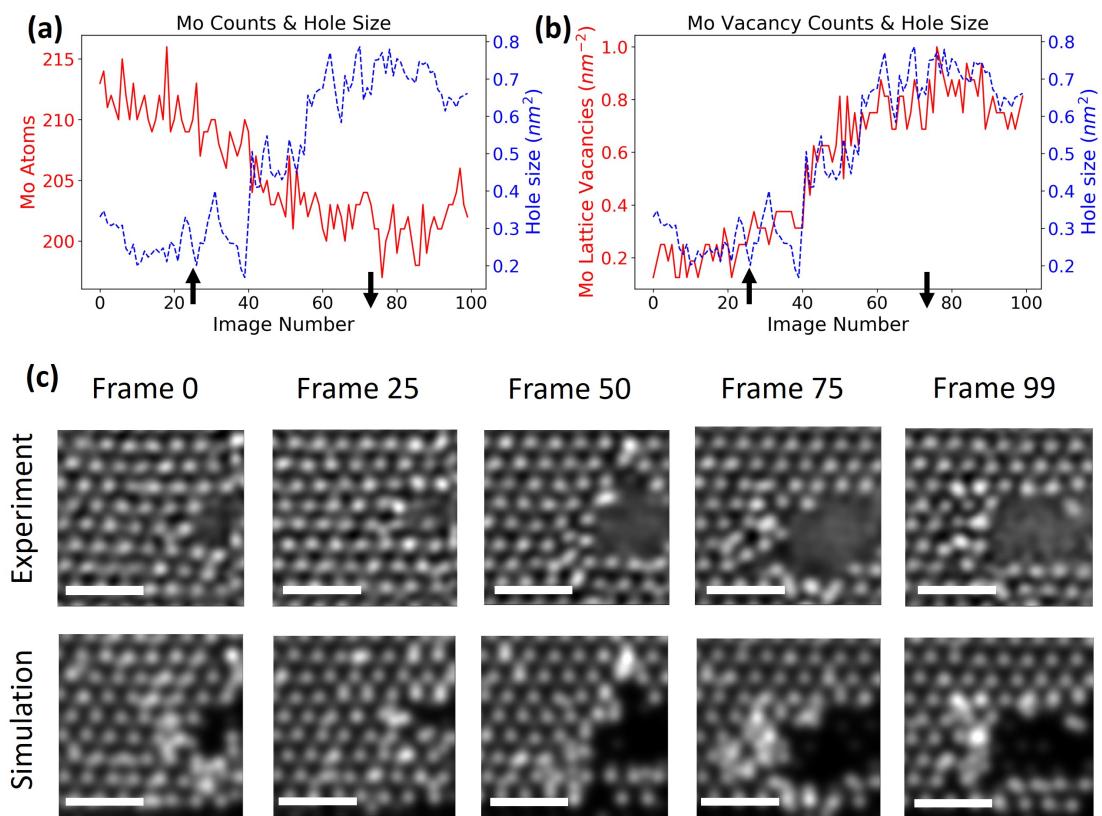


Figure 5.20: (a) Total number of Mo atoms and (b) lattice Mo vacancies per squared nm for each frame in the movie. The lattice hole size per squared nm is overlaid on right y-axis of both (a) and (b). The upward and downward arrows in (a) and (b) indicate the frames at which the dwell time is increased and decreased, respectively. (c) Experimental images and final model refiner image simulations of frames 0, 25, 50, 75 and 99 of the movie. Scale bars are 1 nm.

A straightforward way of comparing the effect of the dose rate is to increase the pixel dwell time of the electron beam while imaging a region, and reducing it after a certain time. A model refiner (MR) analysed section of such a movie is presented in Figure 5.20. During the first 25 frames of the movie the dwell time was set to $5.3 \mu\text{s}$. The dwell time was then quadrupled to $21.2 \mu\text{s}$ for 50 frames. Finally, from frame 75 onwards the dwell time is reduced to its original value. The increase and decrease of the dwell time is indicated by the black up and down arrows in Figure 5.20(a) and Figure 5.20(b). During the higher dwell time imaging, one would expect the lattice damage to increase. In this case, the damage is mostly located in the bottom right region of the movie [see frames in Figure 5.20(c)]. In Figure 5.20(a), the total number of Mo atoms during the movie is plotted with the size of the lattice hole per squared nm, both as a function of the movie frames. As expected, there is an inverse correlation between the number of Mo atoms and the size of the lattice hole. When the dwell time is increased (black upward arrow) the lattice hole's size increases, just as the number of Mo atoms decreases. These trends stop after the dwell time is decreased. After frame 100, the hole size stays at a consistent size for 150 more frames (not shown). In Figure 5.20(b), the lattice Mo vacancies per square nm follows closely the size of the lattice hole. The frames in Figure 5.20(c) reflect the data presented in Figure 5.20(a) and Figure 5.20(b). The experimental images for Figure 5.20(frames 0, 25, 50, 75, and 99) are plotted above their respective MR analysed simulations. The lattice hole in the bottom right region of each frame, initially small, is expanded considerably ($\sim 0.4 \text{ nm}^2$) from Figure 5.20(frame 25) to Figure 5.20(frame 75).

The dose rate clearly has an effect on the lattice. This effect is quantified in the following section by looking at the lattice Mo vacancies per squared nm for various dose rates. Additionally, the dose rate at which the total cumulative dose ceases to effect the sample in a detrimental manner i.e., the detrimental threshold dose rate, is inspected. Histograms of all atom intensities for each movie analysed in the following section can be found in Figure A.13 in Subsection A.4.

Figure 5.21 compares the data from the monochromated (MC) movies. In this context, monochromation of the electron beam is used to lower the beam current by an order of magnitude. Over the ~ 2 minute time-frame analysed by the model refiner (MR), each of the MC movies do not damage. This pattern of non-destructive imaging continues for the extend of each of these MC movies (~ 6 minutes total). The y-axis of Figure 5.21(a) is calibrated per nm^2 , and represents the number of lattice Mo vacancies counted in each frame.

Movie 004, with the highest dose rate of $1.28 \text{ pC nm}^{-2} \text{ s}^{-1}$, is the most un-

stable. However, this is not solely due to the dose rate, as the upper left section of movie 004 initially contains a hole. This hole, easily visible in Figure 5.21(b), remains relatively unchanged throughout the 2 minutes of MR analysis, gaining or losing \sim 3-5 more Mo vacancies (0.2 - 0.35 Mo vacancies nm^{-2}) compared to the initial frame. The frame drifts over the hole, which causes some issues with image registration. Therefore, movie 004 is not used for comparison with high dose rate movies in the next section.

Movies 003 and 005 were imaged with respective dose rates of 1.13 and 0.87 $pC \text{ nm}^{-2} s^{-1}$. As shown in Figure 5.21(c) and Figure 5.21(d), their respective lattices are intact without holes. At 60 s and 80 s, a Mo vacancy forms in movie 003, as indicated by the two highest peaks in the blue, solid line. This single Mo vacancy hole stabilises, fills and reappears several times to various extents throughout the movie. This indicates that at this dose rate the lattice is stable for extended periods of time, and in some cases will ‘heal’ itself which prevents extensive damage. This healing process is most likely the ejected lattice Mo atoms existing as surface adatoms - due to not being fully ejected - and returning to their lattice positions. Mo atoms are discussed in Figure 5.16 in the previous subsection. At the lowest dose rate of 0.87 $pC \text{ nm}^{-2} s^{-1}$, movie 005 returns the lowest average Mo lattice vacancies per squared nm at 0.046 vacancies/ nm^2 . For comparison, movies 003 and 004 have an average of 0.05 and 0.39 vacancies/ nm^2 .

During the first two minutes of these movies, the total cumulative dose is \sim 120 $pC \text{ nm}^{-2}$ (see Figure A.11 in appendices for details). As mentioned, their patterns repeat for the remaining minutes not analysed by the MR. Therefore, for equivalent total doses, these monochromated dose rates were found not to be detrimental to the Se implanted monolayer MoS_2 . These low dose rates were not imaged for the equivalent total doses of the higher dose rates discussed later. This was due to pragmatic reasons, as each movie was imaged for similar overall times (\sim 6 minutes).

To achieve higher dose rates, the monochromating slit was removed. Figure 5.22 shows the resulting lattice Mo vacancies after 130 seconds for these higher dose rates, which range from \sim 4 to \sim 12 $pC \text{ nm}^{-2} s^{-1}$. The most damaged region is in movie 008 (dash-dotted purple line) due to the high dose rate of 12.30 $pC \text{ nm}^{-2} s^{-1}$. This dataset was used for verification of the model refiner (MR), as shown in Figure 5.11 and Figure 5.12 in subsection 5.2.1 above. The initial frame of each movie is shown on the right in Figure 5.22(b-e). In Figure 5.22(b) the lattice is relatively stable with a single Mo vacancy visible on the right side. A hole forms (many Mo vacancies) during movie 008 and increases rapidly from $t=60$ s, tripling in size in the space of another 60 s. In movie 012 (dashed yellow line), a similar

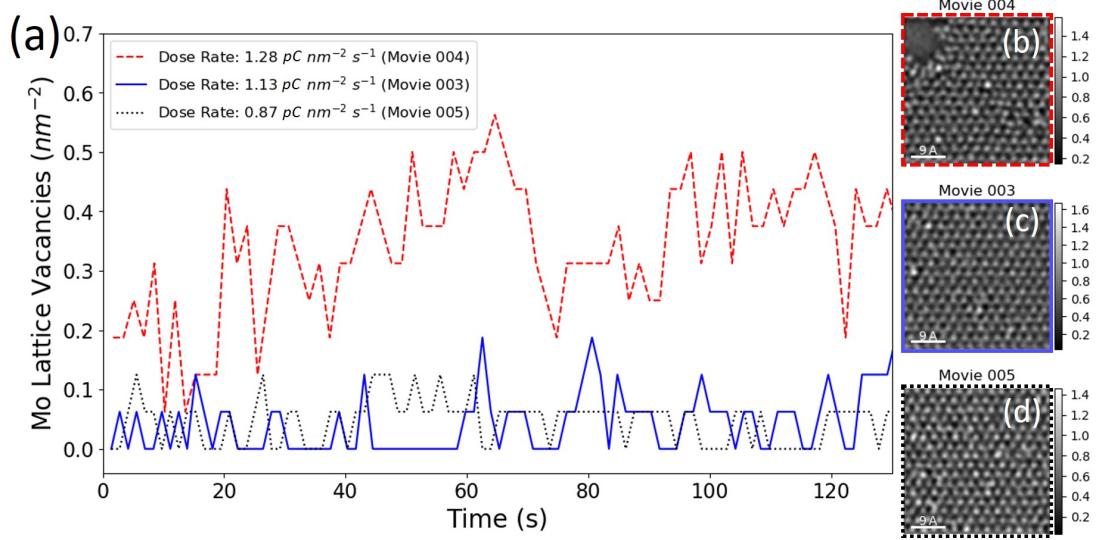


Figure 5.21: (a) Lattice Mo vacancies per squared nm as a function of time for low (monochromated) dose rate movies of Se implanted MoS_2 . Initial frames of (b) movie 004 (dashed red line), movie 003 (solid blue line), and movie 005 (dotted black line). Scale bars are 9 Å.

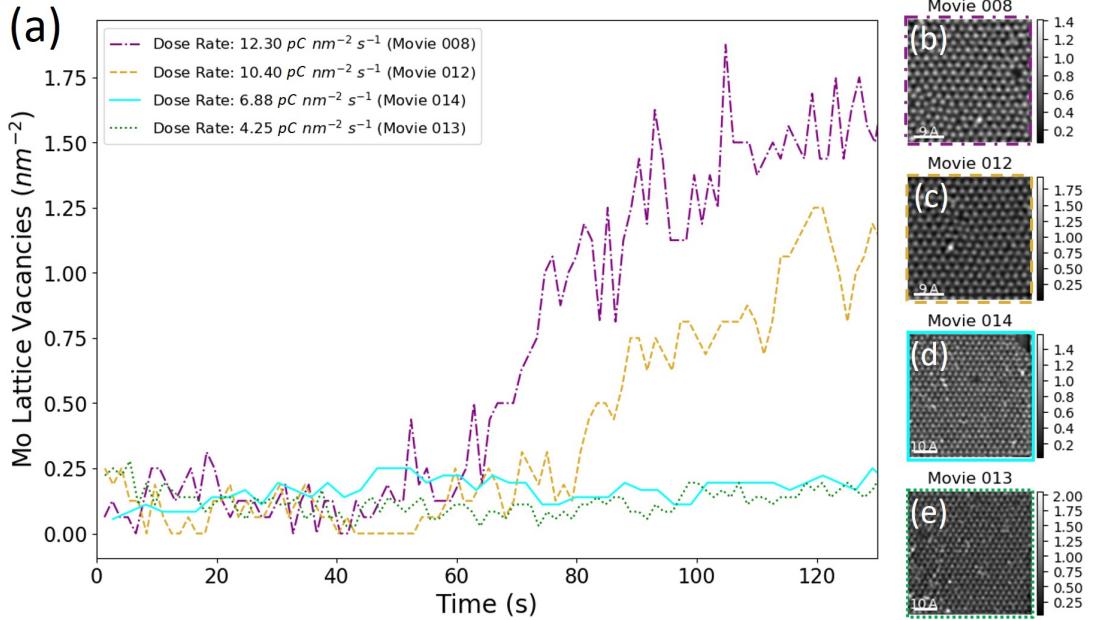


Figure 5.22: (a) Lattice Mo vacancies per squared nm as a function of time for high (non-monochromated) dose rate movies of Se implanted MoS_2 . Initial frames of (b) movie 008 (dash-dotted purple line), movie 012 (dashed yellow line), movie 014 (solid cyan line), and movie 013 (dotted green line). Scale bars (b-c) 9 Å, (d-e) 10 Å.

5.2. Atom Dynamics in Implanted TMDs with the MR Tool

hole trend appears after ~ 20 more seconds. This delay is possibly due to the lower dose rate of $10.40 \text{ pC nm}^{-2} \text{ s}^{-1}$. This is a dose rate difference of $1.9 \text{ pC nm}^{-2} \text{ s}^{-1}$ between movie 008 and 012. It is worth noting that this difference is higher than the dose rate of each movie shown before in Figure 5.21.

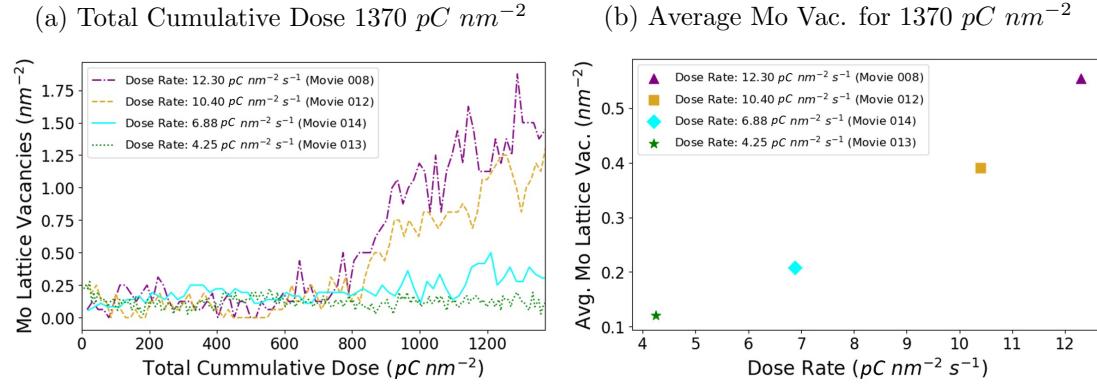


Figure 5.23: (a) Lattice Mo vacancies as a function of the total cumulative dose for each non-monochromated, high dose rate, movie. Comparable time periods are presented in Figure 5.24(a). (b) Average lattice Mo vacancies as a function of the dose rate for each non-monochromated, high dose rate, movie.

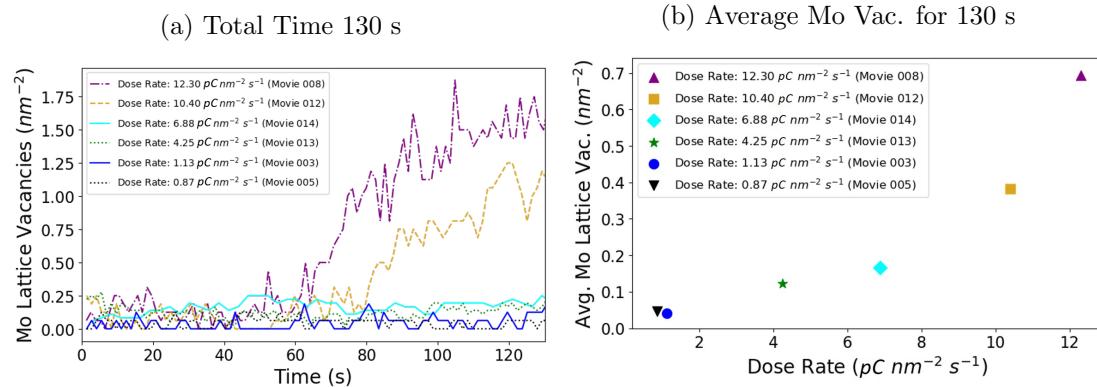


Figure 5.24: (a) Lattice Mo vacancies as a function of time for each movie. (b) Average lattice Mo vacancies as a function of the dose rate for each movie.

For movie 014 (solid cyan line), the number of lattice Mo vacancies is consistent throughout the 130 s time-frame. The hole visible at the top right of Figure 5.22(d) grows steadily throughout the movie, though mostly after 130 s. It should be noted that although the hole grows in movie 014, the monolayer lattice in general becomes defective, with many chalcogen vacancies and lattice distortions. This damage is not quantified as easily as holes. In movie 013 (dotted green line) no holes form during the entire 7 minute movie (only initial 130 s shown here). The monolayer lattice is stable under the beam, as shown by the dotted green line in Figure 5.22(a), and does not change from the initial frame displayed in Figure 5.22(e). It would

seem then, that the lower dose rate of $4.25 \text{ pC nm}^{-2} \text{ s}^{-1}$ does not damage for realistic imaging time-frames, as concluded for the even lower dose rates discussed in Figure 5.21 above.

Of interest is the delay between the hole formation in movies 008 and 012 in Figure 5.22(a). It is possible that this delay in hole formation is due to the fact that it takes longer for the total dose to accumulate in movie 012. Figure 5.23(a) shows the effect of total cumulative dose for each dose rate. In it, the Mo vacancies are plotted as a function of total cumulative dose, rather than time. Low dose rate movies, such as in Figure 5.21, do not accumulate enough total dose for comparison with these high dose rate movies. For movies 008 and 012, it is clear that damage begins at $\sim 800 \text{ pC nm}^{-2}$. Therefore, the delay in Figure 5.22(a) is due to the total cumulative dose. Additionally, movie 014, which seemed stable in Figure 5.22(a), begins to damage around the same total dose as movies 008 and 012, though at a lower rate. The relatively low dose rate movie 013 is again stable throughout. To reach a comparable total cumulative dose, the MR tool analysed ~ 3 times as many frames for movie 013 as the high dose rate movie 008.

The stability of the lattice is reflected well by plotting the average Mo lattice vacancies for the same high dose rates. The low dose rates from Figure 5.21 did not accumulate a comparable total dose, and are therefore not shown here. The lower dose rate of $4.25 \text{ pC nm}^{-2} \text{ s}^{-1}$ in movie 013 produces 0.12 Mo vacancies per squared nm, compared to 0.21 Mo vacancies, 0.39 Mo vacancies and 0.55 Mo vacancies in movies 014, 012 and 008, respectively.

Both low and high dose rates are plotted together as a function of time for a period of 130 s in Figure 5.24(a). Above the dose rate of movie 013 ($4.25 \text{ pC nm}^{-2} \text{ s}^{-1}$), the lattice damages considerably. As the dose rate increases, the trend of the average lattice Mo vacancies in Figure 5.24(b) suggests an exponential relationship. This is physically realistic, as at a very high dose rate the lattice Mo vacancies in the imaging area would rise to its maximum value, resulting in complete destruction of the lattice. Conversely, imaging the same region at extremely low dose rates should result in very few Mo vacancies, such as attempted with compressed sensing reconstruction [418].

The data points measured for average Mo lattice vacancies as a function of both time and total cumulative dose are fitted with exponential regression in Figure 5.25 and Figure 5.26, respectively. The total cumulative dose fit is better than the time fit as indicated by the lower standard errors of regression σ_A and σ_b in Figure 5.26. This is logical, as the total cumulative dose reflects more closely the damage indicator than time. Of course, there is only four data points, and cannot be considered conclusive alone. Together with the $t=130 \text{ s}$ fit in Figure 5.25,

and with the physically realistic trend, it is reasonable to conclude that the average Mo lattice vacancies follow an exponential model as a function of both time and total cumulative dose. Using these models, one can estimate the ‘worst-case’ dose rate scenario for damage to the lattice in the form of Mo vacancies to be $>15 \text{ pC nm}^{-2} \text{ s}^{-1}$. This dose rate would eradicate the lattice to nothing in ~ 60 seconds. Higher dose rates again would accelerate this damage.

From these models and the pattern of Mo vacancies as a function of total cumulative dose, we propose a detrimental threshold dose rate (TDR). Only when this TDR is exceeded is total cumulative dose a factor for consideration, as seen in Figure 5.23(a) above. In other words, if the dose rate is below the TDR, you can imagine the sample for long periods of time without any negative effects. Negative effects include large holes (several Mo vacancies), or numerous lattice distortions (seen in movie 014). From Figure 5.25 the TDR is estimated to be between 4.25 and $6.88 \text{ pC nm}^{-2} \text{ s}^{-1}$ for single Mo atoms in Se implanted monolayer MoS₂, though it likely is similar for pristine MoS₂.

The TDR is explained via the inelastic scattering damage mechanism of radiolysis (ionisation), as 60 keV electrons are below the displacement threshold energy for S in monolayer MoS₂ [see Subsection 2.1.2(Elastic Scattering)]. As noted by Egerton *et al*, mass loss solely due to radiolysis is often exponential [161, 179]. Ionisation damage can affect a semiconductor material such as the 2H monolayer MoS₂ imaged here, as discussed in

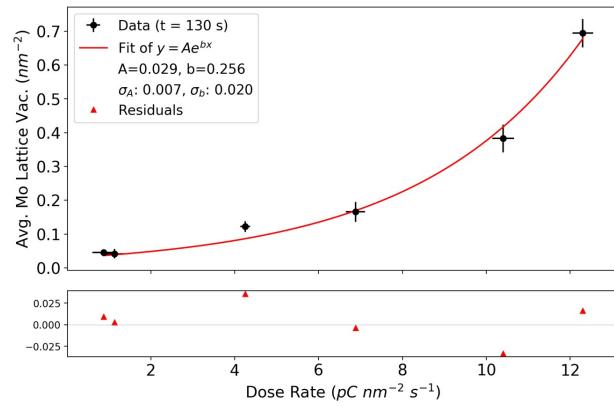


Figure 5.25: Fit of the average Mo lattice vacancies as a function of electron beam dose rate for a total time of 130 s. The equation of the line is $y=0.029e^{0.256x}$.

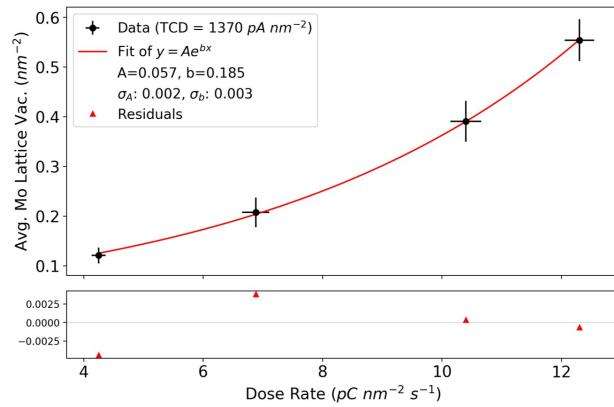


Figure 5.26: Fit of the average Mo lattice vacancies as a function of electron beam dose rate for a total cumulative dose of 1370 pC nm^{-2} . The equation of the line is $y=0.057e^{0.185x}$.

Subsection 2.1.2(Inelastic Scattering). A fast electron excites an electron in the valence band and leaves a positive hole in the valence band (an inner-shell excitation can also occur, though it is less likely [175]). The electron-hole pair can form further electron-hole pairs if not immediately thermalised or emitted as a secondary electron. If the valence band hole is not filled, lattice bonds may be broken by atomic vibrations subsequently weakening the lattice. Further scattering can displace atoms with weakened bonds because of a decrease in displacement threshold energy [175]. Therefore, if the rate at which valence holes are created is equal to the rate at which the valence holes are filled by free electrons, the lattice is stable. In metals, this is not usually an issue, as there is an abundant supply of free electrons (high conductivity). However, once the rate of valence hole creation is higher than valence hole filling, irreparable damage of the lattice is likely. The greater the disparity between the two, the faster and more widespread the damage. This is because as lattice holes form in the monolayer material, the displacement threshold energy decreases for the atoms at the edge of the lattice hole. This, in turn, causes further elastic sputtering of atoms and accelerates the lattice hole growth.

From the above findings and fittings in Figure 5.25 and Figure 5.26, we can conclude that by keeping the dose rate below the TDR of $\sim 4.25 \text{ pC nm}^{-2} \text{ s}^{-1}$, the monolayer sample can be imaged for extended periods of time without destroying the crystal lattice. On the opposite end of the scale, because the relationship between lattice vacancies and dose rate is exponential, increasing the dose rate just several pC above the TDR will unsustainably damage the lattice.

It would be interesting to perform theoretical calculations to confirm the hypothesis that imaging above the TDR can cause lattice holes in monolayer MoS₂. A clearer understanding of the damage mechanisms is needed beyond the above experimental model. Not all analysed movies fit within this model: sample drift over holes (movie 004 above), as well as contamination cause deviations from the model. In the following section, we will discuss the protection the contamination provides the lattice.

Effect of Contamination on Movie Analysis

The effect of surface hydrocarbon contamination on beam-induced lattice damage is inspected by comparing movies 008 and 010. Movie 008 was analysed in the previous sections. A large hole begins to form after ~ 60 s, as shown below by the sharp increase of the purple line in Figure 5.27(a). In movie 008 (dash-dot purple line) and movie 010 (solid teal line), the dose rates are 12.3 and $11.5 \text{ pC nm}^{-2} \text{ s}^{-1}$, respectively.

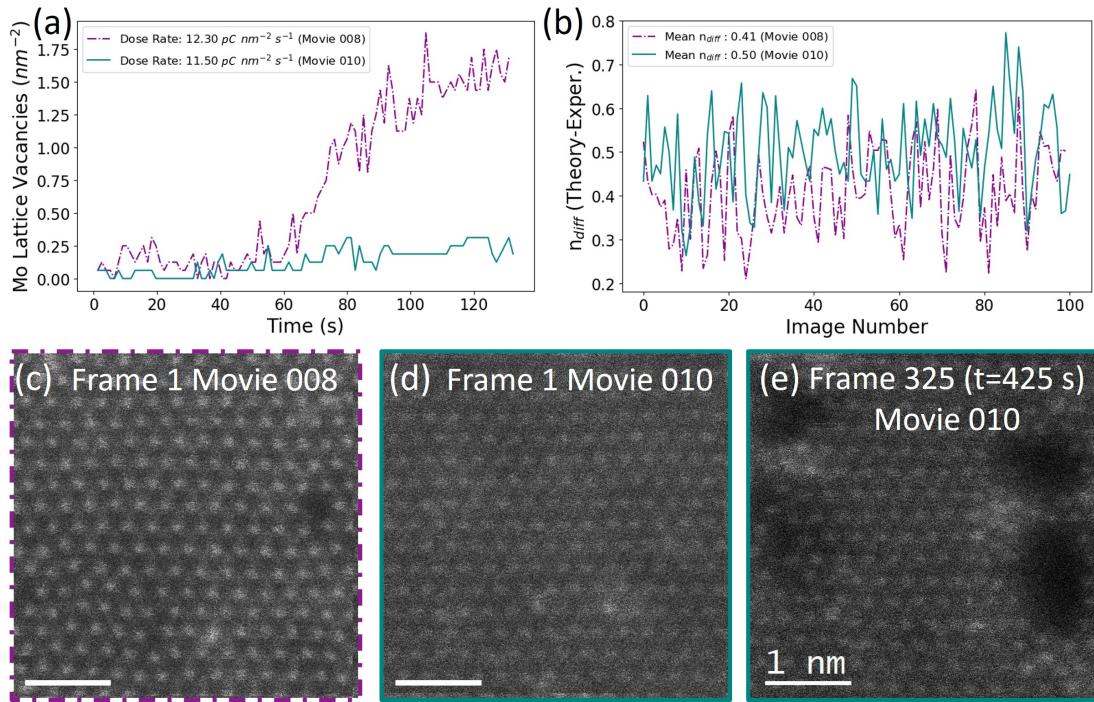


Figure 5.27: (a) Number of lattice Mo vacancies counted per square nm for movie 008 (dash-dot purple line) and movie 010 (solid teal line). (b) The difference, n_{diff} , between the theoretical and experimental Z-contrast exponent n for movies 008 and 010. The mean n_{diff} for each is given in the legend. (c-d) Initial frames used in the MR analysis for movies 008 and 010, respectively. (e) Frame 325 (time=425 s) of movie 010. This frame is of the same area and ~ 293 seconds after the final MR analysed frame in (a). Scale bars are 1 nm.

Both dose rates are above the proposed threshold dose rate (TDR) of $\sim 4.25 \text{ pC nm}^{-2} \text{ s}^{-1}$. One would assume, therefore, that a hole would form in movie 010 in the same manner as seen in movie 008. However, in Figure 5.27(a), within similar time-frames (and total cumulative doses), movie 010 does not form a hole commensurate with the hole in movie 008. This is most likely due to the higher level of hydrocarbon contamination present on the surface of the monolayer in movie 010. The contrast difference between the initial movie frame in Figure 5.27(c) and Figure 5.27(d) is due to contamination. This claim is quantitatively supported by Figure 5.27(b), which is a plot of the difference between the theoretical and ex-

perimental Z-contrast exponent n ($n_{diff} = n_{theory} - n_{exper}$). Lower values of n_{diff} indicate a higher Z-contrast exponent and therefore better match with theory and simulation. The mean value of the n_{diff} is 0.41 and 0.50 for movies 008 and 010, respectively. The higher mean n_{diff} for movie 010 indicates more contamination on the monolayer. This is because higher levels of contamination have a disproportionate effect on the intensity readings of lower atomic number elements, such as sulphur, as discussed in more detail in subsection 4.1.1(Microscopy of Se Implanted MoS₂). The mean n_{diff} for every movie analysed in this chapter is shown in Figure A.12 in Subsection A.4.

Therefore, this contamination has a stabilising effect on the MoS₂ lattice. The stabilising effect of the contamination on the surface of movie 010 was seen in several other contaminated regions during single frame experiments [see Figure 4.5 in subsection 4.1.1(Microscopy of Se Implanted MoS₂)]. The effect only delays the destruction of the lattice, however. Later in movie 010 – an example frame is shown in Figure 5.27(e) – several large holes are created. Frame 325 in Figure 5.27(e) was not included in the MR analysis, as only the initial 100 frames were analysed.

In summary, contamination can help to stabilise the lattice for high dose rates, or at least delay the detrimental effect of a high dose rate. Of course, if one is aiming to image the lattice without damaging it, we would recommend locating a clean region, and imaging under the threshold dose rate measured above ($\sim 4.25 \text{ pC nm}^{-2} \text{ s}^{-1}$).

5.3 Chapter Conclusions - Atom Dynamics in 2D TMDs

In the first part of this chapter, open-source methods for the automatic analysis of Se implanted MoS₂ were developed. The methods are not limited to 2D TMDs. In subsection 5.1.1, two algorithms were designed based on iterative and automatic experimental comparison with scanning transmission electron microscopy (STEM) image simulations. Following this, we established a multi-functional Python class called the Model Refiner (MR) in subsection 5.1.2. The MR tool automatically simulates and calibrates data, and tracks all changes made by the above-mentioned algorithms between experiment and simulation. It also is useful for completing and plotting element quantification. The critical aspect of the developed algorithms and MR tool is their simple integration with the popular HyperSpy and Atomap Python packages.

In the second part of this chapter, the MR tool was employed for the automated analysis of atomic resolution STEM movies of Se implanted MoS₂ monolayers. After verifying the approach to specific steps for the analysis of the movies in subsection 5.2.1(Image Series Automated Analysis Steps), the atom lattice was analysed over time. The energetics of atom movement, adatoms, vacancies and holes were analysed in subsection 5.2.2(Analysis of Lattice Energetics). Following this, subsection 5.2.3(Effect of Electron Beam) investigated the effect of the electron beam on the lattice as a function of electron beam dose rate for specified time-frames and total doses. By quantifying the elements, an exponential relationship between the Mo lattice vacancies and electron beam dose rate was fitted. Additionally, we hypothesise a detrimental threshold dose rate (TDR) of $\sim 4.25 \text{ pC nm}^{-2} \text{ s}^{-1}$, above which the monolayer MoS₂ lattice is damaged irreversibly. Finally, Subsection 5.2.3(Effect of Contamination on Movie Analysis) discussed and quantified the stabilising effect of surface contamination on the monolayer lattice.

In the following chapter, we maintain a similar layout; we begin with the development of automated open-source Python code and subsequently apply this code to the analysis of a different type of 2D material, ferroelectric domain walls.

Chapter 6

Results - Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls

Ferroelectric domains and domain wall (DW) research has become increasingly popular in recent years [419]. In the first part of this chapter, we will discuss the tools developed for visualisation of sub-atomic shifts in Subsection 6.1(Tools for Visualisation of Subatomic Shifts). The developed Python functions are open-source and built on the data structure of NumPy, Atomap, and HyperSpy. They are included in the polarisation module of the TEMUL toolkit [13] and in Sub-section A.6 (TEMUL: Transmission Electron Microscopy University of Limerick). Because they are integrated with popular Python packages, they are expected to be easily accessible for other researchers. Such tools are then utilised for in-depth analysis of atomic resolution scanning transmission electron microscopy (STEM) images of DWs in ferroelectric materials in Subsection 6.2(2D Domain walls).

6.1 Tools for Visualisation of Subatomic Shifts

Open-source Python tools were developed for the analysis of atomic resolution images and visualisation of material properties. These tools are extensible and can be easily adapted by those with basic knowledge of scripting. The majority of the functions discussed in this section are located in the `polarisation.py` module of the TEMUL toolkit on GitHub [13]. The full `polarisation.py` module code can be found in the appendices in subsection A.6.4. These tools are available for free and can be used by anyone. The functions are based on the methods described in subsection 3.4.7(Methods for Analysing Atomic Movement and Polarisation). Table 6.1 shows an overview of each of the most important functions. The name, a brief explanation and the data output are shown in Table 6.1, as well as a link to the functions' subsections.

Table 6.1: The main functions of the `polarisation.py` module [13]. A link, brief explanation and action taken by each function is shown (Hspy refers to Hyperspy).

Function	Explanation	Action/Output
<code>plot_polarisation_vectors</code> (subsection 6.1.1)	Visualise vectors (uv) at given coordinates (xy). Many plotting options	Matplotlib Pyplot
<code>find_polarisation_vectors</code> (subsection 6.1.2)	Calculate the uv components from one list of coordinates to another.	Two Lists: u, v
<code>atom_deviation_from_straight_line_fit</code> (subsection 6.1.3)	Calculate the relative movement of an atom from a chosen zone.	Four lists: x,y,u,v
<code>plot_atom_deviation_from_all_zone_axes</code> (Subsection 6.1.3)	Repeat <code>atom_deviation_from_straight_line_fit</code> for each zone.	Matplotlib Pyplots
<code>combine_atom_deviations_from_zone_axes</code> (subsection 6.1.4)	Add together the vectors from each <code>atom_deviation_from_straight_line_fit</code> .	Four lists: x,y,u,v
<code>get_average_polarisation_in_regions</code> <code>get_average_polarisation_in_regions_square</code> (subsection 6.1.5)	Calculate the average vectors 'blockwise' in an image for lower magnification displays	Four lists: x,y,u,v
<code>delete_atom_planes_from_sublattice</code> (subsection 6.1.6)	Remove unwanted atom planes from a sublattice.	Zone with deleted planes
<code>Strain Map</code> (subsection 6.1.7)	Create a strain map	Hspy Signal 2D
<code>Angle of Zone Axes Map</code> (subsection 6.1.7)	Create a map of the zone axis angles	Hspy Signal 2D
<code>Lattice Parameter Ratio Map</code> (subsection 6.1.7)	Create a map of the ratio of two lattice parameters	Hspy Signal 2D

In each function's subsection, a detailed description, overview of the relevant parameters, short code segment of its use, and visualisation of the output are shown on a test dataset. The test dataset used is simply an example STEM image, wherein the atom positions have been located with the Atomap Python package. In some cases, the technique behind a function will be described in the methodology and referenced accordingly. Additionally, the use of each function for the analysis of atomic resolution STEM images of DW structures in Subsection 6.2(2D Domain walls) is discussed.

The `plot_polarisation_vectors` function will be discussed first as it is used to visualise all other functions in this section. Then, functions that calculate the

atom movement vectors in various ways such as `find_polarisation_vectors`, `atom_deviation_from_straight_line_fit`, and others will be explained. The material property maps will then be presented.

6.1.1 Plot Polarisation Vectors

The `plot_polarisation_vectors` function is the core visualisation tool in the above-mentioned `polarisation.py` Python module developed during the thesis. It allows one to view the movement of atomic columns by supplying to it the original atom coordinates (x, y) and the movement vector components (u, v). Because of this, the functions that are used to calculate polarisation, atomic movement etc., generally output lists of x, y, u , and v values or similar lists. Atom movement and polarisation are defined in the methodology in Subsection 3.4.7(Polarisation & Atomic Movement). This data structures allows one to start by using the Atomap package to find the atomic column coordinates, and to then seamlessly use the `plot_polarisation_vectors` function to visualise sub-Ångström shifts. The `plot_polarisation_vectors` function is utilised for each material studied in Subsection 6.2(2D Domain walls).

The function is easy to use, with only the x, y, u, v lists and the image needed to plot a figure, but is also quite extensible. An example output of these required parameters can be seen in Figure 6.1(a-b). In Figure 6.1(a), there is no background image because the parameter `overlay=False`. When `overlay=True`, the vectors are overlaid on the image, as seen in Figure 6.1(b). When `plot_style='vector'`, an arrow is shown at the top right of the image for scale.

Rather than a single colour, the arrows can be defined by a graduated colour-map by setting the parameter `plot_style='colormap'`, rather than the default `plot_style='vector'`. An example of this is shown in Figure 6.1(c). The default colour-map is viridis, and can be changed to any colour-map in the Matplotlib Python package (or related packages, e.g., colrcet). Another `plot_style` option is a contour-map (`plot_style='contour'`). The contour-map is shown in Figure 6.1(f).

The image sampling, described in Subsection 3.4.2(Image Calibration), can be provided to scale the vector arrows if not already scaled. The `sampling` display units are changeable via the `units` parameter, as seen in line 27 of Code Segment 6.1 and in Figure 6.1(d). The `units` parameter only changes the unit being displayed, e.g., nm, Å, pm, and does not effect the `sampling` number. The vectors can be presented as units vectors by setting the parameter `normalise=True`. This is shown in both Figure 6.1(e) and Figure 6.1(f). Unit vectors are often useful

6.1. Tools for Visualisation of Subatomic Shifts

for visualising the overall uv movement of atomic columns in an image. Each of these descriptions are implemented in an example script in Code Segment 6.1.

```
1 # Example of the plot_polarisation_vectors function
2 import atomap.api as am
3 from temul.polarisation import (
4     combine_atom_deviations_from_zone_axes,
5     plot_polarisation_vectors)
6
7 # get an example atom lattice
8 atom_lattice = am.dummy_data.get_polarization_film_atom_lattice()
9 sublatticeA = atom_lattice.sublattice_list[0]
10 sublatticeA.find_nearest_neighbors()
11 sublatticeA.refine_atom_positions_using_center_of_mass()
12 sublatticeA.construct_zone_axes()
13
14 # Create some u, v components for visualisation
15 x, y, u, v = combine_atom_deviations_from_zone_axes(sublatticeA,
16     save=None)
16 monitor_dpi = 25 # make the image larger than default
17
18 # Basic plot of some atom movement
19 plot_polarisation_vectors(x, y, u, v, image=sublatticeA.image,
20                             normalise=False, save=None,
21                             plot_style='vector', color='r',
22                             overlay=False, title='Vectors',
23                             monitor_dpi=monitor_dpi)
24
25 # Scaled vector plot with red arrows overlaid on the image:
26 plot_polarisation_vectors(x, y, u, v, image=sublatticeA.image,
27                             normalise=False, save=None, units='nm',
28                             title='Vectors on Image', sampling=3.03,
29                             plot_style='vector', color='r',
30                             overlay=True, monitor_dpi=monitor_dpi)
31
32 # vector plot with colormap viridis:
33 plot_polarisation_vectors(x, y, u, v, image=sublatticeA.image,
34                             normalise=False, save=None,
35                             title='Colormap Vectors',
36                             plot_style='colormap', overlay=False,
37                             cmap='viridis', monitor_dpi=monitor_dpi)
38
39 # colormap arrows with example sampling applied:
40 plot_polarisation_vectors(x, y, u, v, image=sublatticeA.image,
41                             sampling=3.03, units='nm',
42                             title='Colormap Scaled Vectors',
```

6.1. Tools for Visualisation of Subatomic Shifts

```
43         normalise=False, plot_style='colormap',
44         overlay=False, save=None, cmap='plasma',
45         monitor_dpi=monitor_dpi)
46
47 # vector plot with colormap and unit vectors:
48 plot_polarisation_vectors(x, y, u, v, image=sublatticeA.image,
49                            normalise=True, save=None,
50                            title='Colormap Unit Vectors',
51                            plot_style='colormap', color='r',
52                            overlay=False, cmap='inferno',
53                            monitor_dpi=monitor_dpi)
54
55 # unit vectors vectors on a plt.contourf map:
56 plot_polarisation_vectors(x, y, u, v, image=sublatticeA.image,
57                            normalise=True, plot_style='contour',
58                            title='Contour Map Unit Vectors',
59                            overlay=False, pivot='middle',
60                            monitor_dpi=monitor_dpi, save=None,
61                            color='darkgray', cmap='viridis')
```

Code Segment 6.1: Example script using the `plot_polarisation_vectors` function.
The outputs from this code is displayed in Figure 6.1

6.1. Tools for Visualisation of Subatomic Shifts

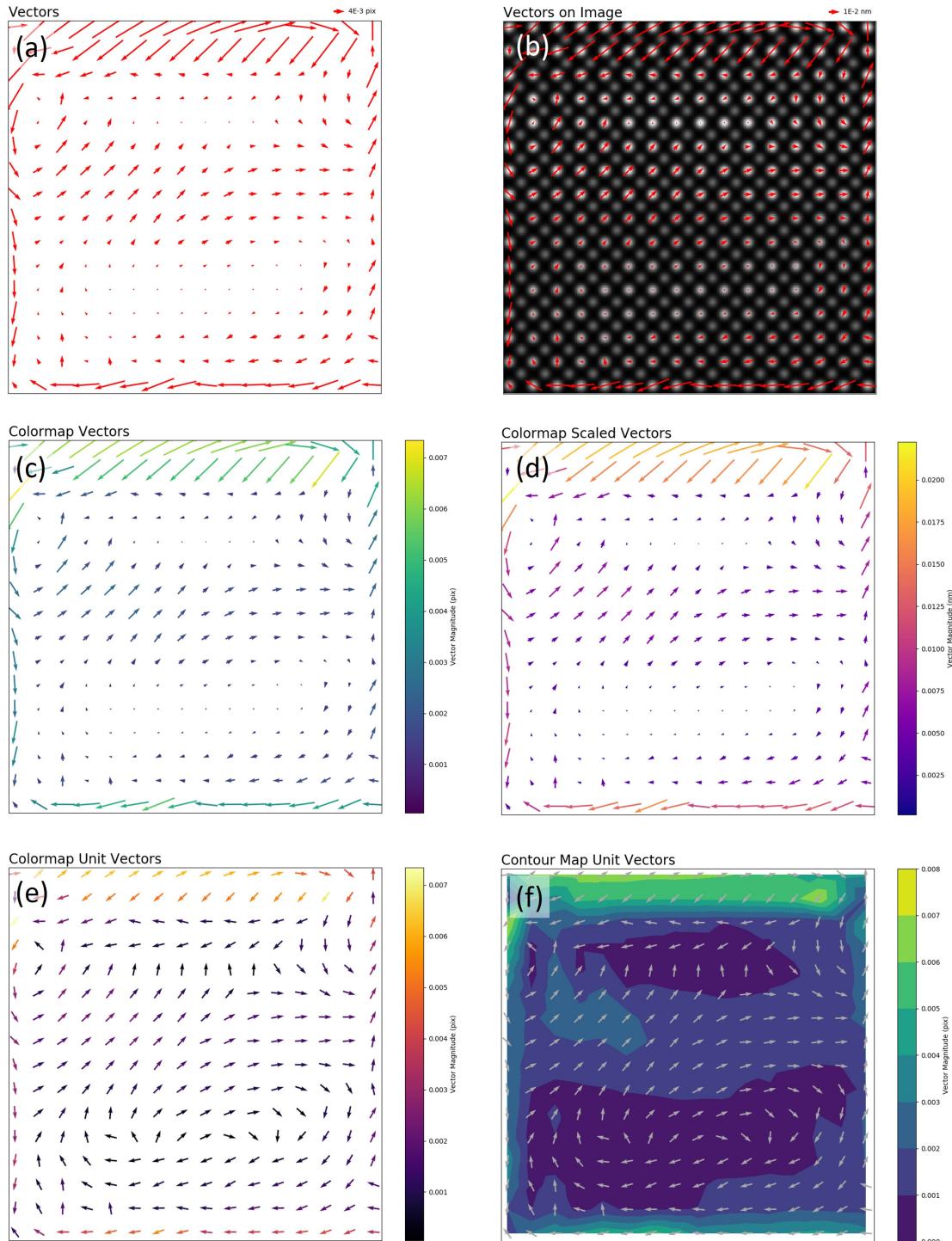


Figure 6.1: Plots of polarisation in an example atomic resolution image showing (a) a basic vector plot, (b) scaled vectors (nm) overlaid on the example image, (c) colormap of the vectors, (d) colormap of the scaled vectors (nm), (e) colormap of unit vectors, and (f) a contour map of unit vectors. The atom position data for all images (a-f) is the same. The script in Code Segment 6.1 above produces these figures.

6.1.2 Find Polarisation Vectors

The `find_polarisation_vectors` function is used to calculate the vector components (u, v) for two lists of atomic column coordinates (x, y). It calculates the u, v distances from `atom_positions_A` to `atom_positions_B`. In the case of the PTO DW junction in subsection 6.2.1, the movement from the ideal coordinate (x_0, y_0) to the actual coordinate (x_1, y_1) defines the atomic movement. However, the polarisation is often the opposite of the movement of the atoms. Atom movement and polarisation are defined in the methodology in Subsection 3.4.7(Polarisation & Atomic Movement). This can easily be calculated by reversing the order of the atom positions input to the `find_polarisation_vectors` function. An example script and output of each is displayed in Code Segment 6.2 and Figure 6.2, respectively.

The `find_polarisation_vectors` function was used to analyse the PTO junction in Figure 6.11 and Boracite in ?? in Subsection 6.2(2D Domain Walls).

```
1
2 ''
3 For the centre point of four atoms in a regular Atom Lattice,
4 the Atomap "get_polarization_from_second_sublattice"
5 function can be used instead.
6 ''
7
8 import numpy as np
9 import atomap.api as am
10 from temul.polarisation import (
11     find_polarisation_vectors,
12     plot_polarisation_vectors)
13
14 atom_lattice = am.dummy_data.get_polarization_film_atom_lattice()
15
16 for sublattice in atom_lattice.sublattice_list:
17     sublattice.find_nearest_neighbors()
18     sublattice.refine_atom_positions_using_center_of_mass()
19
20 sublatticeA = atom_lattice.sublattice_list[0]
21 sublatticeB = atom_lattice.sublattice_list[1]
22 sublatticeA.construct_zone_axes()
23
24 #sublatticeA.plot_planes()
25 zone_axis = sublatticeA.zones_axis_average_distances[2]
26
27 # In future versions, this function will be combined with the
28 # opposite zone for a more robust calculation
```

6.1. Tools for Visualisation of Subatomic Shifts

```
29 mid_points_of_A = sublatticeA.find_missing_atoms_from_zone_vector(
30     zone_axis, vector_fraction=0.5)
31
32 atom_positions_B = np.array(sublatticeB.atom_positions).T
33 x,y = atom_positions_B.T.tolist()
34
35 # distance from sublatticeA mid-point to actual coordinate of
36 # sublatticeB
37 u,v = find_polarisation_vectors(mid_points_of_A, atom_positions_B,
38     save=None)
39
40 plot_polarisation_vectors(x, y, u, v, image=atom_lattice.image,
41     normalise=False, save=None, color='r',
42     plot_style='vector', overlay=True,
43     monitor_dpi=25)
44
45 # distance from actual coordinate of sublatticeB to sublatticeA
46 # mid-point
47 u,v = find_polarisation_vectors(atom_positions_B, mid_points_of_A,
48     save=None)
49
50 plot_polarisation_vectors(x, y, u, v, image=atom_lattice.image,
51     normalise=False, save=None, color='r',
52     plot_style='vector', overlay=True,
53     monitor_dpi=25)
```

Code Segment 6.2: Example script using the `find_polarisation_vectors` function.
The outputs from this code is displayed in Figure 6.2

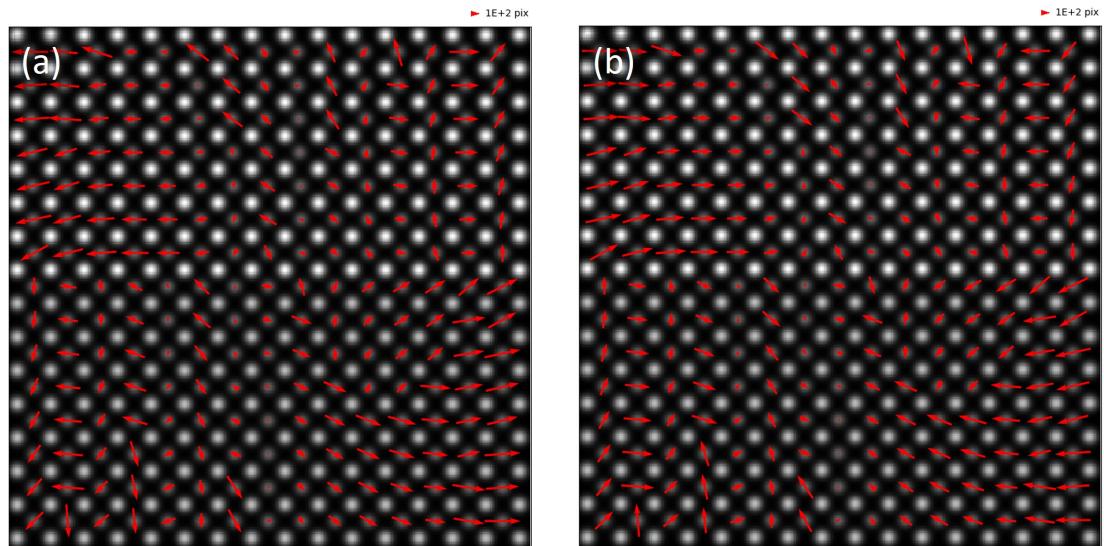


Figure 6.2: (a) Atomic movement calculated by the distance from the ideal, mid-point positions of each of the four atoms in the bright sublattice to the actual positions of the second sublattice (see Code Segment 6.2 Line 34). (b) The opposite vectors of (a) by reversing the `find_polarisation_vectors` parameters (see Code Segment 6.2 Line 42).

6.1.3 Atom Deviation from a Straight Line Fit

One can calculate the relative deviation of an atomic column from the atom planes containing the atomic column in any zone axis direction with the

`atom_deviation_from_straight_line_fit` function. A full description of the calculation is shown in the methodology in Subsection 3.4.7(Atomic Position Deviation from Zone Axes). In short, for all of the atomic columns in each atom plane in an Atomap constructed zone axis, a straight line fit is calculated. The perpendicular distance from each atomic column to its straight line fit is defined as the atom deviation vector. An example of the atom deviation vectors is plotted in Figure 6.3(b), with the example sublattice for reference in Figure 6.3(a). The `axis_number` parameter is 0 by default, but can be any sublattice zone axis index (integer, see Atomap.org for details).

The `atom_deviation_from_straight_line_fit` function is best used in situations where the polarisation across a domain wall (DW) must be calculated with only one main sublattice. This is investigated in ??(Lithium Niobate - LiNbO₃) and in subsection 6.2.1(PbTiO₃ Domain Wall Junctions).

```
1 # Example of the atom_deviation_from_straight_line_fit function
2 from temul.polarisation import (
3     atom_deviation_from_straight_line_fit,
4     plot_polarisation_vectors)
5 from temul.dummy_data import
6     get_distorted_cubic_sublattice_adjustable
7
8 sublattice = get_distorted_cubic_sublattice_adjustable(y_offset=5)
9 sublattice.construct_zone_axes(atom_plane_tolerance=1)
10 #sublattice.plot_planes()
11 sublattice.signal.plot()
12
13 x,y,u,v = atom_deviation_from_straight_line_fit(
14     sublattice, axis_number=0, save=None)
15
16 sublattice.signal.plot()
17 plot_polarisation_vectors(x, y, u, v, image=sublattice.image,
18                             normalise=False, save=None, color='r',
19                             plot_style='vector', overlay=True,
20                             monitor_dpi=35, scale=0.04)
```

Code Segment 6.3: Example script using the function

`atom_deviation_from_straight_line_fit`. The outputs from this code are displayed in Figure 6.3

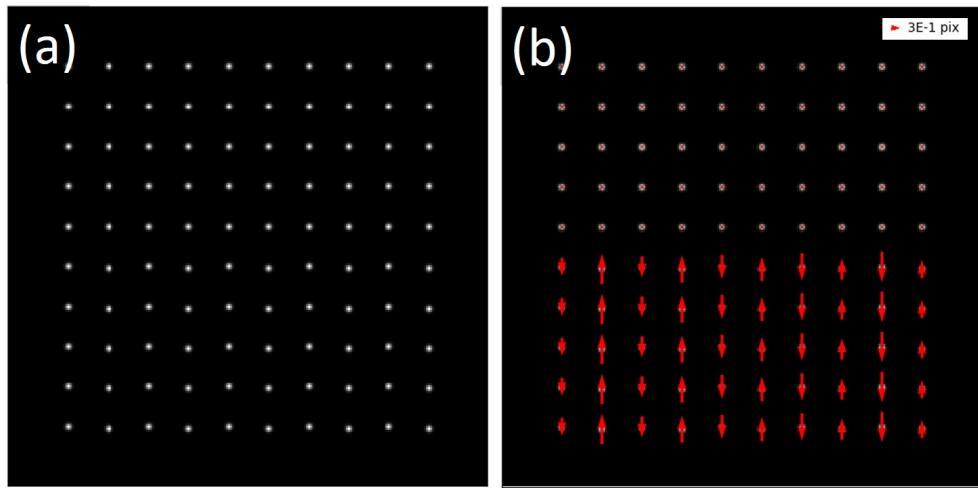


Figure 6.3: (a) Example sublattice with every second column in the bottom half of the image offset in y, creating a zigzag pattern. (b) Visualisation of the deviation of each atomic column from a horizontal straight line fit.

Plot Atom Deviation from all Zone Axes

To quickly view the atomic column deviations from all zone axes in the sublattice, the `plot_atom_deviation_from_all_zone_axes` function is used. Each of the zones in a test dataset is displayed in Figure 6.4(a 0-4) alongside the respective zone axis atomic column deviations in Figure 6.4(b 0-4). This function automatically uses the `plot_polarisation_vectors` function described above in subsection 6.1.3, and therefore includes all of its parameters.

```

1 import atomap.api as am
2 from temul.polarisation import
3     plot_atom_deviation_from_all_zone_axes
4
5 atom_lattice = am.dummy_data.get_polarization_film_atom_lattice()
6 sublatticeA = atom_lattice.sublattice_list[0]
7 sublatticeA.find_nearest_neighbors()
8 sublatticeA.refine_atom_positions_using_center_of_mass()
9 sublatticeA.construct_zone_axes()
10 sublatticeA.plot_planes()
11 plot_atom_deviation_from_all_zone_axes(sublatticeA,
12                                         plot_style='vector', color='r', save=None, monitor_dpi=40)

```

Code Segment 6.4: Example script using the function

`plot_atom_deviation_from_all_zone_axes`. The outputs from this code is displayed in Figure 6.4

6.1. Tools for Visualisation of Subatomic Shifts

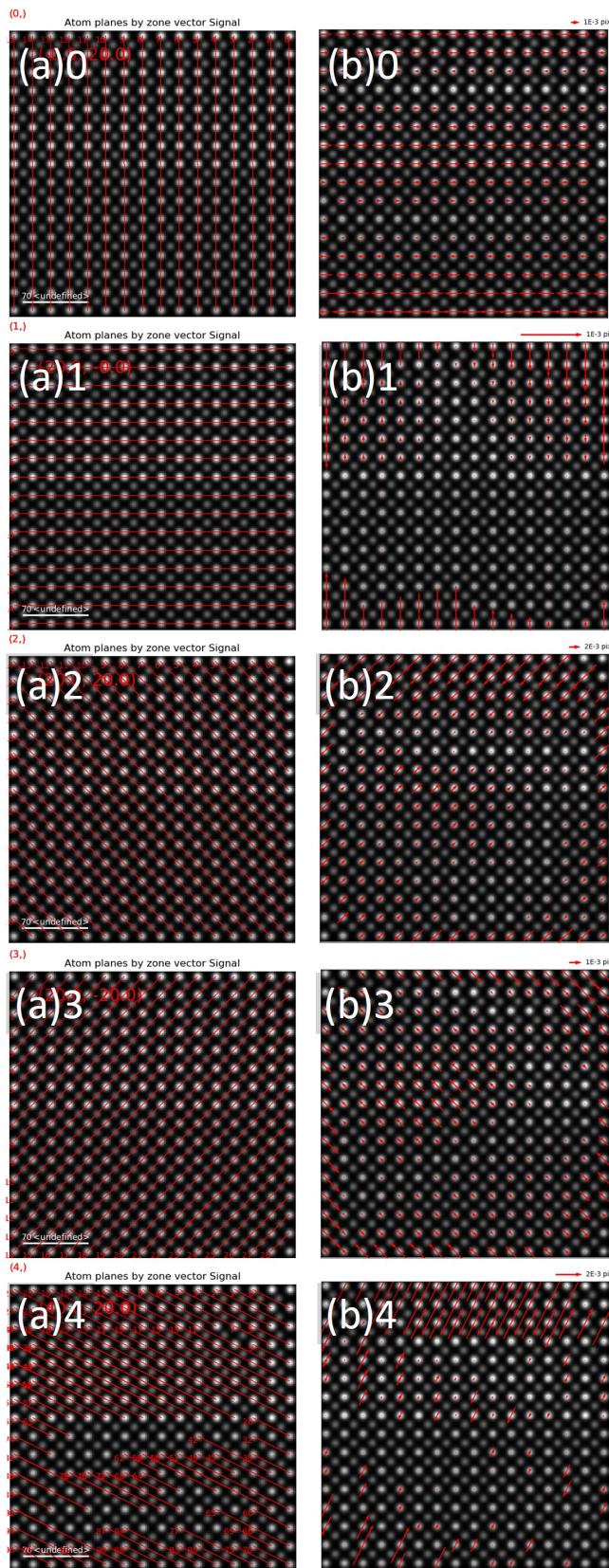


Figure 6.4: (a 0-4) The example sublattice's five zone axes. (b 0-4) The resulting atomic column deviations from each zone axis. Code Segment 6.4 will produce these plots.

6.1.4 Combine Atom Deviations from all Zone Axes

The atomic column deviations calculated by the above-described function `atom_deviation_from_straight_line_fit` and displayed for all zones by the `plot_atom_deviation_from_all_zone_axes` function can be combined in the new function `combine_atom_deviations_from_zone_axes`, as described in Subsection 3.4.7(Multiple Atom Position Deviations). The main parameter `sublattice` is the only required parameter, from which it will calculate and combine all atomic column deviations from all zones, as plotted in Figure 6.5(a).

The parameter `axes` can be set to a list of allowed zone axis indices. For example, if one sets `axes=[0, 2]` the combined atom deviation plot will only use the atomic column deviations from the 0 and 2 zone axes. It can be used to only choose well defined zone axes. For example, the zone axis in Figure 6.4(a4) above does not span across the image correctly, and would therefore not be included when calculating the combined atom deviation i.e., `axes=[0, 1, 2, 3]` in Code Segment 6.5(Line 28). The restricted `axes` output is shown in Figure 6.5(b), and differs only slightly from Figure 6.5(a). When `save` is not set to `None`, this function automatically also uses the `plot_polarisation_vectors` function described above in subsection 6.1.1, and therefore includes all of its parameters.

The function `combine_atom_deviations_from_zone_axes` was used to investigate different types of vortexes in PbTiO_3 Vortexes).

```
1 # Example of the combine_atom_deviations_from_zone_axes function
2 import atomap.api as am
3 from temul.polarisation import (
4     combine_atom_deviations_from_zone_axes,
5     plot_polarisation_vectors)
6 import matplotlib.pyplot as plt
7
8 # get an example atom lattice
9 atom_lattice = am.dummy_data.get_polarization_film_atom_lattice()
10 sublatticeA = atom_lattice.sublattice_list[0]
11 sublatticeA.find_nearest_neighbors()
12 sublatticeA.refine_atom_positions_using_center_of_mass()
13 sublatticeA.construct_zone_axes()
14
15 # Create some u, v components for visualisation
16 x, y, u, v = combine_atom_deviations_from_zone_axes(
17     sublatticeA, save=None)
18
19 # Basic plot of some atom movement
20 plot_polarisation_vectors(x, y, u, v, image=sublatticeA.image,
```

6.1. Tools for Visualisation of Subatomic Shifts

```
21     normalise=True, save=None, cmap='autumn',
22     plot_style='colormap', color='r',
23     overlay=False, title='Combined Deviations',
24     monitor_dpi=25)
25 plt.plot(x, y, 'k.', alpha=0.75)
26
27 # Restrict to the first four zones
28 x, y, u, v = combine_atom_deviations_from_zone_axes(
29     sublatticeA, save=None, axes=[0,1,2,3])
30
31 plot_polarisation_vectors(x, y, u, v, image=sublatticeA.image,
32     normalise=True, save=None, cmap='autumn',
33     plot_style='colormap', color='r',
34     overlay=False, title='Combined Deviations 0-3',
35     monitor_dpi=25)
36 plt.plot(x, y, 'k.', alpha=0.75)
```

Code Segment 6.5: Example script using the function

`combine_atom_deviations_from_zone_axes`. The outputs from this code is displayed in Figure 6.5

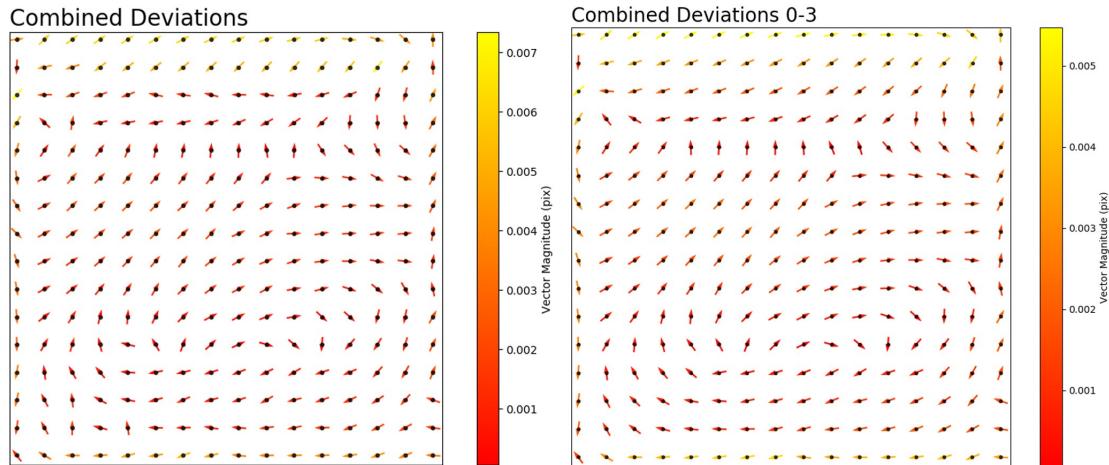


Figure 6.5: The combined atomic column deviations from (a) all zone axes and (b) zone axes 0, 1, 2, and 3 from Figure 6.4 above.

6.1.5 Get the Average Polarisation in n Unit Cells

The movement of atomic columns, regardless of how it is calculated (polarisation, atomic column deviation etc.), can be averaged over sub-regions of an image by the `get_average_polarisation_in_regions` and

`get_average_polarisation_in_regions_square` functions. Both functions take the x, y, u, v lists and the image as required parameters. The number of regions is defined by the `divide_into` parameter. It is set to 8 by default, meaning the x and y axis lengths will both be divided by 8. For

`get_average_polarisation_in_regions`, this will result in rectangular regions for rectangular images. To keep the averaged regions square in a rectangular image, the function `get_average_polarisation_in_regions_square` should be used (the functionality of both will be combined into one function in future versions).

Both versions of the averaging function can be used to see the general structure and possible polarisation in a structure. It is useful because overall trends can be difficult to discern from vectors of each atom in a large image. If the unit cell dimensions are known, the `divide_into` parameter can be adjusted to average over n unit cells. It has been used to full effect for DW analysis in ??(PbTiO₃ Vortexes) and in ??(Boracite).

```
1 # Example of the get_average_polarisation_in_regions function
2 import atomap.api as am
3 from temul.polarisation import (
4     combine_atom_deviations_from_zone_axes,
5     plot_polarisation_vectors, get_average_polarisation_in_regions)
6
7 atom_lattice = am.dummy_data.get_polarization_film_atom_lattice()
8 sublatticeA = atom_lattice.sublattice_list[0]
9 sublatticeA.find_nearest_neighbors()
10 sublatticeA.refine_atom_positions_using_center_of_mass()
11 sublatticeA.construct_zone_axes()
12 monitor_dpi=25
13
14 # Get and plot the original polarisation vectors:
15 x, y, u, v = combine_atom_deviations_from_zone_axes(
16     sublatticeA, save=None)
17 plot_polarisation_vectors(x, y, u, v, image=sublatticeA.image,
18                           normalise=False, save=None,
19                           plot_style='vector', color='r',
20                           overlay=False, title='Actual Vector Arrows',
21                           monitor_dpi=monitor_dpi)
22
```

6.1. Tools for Visualisation of Subatomic Shifts

```
23 # Get and plot the new, averaged polarisation vectors
24 x_new, y_new, u_new, v_new = get_average_polarisation_in_regions(
25     x, y, u, v, image=sublatticeA.image, divide_into=10)
26 plot_polarisation_vectors(x_new, y_new, u_new, v_new,
27     image=sublatticeA.image, normalise=False,
28     save=None, plot_style='vector', color='r',
29     overlay=False, title='Averaged Vector Arrows',
30     monitor_dpi=monitor_dpi)
```

Code Segment 6.6: Example script using the `get_average_polarisation_in_regions` function. The outputs from this code is displayed in Figure 6.6

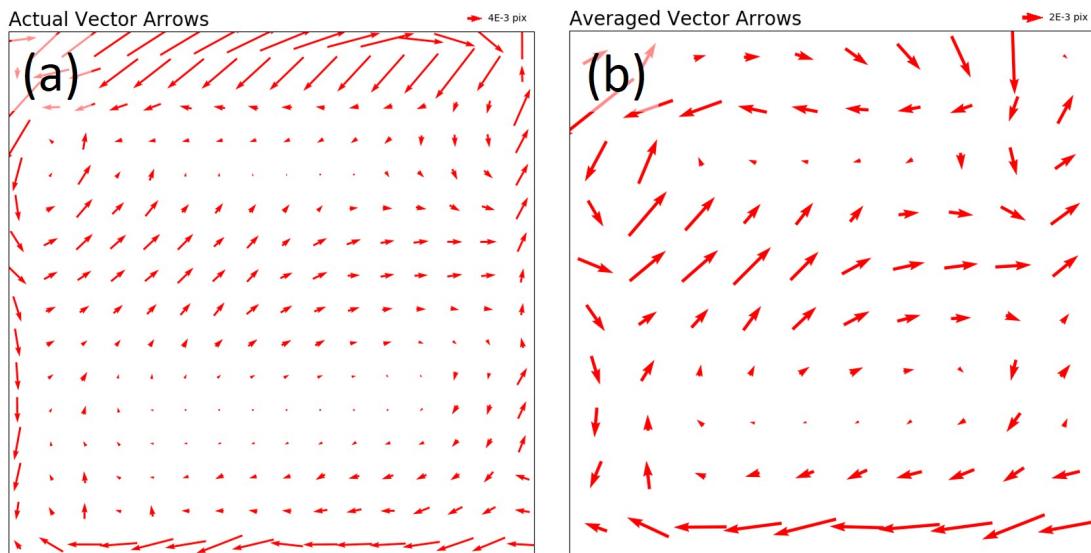


Figure 6.6: Visualisation of vector data in an atomic resolution image with each vector representing (a) an atomic column and (b) the average of several vectors in a region.

6.1.6 Delete Atom Planes from Sublattice Zone

For some images, the initial atom position finding routine in Atomap will detect too many atom positions. This could be due to the brightness of the atomic columns being similar for different atomic column element configurations. If the material's structure is known, this issue can be rectified by removing the atom planes on which the unwanted atoms lie. A new, reduced sublattice can then be created which contains only the desired atomic columns. This is best described graphically, as shown in Figure 6.7. In Figure 6.7(a), all atomic columns along the displayed atom planes (vertical lines) are included in the sublattice. After applying the `delete_atom_planes_from_sublattice` function in Code Segment 6.7(Line 12), every third atom plane, offset from atom plane 1, is deleted. The result is shown in Figure 6.7(b). The atoms along these planes can be used to create a new sublattice.

The main parameters of `delete_atom_planes_from_sublattice` are the `sublattice` from which the `zone_axis_index` atom planes will be deleted. Every nth atom plane to be deleted is set by `divisible_by`. The `offset_from_zero` sets where the deleting of atom planes begins. The `opposite` parameter will inverse the atom planes to be deleted. This function was utilised in the analysis of PbGeO3, which is not discussed in this manuscript.

```
1 # Example of the delete_atom_planes_from_sublattice function
2 import atomap.api as am
3 from temul.polarisation import delete_atom_planes_from_sublattice
4
5 atom_lattice = am.dummy_data.get_polarization_film_atom_lattice()
6 sublatticeA = atom_lattice.sublattice_list[0]
7 sublatticeA.construct_zone_axes()
8
9 # look at the zones and planes before deleting
10 sublatticeA.plot_planes()
11
12 delete_atom_planes_from_sublattice(
13     sublatticeA, zone_axis_index=0,
14     divisible_by=3, offset_from_zero=1)
15
16 # look at the zones and planes after deleting
17 sublatticeA.plot_planes()
```

Code Segment 6.7: Example script using the `delete_atom_planes_from_sublattice` function. The outputs from this code is displayed in Figure 6.7

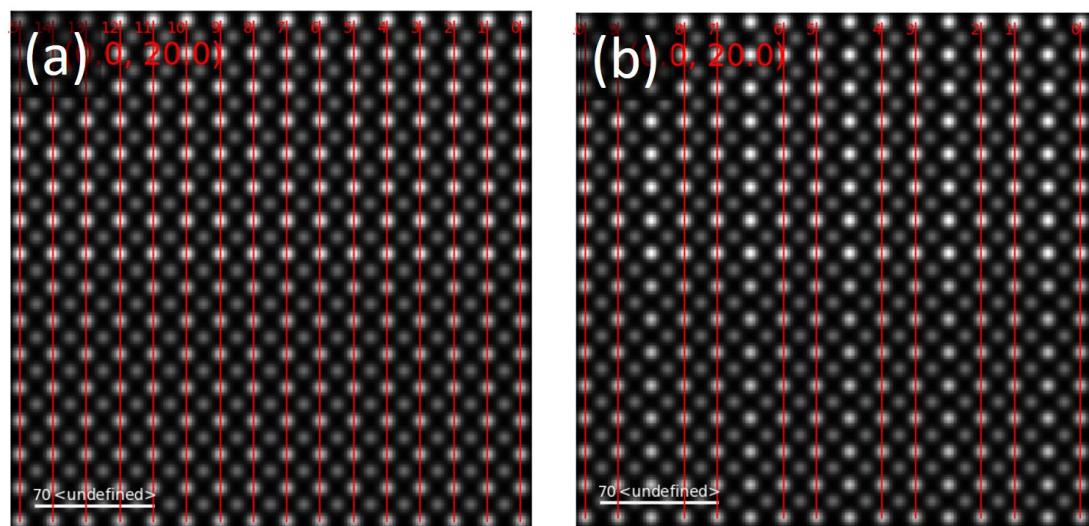


Figure 6.7: Atom planes of the 0 zone axis (vertical lines) of an example sublattice (a) before and (b) after atom plane deletion with the function `delete_atom_planes_from_sublattice`.

6.1.7 Material Property Maps: Strain, Angle of Zone and Lattice Spacing Ratio

The `polarisation.py` module developed during this thesis also visualises several material properties: strain (Figure 6.8), the angle of atom planes in a zone axis (Figure 6.9) and the ratio of the atomic lattice spacings in a zone axis (Figure 6.10). Code Segment 6.8 shows these functions in use and demonstrates the simplicity of the code. All three of the functions were used to understand the properties of domains near a DW confluence in [17].

The function `get_strain_map` calculates the distance between each atom and its neighbouring atom planes and plots the result, as well as returning the signal in the form of a HyperSpy Signal2D. The strain is defined in the methodology in Subsection 3.4.7(Strain). The zone axis is specified by its index with the `zone_axis_index` parameter. It will calculate the absolute strain if both the image `sampling` and `theoretical_value` are defined when calling the function. The `theoretical_value` is the value at which the specified zone axis' strain is zero. If this value is unknown, `theoretical_value` can be set instead to the mean strain calculated in the lattice. All other calculated strain values will be compared as a percentage error from this `theoretical_value`. Figure 6.8 displays a basic and exaggerated example of deformation in a sublattice. In the lower half of Figure 6.8(a), from left to right every second atom is offset in y by three pixels. Figure 6.8(b) highlights the zigzag pattern. In Figure 6.8(c), the strain in the lattice appear in the centre of the image, wherein the regular square unit cell meets the lattice distortion. With an example `theoretical_value` of 2 Å, the bright region in Figure 6.8(c) is calculated as $\sim 2\%$ above 2 Å. In subsection 6.2.1(PbTiO₃ Domain Wall Junctions), the absolute strain of zones near a confluence of DWs is calculated with `get_strain_map`.

The `rotation_of_atom_planes` function calculates the angle of rotation between each atom and its nearest neighbour relative to the horizontal x-axis in a specified zone axis. This function will be expanded in future versions to include better visualisation of angles. The zone axis is specified by its index with the `zone_axis_index` parameter. `angle_offset` may be used to shift all calculated angles by a certain value. This can be thought of as rotating the 0°horizontal axis from which the angles are calculated, or as setting a certain value as the zero angle value to which all others will be compared. This is useful for domain wall analysis, as shown in subsection 6.2.1(PbTiO₃ Domain Wall Junctions). `angle_type` chooses the units (degrees or radians) of the calculated angles, and is by default set to degrees. If the `angle_type='rad'`, then the `angle_offset` value, if specified,

6.1. Tools for Visualisation of Subatomic Shifts

should be set to radians also. Figure 6.9(b) shows the zone axis 2 of the sublattice in Figure 6.9(a). The zigzag pattern of the lower half of the sublattice atoms produces the alternating angle pattern between $\sim 40^\circ$ to $\sim 50^\circ$ in Figure 6.9(c).

The ratio between two different lattice spacings (STEM projection of the atomic structure) is calculated and visualised with the `ratio_of_lattice_spacings` function. It can be used to calculate tetragonality and shearing of unit cells. Both zone axes are specified by its index with the `zone_axis_index_A` and `zone_axis_index_B` parameters. First, the `zone_axis_index_A` is calculated and plotted, followed by `zone_axis_index_B`, as shown for zones 0 and 1 in Figure 6.10(a-b) and Figure 6.10(c-d), respectively. Lattice spacing A is divided by lattice spacing B to find their ratio. This is visualised for zone 0/zone 1 in Figure 6.10(e). If `ideal_ratio_one` is set to True, the calculated ratio values below one will be inversed, setting one as the minimum value. This is useful if one is interested in viewing the shape of the lattice spacing ratio rather than the absolute values, as shown for the c/a lattice ratio in subsection 6.2.1(PbTiO₃ Domain Wall Junctions).

All three of the above-described functions have the following common parameters: sampling, units, `vmin`, `vmax`, `cmap`, `title`, `filename`, and `**kwargs`. `vmin` and `vmax` set the minimum and maximum intensity value of the plot. `cmap` sets the Matplotlib colour-map. The `title` parameter sets the main part of the image heading, and if `filename` is set to a name (string), the .png image and .hspy signal will be saved. The `**kwargs` parameter allows the user to specify plotting keyword arguments from Matplotlib's `imshow` function. For further functionality, the HyperSpy Signal2D is output and can be plotted with various extra parameters as defined by Hyperspy's `plot` function.

```
1 # Example of the get_strain_map function
2 from temul.dummy_data import
3     get_distorted_cubic_sublattice_adjustable
4 from temul.polarisation import (get_strain_map,
5                                 rotation_of_atom_planes,
6                                 ratio_of_lattice_spacings)
7
8 sublatticeA = get_distorted_cubic_sublattice_adjustable(
9     y_offset=3)
10 sublatticeA.construct_zone_axes(atom_plane_tolerance=1)
11 sublatticeA.plot()
12 sublatticeA.plot_planes()
13
14 # get the strain map
15 strain_map_signal = get_strain_map(
16     sublatticeA, zone_axis_index=0, theoretical_value=2,
```

6.1. Tools for Visualisation of Subatomic Shifts

```

16     sampling=0.1, units='Å', title='Strain Map')
17
18
19 # get the rotation map
20 rotation_signal = rotation_of_atom_planes(
21     sublatticeA, zone_axis_index=2, angle_offset=None,
22     angle_type='deg', vmin=35, vmax=55, title='Rotation Map')
23
24
25 # get the ratio of spacings maps
26 ratio_signal = ratio_of_lattice_spacings(sublatticeA, 0, 1,
27     ideal_ratio_one=True, vmin=None, vmax=None,
28     title='Spacings Map')

```

Code Segment 6.8: Example script using
the `get_strain_map`, `rotation_of_atom_planes` and `ratio_of_lattice_spacings` functions. The outputs from these functions are displayed in Figure 6.8, Figure 6.9 and Figure 6.10, respectively.

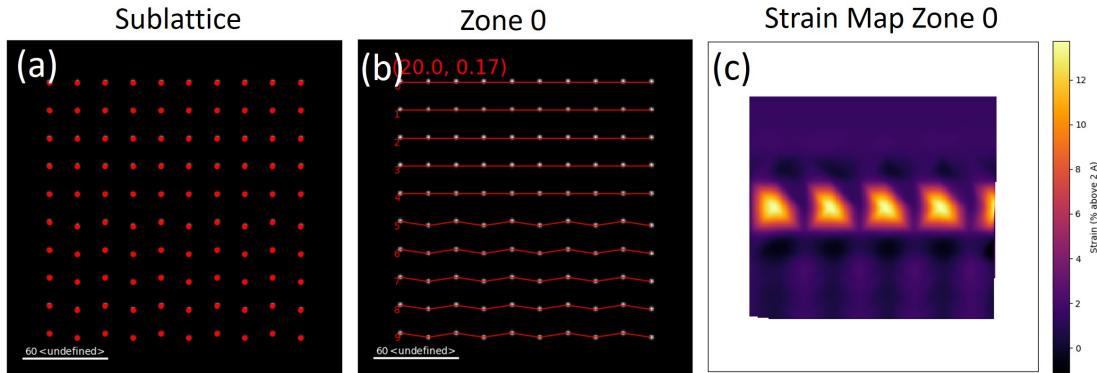


Figure 6.8: (a) Example sublattice with atom y coordinate offsets in the lower half of the image. (b) First zone axis of the sublattice in (a) showing the zig-zag offset pattern. (c) Strain map of the first zone axis, calculated with the `get_strain_map` function.

6.1. Tools for Visualisation of Subatomic Shifts

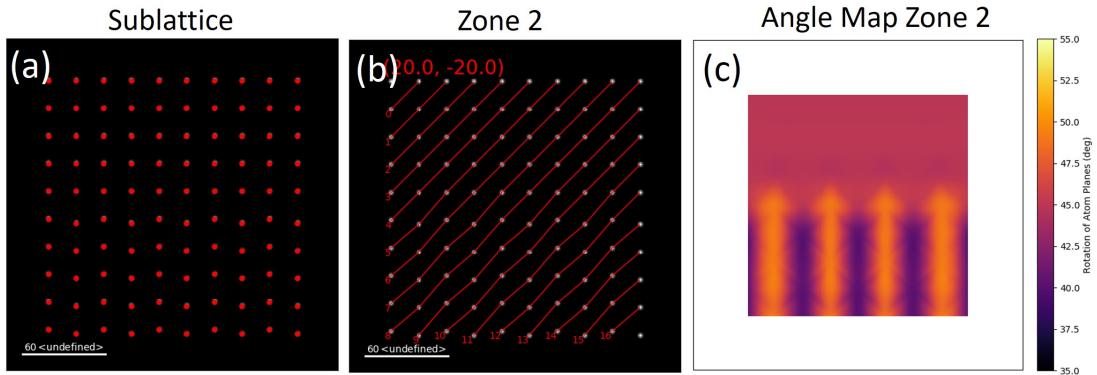


Figure 6.9: (a) Example sublattice with atom y coordinate offsets in the lower half of the image. (b) Third zone axis of the sublattice in (a) showing the zig-zag offset pattern. (c) Angle map of the atom planes in the third zone axis, calculated with the `rotation_of_atom_planes` function.

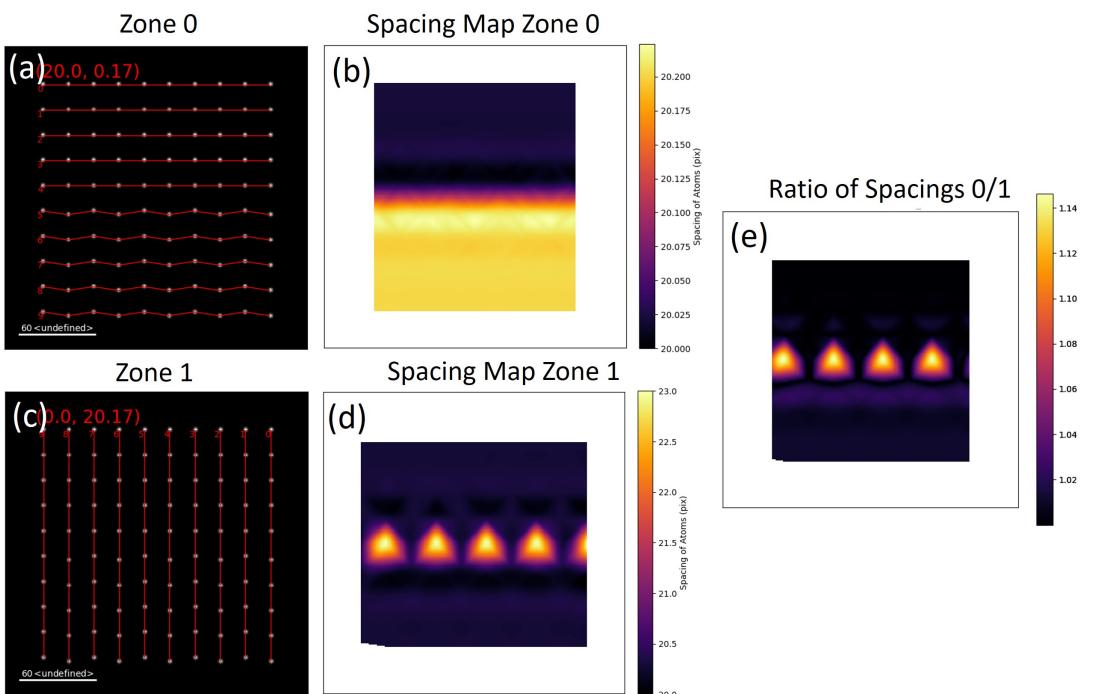


Figure 6.10: (a) First and (c) second zone axis of the sublattice in Figure 6.9(a). Lattice spacing map of the atom spacings along the atom planes in the (b) first and (d) second zone axis, calculated with the function `ratio_of_lattice_spacings`. (e) Ratio map of the lattice spacing maps from (b) and (d).

6.2 2D Domain Walls in Ferroelectric Materials

Certain parts of this section are **redacted** until published. As a result, certain hyperlinks in the thesis will not be displayed correctly.

In the following section we will apply the above-described tools in Subsection 6.1 (Tools for Visualisation of Subatomic Shifts) to analyse domain wall (DW) structures in ferroelectric materials. Both DWs and ferroelectric materials are described in Subsection 2.3(Ferroelectric Materials and Domain Walls), and reviewed well by Catalan *et al* [420], while both atom movement and polarisation are defined in the methodology in Subsection 3.4.7(Polarisation & Atomic Movement). The atom movement is mapped for each material rather than the polarisation, unless otherwise stated. Although, as we will describe, the analyses carried out in this chapter have helped answer complex material science problems, it is worthwhile noting the main aim of the chapter: we wish to highlight the ability and application of the TEMUL toolkit's `polarisation.py` module which is available open-source [13].

6.2.1 PbTiO₃ Domain Wall Junctions

The `polarisation.py` functions were used to analyse the interesting material properties of domain wall (DW) junctions in single crystal lead titanate (PbTiO₃) [17]. A DW junction is a confluence of several DWs. The charged needle point (CNP) domains found in PbTiO₃ (PTO) are of particular interest from a data analysis viewpoint because they are highly strained without dislocations. Coupled with the Atomap Python package, the `polarisation.py` functions described in Subsection 6.1 were utilised to verify the head-to-head (H-H) polarisation of the Ti lattice and quantify Pb vacancies at the PTO junction. The functions were also used to quantify the strain, atom plane rotation and c/a lattice spacing ratio in the regions at and near the DW junction.

The multiple domain structures in the studied PTO junction can be seen in Figure 6.11(a). The coordinates of both the Pb and Ti sublattices were located and refined with Atomap [see subsection 3.4.3(Methods for Atom Position Analysis)]. For the Pb (bright) sublattice, some atom locations needed to be added manually due to the strained structure near the PTO junction, marked in Figure 6.11(a). This in turn creates difficulties for finding the Ti atoms via the zone axes of the Pb sublattice. Though creating a function to handle dislocations is a likely solution to this issue, manually adding and removing atoms with Atomap is sometimes necessary for complex or not-perfect structures.

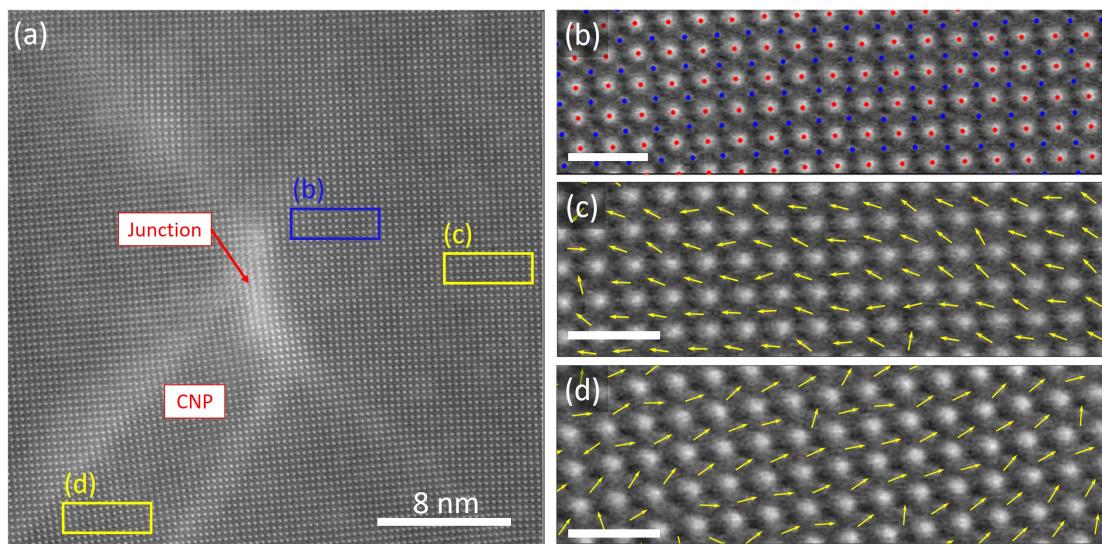


Figure 6.11: (a) Atomic resolution ADF STEM image of a domain wall junction in PTO. The charged needle point (CNP) domain causes the area of high strain i.e., the PTO junction. The rectangular regions inset are magnified in (b), (c) and (d). (b) Sub-Ångström position refinement with Atomap of the bright Pb (red) atoms and Ti (blue) atoms. (c, d) Polarisation mapping of the H-H domain wall with the `polarisation.py` module of the TEMUL toolkit. Scale bars (b-d) are 1 nm.

After position refinement of the Ti atoms with the Atomap Gaussian fitting algorithm, we calculate the difference from this point to their initial ideal position. These ideal positions are defined by their crystal structure as the centre point of the nearest four Pb atoms. Figure 6.11(b) marks the bright Pb sublattice as red dots and the Ti sublattice as blue dots. The PTO junction caused difficulties when mapping the polarisation of the Ti atoms relative to the Pb sublattice for zone axis 1 due to the large atom plane shifts. Therefore, Atomap's easy-to-use `get_polarization_from_second_sublattice` function could not be applied in this case. Instead, the ideal and actual Ti sublattice xy coordinate lists were input to the `find_polarisation_vectors` function. The polarisation of two domains are plotted in Figure 6.11(c, d). In the case of PTO, the polarisation is simply the opposite vector of the atomic shift [17, 336]. Therefore, the polarisation mapping of area (c) and (d) inset in Figure 6.11 verifies that the PTO junction is a H-H DW. Figure A.14 in subsection A.5.1 shows the entire atom lattice and Ti polarisation map.

The function `get_strain_map` quantifies the strain of the first zone axis (index 0) at the PTO junction in Figure 6.12(a), as described in subsection 6.1.7(Material Property Maps: Strain, Angle of Zone and Lattice Spacing Ratio). The strain is defined in the methodology in Subsection 3.4.7(Strain). For this map, the `theoretical_value` of the first zone's atom plane spacing is 3.929 Å[17]. In Figure 6.12(a) -15% and +32% are the minimum (vmin) and maximum (vmax) intensity limits, respectively. From the bottom left CNP domain through the PTO junction in the centre of the image, the strain spreads out to the large domain on the right side. Away from the centre of the PTO junction, near the bottom right, the right side domain is consistently at ~5% strain, similar to that in the CNP domain. The enormous strain at the PTO junction (31%) results in a small area with a lattice spacing (atom plane separation) of 5.147 Å. In the darker, purple domains in Figure 6.12(a), the lattice spacing is reduced to 3.693 Å on average.

Figure 6.13(a-b) shows the same strain map but with two line profiles across different domains and DWs. The line profile positions are displayed in Figure A.15 in subsection A.5.1(PTO Junction). The line profile positions were found via the `add_atoms_with_gui` Atomap function, which has been expanded upon during this thesis [405]. The `compare_images_line_profile_one_image` function in the `signal_plotting.py` of the TEMUL toolkit visualises the line trace region on the strain map in Figure 6.13(b) and the resulting line strain profile in Figure 6.13(a). In the top (solid red line) profile, the mean strain in the CNP domain is at ~5%, as mentioned above. At the PTO junction, the mean strain increases to ~17%. The x-axis nm width of the high strain values does not reflect the actual thickness

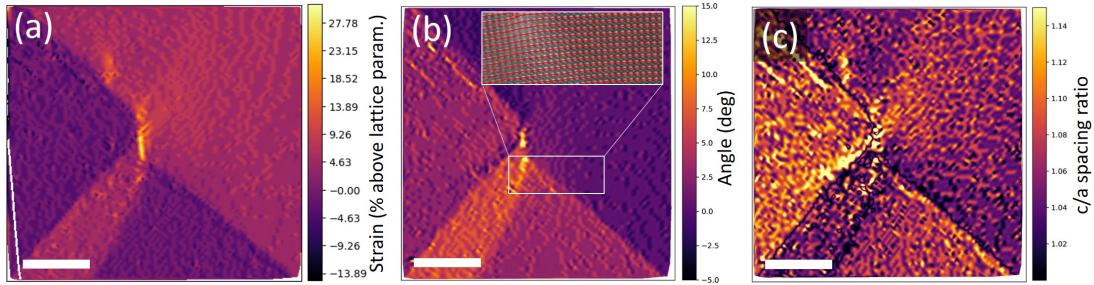


Figure 6.12: (a) Quantified strain map, (b) atom plane angle map, and (c) lattice spacing c/a ratio map of the PTO junction from Figure 6.11(a) using the `polarisation.py` module of the TEMUL toolkit. Inset in (b) is the zone axis 1 direction showing bending across DWs. Scale bars are 8 nm.

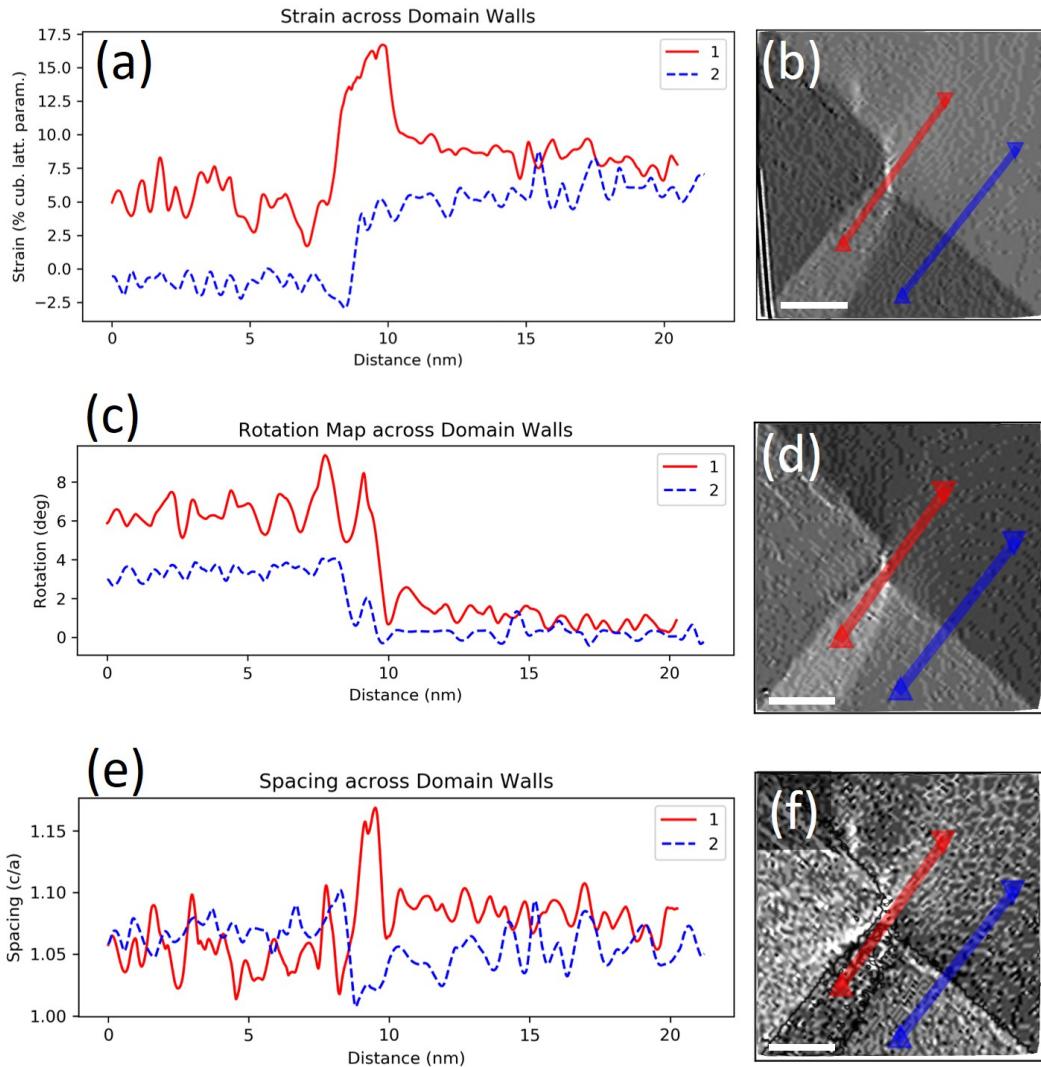


Figure 6.13: Line profiles (a, c, e) from their respective (b) quantified strain map, (d) atom plane angle map and (d) lattice spacing c/a ratio map across the H-H domain wall. The line profiles and maps were visualised with the `polarisation.py` and `signal_plotting.py` modules of the TEMUL toolkit, respectively. The chosen line profile positions are shown in the appendix in Figure A.15. Scale bars are 8 nm.

of the DW in the centre of the PTO junction because the line profile is not taken perpendicular to it. This line trace continues into the right side domain and gradually decreases in mean strain. This decrease is gradual because the strain from the PTO junction extends into the right side domain. The lower (dashed blue line) line profile begins in the low strain domain at the bottom of the image and continues into the right side domain. The jump in strain in Figure 6.13(a) across this DW is a clear example of how useful these comparisons are for understanding the material properties in ferroelectrics.

The `rotation_of_atom_planes` function quantifies the angle an atom plane of a zone axis has with the horizontal axis. More specifically, it calculates the angle between each atom and its consecutive atom along the atom plane, for all atom planes in the zone axis, as described in subsection 6.1.7(Material Property Maps). Figure 6.12(b) shows the second zone axis (index 1) angle in degrees from the horizontal axis with an `angle_offset=-2`. This `angle_offset` essentially sets the average angle in the large right side domain to zero for comparison purposes. The horizontal atom plane direction of zone axis 1 is shown inset in Figure 6.12(b). It shows a clear bend in the atom plane lines from left to right across DWs.

Figure 6.13(c-d) show the large decrease ($\sim 6^\circ\text{-}6.5^\circ$) in angle from the CNP domain (solid red line) and the bottom domain (dashed blue line) to the right side domain. Normally, the solid red line profile from the CNP to right side domain would have an angle mismatch of 0° . In this material, however, there are other 90° rotated CNPs in the right side domain (not shown), which cause an offset of $\sim 6^\circ$ in the right side domain, rather than the normal 0° . More detail on this phenomena can be found in Moore *et al* [17]. Easily mapping the angle of the atom planes relative to each other in this material therefore allows one to understand the overall lattice bending in and near PTO DW junctions.

Figure 6.12(c) shows the c/a lattice spacings ratio and therefore displays the tetragonality of the lattice in and around the PTO junction. The function `ratio_of_lattice_spacings` calculates the spacings by first finding the lattice spacing between each atom in each atom plane for two specified zone axes. Here, the index 0 (vertical) and index 1 (horizontal) zone axes were used. It then divides the first zone axis lattice spacing by the second. This is fully described in subsection 6.1.7(Material Property Maps). The `ideal_ratio_one` parameter was set to True in this case, making the lowest intensity 1 for the map in Figure 6.12(c). This makes all other non-square ratios larger than 1. In each domain the unit cell, which should be a square unit cell, is elongated in either the *c* or *a* direction. Where two domains meet, these elongated unit cells become square unit cells (strain equal in all directions). Therefore, the c/a ratio of 1 - the darkest regions in the image -

6.2. 2D Domain Walls in Ferroelectric Materials

locate the DWs themselves. This function allows one to directly locate the centre line of each DW in a structure. The line profiles in Figure 6.13(e) are taken over the line traces in Figure 6.13(f). At ~ 8.5 nm along the bottom line trace (dashed blue line), the line profile goes to a ratio of 1 as it crosses the DW.

In DW materials, the `get_strain_map`, `rotation_of_atom_planes` and `ratio_of_lattice_spacings` functions can visualise material properties quantitatively and quickly. Coupled with the functions in the `signal_plotting.py` module, the `polarisation.py` functions provide in-depth means for atomic resolution image analysis. Of course, the better the imaging and data, the better the atomic coordinate finding and refining can be with Atomap.

The `polarisation.py` module of the TEMUL toolkit can be expanded considerably. The line profiles shown above in Figure 6.13 can be traced over any signal, and so any other developed maps could be analysed in the same manner. Another addition to the toolkit for the `polarisation.py` module would be a function which returns statistics on a specified region of the image. For example, one could specify a domain in a strain map and the function would return the maximum, minimum, average etc., statistics of the strain from the domain.

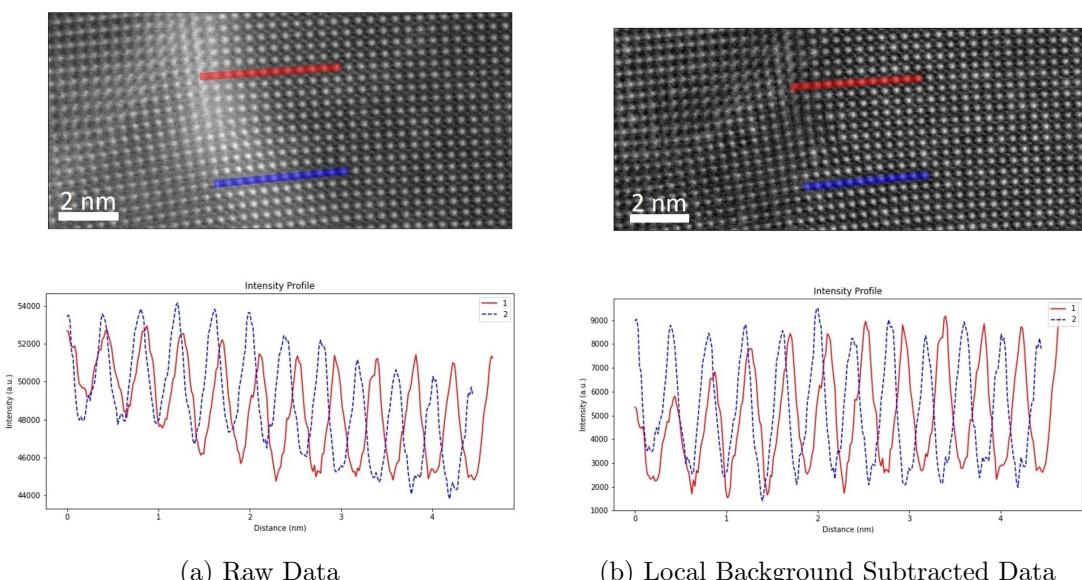


Figure 6.14: Line intensity profiles along two parts of the (a) raw and (b) local background subtracted images from the central PTO DW junction in Figure 6.11(a). The top (solid red) and bottom (dashed blue) line profiles trace over the junction and the right side domain, respectively.

In the H-H DW between the CNP domain and the large right side domain, it was hypothesised that the depolarising electric field traps negative carriers. To verify this, the `get_cell_image` function in the `signal_processing.py` module was used to remove the local background closest to each Pb atom in the image

in Figure 6.11(a) above. By setting the `reduce_func=np.min`, the minimum background intensity in the region around each atom is subtracted from each pixel in that region, as described in Subsection 3.4.3(Background Intensity). The high contrast at the PTO junction is not due to more atoms in the atomic columns. It is due to the high strain between the CNP and right side domains. This de-channels the probe, causing additional scattering and brightness at the junction, as described by Moore *et al* [17].

To visualise the difference before and after local background subtraction the `compare_images_line_profile_one_image` function in the `signal_plotting.py` module was used. Figure 6.14 shows the centre of the PTO junction before and after background subtraction. The line profile comparison between the centre of the junction (solid red line) and below the junction (dashed blue line) in Figure 6.14(b) is much clearer than in Figure 6.14(a). It shows a lower intensity over the atomic columns in the solid red line in Figure 6.14(b), which indicates vacancies in the material near the PTO junction. The $\sim 45\%$ Pb vacancies (V_{Pb}^{-2}) create a negative charge build up as described by Moore *et al* [17]. Therefore, the tools developed in the TEMUL toolkit also verified that the depolarising electric field traps negative carriers in the PTO junction.

6.3 Chapter Conclusions - Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls

In the first part of this chapter, we developed several accessible functions via open-source Python code for the analysis of atomic columns. The functions in Subsection 6.1(Tools for Visualisation of Subatomic Shifts) are built on the data structure of widely used Python packages (Hyperspy, Atomap, Matplotlib) and therefore should be useful for other researchers.

In the second part of this chapter, polarisation in domains and across domain walls (DWs) in different ferroelectric materials was investigated using the functions developed in Subsection 6.1(Tools for Visualisation of Subatomic Shifts).

Redacted

Chapter 7

Conclusions & Future Directions

7.1 Overall Conclusions

Overall, we have developed and implemented open-source methods to analyse atomic resolution annular dark field (ADF) scanning transmission electron (STEM) images. These methods can be found in the TEMUL toolkit on GitHub [13]. Implementation of known analysis methods were described in Subsection 3.4(Signal and Data Analysis). In Chapter 4(Atomic Characterisation of Ion Implanted 2D TMDs), these methods were utilised for the analysis of single Se atoms introduced into monolayer MoS₂ via low energy ion implantation. In Chapter 5(Atom Dynamics in 2D TMDs), by utilising fast image simulation software, methods were designed for automated iterative analysis of atomic resolution ADF STEM movies. By applying these methods to Se implanted MoS₂ monolayers, lattice Mo vacancies were quantified. The effect of dose rate, total cumulative dose, and time on the creation of such vacancies was investigated, and an exponential relationship was found [see subsection 5.2.3(Effect of Electron Beam on TMD Monolayers)]. Additionally, we hypothesise a detrimental threshold dose rate, above which the monolayer MoS₂ lattice is damaged irreversibly. Further methods for the analysis of polarisation and movement of atomic columns were developed in Chapter 6(Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls). These methods were successfully used to analyse polarisation of domain walls (DWs) and domains in ferroelectric materials.

Below we reiterate the conclusions stated in their respective chapters.

In Chapter 4(Atomic Characterisation of Ion Implanted 2D TMDs), low energy ion implantation into 2D TMDs has been shown for both Se (substitutional) and Au (nanoparticle formation). ADF STEM, image simulation and EELS were employed to verify the effects of ion implantation on the surface of monolayer and few-layer

7.1. Overall Conclusions

MoS_2 and MoSe_2 . The varied results from 10 eV Se and Au ion implantation show how vital it is to investigate the atomic structure of such implanted materials, but also the wide range of possibilities low energy ion implantation can provide.

Se ion implantation has been verified to create substitutional anti-sites and vacancies in monolayer MoS_2 , as was discussed in detail in subsection 4.1.1(Microscopy of Se implanted MoS_2). Image simulation verified the Z-contrast intensity arising from single substitutional Se atoms and vacancies. The energy required to create anti-sites and vacancies was calculated via DFT [work completed by collaborators, see Figure 5.13 in subsection 5.2.2(Analysis of Lattice Energetics)]. Single atom chemical analysis via core-loss EELS, coupled with image simulation, confirmed successful Se implantation in Subsection 4.1.2. In Subsection 4.1.2, low-loss EELS showed the change in band gap and exciton energies in monolayer and few-layer MoS_2 due to Se implantation. For the $\text{MoS}_{1.87}\text{Se}_{0.13}$ alloy created (5.56% Se doping), the band gap changed by 0.09 eV. Overall, Se ion implantation into a monolayer TMD has been successfully demonstrated. It was used to manipulate the band structure of MoS_2 , similar to other doping methods, such as CVT [50] and CVD [78].

Au ion implantation at low energies provides a method for depositing Au nanoparticles (NPs) on the surface of monolayer and few-layer material without causing lattice damage, as verified by ADF STEM in Subsection 4.2. It is expected by increasing the total ion dose (fluence), one can increase the size of the Au NPs, and vice versa. The most exciting possible use of Au implantation on 2D TMDs is for truly nano-scale contacts for various electronic devices. The Au NPs were confirmed to be conductive on the TMD monolayer surface via DFT calculations. By controlling the ion beam, one can envisage patterning of conductive, ultra-thin 1-5 nm Au contacts on the monolayer and few-layer material surface.

In Chapter 5(Atom Dynamics in 2D TMDs), open-source methods for the automatic analysis of Se implanted MoS_2 were developed. In subsection 5.1.1, two algorithms were designed based on iterative and automatic experimental comparison with scanning transmission electron microscopy (STEM) image simulations. Following this, we established a multi-functional Python class called the Model Refiner (MR) in subsection 5.1.2. The MR tool automatically simulates and calibrates data, and tracks all changes made by the above-mentioned algorithms between experiment and simulation. It also is useful for completing and plotting element quantification. The critical aspect of the developed algorithms and MR tool is their simple integration with the popular HyperSpy and Atomap Python packages. Additionally, the methods can be applied to analyse many different

materials and images.

In Subsection 5.2(Atom Dynamics in Implanted TMDs with the MR Tool), the MR tool was employed for the automated analysis of atomic resolution STEM movies of Se implanted MoS₂ monolayers. After verifying the approach to specific steps for the analysis of the movies in subsection 5.2.1(Image Series Automated Analysis Steps), the atom lattice was analysed over time. The energetics of atom movement, adatoms, vacancies and holes were analysed in subsection 5.2.2(Analysis of Lattice Energetics). Following this, subsection 5.2.3(Effect of Electron Beam) investigated the effect of the electron beam on the lattice as a function of electron beam dose rate for specified time-frames and total doses. By quantifying the elements, an exponential relationship between the Mo lattice vacancies and electron beam dose rate was fitted. Additionally, we hypothesise a detrimental threshold dose rate (TDR) of $\sim 4.25 \text{ pC nm}^{-2} \text{ s}^{-1}$, above which the monolayer MoS₂ lattice is damaged irreversibly. Finally, Subsection 5.2.3(Effect of Contamination on Movie Analysis) discussed and quantified the stabilising effect of surface contamination on the monolayer lattice.

In Chapter 6(Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls), we developed several accessible functions via open-source Python code for the analysis of atomic columns. The functions in Subsection 6.1(Tools for Visualisation of Subatomic Shifts) are built on the data structure of widely used Python packages (Hyperspy, Atomap, Matplotlib) and therefore should be useful for other researchers. These tools were then used to investigate polarisation in domains and across domain walls (DWs) in different ferroelectric materials. **Redacted**

7.2 Future Directions

Successful low energy ion implantation of two-dimensional (2D) transition metal dichalcogenides (TMDs) was verified during this project [10]. Recently, other groups have utilised ion implantation of WS₂ for the creation of Janus monolayers [96]. Because such novel structures can be created with the technique, we expect this field to grow in the coming years. Additionally, low energy implantation of large atomic number metals such as Au is likely to create similar nanoparticle (NP) structures non-destructively, as demonstrated in Chapter 4(Atomic Characterisation of Ion Implanted 2D TMDs). Because of the top-down nature of ion implantation, ion-beam lithography may allow for nano-fabrication of atomically bonded contact design on monolayer TMDs. A preceding doping method, such as the Se implantation in this work, would allow two-step device creation in the

implanter to become a reality.

Reliable simulation of hydrocarbon contamination is essential for the analysis of 2D materials, especially those with low Z atoms, as discussed in subsection 4.1.1(Microscopy of Se implanted MoS₂). Phase contrast imaging can be used to detect regions of contamination clearly, and could be combined with annular dark field (ADF) scanning transmission electron microscopy (STEM) imaging. By averaging many density functional theory (DFT) calculations of the positions and orientation of hydrocarbons on the 2D surface, it would be possible to quantify the surface contamination. This would lead to a standard of including such hydrocarbons in image simulations for better quantification.

For image simulation, a Laplacian and a Laplacian + Gaussian filter could be added to the `compare_two_image_and_create_filtered_image` function of the `signal_processing.py` module. These could also be added for general image filtering functions. For the automated `double_gaussian_fft_filter` function, a function that detects the main diffraction spots of the input material's image would be quite useful. Complementary to this, a method for linking to a database of these values in the function would be convenient.

In this work, automated open-source analysis routines were designed for atomic resolution images. With the recent development of popular Python packages, analysis methods have been opened to many researchers, new and experienced. We expect Hyperspy, Atomap, and similar packages to become staple parts of the electron microscopy community in the coming years. The work carried out in this thesis will be implemented into these packages as far as reasonably possible. Some of the developed methods have already been added to Atomap [405] and the extensive Python image analysis library Scikit-Image [394]. We will now discuss some improvements that could be made to the analysis tools developed and utilised during this work.

The `polarisation.py` module of the TEMUL toolkit is accessible for beginners and advanced users alike. We expect its flexibility to be useful for other researchers. In future, using the developed tools on experimental data in real time would further expand their usefulness and solidify their role in understanding complex material features. Image analysis performed live on atomic resolution data coming from the microscope might inform the researcher of chemical composition, interesting regions, sample thickness, and damage to the sample depending on microscope parameters, such as dose or dose rate. For ferroelectric materials, one can envisage live domain mapping at both low magnifications [346] and atomic resolution, speeding up the rate at which data can be acquired and decreasing the number of repeated experiments.

7.2. Future Directions

There are many possible avenues for further development of the Model Refiner class (MR tool), which was used in this thesis to quantify and track atoms automatically. Comparison of experimental data with image simulation based on an absolute scale relative to the intensity of the incident electron beam should be, and can easily be, implemented. This addition is vital for quantification of 3D crystals lacking a consistent internal reference, such as the single Mo atoms in monolayer MoS₂. A full atom/adatom tracker would allow for advanced insight into material properties and would complement similar DFT studies. The Occam's razor approach discussed in subsection 5.2.2(Analysis of Lattice Energetics, Ad-atoms and Vacancies with the MR Tool) could be used, wherein specific atom movement, stability, and vacancy formation would be defined and ranked based on probability, informed by DFT calculations. The atom movement tracking would automatically output maps of each atom's location and the overall movement per atomic species, and therefore provide vital information on the dynamics of single atoms and atomic columns. The likelihood of the element configurations detected by the program (where the atoms sit in the column) would be taken from DFT [166] or by advanced imaging techniques [8].

The MR tool, and indeed the entire TEMUL toolkit, is not limited to the analysis of 2D materials, and was tested on atomic resolution images of nanoparticles and FIB-lamellas. Understanding the limit of element quantification is possible with the MR tool, as it is integrated with PyPrismatic. By varying image simulation and sample parameters, such as aberrations and sample tilt, forward modelling processes could be incorporated into the MR tool to truly find the best match image simulation for experimental data. This process would be easily accessible within the simulation and refinement functions as parameter ranges and increments. For example, the sample tilt could be given as a range from -5° to +5° with 1° increments to investigate the effect of tilt on the resulting comparison between image simulation and experiment. An option would be to have this function input parameter as a tuple, e.g., (-5, +5, 1). It is possible to simulate ranges of inputs of many parameters due to the fast image simulation software provided in PyPrismatic. For ferroelectric materials, such as those discussed in Chapter 6(Subatomic Analysis of Atom Movement & Polarisation in 2D Domain Walls), understanding the effect of sample parameters, especially crystal tilt, on the mapping of atomic movement and polarisation is vital. By quantifying the elements in each atomic column in experimental data, such as completed for 2D TMDs in this thesis or for other materials [120], a 3D atomic model could be created. This model would be refined using the MR tool in a forward modelling manner as above-described. In the case of ferroelectric materials, various image and sample parameters (e.g.,

7.2. Future Directions

crystal tilt) should be varied to understand the effect of each on the measurement of sub-Ångström atomic shifts, strain and therefore macroscopic polarisation.

Finally, the Model Refiner could be further streamlined with the Atomic Simulation Environment (ASE) Python package for automated molecular dynamics (MD) relaxation of created structures, as well as in-depth DFT to reveal novel material properties, all within the same Python terminal. This process would feed directly into online repositories, such as the Materials Data Bank (www.materialsdatabank.org). There, the quantified models could be accessed by researchers and citizen scientists alike, increasing the usefulness of such quantification and providing a platform for further understanding of material properties.

References

- [1] Z Wang and A Bovik. “Mean Squared Error: Love It or Leave It?” In: *IEEE Signal Processing Magazine* 98 (2009). DOI: 10.1109/MSP.2008.930649. URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4775883>.
- [2] Francisco de la Peña. “Advanced methods for Electron Energy Loss Spectroscopy core-loss analysis”. In: PhD thesis (2010), p. 181. DOI: 10.1186/1752-0509-4-158. URL: <https://www.stem.lps.u-psud.fr/sites/default/files/F%20de%20201a%20Pena%20Thesis.pdf>.
- [3] Francisco de la Peña et al. “hyperspy/hyperspy: v1.4.2”. In: (June 2019). DOI: 10.5281/ZENODO.3249885. URL: <https://doi.org/10.5281/zenodo.3249885#XSSTI83yN51.mendeley>.
- [4] Magnus Nord et al. “Atomap: a new software tool for the automated analysis of atomic resolution images using two-dimensional Gaussian fitting”. In: *Advanced Structural and Chemical Imaging* 3.1 (2017), p. 9. DOI: 10.1186/s40679-017-0042-5. URL: <http://ascimaging.springeropen.com/articles/10.1186/s40679-017-0042-5>.
- [5] Ondrej L. Krivanek et al. “Gentle STEM: ADF imaging and EELS at low primary energies”. In: *Ultramicroscopy* 110.8 (2010), pp. 935–945. DOI: 10.1016/j.ultramic.2010.02.007.
- [6] Recep Zan et al. “Metal - Graphene Interaction Studied via Atomic Resolution Scanning”. In: *Nano letters* 11 (2011), pp. 1087–1092. DOI: dx.doi.org/10.1021/nl103980h.
- [7] F. S. Hage et al. “Single-atom vibrational spectroscopy in the scanning transmission electron microscope”. In: *Science* 367.6482 (2020), pp. 1124–1127. DOI: 10.1126/science.aaz5312.
- [8] Xuezeng Tian et al. “Correlating the three-dimensional atomic defects and electronic properties of two-dimensional transition metal dichalcogenides”. In: *Nature Materials* (2020). DOI: 10.1038/s41563-020-0636-5. URL: <http://dx.doi.org/10.1038/s41563-020-0636-5>.

REFERENCES

- [9] EN O'Connell et al. "Utilizing an Open-Source Workflow for the Analysis of Atom Dynamics in Two-Dimensional Materials". In: *Microscopy and Microanalysis* 25.S2 (Aug. 2019), pp. 136–137. DOI: 10.1017/S1431927619001417. URL: https://www.cambridge.org/core/product/identifier/S1431927619001417/type/journal_article.
- [10] U. Bangert et al. "Ion-beam modification of 2-D materials - single implant atom analysis via annular dark-field electron microscopy". In: *Ultramicroscopy* 176.August 2016 (2017), pp. 31–36. DOI: 10.1016/j.ultramic.2016.12.011.
- [11] Colin Ophus. "A fast image simulation algorithm for scanning transmission electron microscopy". In: *Advanced Structural and Chemical Imaging* 3.1 (Dec. 2017), p. 13. DOI: 10.1186/s40679-017-0046-1. URL: <http://ascimaging.springeropen.com/articles/10.1186/s40679-017-0046-1>.
- [12] Alan Pryor, Colin Ophus, and Jianwei Miao. "A streaming multi-GPU implementation of image simulation algorithms for scanning transmission electron microscopy". In: *Advanced Structural and Chemical Imaging* 3.1 (Dec. 2017), p. 15. DOI: 10.1186/s40679-017-0048-z. URL: <http://ascimaging.springeropen.com/articles/10.1186/s40679-017-0048-z>.
- [13] Eoghan Noel O'Connell, Michael Hennessy, and Eoin Moynihan. "TEMUL Toolkit: v0.1-alpha". In: *Zenodo* (2020). DOI: 10.5281/zenodo.3832143. URL: github.com/PinkShnack/TEMUL.
- [14] M. Conroy et al. "Investigating Ferroelectric Domain and Domain Wall Dynamics at Atomic Resolution by TEM/STEM in situ Heating and Biasing". In: *Microscopy and Microanalysis* 25.S2 (2019), pp. 1882–1883. DOI: 10.1017/s1431927619010146.
- [15] M. Conroy et al. "Atomic-Scale Characterization of Ferro-Electric Domains in Lithium Niobate-revealing the Electronic Properties of Domain Walls". In: *Microscopy and Microanalysis* 25.S2 (2019), pp. 576–577. DOI: 10.1017/s1431927619003611.
- [16] M Conroy et al. "Probing the Dynamics of Topologically Protected Charged Ferroelectric Domain Walls with the Electron Beam at the Atomic Scale". In: *Microscopy and Microanalysis (Accepted)* (2020).
- [17] K. Moore et al. "Highly Charged 180 ° Head-to-Head Domain Walls in PbTiO₃". In: *Nature Communications Physics (Submitted)* (2020).
- [18] Marios Hadjimichael et al. "Direct imaging of artificially-layered ferroelectric domain supercrystals". In: *Nature materials (Submitted)* (2020).
- [19] U Bangert et al. "Ion Implantation of Graphene-Toward IC Compatible Technologies." In: *Nano letters* 13.10 (2013), pp. 4902–4907. DOI: 10.1021/nl402812y. URL: <http://www.ncbi.nlm.nih.gov/pubmed/24059439>.

REFERENCES

- [20] Demie Kepaptsoglou et al. “Electronic Structure Modification of Ion Implanted Graphene : The Spectroscopic Signatures of p- and n-Type Doping”. In: *ACS Nano* 9.11 (2015), pp. 11398–11407. DOI: 10.1021/acsnano.5b05305.
- [21] Ondrej L. Krivanek et al. “Atom-by-atom structural and chemical analysis by annular dark-field electron microscopy”. In: *Nature* 464.7288 (2010), pp. 571–574. DOI: 10.1038/nature08879. URL: <http://www.nature.com/doifinder/10.1038/nature08879>.
- [22] A. V. Crewe, J. Wall, and J. Langmore. “Visibility of Single Atoms”. In: *Science (New York, N.Y.)* 168 (1970), pp. 1338–1340. DOI: 10.1103/PhysRevB.84.073408.
- [23] Rolf Erni et al. “Atomic-Resolution Imaging with a Sub-50-pm Electron Probe”. In: *Physical Review Letters* 102.9 (Mar. 2009), p. 096101. DOI: 10.1103/PhysRevLett.102.096101. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.102.096101>.
- [24] M. M.J. Treacy, A Howie, and C. J. Wilson. “Z contrast of platinum and palladium catalysts”. In: *Philosophical Magazine A: Physics of Condensed Matter, Structure, Defects and Mechanical Properties* 38.5 (Nov. 1978), pp. 569–585. DOI: 10.1080/01418617808239255. URL: <http://www.tandfonline.com/doi/abs/10.1080/01418617808239255>.
- [25] S. J. Pennycook and L.A. Boatner. “Chemically sensitive structure-imaging with a scanning transmission electron microscope”. In: *Nature* 336 (1988).
- [26] S. J. Pennycook and P. D. Nellist. “Scanning transmission electron microscopy”. In: *Impact of Electron Scanning Probe Microscopy on Materials Research* (1999), pp. 161–208. DOI: 10.1007/978-94-011-4451-3.
- [27] S. J. Pennycook. “Structure determination through Z-contrast microscopy”. In: *Advances in Imaging and Electron Physics* 123.C (2002), pp. 173–206. DOI: 10.1016/S1076-5670(02)80063-5.
- [28] David B Williams and C Barry Carter. *The Transmission Electron Microscope*. 2009, pp. 3–22. DOI: 10.1007/978-0-387-76501-3__1. URL: http://www.springer.com/materials/characterization+&+evaluation/book/978-0-387-76500-6%5Cnhttp://link.springer.com/10.1007/978-0-387-76501-3_1.
- [29] Michael M.J. Treacy. “Z dependence of electron scattering by single atoms into annular dark-field detectors”. In: *Microscopy and Microanalysis* 17.6 (2011), pp. 847–858. DOI: 10.1017/S1431927611012074.

REFERENCES

- [30] Ondrej L. Krivanek et al. “Scanning transmission electron microscopy: Albert Crewe’s vision and beyond”. In: *Ultramicroscopy* 123 (2012), pp. 90–98. DOI: 10.1016/j.ultramic.2012.04.004. URL: <http://dx.doi.org/10.1016/j.ultramic.2012.04.004>.
- [31] A.V. Crewe, J.P. Langmore, and M.S. Isaacson. “Resolution and contrast in the STEM”. In: *Physical Aspects of Electron Microscopy and Microbeam Analysis*. Ed. by B.M. Siegel and D.R. Beaman. John Wiley & Sons Inc. New York, 1975, pp. 47–62.
- [32] A. Howie. “Image Contrast And Localized Signal Selection Techniques”. In: *Journal of Microscopy* 117.1 (1979), pp. 11–23. DOI: 10.1111/j.1365-2818.1979.tb00228.x. URL: <http://onlinelibrary.wiley.com/doi/10.1111/j.1365-2818.1979.tb00228.x/abstract>.
- [33] F S Hage et al. “Phonon Spectroscopy at Atomic Resolution”. In: *Physical Review Letters* 122 (2019). DOI: 10.1103/PhysRevLett.122.016103. URL: <https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.122.016103>.
- [34] B D Forbes et al. “Quantum mechanical model for phonon excitation in electron diffraction and imaging using a Born-Oppenheimer approximation”. In: *Physical Review B - Condensed Matter and Materials Physics* 82.10 (2010). DOI: 10.1103/PhysRevB.82.104103. URL: <https://journals.aps.org/prb/pdf/10.1103/PhysRevB.82.104103>.
- [35] Earl J Kirkland. *Advanced Computing in Electron Microscopy*. Second. Springer New York, 2010, pp. 1–286. DOI: 10.1017/CBO9781107415324.004. URL: people.ccmr.cornell.edu/%CB%9Ckirkland.
- [36] J. M. Cowley and A. F. Moodie. “The scattering of electrons by atoms and crystals. I. A new theoretical approach”. In: *Acta Crystallographica* 10.10 (1957), pp. 609–619. DOI: 10.1107/s0365110x57002194.
- [37] B.D.A. Levin et al. “4D STEM with a direct electron detector”. In: *Wiley Analytical Science* (2020). DOI: 10.1002/was.00010003. URL: <https://analyticalscience.wiley.com/do/10.1002/was.00010003>.
- [38] Stephen J. Pennycook and Peter D Nellist, eds. *Scanning Transmission Electron Microscopy - Imaging and Analysis*. 2011.
- [39] Eiji Okunishi, Hidetaka Sawada, and Yukihito Kondo. “Experimental study of annular bright field (ABF) imaging using aberration-corrected scanning transmission electron microscopy (STEM)”. In: *Micron* 43.4 (2012), pp. 538–544. DOI: 10.1016/j.micron.2011.10.007. URL: <http://dx.doi.org/10.1016/j.micron.2011.10.007>.
- [40] K Urban. “Transmission Electron Microscopy II Dark Field (Annular Detector Techniques) in STEM”. In: December (2018).

REFERENCES

- [41] Duncan Alexander. “Scanning TEM (STEM): Imaging & Resolution”. In: (2007), pp. 7–8.
- [42] Shunsuke Yamashita et al. “Atomic number dependence of Z contrast in scanning transmission electron microscopy”. In: *Scientific Reports* 8.1 (2018), pp. 1–7. DOI: 10.1038/s41598-018-30941-5. URL: <http://dx.doi.org/10.1038/s41598-018-30941-5>.
- [43] Lars P. Hansen et al. “Atomic-scale edge structures on industrial-style MoS₂ nanocatalysts”. In: *Angewandte Chemie - International Edition* 50.43 (2011), pp. 10153–10156. DOI: 10.1002/anie.201103745.
- [44] I. Lobato and D. Van Dyck. “An accurate parameterization for scattering factors, electron densities and electrostatic potentials for neutral atoms that obey all physical constraints”. In: *Acta Crystallographica Section A: Foundations and Advances* 70 (2014), pp. 636–649. DOI: 10.1107/S205327331401643X.
- [45] James M Lebeau et al. “Standardless Atom Counting in Scanning Transmission Electron Microscopy”. In: *Nano letters* 10.11 (2010), pp. 4405–4408. DOI: 10.1021/nl102025s. URL: <https://pubs.acs.org/sharingguidelines>.
- [46] S. D. Findlay and J. M. LeBeau. “Detector non-uniformity in scanning transmission electron microscopy”. In: *Ultramicroscopy* 124 (2013), pp. 52–60. DOI: 10.1016/j.ultramic.2012.09.001. URL: <http://dx.doi.org/10.1016/j.ultramic.2012.09.001>.
- [47] Sean Hillyard and John Silcox. “Detector geometry, thermal diffuse scattering and strain effects in ADF STEM imaging”. In: *Ultramicroscopy* 58.1 (1995), pp. 6–17. DOI: 10.1016/0304-3991(94)00173-K.
- [48] James M. LeBeau and Susanne Stemmer. “Experimental quantification of annular dark-field images in scanning transmission electron microscopy”. In: *Ultramicroscopy* 108.12 (2008), pp. 1653–1658. DOI: 10.1016/j.ultramic.2008.07.001.
- [49] Recep Zan et al. “Graphene reknits its holes”. In: *Nano Letters* 12.8 (2012), pp. 3936–3940. DOI: 10.1021/nl300985q.
- [50] Yung Chang Lin et al. “Properties of individual dopant atoms in single-layer MoS₂: Atomic structure, migration, and enhanced reactivity”. In: *Advanced Materials* 26.18 (2014), pp. 2857–2861. DOI: 10.1002/adma.201304985.
- [51] Wu Zhou et al. “Direct Determination of the Chemical Bonding of Individual Impurities in Graphene”. In: *Physical Review Letters* 109.20 (Nov. 2012), p. 206803. DOI: 10.1103/PhysRevLett.109.206803. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.109.206803>.

REFERENCES

- [52] Quentin M Ramasse et al. “Probing the bonding and electronic structure of single atom dopants in graphene with electron energy loss spectroscopy.” In: *Nano letters* 13.10 (2013), pp. 4989–95. DOI: 10.1021/nl304187e. URL: <http://www.ncbi.nlm.nih.gov/pubmed/23259533>.
- [53] Toma Susi, Jannik C. Meyer, and Jani Kotakoski. “Manipulating low-dimensional materials down to the level of single atoms with electron irradiation”. In: *Ultramicroscopy* 180 (Sept. 2017), pp. 163–172. DOI: 10.1016/J.ULTRAMIC.2017.03.005. URL: <https://www.sciencedirect.com/science/article/pii/S0304399117300888>.
- [54] Mukesh Tripathi et al. “Electron-Beam Manipulation of Silicon Dopants in Graphene”. In: *Nano Letters* 18.8 (2018), pp. 5319–5323. DOI: 10.1021/acs.nanolett.8b02406. URL: <http://pubs.acs.org/doi/10.1021/acs.nanolett.8b02406>.
- [55] Mukesh Tripathi et al. “Supplementary - Electron-Beam Manipulation of Silicon Dopants in Graphene”. In: *Nano Letters* 18.8 (2018), pp. 5319–5323. DOI: 10.1021/acs.nanolett.8b02406.
- [56] Toma Susi et al. “Single-atom spectroscopy of phosphorus dopants implanted into graphene Related content”. In: *2D Materials* 4 (2017), p. 021013. DOI: 10.1088/2053-1583/aa5e78. URL: <https://iopscience.iop.org/article/10.1088/2053-1583/aa5e78/pdf>.
- [57] Mukesh Tripathi et al. “Implanting Germanium into Graphene”. In: *ACS Nano* 12 (2018), pp. 4641–4647. DOI: 10.1021/acsnano.8b01191. URL: www.acsnano.org.
- [58] Alex W. Robertson et al. “Dynamics of Single Fe Atoms in Graphene Vacancies”. In: *Nano Letters* 13.4 (Apr. 2013), pp. 1468–1475. DOI: 10.1021/nl304495v. URL: <https://pubs.acs.org/doi/10.1021/nl304495v>.
- [59] Alexander V. Markevich et al. “Dynamic Behavior of Single Fe Atoms Embedded in Graphene”. In: *The Journal of Physical Chemistry C* 120.38 (Sept. 2016), pp. 21998–22003. DOI: 10.1021/acs.jpcc.6b06554. URL: <https://pubs.acs.org/doi/10.1021/acs.jpcc.6b06554>.
- [60] Hongtao Wang et al. “Doping monolayer graphene with single atom substitutions”. In: *Nano Letters* 12.1 (Jan. 2012), pp. 141–144. DOI: 10.1021/nl2031629.
- [61] Recep Zan et al. “Interaction of metals with suspended graphene observed by transmission electron microscopy”. In: *Journal of Physical Chemistry Letters* 3.7 (2012), pp. 953–958. DOI: 10.1021/jz201653g. URL: <http://pubs.acs.org/doi/abs/10.1021/jz201653g>.

REFERENCES

- [62] Christian Kisielowski et al. “Real-time sub-Ångstrom imaging of reversible and irreversible conformations in rhodium catalysts and graphene”. In: *Physical Review B - Condensed Matter and Materials Physics* 88.2 (July 2013), p. 024305. DOI: [10.1103/PhysRevB.88.024305](https://doi.org/10.1103/PhysRevB.88.024305).
- [63] Ondrej L Krivanek et al. “supplementary: Atom-by-atom structural and chemical analysis by annular dark-field electron microscopy.” In: *Nature* 464.7288 (2010), pp. 571–574. DOI: [10.1038/nature08879](https://doi.org/10.1038/nature08879). URL: <http://www.ncbi.nlm.nih.gov/pubmed/20336141%5Cnhttp://www.nature.com/doifinder/10.1038/nature08879>.
- [64] Ang Yu Lu et al. “Janus monolayers of transition metal dichalcogenides”. In: *Nature Nanotechnology* 12.8 (2017), pp. 744–749. DOI: [10.1038/nnano.2017.100](https://doi.org/10.1038/nnano.2017.100).
- [65] Yongji Gong et al. “Band gap engineering and layer-by-layer mapping of selenium-doped molybdenum disulfide”. In: *Nano Letters* 14.2 (2014), pp. 442–449. DOI: [10.1021/nl4032296](https://doi.org/10.1021/nl4032296).
- [66] Toma Susi et al. “Silicon–Carbon Bond Inversions Driven by 60-keV Electrons in Graphene”. In: *Physical Review Letters* 113.11 (2014), p. 115501. DOI: [10.1103/PhysRevLett.113.115501](https://doi.org/10.1103/PhysRevLett.113.115501). URL: <http://link.aps.org/doi/10.1103/PhysRevLett.113.115501>.
- [67] E. H. Åhlgren, A. Markevich, and E. Besley. “Atomistic Simulations of the Efficiencies of Ge and Pt Ion Implantation into Graphene”. In: *Journal of Physical Chemistry C* 122.44 (2018), pp. 25700–25708. DOI: [10.1021/acs.jpcc.8b07306](https://doi.org/10.1021/acs.jpcc.8b07306).
- [68] K. Ishizuka and E. Abe. *DeConvHAADF, In Instrumentation and Methodology, Proceedings of the 13th European Microscopy Congress; Antwerp, Belgium.* 2004. URL: <https://www.hremresearch.com>.
- [69] Francisco de la Peña et al. “hyperspy/hyperspy: HyperSpy v1.5.2”. In: (Sept. 2019). DOI: [10.5281/ZENODO.3396791](https://doi.org/10.5281/ZENODO.3396791). URL: https://doi.org/10.5281/zenodo.3396791#XosKK8ENo_c.mendeley.
- [70] Masashi Watanabe, Eiji Okunishi, and Kazuo Ishizuka. “Analysis of spectrum-imaging datasets in atomic-resolution electron microscopy”. In: *Microscopy and Analysis-UK* 23.7 (2009), pp. 5–7. URL: <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Analysis+of+Spectrum-Imaging+Datasets+in+Atomic-Resolution+Electron+Microscopy#0>.
- [71] Masashi Watanabe. *MSA for DigitalMicrograph*. 2014. URL: <https://www.hremresearch.com/index.html>.
- [72] F. S. Hage et al. “SUP-Single-atom vibrational spectroscopy in the scanning transmission electron microscope”. In: *Science* 367.6482 (2020), pp. 1120–1124. DOI: [10.1126/science.aba1136](https://doi.org/10.1126/science.aba1136).

REFERENCES

- [73] Christoph Koch. *DETERMINATION OF CORE STRUCTURE PERIODICITY AND POINT DEFECT DENSITY ALONG DISLOCATIONS* by. Tech. rep. Arizona State University, 2002. URL: https://www.physics.hu-berlin.de/en/seminars/images/koch02_phdthesis.pdf.
- [74] O. Lehtinen et al. “Atomic scale study of the life cycle of a dislocation in graphene from birth to annihilation”. In: *Nature Communications* 4 (2013), pp. 1–7. DOI: 10.1038/ncomms3098. URL: <http://dx.doi.org/10.1038/ncomms3098>.
- [75] Stewart J. Clark et al. “First principles methods using CASTEP”. In: *Zeitschrift für Kristallographie - Crystalline Materials* 220.5/6 (Jan. 2005). DOI: 10.1524/zkri.220.5.567.65075. URL: <http://www.degruyter.com/view/j/zkri.2005.220.issue-5-6/zkri.220.5.567.65075/zkri.220.5.567.65075.xml>.
- [76] P.A. Stadelmann. “EMS - a software package for electron diffraction analysis and HREM image simulation in materials science”. In: *Ultramicroscopy* 21.2 (Jan. 1987), pp. 131–145. DOI: 10.1016/0304-3991(87)90080-5. URL: <https://linkinghub.elsevier.com/retrieve/pii/0304399187900805>.
- [77] Lan Nguyen et al. “Atomic Defects and Doping of Monolayer NbSe₂”. In: *ACS Nano* 11.3 (2017), pp. 2894–2904. DOI: 10.1021/acsnano.6b08036.
- [78] Luiz H.G. Tizei et al. “Exciton Mapping at Subwavelength Scales in Two-Dimensional Materials”. In: *Physical Review Letters* 114.10 (2015), pp. 1–5. DOI: 10.1103/PhysRevLett.114.107601.
- [79] Yongji Gong et al. “Vertical and in-plane heterostructures from WS₂/MoS₂ monolayers”. In: *Nature Materials* 13.12 (Dec. 2014), pp. 1135–1142. DOI: 10.1038/nmat4091. URL: <http://www.nature.com/articles/nmat4091>.
- [80] Bernhard Schaffer. *StackBuilder*. 2019. URL: <http://digitalmicrograph-scripting.tavernmaker.de>.
- [81] Chunming Huang et al. “Lateral heterojunctions within monolayer MoSe₂–WSe₂ semiconductors”. In: *Nature Materials* 13.12 (Dec. 2014), pp. 1096–1101. DOI: 10.1038/nmat4064. URL: <http://www.nature.com/articles/nmat4064>.
- [82] Total Resolution. *MacTempasX*. 2019.
- [83] G. Kresse and J. Furthmüller. “Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set”. In: *Physical Review B* 54.16 (Oct. 1996), pp. 11169–11186. DOI: 10.1103/PhysRevB.54.11169. URL: <https://link.aps.org/doi/10.1103/PhysRevB.54.11169>.
- [84] Alex W. Robertson et al. “Atomic Structure and Spectroscopy of Single Metal (Cr, V) Substitutional Dopants in Monolayer MoS₂”. In: *ACS Nano* 10.11 (2016), pp. 10227–10236. DOI: 10.1021/acsnano.6b05674.

REFERENCES

- [85] Guoliang Liu et al. “MoS₂ monolayer catalyst doped with isolated Co atoms for the hydrodeoxygenation reaction”. In: *Nature Chemistry* 9.8 (2017), pp. 810–816. DOI: 10.1038/NCHEM.2740. URL: <http://dx.doi.org/10.1038/nchem.2740>.
- [86] N Bonnet, N Brun, and C Colliex. “Extracting information from sequences of spatially resolved EELS spectra using multivariate statistical analysis”. In: *Ultramicroscopy* 77.3-4 (July 1999), pp. 97–112. DOI: 10.1016/S0304-3991(99)00042-X. URL: <https://www.sciencedirect.com/science/article/pii/S030439919900042X?via%3Dihub>.
- [87] Yuanyuan Zhu et al. “Visualizing the stoichiometry of industrial-style Co-Mo-S catalysts with single-atom sensitivity”. In: *Angewandte Chemie - International Edition* 53.40 (2014), pp. 10723–10727. DOI: 10.1002/anie.201405690.
- [88] Paolo Giannozzi et al. “QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials”. In: *Journal of Physics: Condensed Matter* 21.39 (Sept. 2009), p. 395502. DOI: 10.1088/0953-8984/21/39/395502. URL: <http://stacks.iop.org/0953-8984/21/i=39/a=395502?key=crossref.c21336c286fa6d3db893262ae3f6e151>.
- [89] Gongxun Bai et al. “2D Layered Materials of Rare-Earth Er-Doped MoS₂ with NIR-to-NIR Down- and Up-Conversion Photoluminescence”. In: *Advanced Materials* (2016), pp. 7472–7477. DOI: 10.1002/adma.201601833.
- [90] Jian Gao et al. “Transition-Metal Substitution Doping in Synthetic Atomically Thin Semiconductors”. In: *Advanced Materials* 28.44 (2016), pp. 9735–9743. DOI: 10.1002/adma.201601104.
- [91] Stéfan van der Walt et al. “scikit-image: image processing in Python”. In: *PeerJ* 2 (June 2014), e453. DOI: 10.7717/peerj.453. URL: <https://peerj.com/articles/453>.
- [92] Vidya Kochat et al. “Re Doping in 2D Transition Metal Dichalcogenides as a New Route to Tailor Structural Phases and Induced Magnetism”. In: *Advanced Materials* 29.43 (2017), pp. 1–8. DOI: 10.1002/adma.201703754.
- [93] Huashan Li et al. “Atomic Structure and Dynamics of Single Platinum Atom Interactions with Monolayer MoS₂”. In: *ACS Nano* 11.3 (Mar. 2017), pp. 3392–3403. DOI: 10.1021/acsnano.7b00796. URL: <https://pubs.acs.org/doi/10.1021/acsnano.7b00796>.
- [94] H.-J. Kim et al. “Enhanced electrical and optical properties of single-layered MoS₂ by incorporation of aluminum”. In: *Nano Research* (2017), pp. 1–10. DOI: 10.1007/s12274-017-1682-4.

REFERENCES

- [95] Guoli Yin et al. “Hydrogen-assisted post-growth substitution of tellurium into molybdenum Hydrogen-assisted post-growth substitution of tellurium into molybdenum disul fi de monolayers with tunable compositions”. In: *Nanotechnology* 29.14 (2018), p. 145603. DOI: 10.1088/1361-6528/aaabe8. URL: <http://dx.doi.org/10.1088/1361-6528/aaabe8>.
- [96] Yu-Chuan Lin et al. “Low Energy Implantation into Transition Metal Dichalcogenide Monolayers to Form Janus Structures”. In: *ACS Nano* (2020), acsnano.9b10196. DOI: 10.1021/acsnano.9b10196. URL: <https://pubs.acs.org/doi/10.1021/acsnano.9b10196>.
- [97] Shogo Sasaki et al. “Growth and optical properties of Nb-doped WS₂ monolayers”. In: *Applied Physics Express* 9.7 (July 2016), p. 071201. DOI: 10.7567/APEX.9.071201. URL: <http://stacks.iop.org/1882-0786/9/i=7/a=071201?key=crossref.fcd0f3f08c7cdfc6f0fae95286ccbea0>.
- [98] Shanshan Wang et al. “Atomic structure and formation mechanism of sub-nanometer pores in 2D monolayer MoS₂”. In: *Nanoscale* 9.19 (May 2017), pp. 6417–6426. DOI: 10.1039/C7NR01127J. URL: <http://xlink.rsc.org/?DOI=C7NR01127J>.
- [99] Ryan J. Wu, Michael L. Odlyzko, and K. Andre Mkhoyan. “Determining the thickness of atomically thin MoS₂ and WS₂ in the TEM”. In: *Ultramicroscopy* 147 (2014), pp. 8–20. DOI: 10.1016/j.ultramic.2014.05.007. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0304399114001077>.
- [100] Ryosuke Senga et al. “Atomic structure and dynamic behaviour of truly one-dimensional ionic chains inside carbon nanotubes”. In: *Nature Materials* 13.11 (2014), pp. 1050–1054. DOI: 10.1038/nmat4069.
- [101] Toma Susi et al. *Towards atomically precise manipulation of 2D nanostructures in the electron microscope*. Sept. 2017. DOI: 10.1088/2053-1583/aa878f. URL: <http://stacks.iop.org/2053-1583/4/i=4/a=042004?key=crossref.ead42ba069c2b438d855da4a962552df>.
- [102] J. Cui et al. “The origin of atomic displacements in HAADF images of the tilted specimen”. In: *Ultramicroscopy* 182 (2017), pp. 156–162. DOI: 10.1016/j.ultramic.2017.07.007. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0304399117300931>.
- [103] Pierre-Simon Laplace. “Philosophical essay on probabilities. Transl. from the 5th French edition”. In: ed. by Andrew I. Dale. Gauthier-Villar, 1886. URL: <http://www.medicine.mcgill.ca/epidemiology/hanley/bios601/GaussianModel/HistoryCentralLimitTheorem.pdf>.

REFERENCES

- [104] Hans Fischer. *A History of the Central Limit Theorem*. Springer New York Dordrecht Heidelberg London, 2011. URL: <http://www.medicine.mcgill.ca/epidemiology/hanley/bios601/GaussianModel/HistoryCentralLimitTheorem.pdf>.
- [105] Andreas Beyer et al. “Influence of spatial and temporal coherences on atomic resolution high angle annular dark field imaging”. In: *Ultramicroscopy* 169 (2016), pp. 1–10. DOI: 10.1016/j.ultramic.2016.06.006. URL: <http://dx.doi.org/10.1016/j.ultramic.2016.06.006>.
- [106] Ryo Ishikawa et al. “Three-dimensional location of a single dopant with atomic precision by aberration-corrected scanning transmission electron microscopy”. In: *Nano Letters* 14.4 (2014), pp. 1903–1908. DOI: 10.1021/nl500564b. URL: <https://pubs.acs.org/sharingguidelines>.
- [107] Andreas Rosenauer et al. “Composition mapping in InGaN by scanning transmission electron microscopy”. In: *Ultramicroscopy* 111.8 (July 2011), pp. 1316–1327. DOI: 10.1016/j.ultramic.2011.04.009.
- [108] G Voronoi. “Nouvelles applications des paramètres continus à la théorie des formes quadratiques”. In: *Journal für die Reine und Angewandte Mathematik* 133 (1908), pp. 97–178.
- [109] Franz Aurenhammer. “Voronoi diagrams—a survey of a fundamental geometric data structure”. In: *ACM Computing Surveys (CSUR)* 23.3 (Jan. 1991), pp. 345–405. DOI: 10.1145/116873.116880.
- [110] Tim Grieb et al. “Determination of the chemical composition of GaNAs using STEM HAADF imaging and STEM strain state analysis”. In: *Ultramicroscopy* 117 (2012), pp. 15–23. DOI: 10.1016/j.ultramic.2012.03.014.
- [111] P. Kükelhan et al. “Simultaneous determination of local thickness and composition for ternary III-V semiconductors by aberration-corrected STEM”. In: *Ultramicroscopy* 201.March (2019), pp. 49–57. DOI: 10.1016/j.ultramic.2019.03.005. URL: <https://doi.org/10.1016/j.ultramic.2019.03.005>.
- [112] P. Kükelhan et al. “Segregation at interfaces in (GaIn)As/Ga(AsSb)/(GaIn)As quantum well heterostructures explored by atomic resolution STEM”. In: *Journal of Crystal Growth* 524.August (2019), p. 125180. DOI: 10.1016/j.jcrysgro.2019.125180. URL: <https://doi.org/10.1016/j.jcrysgro.2019.125180>.
- [113] L. Duscek et al. “Composition determination of semiconductor alloys towards atomic accuracy by HAADF-STEM”. In: *Ultramicroscopy* 200 (May 2019), pp. 84–96. DOI: 10.1016/j.ultramic.2019.02.009.

REFERENCES

- [114] Ryo Ishikawa et al. “Direct observation of dopant atom diffusion in a bulk semiconductor crystal enhanced by a large size mismatch”. In: *Physical Review Letters* 113.15 (2014), p. 55. DOI: 10.1103/PhysRevLett.113.155501. URL: <https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.113.155501>.
- [115] S. Van Aert et al. “Procedure to count atoms with trustworthy single-atom sensitivity”. In: *Physical Review B - Condensed Matter and Materials Physics* 87.6 (2013), pp. 1–6. DOI: 10.1103/PhysRevB.87.064107.
- [116] H. E et al. “Probe integrated scattering cross sections in the analysis of atomic resolution HAADF STEM images”. In: *Ultramicroscopy* 133 (2013), pp. 109–119. DOI: 10.1016/j.ultramic.2013.07.002.
- [117] A. De Backer et al. “Atom counting in HAADF STEM using a statistical model-based approach: Methodology, possibilities, and inherent limitations”. In: *Ultramicroscopy* 134.2013 (2013), pp. 23–33. DOI: 10.1016/j.ultramic.2013.05.003. URL: <http://dx.doi.org/10.1016/j.ultramic.2013.05.003>.
- [118] Lewys Jones et al. “Rapid estimation of catalyst nanoparticle morphology and atomic-coordination by high-resolution Z-contrast electron microscopy”. In: *Nano Letters* 14.11 (2014), pp. 6336–6341. DOI: 10.1021/nl502762m.
- [119] Aakash M. Varambhia et al. “Quantifying a Heterogeneous Ru Catalyst on Carbon Black Using ADF STEM”. In: *Particle and Particle Systems Characterization* 33.7 (July 2016), pp. 438–444. DOI: 10.1002/ppsc.201600067.
- [120] Sandra Van Aert et al. “Control of Knock-On Damage for 3D Atomic Scale Quantification of Nanostructures: Making Every Electron Count in Scanning Transmission Electron Microscopy”. In: *Physical Review Letters* 122 (2019). DOI: 10.1103/PhysRevLett.122.066101.
- [121] A. De Backer et al. “StatSTEM: An efficient approach for accurate and precise model-based quantification of atomic resolution electron microscopy images”. In: *Ultramicroscopy* 171 (2016), pp. 104–116. DOI: 10.1016/j.ultramic.2016.08.018.
- [122] Annelies De wael et al. “Hybrid statistics-simulations based method for atom-counting from ADF STEM images”. In: *Ultramicroscopy* 177 (June 2017), pp. 69–77. DOI: 10.1016/j.ultramic.2017.01.010.
- [123] Annelies De wael et al. “Measuring Dynamic Structural Changes of Nanoparticles at the Atomic Scale Using Scanning Transmission Electron Microscopy”. In: *Physical Review Letters* 124.10 (2020), p. 106105. DOI: 10.1103/physrevlett.124.106105. URL: <https://doi.org/10.1103/PhysRevLett.124.106105>.

REFERENCES

- [124] Ryosuke Senga and Kazu Suenaga. “Single-atom detection of light elements: Imaging or spectroscopy?” In: *Ultramicroscopy* 180 (2017), pp. 150–155. DOI: 10.1016/j.ultramic.2016.12.007. URL: <http://dx.doi.org/10.1016/j.ultramic.2016.12.007>.
- [125] R F Egerton. *Electron Energy-Loss Spectroscopy in the Electron Microscope*. 2011, pp. 231–232. DOI: 10.1088/0034-4885/72/1/016502.
- [126] M Varela et al. “Spectroscopic imaging of single atoms within a bulk solid”. In: *Physical Review Letters* 92.9 (2004). DOI: 10.1103/PhysRevLett.92.095502. URL: <https://journals.aps.org/prl/pdf/10.1103/PhysRevLett.92.095502>.
- [127] L. A. Grunes R. D. Leapman and P. L. Fejesf. “Study of the L23 edges in the 3d transition metals and their oxides by electron-energy-loss spectroscopy with comparisons to theory”. In: *Physical Review* 26.1 (1982).
- [128] D. A. Muller et al. “Atomic-scale chemical imaging of composition and bonding by aberration-corrected microscopy”. In: *Microscopy and Microanalysis* 14.SUPPL. 2 (2008), pp. 132–133. DOI: 10.1017/S1431927608087448.
- [129] Ryosuke Senga and Kazu Suenaga. “Single-atom electron energy loss spectroscopy of light elements”. In: *Nature Communications* 6 (2015), pp. 1–6. DOI: 10.1038/ncomms8943.
- [130] Kazu Suenaga et al. “Visualizing and identifying single atoms using electron energy-loss spectroscopy with low accelerating voltage”. In: *Nature Chemistry* 1.5 (2009), pp. 415–418. DOI: 10.1038/nchem.282. URL: <http://dx.doi.org/10.1038/nchem.282>.
- [131] Zheng Liu et al. “Identification of active atomic defects in a monolayered tungsten disulphide nanoribbon”. In: *Nature Communications* 2.1 (2011). DOI: 10.1038/ncomms1224.
- [132] Kazu Suenaga and Masanori Koshino. “Atom-by-atom spectroscopy at graphene edge”. In: *Nature* 468.7327 (2010), pp. 1088–1090. DOI: 10.1038/nature09664.
- [133] Fredrik Sydow Hage et al. “Local Plasmon Engineering in Doped Graphene”. In: *ACS Nano* 12 (2018), pp. 1837–1848. DOI: 10.1021/acsnano.7b08650. URL: www.acsnano.org.
- [134] Fredrik S Hage et al. “Nanoscale momentum-resolved vibrational spectroscopy”. In: *Science Advances* 4.June (2018), pp. 1–7.
- [135] T. Malis, S. C. Cheng, and R. F. Egerton. “EELS log-ratio technique for specimen-thickness measurement in the TEM”. In: *Journal of Electron Microscopy Technique* 8.2 (Feb. 1988), pp. 193–200. DOI: 10.1002/jemt.1060080206. URL: <http://doi.wiley.com/10.1002/jemt.1060080206>.

REFERENCES

- [136] F. J. García de Abajo. “Optical excitations in electron microscopy”. In: *Reviews of Modern Physics* 82.1 (2010), pp. 209–275. DOI: 10.1103/RevModPhys.82.209.
- [137] Wu Zhou et al. “Atomically localized plasmon enhancement in monolayer graphene”. In: *Nature Nanotechnology* 7.3 (2012), pp. 161–165. DOI: 10.1038/nnano.2011.252. URL: www.nature.com/naturenanotechnology.
- [138] Jiadan Lin et al. “Plasmonic enhancement of photocurrent in MoS₂field-effect-transistor”. In: *Applied Physics Letters* 102.20 (2013), pp. 2011–2014. DOI: 10.1063/1.4807658.
- [139] Alejandro Molina-Sánchez et al. “Effect of spin-orbit interaction on the optical spectra of single-layer, double-layer, and bulk MoS₂”. In: *Physical Review B - Condensed Matter and Materials Physics* 88.4 (July 2013), p. 045412. DOI: 10.1103/PhysRevB.88.045412.
- [140] Karl Pearson. “LIII. On lines and planes of closest fit to systems of points in space”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (Nov. 1901), pp. 559–572. DOI: 10.1080/14786440109462720.
- [141] H. Hotelling. “Analysis of a complex of statistical variables into principal components”. In: *Journal of Educational Psychology* 24.6 (Sept. 1933), pp. 417–441. DOI: 10.1037/h0071325.
- [142] Pierre Trebbia and Noël Bonnet. “EELS elemental mapping with unconventional methods I. Theoretical basis: Image analysis with multivariate statistics and entropy concepts”. In: *Ultramicroscopy* 34.3 (Dec. 1990), pp. 165–178. DOI: 10.1016/0304-3991(90)90070-3.
- [143] Aapo Hyvärinen. “Fast and robust fixed-point algorithms for independent component analysis”. In: *IEEE Transactions on Neural Networks* 10.3 (1999), pp. 626–634. DOI: 10.1109/72.761722.
- [144] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja. *Independent component analysis*. John Wiley & Sons Inc. New York, 2001.
- [145] Noël Bonnet and Danielle Nuzillard. “Independent component analysis: A new possibility for analysing series of electron energy loss spectra”. In: *Ultramicroscopy* 102.4 (2005), pp. 327–337. DOI: 10.1016/j.ultramic.2004.11.003.
- [146] José M P Nascimento and José M Bioucas Dias. “Vertex component analysis: A fast algorithm to unmix hyperspectral data”. In: *IEEE Transactions on Geoscience and Remote Sensing* 43.4 (2005), pp. 898–910. DOI: 10.1109/TGRS.2005.844293.
- [147] Nicolas Dobigeon et al. “Joint Bayesian endmember extraction and linear unmixing for hyperspectral imagery”. In: *IEEE Transactions on Signal Processing* 57.11 (2009), pp. 4355–4368. DOI: 10.1109/TSP.2009.2025797.

REFERENCES

- [148] I.T. Jolliffe. *Principal Component Analysis*. Springer, 2002. URL: <https://www.springer.com/gp/book/9780387954424>.
- [149] Nirmal Keshava and John F. Mustard. “Spectral unmixing”. In: *IEEE Signal Processing Magazine* 19.1 (2002), pp. 44–57. DOI: 10.1109/79.974727.
- [150] Paul Cueva et al. “Data Processing for Atomic Resolution Electron Energy Loss Spectroscopy”. In: *Microscopy and Microanalysis* 18 (2012), pp. 667–675.
- [151] David C. Joy and Dennis M. Maher. “The quantitation of electron energy loss spectra”. In: *Journal of Microscopy* 124.1 (Oct. 1981), pp. 37–48. DOI: 10.1111/j.1365-2818.1981.tb01303.x. URL: <http://doi.wiley.com/10.1111/j.1365-2818.1981.tb01303.x>.
- [152] Michael R. Keenan and Paul G. Kotula. “Accounting for Poisson noise in the multivariate analysis of ToF-SIMS spectrum images”. In: *Surface and Interface Analysis* 36.3 (2004), pp. 203–212. DOI: 10.1002/sia.1657.
- [153] F. de la Peña et al. “Mapping titanium and tin oxide phases using EELS: An application of independent component analysis”. In: *Ultramicroscopy* 111.2 (2011), pp. 169–176. DOI: 10.1016/j.ultramic.2010.10.001.
- [154] Jakob Spiegelberg and Ján Rusz. “Can we use PCA to detect small signals in noisy data?” In: *Ultramicroscopy* 172.October 2016 (2016), pp. 40–46. DOI: 10.1016/j.ultramic.2016.10.008. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0304399116302480>.
- [155] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. URL: <http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html>.
- [156] K. W. Urban et al. “Studying Atomic Structures by Aberration-Corrected Transmission Electron Microscopy”. In: *Science* 321.5888 (2008), pp. 506–510. DOI: 10.1126/science.1152800. URL: <http://www.ncbi.nlm.nih.gov/pubmed/18653874> %0Ahttp://www.sciencemag.org/cgi/doi/10.1126/science.1152800.
- [157] D. Cherns, F. J. Minter, and R. S. Nelson. “Sputtering in the high voltage electron microscope”. In: *Nuclear Instruments and Methods* 132.C (Jan. 1976), pp. 369–376. DOI: 10.1016/0029-554X(76)90760-6.
- [158] L. W. Hobbs. “Radiation Effects in Analysis of Inorganic Specimens by TEM”. In: *Introduction to Analytical Electron Microscopy*. 1979, pp. 437–480. DOI: 10.1007/978-1-4757-5581-7{_}17.
- [159] MR McCartney et al. “Electron-beam-induced reactions at transition-metal oxide surfaces”. In: *Vacuum* 42.4 (Jan. 1991), pp. 301–308. DOI: 10.1016/0042-207X(91)90042-H.

REFERENCES

- [160] R F Egerton, P Li, and M Malac. “Radiation damage in the TEM and SEM”. In: *Micron* 35 (2004), pp. 399–409. DOI: 10.1016/j.micron.2004.02.003. URL: https://ac.els-cdn.com/S0968432804000381/1-s2.0-S0968432804000381-main.pdf?_tid=8376a32a-0765-11e8-a9e6-0000aab0f27&acdnat=1517499501_2a49c82aac9f58cc7af807b971c72046.
- [161] R. F. Egerton et al. “Basic questions related to electron-induced sputtering in the TEM”. In: *Ultramicroscopy* 110.8 (July 2010), pp. 991–997. DOI: 10.1016/j.ultramic.2009.11.003. URL: <https://www.sciencedirect.com/science/article/pii/S0304399109002472?via%3Dihub>.
- [162] Florian Banhart. “Irradiation effects in carbon nanostructures”. In: *Reports on Progress in Physics* 62.8 (Aug. 1999), pp. 1181–1221. DOI: 10.1088/0034-4885/62/8/201. URL: <http://stacks.iop.org/0034-4885/62/i=8/a=201?key=crossref.59dc06970256651c592563e84070dc05>.
- [163] D. Cherns, M. W. Finnis, and M. D. Matthews. “Sputtering of gold foils in a high voltage electron microscope A comparison of theory and experiment”. In: *Philosophical Magazine* 35.3 (1977), pp. 693–714. DOI: 10.1080/14786437708235999.
- [164] J. Kotakoski et al. “From Point Defects in Graphene to Two-Dimensional Amorphous Carbon”. In: *Physical Review Letters* 106.10 (Mar. 2011), p. 105505. DOI: 10.1103/PhysRevLett.106.105505. URL: <https://link.aps.org/doi/10.1103/PhysRevLett.106.105505>.
- [165] J. Kotakoski et al. “Electron knock-on damage in hexagonal boron nitride monolayers”. In: *Physical Review B - Condensed Matter and Materials Physics* 82.11 (Sept. 2010), p. 113404. DOI: 10.1103/PhysRevB.82.113404.
- [166] Hannu Pekka Komsa et al. “Two-dimensional transition metal dichalcogenides under electron irradiation: Defect production and doping”. In: *Physical Review Letters* 109.3 (2012), pp. 1–5. DOI: 10.1103/PhysRevLett.109.035503.
- [167] Hannu Pekka Komsa et al. “From point to extended defects in two-dimensional MoS₂: Evolution of atomic structure under electron irradiation”. In: *Physical Review B - Condensed Matter and Materials Physics* 88.3 (2013), pp. 1–8. DOI: 10.1103/PhysRevB.88.035301.
- [168] Anthony Yoshimura et al. “First-principles simulation of local response in transition metal dichalcogenides under electron irradiation”. In: *Nanoscale* 10.5 (2018), pp. 2388–2397. DOI: 10.1039/C7NR07024A. URL: <http://xlink.rsc.org/?DOI=C7NR07024A>.
- [169] John P. Perdew, Kieron Burke, and Matthias Ernzerhof. “Generalized gradient approximation made simple”. In: *Physical Review Letters* 77.18 (Oct. 1996), pp. 3865–3868. DOI: 10.1103/PhysRevLett.77.3865.

REFERENCES

- [170] Jiří Klimeš, David R. Bowler, and Angelos Michaelides. “Chemical accuracy for the van der Waals density functional”. In: *Journal of Physics Condensed Matter* 22.2 (2010). DOI: 10.1088/0953-8984/22/2/022201.
- [171] Jiří Klimeš, David R. Bowler, and Angelos Michaelides. “Van der Waals density functionals applied to solids”. In: *Physical Review B - Condensed Matter and Materials Physics* 83.19 (May 2011), p. 195131. DOI: 10.1103/PhysRevB.83.195131.
- [172] D. M. Ceperley and B. J. Alder. “Ground state of the electron gas by a stochastic method”. In: *Physical Review Letters* 45.7 (Aug. 1980), pp. 566–569. DOI: 10.1103/PhysRevLett.45.566.
- [173] Y. Kudriavtsev et al. “Calculation of the surface binding energy for ion sputtered particles”. In: *Applied Surface Science* 239.3-4 (2005), pp. 273–278. DOI: 10.1016/j.apsusc.2004.06.014.
- [174] Lars P. Hansen et al. “Supplementary - Atomic-Scale Edge Structures on Industrial-Style MoS₂ Nanocatalysts”. In: *Angewandte Chemie - International Edition* 50.43 (2012).
- [175] R. F. Egerton. “Control of radiation damage in the TEM”. In: *Ultramicroscopy* 127 (2013), pp. 100–108. DOI: 10.1016/j.ultramic.2012.07.006. URL: <http://dx.doi.org/10.1016/j.ultramic.2012.07.006>.
- [176] Liqiong Zhang et al. “A study on monolayer MoS₂ doping at the S site via the first principle calculations”. In: *Physica E: Low-dimensional Systems and Nanostructures* 94.June (2017), pp. 47–52. DOI: 10.1016/j.physe.2017.06.028. URL: <http://dx.doi.org/10.1016/j.physe.2017.06.028> %A <http://linkinghub.elsevier.com/retrieve/pii/S1386947717308275>.
- [177] R. F. Egerton, P. A. Crozier, and P. Rice. “Electron energy-loss spectroscopy and chemical change”. In: *Ultramicroscopy* 23.3-4 (Jan. 1987), pp. 305–312. DOI: 10.1016/0304-3991(87)90240-3.
- [178] Silvan Kretschmer et al. “Structural Transformations in Two-Dimensional Transition-Metal Dichalcogenide MoS₂ under an Electron Beam: Insights from First-Principles Calculations”. In: *The Journal of Physical Chemistry Letters* 8.13 (July 2017), pp. 3061–3067. DOI: 10.1021/acs.jpclett.7b01177. URL: <https://pubs.acs.org/doi/10.1021/acs.jpclett.7b01177>.
- [179] I. G. Salisbury et al. “Nanometer scale electron beam lithography in inorganic materials”. In: *Applied Physics Letters* 45.12 (1984), pp. 1289–1291. DOI: 10.1063/1.95115.
- [180] Richard Henderson and Robert M. Glaeser. “Quantitative analysis of image contrast in electron micrographs of beam-sensitive crystals”. In: *Ultramicroscopy* 16.2 (Jan. 1985), pp. 139–150. DOI: 10.1016/0304-3991(85)90069-5.

REFERENCES

- [181] Bo Wu and Andrew R. Neureuther. “Energy deposition and transfer in electron-beam lithography”. In: *Journal of Vacuum Science and Technology B: Microelectronics and Nanometer Structures* 19.6 (2001), pp. 2508–2511. DOI: 10.1116/1.1421548.
- [182] Toma Susi, Jannik C. Meyer, and Jani Kotakoski. “Quantifying transmission electron microscopy irradiation effects using two-dimensional materials”. In: *Nature Reviews Physics* 1.6 (2019), pp. 397–405. DOI: 10.1038/s42254-019-0058-y. URL: <http://dx.doi.org/10.1038/s42254-019-0058-y>.
- [183] Recep Zan et al. “Control of Radiation Damage in MoS₂ by Graphene Encapsulation.” In: *ACS nano* 7.11 (2013), pp. 10167–74. DOI: 10.1021/nn4044035. URL: <http://www.ncbi.nlm.nih.gov/pubmed/24116975>.
- [184] Gerardo Algara-Siller et al. “The pristine atomic structure of MoS₂ monolayer protected from electron radiation damage by graphene”. In: *Applied Physics Letters* 103.20 (2013). DOI: 10.1063/1.4830036.
- [185] Aaron C Johnston-Peck et al. “Dose-rate-dependent damage of cerium dioxide in the scanning transmission electron microscope”. In: *Ultramicroscopy* 170 (2016), pp. 1–9. DOI: 10.1016/j.ultramic.2016.07.002. URL: <http://dx.doi.org/10.1016/j.ultramic.2016.07.002>.
- [186] Nan Jiang and John C.H. Spence. “On the dose-rate threshold of beam damage in TEM”. In: *Ultramicroscopy* 113 (2012), pp. 77–82. DOI: 10.1016/j.ultramic.2011.11.016. URL: <http://dx.doi.org/10.1016/j.ultramic.2011.11.016>.
- [187] A. N. Broers. “Resolution limits for electron-beam lithography”. In: *IBM Journal of Research and Development* 32.4 (1988), pp. 502–513. DOI: 10.1147/rd.324.0502.
- [188] Xiaoxu Zhao et al. “Engineering and modifying two-dimensional materials by electron beams”. In: *MRS Bulletin* 42.09 (Sept. 2017), pp. 667–676. DOI: 10.1557/mrs.2017.184. URL: https://www.cambridge.org/core/product/identifier/S0883769417001841/type/journal_article.
- [189] U. Kaiser et al. “Transmission electron microscopy at 20kV for imaging and spectroscopy”. In: *Ultramicroscopy* 111.8 (2011), pp. 1239–1246. DOI: 10.1016/j.ultramic.2011.03.012.
- [190] L. Jones. “Quantitative ADF STEM: Acquisition, analysis and interpretation”. In: *IOP Conference Series: Materials Science and Engineering* 109.1 (2016). DOI: 10.1088/1757-899X/109/1/012008.
- [191] K. S. Novoselov et al. “Electric Field Effect in Atomically Thin Carbon Films”. In: *Science* 306 (2004), pp. 666–669.

REFERENCES

- [192] Filippo Pizzocchero et al. “The hot pick-up technique for batch assembly of van der Waals heterostructures”. In: *Nature Communications* 7.May (2016). DOI: 10.1038/ncomms11894.
- [193] Maria O’Brien et al. “Transition Metal Dichalcogenide Growth via Close Proximity Precursor Supply”. In: *Scientific Reports* 4.1 (May 2015), p. 7374. DOI: 10.1038/srep07374. URL: <http://www.nature.com/articles/srep07374>.
- [194] Luca Banszerus et al. “Ultrahigh-mobility graphene devices from chemical vapor deposition on reusable copper”. In: *Science Advances* 1.6 (July 2015), e1500222. DOI: 10.1126/sciadv.1500222.
- [195] Gerardo Algara-Siller et al. “Dry-cleaning of graphene”. In: *Applied Physics Letters* 104.15 (2014), pp. 1–5. DOI: 10.1063/1.4871997.
- [196] Ahmad Ehteshamul Islam et al. “Atomic level cleaning of poly-methyl-methacrylate residues from the graphene surface using radiolized water at high temperatures”. In: *Applied Physics Letters* 111.10 (2017). DOI: 10.1063/1.5001479. URL: <http://dx.doi.org/10.1063/1.5001479>.
- [197] Yuehui Jia et al. “Toward High Carrier Mobility and Low Contact Resistance: Laser Cleaning of PMMA Residues on Graphene Surfaces”. In: *Nano-Micro Letters* 8.4 (2016), pp. 336–346. DOI: 10.1007/s40820-016-0093-5.
- [198] Mukesh Tripathi et al. “Cleaning graphene: Comparing heat treatments in air and in vacuum”. In: *Physica Status Solidi - Rapid Research Letters* 11.8 (2017). DOI: 10.1002/pssr.201700124.
- [199] D. G. Purdie et al. “Cleaning interfaces in layered materials heterostructures”. In: *Nature Communications* 9.1 (2018), pp. 1–12. DOI: 10.1038/s41467-018-07558-3. URL: <http://dx.doi.org/10.1038/s41467-018-07558-3>.
- [200] Yung Chang Lin et al. “Atomic mechanism of the semiconducting-to-metallic phase transition in single-layered MoS₂”. In: *Nature Nanotechnology* 9.5 (2014), pp. 391–396. DOI: 10.1038/nnano.2014.64.
- [201] Nikolaos Peltekis et al. “The effect of downstream plasma treatments on graphene surfaces”. In: *Carbon* 50.2 (2012), pp. 395–403. DOI: 10.1016/j.carbon.2011.08.052. URL: <http://dx.doi.org/10.1016/j.carbon.2011.08.052>.
- [202] Achint Jain et al. “Minimizing residues and strain in 2D materials transferred from PDMS”. In: *Nanotechnology* 29.26 (2018). DOI: 10.1088/1361-6528/aabd90.
- [203] K S Novoselov et al. “Two-dimensional atomic crystals.” In: *Proceedings of the National Academy of Sciences of the United States of America* 102.30 (July 2005), pp. 10451–3. DOI: 10.1073/pnas.0502848102. URL: <http://www.pnas.org/content/102/30/10451.abstract>.

REFERENCES

- [204] Kin Fai Mak et al. “Atomically Thin MoS₂: A New Direct-Gap Semiconductor”. In: *Physical Review Letters* 105.13 (2010), p. 136805. DOI: 10.1103/PhysRevLett.105.136805. URL: <http://link.aps.org/doi/10.1103/PhysRevLett.105.136805>.
- [205] B. Radisavljevic et al. “Single-layer MoS₂ transistors”. In: *Nature nanotechnology* 6.3 (2011), pp. 147–50. DOI: 10.1038/nnano.2010.279. arXiv: 0402594v3 [cond-mat]. URL: <http://www.nature.com/doifinder/10.1038/nnano.2010.279> Cn <http://www.ncbi.nlm.nih.gov/pubmed/21278752>.
- [206] Sunkook Kim et al. “High-mobility and low-power thin-film transistors based on multilayer MoS₂ crystals.” In: *Nature communications* 3 (2012), p. 1011. DOI: 10.1038/ncomms2018. arXiv: 0402130v1 [cond-mat]. URL: <http://www.ncbi.nlm.nih.gov/pubmed/22910357>.
- [207] Dominik Lembke and Andras Kis. “Breakdown of high-performance monolayer MoS₂ transistors”. In: *ACS Nano* 6.11 (2012), pp. 10070–10075. DOI: 10.1021/nn303772b.
- [208] Kin Fai Mak et al. “Control of valley polarization in monolayer MoS₂ by optical helicity”. In: *Nature Nanotechnology* 7.8 (2012), pp. 494–498. DOI: 10.1038/nnano.2012.96.
- [209] Hyung-youl Park et al. “Wide-Range Controllable n-Doping of Molybdenum Disulfide (MoS₂) through Thermal and Optical Activation”. In: 3 (2015), pp. 2368–2376. DOI: 10.1021/acsnano.5b00153.
- [210] Hualing Zeng et al. “Valley polarization in MoS₂ monolayers by optical pumping”. In: *Nature Nanotechnology* 7.8 (June 2012), pp. 490–493. DOI: 10.1038/nnano.2012.95. URL: <http://www.nature.com/doifinder/10.1038/nnano.2012.95>.
- [211] Hee Sung Lee et al. “MoS₂ Nanosheet Phototransistors with Thickness-Modulated Optical Energy Gap”. In: *Nano Lett.* 12.7 (2012), pp. 3695–3700.
- [212] Zongyou Yin et al. “Single-Layer MoS₂ Phototransistors”. In: (2011).
- [213] Simone Bertolazzi, Jacopo Brivio, and Andras Kis. “Stretching and Breaking of Ultrathin MoS₂”. In: *ACS Nano* 5.12 (2011), pp. 9703–9709.
- [214] C. Rice et al. “Raman-scattering measurements and first-principles calculations of strain-induced phonon shifts in monolayer MoS₂”. In: *Physical Review B* 87.8 (2013), p. 081307. DOI: 10.1103/PhysRevB.87.081307. URL: <http://link.aps.org/doi/10.1103/PhysRevB.87.081307>.
- [215] Woong Choi et al. “High-Detectivity Multilayer MoS₂ Phototransistors with Spectral Response from Ultraviolet to Infrared”. In: *Advanced Materials* 24.43 (Nov. 2012), pp. 5832–5836. DOI: 10.1002/adma.201201909. URL: <http://doi.wiley.com/10.1002/adma.201201909>.

REFERENCES

- [216] R a Bromley, R B Murray, and a D Yoffe. “The band structures of some transition metal dichalcogenides. III. Group VIA: trigonal prism materials”. In: *Journal of Physics C: Solid State Physics* 5 (1972), pp. 759–778. DOI: 10.1088/0022-3719/5/7/007.
- [217] L. F. Mattheiss. “Energy bands for 2H-NbSe₂ and 2H-MoS₂”. In: *Physical Review Letters* 30.17 (Apr. 1973), pp. 784–787. DOI: 10.1103/PhysRevLett.30.784.
- [218] R. Coehoorn et al. “Electronic structure of MoSe₂, MoS₂, and WSe₂. I. Band-structure calculations and photoelectron spectroscopy”. In: *Physical Review B* 35.12 (1987), pp. 6195–6202. DOI: 10.1103/PhysRevB.35.6195.
- [219] Th Böker et al. “Band structure of (formula presented) (formula presented) and (formula presented) Angle-resolved photoelectron spectroscopy and ab initio calculations”. In: *Physical Review B - Condensed Matter and Materials Physics* 64.23 (Nov. 2001), p. 235305. DOI: 10.1103/PhysRevB.64.235305.
- [220] Alexander V. Kolobov and Junji Tominaga. *Two-Dimensional Transition-Metal Dichalcogenides*. Vol. 239. 2016. DOI: 10.1007/978-3-319-31450-1. URL: <http://link.springer.com/10.1007/978-3-319-31450-1>.
- [221] L. F. Mattheiss. “Band structures of transition-metal-dichalcogenide layer compounds”. In: *Physical Review B* 8.8 (1973), pp. 3719–3740. DOI: 10.1103/PhysRevB.8.3719.
- [222] T. Björkman et al. “Van der Waals bonding in layered compounds from advanced density-functional first-principles calculations”. In: *Physical Review Letters* 108.23 (2012), pp. 1–5. DOI: 10.1103/PhysRevLett.108.235502.
- [223] Changgu Lee et al. “Measurement of the elastic properties and intrinsic strength of monolayer graphene”. In: *Science* 321.5887 (2008), pp. 385–388. DOI: 10.1126/science.1157996.
- [224] Dimitrios G. Papageorgiou, Ian A. Kinloch, and Robert J. Young. “Mechanical properties of graphene and graphene-based nanocomposites”. In: *Progress in Materials Science* 90 (Oct. 2017), pp. 75–127. DOI: 10.1016/j.pmatsci.2017.07.004.
- [225] Jiang Pu et al. “Highly flexible MoS₂ thin-film transistors with ion gel dielectrics”. In: *Nano Letters* 12.8 (Aug. 2012), pp. 4013–4017. DOI: 10.1021/nl301335q.
- [226] Fang Wang et al. “Strain-induced phonon shifts in tungsten disulfide nanoplatelets and nanotubes”. In: *2D Materials* 4.1 (2016). DOI: doi:10.1088/2053-1583/4/1/015007. URL: <http://iopscience.iop.org/article/10.1088/2053-1583/4/1/015007/pdf>.

REFERENCES

- [227] A. M. Dadgar et al. “Strain Engineering and Raman Spectroscopy of Monolayer Transition Metal Dichalcogenides”. In: *Chemistry of Materials* 30.15 (2018), pp. 5148–5155. DOI: 10.1021/acs.chemmater.8b01672.
- [228] Sangyeon Pak et al. “Strain-Mediated Interlayer Coupling Effects on the Excitonic Behaviors in an Epitaxially Grown MoS₂/WS₂ van der Waals Heterobilayer”. In: *Nano Letters* 17.9 (2017), pp. 5634–5640. DOI: 10.1021/acs.nanolett.7b02513.
- [229] Liangbo Liang and Vincent Meunier. “First-principles Raman spectra of MoS₂, WS₂ and their heterostructures”. In: *Nanoscale* 6.10 (2014), p. 5394. DOI: 10.1039/c3nr06906k. URL: <http://xlink.rsc.org/?DOI=c3nr06906k>.
- [230] H. Terrones et al. “New First Order Raman-active Modes in Few Layered Transition Metal Dichalcogenides”. In: *Scientific Reports* 4 (2014), pp. 1–9. DOI: 10.1038/srep04215.
- [231] Yi Zhang et al. “Direct observation of the transition from indirect to direct bandgap in atomically thin epitaxial MoSe₂”. In: *Nature Nanotechnology* 9.2 (2014), pp. 111–115. DOI: 10.1038/nnano.2013.277.
- [232] Kapildeb Dolui, Ivan Rungger, and Stefano Sanvito. “Origin of the n-type and p-type conductivity of MoS₂ monolayers on a SiO₂ substrate”. In: *Physical Review B - Condensed Matter and Materials Physics* 87.16 (2013), pp. 1–7. DOI: 10.1103/PhysRevB.87.165402.
- [233] A. H. Castro Neto et al. “The electronic properties of graphene”. In: *Reviews of Modern Physics* 81.1 (2009), pp. 109–162. DOI: 10.1103/RevModPhys.81.109. URL: <http://link.aps.org/doi/10.1103/RevModPhys.81.109>?citations?view_op=view_citation&continue=/scholar?hl=en&start=100&as_sdt=0,5&scilib=1&citilm=1&citation_for_view=fLRxLRkAAAAJ:DJbc18HfkQkC&hl=en&oi=p.
- [234] Julia Gusakova et al. “Electronic Properties of Bulk and Monolayer TMDs: Theoretical Study Within DFT Framework (GVJ-2e Method)”. In: *Physica Status Solidi (A) Applications and Materials Science* 214.12 (2017), pp. 1–7. DOI: 10.1002/pssa.201700218.
- [235] Britton W.H. Baugher et al. “Intrinsic electronic transport properties of high-quality monolayer and bilayer MoS₂”. In: *Nano Letters* 13.9 (Sept. 2013), pp. 4212–4216. DOI: 10.1021/nl401916s.
- [236] Dmitry Ovchinnikov et al. “Electrical Transport Properties of Single-Layer WS₂”. In: *ACS Nano* 8.8 (Aug. 2014), pp. 8174–8181. DOI: 10.1021/nn502362b. URL: <http://pubs.acs.org/doi/abs/10.1021/nn502362b>.

REFERENCES

- [237] Francesco Reale et al. “High-Mobility and High-Optical Quality Atomically Thin WS 2”. In: *Scientific Reports* 7.1 (2017), pp. 1–10. DOI: 10.1038/s41598-017-14928-2.
- [238] Wei Zheng et al. “Patterned Growth of P-Type MoS₂ Atomic Layers Using Sol-Gel as Precursor”. In: *Advanced Functional Materials* (2016), pp. 6371–6379. DOI: 10.1002/adfm.201602494.
- [239] Andrey N Enyashin et al. “Line Defects in Molybdenum Disulfide Layers”. In: *J. Phys* 117.20 (2013), pp. 10842–10848. DOI: <https://doi.org/10.1021/jp403976d>. URL: <https://pubs.acs.org/doi/pdf/10.1021/jp403976d>.
- [240] Jason K. Ellis, Melissa J. Lucero, and Gustavo E. Scuseria. “The indirect to direct band gap transition in multilayered MoS₂ as predicted by screened hybrid density functional theory”. In: *Applied Physics Letters* 99.26 (2011). DOI: 10.1063/1.3672219.
- [241] Sefaattin Tongay et al. “Thermally Driven Crossover from Indirect toward Direct Bandgap in 2D Semiconductors: MoSe₂ versus MoS₂”. In: *Nano Letters* 12.11 (Nov. 2012), pp. 5576–5580. DOI: 10.1021/nl302584w. URL: <http://pubs.acs.org/doi/10.1021/nl302584w>.
- [242] Gmelin Handbook of Inorganic and Organometallic Chemistry. 8th ed. Vo. Springer-Verlag, Berlin, 1995.
- [243] K. K. Kam and B. A. Parkinson. “Detailed photocurrent spectroscopy of the semiconducting group VI transition metal dichalcogenides”. In: *Journal of Physical Chemistry* 86.4 (1982), pp. 463–467. DOI: 10.1021/j100393a010.
- [244] Drew W. Latzke et al. “Electronic structure, spin-orbit coupling, and interlayer interaction in bulk MoS₂ and WS₂”. In: *Physical Review B - Condensed Matter and Materials Physics* 91.23 (June 2015), p. 235202. DOI: 10.1103/PhysRevB.91.235202.
- [245] Roland Scheer and Hans-Werner Schock. *Chalcogenide photovoltaics: physics, technologies, and thin film devices*. John Wiley & Sons, 2011.
- [246] Lu Wang, Zdenek Sofer, and Martin Pumera. “Will Any Crap We Put into Graphene Increase Its Electrocatalytic Effect?” In: *ACS Nano* 14.1 (2020), pp. 21–25. DOI: 10.1021/acsnano.9b00184.
- [247] J. Derek Woolins and Risto S. Laitinen. *Selenium and Tellurium Chemistry; from small molecules to biomolecules and materials*. Vol. 53. 9. 2013, pp. 1689–1699. DOI: 10.1017/CBO9781107415324.004.
- [248] Silvan Kretschmer et al. “Supported Two-Dimensional Materials under Ion Irradiation: The Substrate Governs Defect Production”. In: *ACS Applied Materials & Interfaces* 10.36 (Sept. 2018), pp. 30827–30836. DOI: 10.1021/acsami.8b08471. URL: <http://pubs.acs.org/doi/10.1021/acsami.8b08471>.

REFERENCES

- [249] Shanshan Wang et al. “Detailed Atomic Reconstruction of Extended Line Defects in Monolayer MoS₂”. In: *ACS Nano* 10.5 (May 2016), pp. 5419–5430. DOI: 10.1021/acsnano.6b01673.
- [250] A. V. Krasheninnikov and K. Nordlund. “Ion and electron irradiation-induced effects in nanostructured materials”. In: *Journal of Applied Physics* 107.7 (Apr. 2010), p. 071301. DOI: 10.1063/1.3318261. URL: <http://aip.scitation.org/doi/10.1063/1.3318261>.
- [251] Julian Klein et al. “Atomistic defect states as quantum emitters in monolayer MoS₂”. In: *arXiv* arXiv:1901 (2019). URL: <http://arxiv.org/abs/1901.01042>.
- [252] Roland Kozubek et al. “Perforating Freestanding Molybdenum Disulfide Monolayers with Highly Charged Ions”. In: *Journal of Physical Chemistry Letters* 10.5 (2019), pp. 904–910. DOI: 10.1021/acs.jpclett.8b03666.
- [253] Marika Schleberger and Jani Kotakoski. “2D Material Science: Defect Engineering by Particle Irradiation”. In: *Materials* 11.10 (2018), p. 1885. DOI: 10.3390/ma11101885. URL: <http://www.mdpi.com/1996-1944/11/10/1885>.
- [254] Amin Azizi et al. “Defect Coupling and Sub-Angstrom Structural Distortions in W_{1-x}Mo_xS₂ Monolayers”. In: *Nano Letters* (2017), acs.nanolett.6b05045. DOI: 10.1021/acs.nanolett.6b05045. URL: <http://pubs.acs.org/doi/abs/10.1021/acs.nanolett.6b05045>.
- [255] Philipp Tonndorf et al. “Single-photon emission from localized excitons in an atomically thin semiconductor”. In: *Optica* 2.4 (2015), p. 347. DOI: 10.1364/OPTICA.2.000347. URL: <http://www.opticsinfobase.org/optica/abstract.cfm?URI=optica-2-4-347>.
- [256] M. Koperski et al. “Single photon emitters in exfoliated WSe₂ structures”. In: *Nature Nanotechnology* 10.6 (May 2015), pp. 503–506. DOI: 10.1038/nnano.2015.67. URL: <http://www.nature.com/doifinder/10.1038/nnano.2015.67>.
- [257] Johannes Kern et al. “Nanoscale Positioning of Single-Photon Emitters in Atomically Thin WSe₂”. In: *Advanced Materials* (2016), pp. 7101–7105. DOI: 10.1002/adma.201600560.
- [258] Jinhua Hong et al. “Exploring atomic defects in molybdenum disulphide monolayers”. In: *Nature Communications* 6 (2015), pp. 1–8. DOI: 10.1038/ncomms7293. URL: <http://dx.doi.org/10.1038/ncomms7293>.
- [259] Junhao Lin et al. “Flexible metallic nanowires with self-adaptive contacts to semiconducting transition-metal dichalcogenide monolayers”. In: *Nature Nanotechnology* 9.6 (2014), pp. 436–442. DOI: 10.1038/nnano.2014.81. URL: <http://dx.doi.org/10.1038/nnano.2014.81>.

REFERENCES

- [260] André K Geim and I V Grigorieva. “Van der Waals heterostructures.” In: *Nature* 499.7459 (2013), pp. 419–25. DOI: 10.1038/nature12385. URL: <http://www.ncbi.nlm.nih.gov/pubmed/23887427>.
- [261] F Withers et al. “Light-emitting diodes by band-structure engineering in van der Waals heterostructures”. In: *Nat Mater* 14.3 (2015), pp. 301–306. DOI: 10.1038/nmat4205. URL: <http://www.ncbi.nlm.nih.gov/pubmed/25643033>.
- [262] Goki Eda et al. “Coherent atomic and electronic heterostructures of single-layer MoS 2”. In: *ACS Nano* 6.8 (2012), pp. 7311–7317. DOI: 10.1021/nn302422x.
- [263] Yung-Chang Lin et al. “Atomic mechanism of phase transition between metallic and semiconducting MoS 2 single-layers”. In: *arXiv* arXiv:1310 (2013). URL: <https://arxiv.org/ftp/arxiv/papers/1310/1310.2363.pdf>.
- [264] Yulan Dong et al. “Effect of sulphur vacancy and interlayer interaction on the electronic structure and spin splitting of bilayer MoS 2”. In: *Journal of Physics: Condensed Matter* 30.12 (2018), p. 125302. DOI: 10.1088/1361-648X/aaad3b. URL: <http://stacks.iop.org/0953-8984/30/i=12/a=125302?key=crossref.97cbca84cf3a57e312b504ab36ee59bd>.
- [265] S. J. Haigh et al. “Cross-sectional imaging of individual layers and buried interfaces of graphene-based heterostructures and superlattices”. In: *Nature Materials* 11.9 (Sept. 2012), pp. 764–767. DOI: 10.1038/nmat3386. URL: <http://www.nature.com/articles/nmat3386>.
- [266] K S Novoselov et al. “2D materials and van der Waals heterostructures”. In: *Science (New York, N.Y.)* (2016). DOI: 10.1126/science.aac9439.
- [267] Amin Azizi et al. “Freestanding van der Waals Heterostructures of Graphene and Transition Metal Dichalcogenides”. In: 5 (2015), pp. 1–5. DOI: 10.1021/acsnano.5b01677.
- [268] Deep Jariwala, Tobin J. Marks, and Mark C. Hersam. “Mixed-dimensional van der Waals heterostructures”. In: *Nature Materials* 16.2 (2017), pp. 170–181. DOI: 10.1038/nmat4703. URL: <http://dx.doi.org/10.1038/nmat4703>.
- [269] C R Dean et al. “Boron nitride substrates for high-quality graphene electronics.” In: *Nature nanotechnology* 5.10 (2010), pp. 722–726. DOI: 10.1038/nnano.2010.172.
- [270] Xu Cui et al. “Low-temperature Ohmic contact to monolayer MoS₂ by van der Waals bonded Co/h-BN electrodes”. In: *Nano Letters* (2017), acs.nanolett.7b01536. DOI: 10.1021/acs.nanolett.7b01536. URL: <http://pubs.acs.org/doi/abs/10.1021/acs.nanolett.7b01536>.

REFERENCES

- [271] Roland Gillen, John Robertson, and Janina Maultzsch. “Indirect doping effects from impurities in MoS₂/BN heterostructures”. In: *Physical Review B* 90.7 (2014), p. 075437. DOI: 10.1103/PhysRevB.90.075437. URL: <http://link.aps.org/doi/10.1103/PhysRevB.90.075437>.
- [272] John Mann et al. “2-Dimensional transition metal dichalcogenides with tunable direct band gaps: MoS₂(1-x)Se_{2x} monolayers”. In: *Advanced Materials* 26.9 (2014), pp. 1399–1404. DOI: 10.1002/adma.201304389.
- [273] Xidong Duan et al. “Lateral epitaxial growth of two-dimensional layered semiconductor heterojunctions”. In: *Nature Materials* 9 (2014), pp. 1024–1030. DOI: 10.1038/NNANO.2014.222. URL: www.nature.com/naturenanotechnology.
- [274] Ruchun Li et al. “Nitrogen doped MoS₂ nanosheets synthesized via a low-temperature process as electrocatalysts with enhanced activity for hydrogen evolution reaction”. In: *Journal of Power Sources* 356 (July 2017), pp. 133–139. DOI: 10.1016/j.jpowsour.2017.04.060. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0378775317305554>.
- [275] Qingliang Feng et al. “Growth of large-area 2D MoS₂(1-x)Se_{2x} semiconductor alloys”. In: *Advanced Materials* 26.17 (2014), pp. 2648–2653. DOI: 10.1002/adma.201306095.
- [276] Ji-Young Noh et al. “Deep-to-shallow level transition of Re and Nb dopants in monolayer MoS₂ with dielectric environments”. In: *Physical Review B* 92.11 (Sept. 2015), p. 115431. DOI: 10.1103/PhysRevB.92.115431. URL: <https://link.aps.org/doi/10.1103/PhysRevB.92.115431>.
- [277] Joonki Suh et al. “Doping against the native propensity of MoS₂: Degenerate hole doping by cation substitution”. In: *Nano Letters* 14.12 (2014), pp. 6976–6982. DOI: 10.1021/nl503251h.
- [278] S. Lu et al. “Tunable redox potential of nonmetal doped monolayer MoS₂: First principle calculations”. In: *Applied Surface Science* 384 (2016), pp. 360–367. DOI: 10.1016/j.apsusc.2016.05.038. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0169433216310467>.
- [279] Xiao-Li Fan, Yu-Rong An, and Wen-Jun Guo. “Ferromagnetism in Transitional Metal-Doped MoS₂ Monolayer”. In: *Nanoscale Research Letters* 11.1 (2016), p. 154. DOI: 10.1186/s11671-016-1376-y. URL: <http://www.nanoscalereslett.com/content/11/1/154>.
- [280] Jieqiong Wang et al. “Robust ferromagnetism in Mn-doped MoS₂ nanostructures”. In: *Applied Physics Letters* 109.9 (2016), pp. 1–6. DOI: 10.1063/1.4961883. URL: <http://dx.doi.org/10.1063/1.4961883>.

REFERENCES

- [281] Yaping Miao et al. “Tuning of electronic states and magnetic polarization in monolayered MoS₂ by codoping with transition metals and nonmetals”. In: *Journal of Materials Science* 51.20 (2016), pp. 9514–9525. DOI: 10.1007/s10853-016-0195-y. URL: <http://link.springer.com/10.1007/s10853-016-0195-y>.
- [282] J. Karthikeyan et al. “Which Transition Metal Atoms Can Be Embedded into Two-Dimensional Molybdenum Dichalcogenides and Add Magnetism?” In: *Nano Letters* 19.7 (2019), pp. 4581–4587. DOI: 10.1021/acs.nanolett.9b01555.
- [283] Kapildeb Dolui et al. “Possible doping strategies for MoS₂ monolayers: An ab initio study”. In: *Physical Review B - Condensed Matter and Materials Physics* 88.7 (Aug. 2013), p. 075420. DOI: 10.1103/PhysRevB.88.075420.
- [284] Deniz Çakır, Hasan Sahin, and François M. Peeters. “Doping of rhenium disulfide monolayers: a systematic first principles study”. In: *Phys. Chem. Chem. Phys.* 16.31 (July 2014), pp. 16771–16779. DOI: 10.1039/C4CP02007C. URL: <http://xlink.rsc.org/?DOI=C4CP02007C>.
- [285] Qingyi Zeng et al. “The transport properties of the Phosphorus and Chlorine doped single layer MoS₂ pn junctions: A first-principles study”. In: *Solid State Communications* 246 (2016), pp. 82–87. DOI: 10.1016/j.ssc.2016.07.021.
- [286] Dongwei Ma et al. “The adsorption of CO and NO on the MoS₂ monolayer doped with Au, Pt, Pd, or Ni: A first-principles study”. In: *Applied Surface Science* 383 (2016), pp. 98–105. DOI: 10.1016/j.apsusc.2016.04.171. URL: <http://www.sciencedirect.com/science/article/pii/S0169433216309606>.
- [287] Sehoon Oh et al. “Stability, efficiency, and mechanism of \$n\$-type doping by hydrogen adatoms in two-dimensional transition metal dichalcogenides”. In: *PHYSICAL REVIEW B* 100 (2019), p. 85416. DOI: 10.1103/PhysRevB.100.085416. URL: <https://journals.aps.org/prb/pdf/10.1103/PhysRevB.100.085416>.
- [288] Ji Young Noh, Hanchul Kim, and Yong Sung Kim. “Stability and electronic structures of native defects in single-layer MoS₂”. In: *Physical Review B - Condensed Matter and Materials Physics* 89.20 (2014), pp. 1–12. DOI: 10.1103/PhysRevB.89.205417.
- [289] Hannu Pekka Komsa and Arkady V. Krasheninnikov. “Native defects in bulk and monolayer MoS₂ from first principles”. In: *Physical Review B - Condensed Matter and Materials Physics* 91.12 (2015), pp. 1–17. DOI: 10.1103/PhysRevB.91.125304.
- [290] Yuda Zhao et al. “Doping, Contact and Interface Engineering of Two-Dimensional Layered Transition Metal Dichalcogenides Transistors”. In: *Advanced Functional Materials* 27.19 (2017). DOI: 10.1002/adfm.201603484.

REFERENCES

- [291] Yumeng Shi et al. “Selective decoration of Au nanoparticles on monolayer MoS₂ single crystals.” In: *Scientific reports* 3 (2013), p. 1839. DOI: 10.1038/srep01839. URL: <http://www.ncbi.nlm.nih.gov/pubmed/23670611> %20<http://www.ncbi.nlm.nih.gov/articlerender.fcgi?artid=PMC3653143>.
- [292] Biswanath Chakraborty et al. “Symmetry-dependent phonon renormalization in monolayer MoS₂ transistor”. In: *RAPID COMMUNICATIONS PHYSICAL REVIEW B* 85.4 (2012), pp. 161403–161404. DOI: 10.1103/PhysRevB.85.161403. URL: <https://journals.aps.org/prb/pdf/10.1103/PhysRevB.85.161403>.
- [293] T S Sreeprasad et al. “Controlled, Defect-Guided, Metal-Nanoparticle-Incorporation onto MoS₂ via Chemical and Microwave Routes: Electrical, Thermal, and Structural Properties.” In: *Nano letters* (2013). DOI: 10.1021/nl402278y. URL: <http://www.ncbi.nlm.nih.gov/pubmed/23927716>.
- [294] Rajesh Kapper et al. “Phase-engineered low-resistance contacts for ultrathin MoS₂ transistors”. In: *Nature Materials* 13.12 (2014), pp. 1128–1134. DOI: 10.1038/nmat4080.
- [295] Seunghyun Lee et al. “Statistical Study on the Schottky Barrier Reduction of Tunneling Contacts to CVD Synthesized MoS₂”. In: *Nano Letters* 16.1 (2016), pp. 276–281. DOI: 10.1021/acs.nanolett.5b03727.
- [296] Johannes Kern et al. “Nanoantenna-Enhanced Light-Matter Interaction in Atomically Thin WS₂”. In: *ACS Photonics* 2.9 (2015), pp. 1260–1265. DOI: 10.1021/acspophotonics.5b00123.
- [297] Jie Chao Wang et al. “A MoS₂-based system for efficient immobilization of hemoglobin and biosensing applications”. In: *Nanotechnology* 26.27 (2015), p. 274005. DOI: 10.1088/0957-4484/26/27/274005. URL: <http://dx.doi.org/10.1088/0957-4484/26/27/274005>.
- [298] Ziwei Li et al. “Active Light Control of the MoS₂ Monolayer Exciton Binding Energy”. In: *ACS Nano* 9.10 (Oct. 2015), pp. 10158–10164. DOI: 10.1021/acsnano.5b03764. URL: <http://pubs.acs.org/doi/10.1021/acsnano.5b03764>.
- [299] Denis G Baranov et al. “Novel Nanostructures and Materials for Strong Light-Matter Interactions”. In: *ACS Photonics* 5.1 (2017), pp. 24–42. DOI: 10.1021/acspophotonics.7b00674. URL: <https://pubs.acs.org/sharingguidelines>.
- [300] Serkan Butun, Sefaattin Tongay, and Koray Aydin. “Enhanced Light Emission from Large-Area Monolayer MoS₂ Using Plasmonic Nanodisc Arrays”. In: *Nano Letters* 15.4 (2015), pp. 2700–2704. DOI: 10.1021/acs.nanolett.5b00407.

REFERENCES

- [301] Bumsu Lee et al. “Fano resonance and spectrally modified photoluminescence enhancement in monolayer MoS₂ integrated with plasmonic nanoantenna array”. In: *Nano Letters* 15.5 (2015), pp. 3646–3653. DOI: 10.1021/acs.nanolett.5b01563.
- [302] I. Abid et al. “Resonant surface plasmon-exciton interaction in hybrid MoSe₂@Au nanostructures”. In: *Nanoscale* 8.15 (2016), pp. 8151–8159. DOI: 10.1039/c6nr00829a.
- [303] Xu Zhao et al. “Reversible and fast Na-ion storage in MoO₂/MoSe₂ heterostructures for high energy-high power Na-ion capacitors”. In: *Energy Storage Materials* 12 (May 2018), pp. 241–251. DOI: 10.1016/j.ensm.2017.12.015. URL: <https://www.sciencedirect.com/science/article/pii/S2405829717304063?via%3Dihub>.
- [304] J.S. Williams. “Ion implantation of semiconductors”. In: *Materials Science and Engineering: A* 253.1-2 (Sept. 1998), pp. 8–15. DOI: 10.1016/S0921-5093(98)00705-9. URL: <https://www.sciencedirect.com/science/article/pii/S0921509398007059>.
- [305] Jiali Li et al. “Ion-beam sculpting at nanometre length scales”. In: *Nature* 412.6843 (July 2001), pp. 166–169. DOI: 10.1038/35084037. URL: <http://www.nature.com/articles/35084037>.
- [306] M. Uhrmacher and H. Hofsäss. “Ion accelerator facilities at the University of Göttingen”. In: *Nuclear Instruments and Methods in Physics Research, Section B: Beam Interactions with Materials and Atoms*. Vol. 240. 1-2. Oct. 2005, pp. 48–54. DOI: 10.1016/j.nimb.2005.06.087.
- [307] E. H. Åhlgren, J. Kotakoski, and a. V. Krasheninnikov. “Atomistic simulations of the implantation of low-energy boron and nitrogen ions into graphene”. In: *Physical Review B - Condensed Matter and Materials Physics* 83.11 (2011), pp. 1–7. DOI: 10.1103/PhysRevB.83.115424.
- [308] Philip Willke et al. “Doping of Graphene by Low-Energy Ion Beam Implantation: Structural, Electronic, and Transport Properties”. In: *Nano Letters* 15.8 (2015), pp. 5110–5115. DOI: 10.1021/acs.nanolett.5b01280.
- [309] E. H. Åhlgren et al. “Ion irradiation tolerance of graphene as studied by atomistic simulations”. In: *Applied Physics Letters* 100.23 (June 2012), p. 233108. DOI: 10.1063/1.4726053. URL: <http://aip.scitation.org/doi/10.1063/1.4726053>.
- [310] Yijun Xu et al. “Investigation of the effect of low energy ion beam irradiation on mono-layer graphene”. In: *AIP Advances* 3.7 (July 2013), p. 072120. DOI: 10.1063/1.4816715. URL: <http://aip.scitation.org/doi/10.1063/1.4816715>.
- [311] Beidou Guo et al. “Controllable N-Doping of Graphene”. In: *Nano Letters* 10.12 (2010), pp. 4975–4980. DOI: 10.1021/nl103079j. URL: <http://pubs.acs.org/doi/abs/10.1021/nl103079j>.

REFERENCES

- [312] O. Lehtinen et al. “Effects of ion bombardment on a two-dimensional target: Atomistic simulations of graphene irradiation”. In: *Physical Review B* 81.15 (2010), p. 153401. DOI: 10.1103/PhysRevB.81.153401. URL: <http://link.aps.org/doi/10.1103/PhysRevB.81.153401>.
- [313] Hongtao Wang et al. “Doping Monolayer Graphene with Single Atom Substitutions”. In: *Nano Letters* 12.1 (Jan. 2012), pp. 141–144. DOI: 10.1021/nl2031629. URL: <https://pubs.acs.org/doi/10.1021/nl2031629>.
- [314] E. H. Åhlgren et al. “Structural manipulation of the graphene/metal interface with Ar+ irradiation”. In: *Physical Review B - Condensed Matter and Materials Physics* 88.15 (2013), pp. 1–5. DOI: 10.1103/PhysRevB.88.155419.
- [315] M. M. Ugeda et al. “Point defects on graphene on metals”. In: *Physical Review Letters* 107.11 (2011), pp. 1–5. DOI: 10.1103/PhysRevLett.107.116803.
- [316] Miguel M. Ugeda et al. “Electronic and structural characterization of divacancies in irradiated graphene”. In: *Physical Review B* 85.12 (Mar. 2012), p. 121402. DOI: 10.1103/PhysRevB.85.121402. URL: <https://link.aps.org/doi/10.1103/PhysRevB.85.121402>.
- [317] Janine Schwestka et al. “A versatile ion beam spectrometer for studies of ion interaction with 2D materials”. In: *Review of Scientific Instruments* 89.8 (2018). DOI: 10.1063/1.5037798.
- [318] Charlotte Herbig et al. “Xe irradiation of graphene on Ir(111): From trapping to blistering”. In: *Physical Review B* 92.8 (Aug. 2015), p. 085429. DOI: 10.1103/PhysRevB.92.085429. URL: <https://link.aps.org/doi/10.1103/PhysRevB.92.085429>.
- [319] J. Kotakoski et al. “Irradiation-assisted substitution of carbon atoms with nitrogen and boron in single-walled carbon nanotubes”. In: *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms* 228.1-4 (Jan. 2005), pp. 31–36. DOI: 10.1016/J.NIMB.2004.10.018. URL: <https://www.sciencedirect.com/science/article/pii/S0168583X04010791?via%3Dihub>.
- [320] J. Kotakoski et al. “B and N ion implantation into carbon nanotubes: Insight from atomistic simulations”. In: *Physical Review B* 71.20 (May 2005), p. 205408. DOI: 10.1103/PhysRevB.71.205408. URL: <https://link.aps.org/doi/10.1103/PhysRevB.71.205408>.
- [321] A. Tolvanen et al. “Relative abundance of single and double vacancies in irradiated single-walled carbon nanotubes”. In: *Applied Physics Letters* 91.17 (Oct. 2007), p. 173109. DOI: 10.1063/1.2800807. URL: <http://aip.scitation.org/doi/10.1063/1.2800807>.

REFERENCES

- [322] A. V. Krasheninnikov et al. “Stability of carbon nanotubes under electron irradiation: Role of tube diameter and chirality”. In: *Physical Review B* 72.12 (Sept. 2005), p. 125428. DOI: 10.1103/PhysRevB.72.125428. URL: <https://link.aps.org/doi/10.1103/PhysRevB.72.125428>.
- [323] A. Zobelli et al. “Electron knock-on cross section of carbon and boron nitride nanotubes”. In: *Physical Review B* 75.24 (June 2007), p. 245402. DOI: 10.1103/PhysRevB.75.245402. URL: <https://link.aps.org/doi/10.1103/PhysRevB.75.245402>.
- [324] Xiaofei Liu et al. “Top-down fabrication of sub-nanometre semiconducting nanoribbons derived from molybdenum disulfide sheets”. In: *Nature Communications* 4 (2013), p. 1776. DOI: 10.1038/ncomms2803. URL: <http://dx.doi.org/10.1038/ncomms2803>.
- [325] Ankur Nipane et al. “Few-Layer MoS₂ p-Type Devices Enabled by Selective Doping Using Low Energy Phosphorus Implantation”. In: *ACS Nano* 10.2 (2016), pp. 2128–2137. DOI: 10.1021/acsnano.5b06529.
- [326] Ziqi Li and Feng Chen. “Ion beam modification of two-dimensional materials: Characterization, properties, and applications”. In: *Applied Physics Reviews* 4.1 (Mar. 2017), p. 011103. DOI: 10.1063/1.4977087. URL: <http://aip.scitation.org/doi/10.1063/1.4977087>.
- [327] Lukas Kuerten et al. “Local control of improper ferroelectric domains in YMnO₃”. In: *arXiv* (2020). URL: <http://arxiv.org/abs/2004.06339>.
- [328] Bas B. Van Aken et al. “The origin of ferroelectricity in magnetoelectric YMnO₃”. In: *Nature Materials* 3.3 (2004), pp. 164–170. DOI: 10.1038/nmat1080.
- [329] S. Stemmer et al. “Atomistic structure of 90° domain walls in ferroelectric Pb-TiO₃ thin films”. In: *Philosophical Magazine A: Physics of Condensed Matter, Structure, Defects and Mechanical Properties* 71.3 (1995), pp. 713–724. DOI: 10.1080/01418619508244477.
- [330] C. L. Jia, M. Lentzen, and K. Urban. “Atomic-resolution imaging of oxygen in perovskite ceramics”. In: *Science* 299.5608 (2003), pp. 870–873. DOI: 10.1126/science.1079121.
- [331] Doron Shilo, Guruswami Ravichandran, and Kaushik Bhattacharya. “Investigation of twin-wall structure at the nanometre scale using atomic force microscopy”. In: *Nature Materials* 3.7 (June 2004), pp. 453–457. DOI: 10.1038/nmat1151.
- [332] Julia A. Mundy et al. “Atomically engineered ferroic layers yield a room-temperature magnetoelectric multiferroic”. In: *Nature* 537.7621 (2016), pp. 523–527. DOI: 10.1038/nature19343. URL: <http://dx.doi.org/10.1038/nature19343>.

REFERENCES

- [333] Y. L. Tang et al. “Observation of a periodic array of flux-closure quadrants in strained ferroelectric PbTiO₃ films”. In: *Science* 348.6234 (2015), pp. 547–551. DOI: 10.1126/science.1259869.
- [334] Megan E. Holtz et al. “Topological Defects in Hexagonal Manganites: Inner Structure and Emergent Electrostatics”. In: *Nano Letters* 17.10 (2017), pp. 5883–5890. DOI: 10.1021/acs.nanolett.7b01288.
- [335] K. Du et al. “Manipulating topological transformations of polar structures through real-time observation of the dynamic polarization evolution”. In: *Nature Communications* 10.1 (2019), pp. 1–8. DOI: 10.1038/s41467-019-12864-5.
- [336] Chun Lin Jia et al. “Atomic-scale study of electric dipoles near charged and uncharged domain walls in ferroelectric films”. In: *Nature Materials* 7.1 (2008), pp. 57–61. DOI: 10.1038/nmat2080.
- [337] Julie Gonnissen et al. “Direct Observation of Ferroelectric Domain Walls in LiNbO₃: Wall-Meanders, Kinks, and Local Electric Charges”. In: *Advanced Functional Materials* 26.42 (2016), pp. 7599–7604. DOI: 10.1002/adfm.201603489.
- [338] A. K. Yadav et al. “Observation of polar vortices in oxide superlattices”. In: *Nature* 530.7589 (2016), pp. 198–201. DOI: 10.1038/nature16463.
- [339] Haidong Lu et al. “Electrical Tunability of Domain Wall Conductivity in LiNbO₃ Thin Films”. In: *Advanced Materials* 31.48 (2019), pp. 1–7. DOI: 10.1002/adma.201902890.
- [340] Julie Gonnissen et al. “SUP - Direct Observation of Ferroelectric Domain Walls in LiNbO₃: Wall-Meanders, Kinks, and Local Electric Charges”. In: *Advanced Functional Materials* 26.42 (2016), pp. 7599–7604. DOI: 10.1002/adfm.201603489.
- [341] F. T. Huang et al. “Delicate balance between ferroelectricity and antiferroelectricity in hexagonal InMnO₃”. In: *Physical Review B - Condensed Matter and Materials Physics* 87.18 (2013). DOI: 10.1103/PhysRevB.87.184109.
- [342] Benjamin H. Savitzky et al. “Bending and breaking of stripes in a charge ordered manganite”. In: *Nature Communications* 8.1 (2017), pp. 1–6. DOI: 10.1038/s41467-017-02156-1. URL: <http://dx.doi.org/10.1038/s41467-017-02156-1>.
- [343] Shaobo Cheng et al. “Interface reconstruction with emerging charge ordering in hexagonal manganite”. In: *Science Advances* 4.5 (2018), pp. 1–7. DOI: 10.1126/sciadv.aar4298.
- [344] Ian MacLaren and Quentin M. Ramasse. “Aberration-corrected scanning transmission electron microscopy for atomic-resolution studies of functional oxides”. In: *International Materials Reviews* 59.3 (2014), pp. 115–131. DOI: 10.1179/1743280413Y.0000000026.

REFERENCES

- [345] K. Moore et al. “Rapid Fourier Masked Domain Mapping to Reveal Head to Head Charged Domain Walls in Lead Titanate”. In: *Microscopy and Microanalysis* 25.S2 (2019), pp. 974–975. DOI: 10.1017/s1431927619005609.
- [346] K. Moore, M. Conroy, and U. Bangert. “Rapid polarization mapping in ferroelectrics using Fourier masking”. In: *Journal of Microscopy* (2020). DOI: 10.1111/jmi.12876.
- [347] James P. V. McConville et al. “Ferroelectric Domain Wall Memristor”. In: *Advanced Functional Materials* (2019), pp. 1–577. DOI: 10.1002/9781118998977.
- [348] Raymond G.P. McQuaid et al. “Injection and controlled motion of conducting domain walls in improper ferroelectric Cu-Cl boracite”. In: *Nature Communications* 8 (May 2017). DOI: 10.1038/ncomms15105.
- [349] K. S. Novoselov. “Nobel Lecture: Graphene: Materials in the Flatland”. In: *Reviews of Modern Physics* 83.3 (Aug. 2011), pp. 837–849. DOI: 10.1103/RevModPhys.83.837. URL: <https://link.aps.org/doi/10.1103/RevModPhys.83.837>.
- [350] A.J.M. Giesbers et al. “Nanolithography and manipulation of graphene using an atomic force microscope”. In: *Solid State Communications* 147.9-10 (Sept. 2008), pp. 366–369. DOI: 10.1016/J.SSC.2008.06.027. URL: <https://www.sciencedirect.com/science/article/pii/S0038109808003608>.
- [351] G. Deokar et al. “Towards high quality CVD graphene growth and transfer”. In: *Carbon* 89 (Aug. 2015), pp. 82–92. DOI: 10.1016/J.CARBON.2015.03.017. URL: <https://www.sciencedirect.com/science/article/pii/S0008622315002134>.
- [352] Roberto Muñoz and Cristina Gómez-Aleixandre. “Review of CVD Synthesis of Graphene”. In: *Chemical Vapor Deposition* 19.10-11-12 (Dec. 2013), pp. 297–322. DOI: 10.1002/cvde.201300051. URL: <http://doi.wiley.com/10.1002/cvde.201300051>.
- [353] Xuesong Li et al. “Large-area synthesis of high-quality and uniform graphene films on copper foils.” In: *Science (New York, N.Y.)* 324.5932 (June 2009), pp. 1312–4. DOI: 10.1126/science.1171245. URL: <http://www.ncbi.nlm.nih.gov/pubmed/19423775>.
- [354] Yi-Hsien Lee et al. “Synthesis of Large-Area MoS₂ Atomic Layers with Chemical Vapor Deposition”. In: *Advanced Materials* 24.17 (May 2012), pp. 2320–2325. DOI: 10.1002/adma.201104798. URL: <http://doi.wiley.com/10.1002/adma.201104798>.
- [355] Yenny Hernandez et al. “High-yield production of graphene by liquid-phase exfoliation of graphite”. In: *Nature Nanotechnology* 3.9 (Sept. 2008), pp. 563–568. DOI: 10.1038/nnano.2008.215. URL: <http://www.nature.com/articles/nnano.2008.215>.

REFERENCES

- [356] Jonathan N Coleman et al. “Two-dimensional nanosheets produced by liquid exfoliation of layered materials.” In: *Science (New York, N.Y.)* 331.6017 (2011), pp. 568–571. DOI: 10.1126/science.1194975.
- [357] Keith R. Paton et al. “Scalable production of large quantities of defect-free few-layer graphene by shear exfoliation in liquids”. In: *Nature Materials* 13.6 (June 2014), pp. 624–630. DOI: 10.1038/nmat3944. URL: <http://www.nature.com/articles/nmat3944>.
- [358] Eswaraiah Varlla et al. “Large-Scale Production of Size-Controlled MoS₂ Nanosheets by Shear Exfoliation”. In: *Chemistry of Materials* 27.3 (Feb. 2015), pp. 1129–1139. DOI: 10.1021/cm5044864. URL: <https://pubs.acs.org/doi/10.1021/cm5044864>.
- [359] Edwin W. Lee et al. “Growth and electrical characterization of two-dimensional layered MoS₂/SiC heterojunctions”. In: *Applied Physics Letters* 105.20 (Nov. 2014), p. 203504. DOI: 10.1063/1.4901048. URL: <http://aip.scitation.org/doi/10.1063/1.4901048>.
- [360] Gustavo M. Morales et al. “High-quality few layer graphene produced by electrochemical intercalation and microwave-assisted expansion of graphite”. In: *Carbon* 49.8 (July 2011), pp. 2809–2816. DOI: 10.1016/J.CARBON.2011.03.008. URL: <https://www.sciencedirect.com/science/article/pii/S0008622311001849>.
- [361] Wonbong Choi et al. “Synthesis of Graphene and Its Applications: A Review”. In: *Critical Reviews in Solid State and Materials Sciences* 35.1 (Feb. 2010), pp. 52–71. DOI: 10.1080/10408430903505036. URL: <http://www.tandfonline.com/doi/abs/10.1080/10408430903505036>.
- [362] Minzhen Cai et al. “Methods of graphite exfoliation”. In: *Journal of Materials Chemistry* 22.48 (Nov. 2012), p. 24992. DOI: 10.1039/c2jm34517j. URL: <http://xlink.rsc.org/?DOI=c2jm34517j>.
- [363] Andrew J. Mannix et al. “Synthesis and chemistry of elemental 2D materials”. In: *Nature Reviews Chemistry* 1.2 (Feb. 2017), p. 0014. DOI: 10.1038/s41570-016-0014. URL: <http://www.nature.com/articles/s41570-016-0014>.
- [364] Joseph Halim et al. “Synthesis and Characterization of 2D Molybdenum Carbide (MXene)”. In: *Advanced Functional Materials* 26.18 (May 2016), pp. 3118–3127. DOI: 10.1002/adfm.201505328. URL: <http://doi.wiley.com/10.1002/adfm.201505328>.
- [365] Grégory F. Schneider et al. “Wedging transfer of nanostructures”. In: *Nano Letters* 10.5 (2010), pp. 1912–1916. DOI: 10.1021/nl1008037.
- [366] P. J. Zomer et al. “A transfer technique for high mobility graphene devices on commercially available hexagonal boron nitride”. In: *Applied Physics Letters* 99.23 (2011), pp. 98–101. DOI: 10.1063/1.3665405.

REFERENCES

- [367] Andres Castellanos-Gomez et al. “Deterministic transfer of two-dimensional materials by all-dry viscoelastic stamping”. In: *2D Materials* 1.1 (2014), p. 011002. DOI: 10.1088/2053-1583/1/1/011002. URL: <http://stacks.iop.org/2053-1583/1/i=1/a=011002?key=crossref.4a11185bed3e80db2f125d0422bf4c4b>.
- [368] Andres Castellanos-Gomez et al. “Supplementary - Deterministic transfer of two-dimensional materials by all-dry viscoelastic stamping”. In: *2D Materials* 1.1 (2014), p. 011002. DOI: 10.1088/2053-1583/1/1/011002.
- [369] Changgu Lee et al. “Anomalous lattice vibrations of single-and few-layer MoS₂”. In: *ACS nano* 4.5 (2010), pp. 2695–700. DOI: 10.1021/nn1003937. URL: <http://www.ncbi.nlm.nih.gov/pubmed/20392077%5Cnhttp://pubs.acs.org/doi/abs/10.1021/nn1003937>.
- [370] Florian Winkler. “Masters Thesis - Structural and Electronic Characterization of Two-Dimensional Layers of WSe₂”. In: (2014).
- [371] M. R. Müller et al. “Visibility of two-dimensional layered materials on various substrates”. In: *Journal of Applied Physics* 118.14 (2015), p. 145305. DOI: 10.1063/1.4930574. URL: <http://scitation.aip.org/content/aip/journal/jap/118/14/10.1063/1.4930574>.
- [372] U Bangert et al. “Scanning Transmission Electron Microscopy and Spectroscopy of Suspended Graphene”. In: *Physics and Applications of Graphene - Experiments*. Ed. by Sergey Mikhailov. InTech, 2011. Chap. 15, pp. 337–408. URL: <http://www.intechopen.com/books/physics-and-applications-of-graphene-experiments/scanning-transmission-%20electron-microscopy-and-spectroscopy-of-suspended-graphene>.
- [373] Ted Pella. *QUANTIFOIL TEM Substrates*. URL: https://www.tedpella.com/calibration_html/QUANTIFOIL_TEM_Substrates.htm.
- [374] HQ Graphene. *2D Transfer Apparatus*. 2020. URL: <http://www.hqgraphene.com/TransferSystem.php>.
- [375] Reuven Y. Rubinstein and Dirk P. Kroese. *Simulation and the Monte Carlo Method (Wiley Series in Probability and Statistics)*. 2nd ed. 2007. DOI: 10.5555/1349778.
- [376] Blueprint and Gatan. *STEM EELS Detectors*. 2020. URL: <https://www.bluescientific.com/nordic-products/gatan/advanced-stem-detectors/>.
- [377] Sergei Lopatin et al. “Optimization of Monochromated TEM for Ultimate Resolution Imaging and Ultrahigh Resolution Electron Energy Loss Spectroscopy”. In: *Ultramicroscopy* 184 (2017), pp. 109–115. DOI: 10.1016/j.ultramic.2017.08.016. URL: <http://linkinghub.elsevier.com/retrieve/pii/S0304399117302073>.

REFERENCES

- [378] Y. Zhu et al. “Imaging single atoms using secondary electrons with an aberration-corrected electron microscope”. In: *Nature Materials* 8.10 (2009), pp. 808–812. DOI: 10.1038/nmat2532.
- [379] H. G. Brown, A. J. D’Alfonso, and L. J. Allen. “Secondary electron imaging at atomic resolution using a focused coherent electron probe”. In: *Physical Review B - Condensed Matter and Materials Physics* 87.5 (2013), pp. 1–8. DOI: 10.1103/PhysRevB.87.054102.
- [380] J. Barthel. “Dr. Probe: A software for high-resolution STEM image simulation”. In: *Ultramicroscopy* 193 (Oct. 2018), pp. 1–11. DOI: 10.1016/j.ultramic.2018.06.003. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0304399118301402>.
- [381] P. Goodman and A. F. Moodie. “Numerical evaluations of N-beam wave functions in electron scattering by the multi-slice method”. In: *Acta Crystallographica Section A* 30.2 (1974), pp. 280–290. DOI: 10.1107/S056773947400057X.
- [382] R. F. Loane, P. Xu, and J. Silcox. “Thermal vibrations in convergent-beam electron diffraction”. In: *Acta Crystallographica Section A* 47.3 (1991), pp. 267–278. DOI: 10.1107/S0108767391000375.
- [383] David A. Muller et al. “Simulation of thermal diffuse scattering including a detailed phonon dispersion curve”. In: *Ultramicroscopy* 86 (2001), pp. 371–380. URL: <http://eprints.iisc.ernet.in/20294/>.
- [384] Colin Ophus. *Private Communication*. 2018.
- [385] C. B. Boothroyd. “Why don’t high-resolution simulations and images match?” In: *Journal of Microscopy* 190.1-2 (Apr. 1998), pp. 99–108. DOI: 10.1046/j.1365-2818.1998.2910843.x. URL: <http://doi.wiley.com/10.1046/j.1365-2818.1998.2910843.x>.
- [386] Colin Ophus, Jim Ciston, and Chris T. Nelson. *Correcting nonlinear drift distortion of scanning probe and scanning transmission electron microscopies from image pairs with orthogonal scan directions*. 2016. DOI: 10.1016/j.ultramic.2015.12.002.
- [387] Benjamin H. Savitzky et al. “Image registration of low signal-to-noise cryo-STEM data”. In: *Ultramicroscopy* 191 (Aug. 2018), pp. 56–65. DOI: 10.1016/j.ultramic.2018.04.008. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0304399117304369>.
- [388] Manuel Guizar-Sicairos, Samuel T. Thurman, and James R. Fienup. “Efficient subpixel image registration algorithms”. In: *Optics Letters* 33.2 (Jan. 2008), p. 156. DOI: 10.1364/ol.33.000156.
- [389] E L Lehmann and George Casella Springer. *Theory of Point Estimation, Second Edition*. 1999.

REFERENCES

- [390] Zhou Wang et al. *Image Quality Assessment: From Error Visibility to Structural Similarity*. Tech. rep. 4. 2004. URL: [http://www.cns.nyu.edu/~lcv/ssim/..](http://www.cns.nyu.edu/~lcv/ssim/)
- [391] Alireza Nasiri Avanaki. “Exact Global Histogram Specification Optimized for Structural Similarity”. In: *Optical Review* 16.6 (2009), pp. 613–621.
- [392] Lewys Jones et al. “Smart Align—a new tool for robust non-rigid registration of scanning microscope data”. In: *Advanced Structural and Chemical Imaging* 1.1 (2015), p. 8. DOI: 10.1186/s40679-015-0008-4. URL: <http://www.ascimaging.com/content/1/1/8>.
- [393] Eoghan O’Connell. *Add a reduction operation for profile_line #4206*. 2019. URL: <https://github.com/scikit-image/scikit-image/pull/4206>.
- [394] Eoghan O’Connell. *scikit-image commit e75a94a54b37e7bda8586b255c74dba3d36de8ab*. 2019.
- [395] Stuart Berg et al. “Ilastik: Interactive Machine Learning for (Bio)Image Analysis”. In: *Nature Methods* 16.December (2019). DOI: 10.1038/s41592-019-0582-9. URL: <http://dx.doi.org/10.1038/s41592-019-0582-9>.
- [396] Christoph Sommer et al. “Ilastik: Interactive learning and segmentation toolkit”. In: *Proceedings - International Symposium on Biomedical Imaging* (2011), pp. 230–233. DOI: 10.1109/ISBI.2011.5872394.
- [397] Koichi Momma and Fujio Izumi. “VESTA 3 for three-dimensional visualization of crystal, volumetric and morphology data”. In: *Journal of Applied Crystallography* 44.6 (2011), pp. 1272–1276. DOI: 10.1107/S0021889811038970.
- [398] Ask Hjorth Larsen et al. “The atomic simulation environment-a Python library for working with atoms Related content ATK-forceField: a new generation molecular dynamics software package”. In: *Journal of Physics: Condensed Matter TOPICAL REVIEW* 29 (2017). DOI: 10.1088/1361-648X/aa680e. URL: <https://doi.org/10.1088/1361-648X/aa680e>.
- [399] Earl J Kirkland. “Image Simulation in Transmission Electron Microscopy”. In: *Simulation* (2006), pp. 1–14.
- [400] Nicola A Spaldin. “A beginner’s guide to the modern theory of polarization”. In: *Journal of Solid State Chemistry* 195 (2012), pp. 2–10. URL: https://www.youtube.com/channel/UC_7bwXJs8-ayCatrEhzgeEg.
- [401] Aapo Hyvärinen and Erkki Oja. “A Fast Fixed-Point Algorithm for Independent Component Analysis”. In: *Neural Computation* 9.7 (Oct. 1997), pp. 1483–1492. DOI: 10.1162/neco.1997.9.7.1483. URL: <http://www.mitpressjournals.org/doi/10.1162/neco.1997.9.7.1483>.
- [402] David Mitchell. *DM Scripting*. 2020. URL: <http://www.dmscripting.com/>.
- [403] ACEMN. *DM Script Database*. 2019.

REFERENCES

- [404] Eileen Courtney. *Private Communication*. 2019.
- [405] E.N. O'Connell. *atomap-commit 5ba307b5*. 2019. URL: https://gitlab.com/atomap/atomap/-/merge_requests/58/commits.
- [406] E.N. O'Connell. *Atomap-issue-z-coord*. 2019. URL: <https://gitlab.com/atomap/atomap/-/issues/93>.
- [407] Johannes Schindelin et al. “The ImageJ ecosystem: An open platform for biomedical image analysis”. In: *Molecular Reproduction and Development* 82.7-8 (2015), pp. 518–529. DOI: [10.1002/mrd.22489](https://doi.org/10.1002/mrd.22489).
- [408] Thomas Kluyver et al. “Jupyter Notebooks—a publishing format for reproducible computational workflows”. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas* (2016), pp. 87–90. DOI: [10.3233/978-1-61499-649-1-87](https://doi.org/10.3233/978-1-61499-649-1-87).
- [409] B. P. Abbott et al. “Observation of Gravitational Waves from a Binary Black Hole Merger”. In: *Physical Review Letters* 116.6 (Feb. 2016), p. 061102. DOI: [10.1103/PhysRevLett.116.061102](https://doi.org/10.1103/PhysRevLett.116.061102). URL: <https://link.aps.org/doi/10.1103/PhysRevLett.116.061102>.
- [410] Jkanner; and Ajwcaltech; *LOSC_Event_tutorial*. 2017. URL: https://github.com/losc-tutorial/LOSC_Event_tutorial.
- [411] Bernhard Schaffer, Werner Grogger, and Gerald Kothleitner. “Automated spatial drift correction for EFTEM image series”. In: *Ultramicroscopy* 102.1 (Dec. 2004), pp. 27–36. DOI: [10.1016/j.ultramic.2004.08.003](https://doi.org/10.1016/j.ultramic.2004.08.003).
- [412] EN O'Connell. *Atomap Extension - Issue*. 2019. URL: <https://gitlab.com/atomap/atomap/issues/93>.
- [413] A. Nalin Mehta et al. *Structural and low-loss characterization of synthetic 2D-TMDs*. Tech. rep. 2017.
- [414] Hannah Catherine Nerl et al. “Probing the local nature of excitons and plasmons in few-layer MoS₂”. In: *npj 2D Materials and Applications* 1.1 (2017), p. 2. DOI: [10.1038/s41699-017-0003-9](https://doi.org/10.1038/s41699-017-0003-9). URL: <http://www.nature.com/articles/s41699-017-0003-9>.
- [415] A S Barnard. “Modelling of nanoparticles: approaches to morphology and evolution”. In: *Reports on Progress in Physics* 73.8 (Aug. 2010), p. 086502. DOI: [10.1088/0034-4885/73/8/086502](https://doi.org/10.1088/0034-4885/73/8/086502). URL: <http://stacks.iop.org/0034-4885/73/i=8/a=086502?key=crossref.50494ec20b122d99c1f571b6aae55855>.
- [416] Hui Li et al. “Magic-Number Gold Nanoclusters with Diameters from 1 to 3.5 nm: Relative Stability and Catalytic Activity for CO Oxidation”. In: *Nano Letters* 15.1 (Jan. 2015), pp. 682–688. DOI: [10.1021/nl504192u](https://doi.org/10.1021/nl504192u). URL: <http://pubs.acs.org/doi/10.1021/nl504192u>.

REFERENCES

- [417] Hannu Pekka Komsa and Arkady V Krasheninnikov. “Effects of confinement and environment on the electronic structure and exciton binding energy of MoS₂ from first principles”. In: *Physical Review B - Condensed Matter and Materials Physics* 86.24 (2012), p. 241201. DOI: 10.1103/PhysRevB.86.241201. URL: <https://journals.aps.org/prb/pdf/10.1103/PhysRevB.86.241201>.
- [418] Andrew Stevens et al. “The potential for Bayesian compressive sensing to significantly reduce electron dose in high-resolution STEM images”. In: *Microscopy* 63.1 (Feb. 2014), pp. 41–51. DOI: 10.1093/jmicro/dft042.
- [419] Nicola A. Spaldin. “Multiferroics: Past, present, and future”. In: *MRS Bulletin* 42.5 (2017), pp. 385–389. DOI: 10.1557/mrs.2017.86.
- [420] G. Catalan et al. “Domain wall nanoelectronics”. In: *Reviews of Modern Physics* 84.1 (2012), pp. 119–156. DOI: 10.1103/RevModPhys.84.119.
- [421] J. P. Velev, S. S. Jaswal, and E. Y. Tsymbal. “Multi-ferroic and magnetoelectric materials and interfaces”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 369.1948 (2011), pp. 3069–3097. DOI: 10.1098/rsta.2010.0344.

Appendix A

Appendix

A.1 Literature Review

A.1.1 Ferroic Domains

A multiferroic material is one which exhibits two or more ferroic properties: ferroelasticity, ferroelectricity and ferromagnetism [419]. These ferroic properties are referred to as primary order parameters in multiferroics [421]. Primary order parameters can point in one of two directions, also known as polarities. The order parameters are strain for ferroelastics, polarisation for ferroelectric, and magnetism for ferromagnets. The order parameters are stable when the sample is not exposed to external electric or magnetic fields, but their polarities can be switched by applying such a field [420]. Theoretically, each polarity is energetically equivalent and therefore has equal probability of appearing in the material. Domains are formed when each polarity forms in separate regions. At the border between the domains lie the domain boundaries, also known as domain walls (DWs).

A.2 Methodology

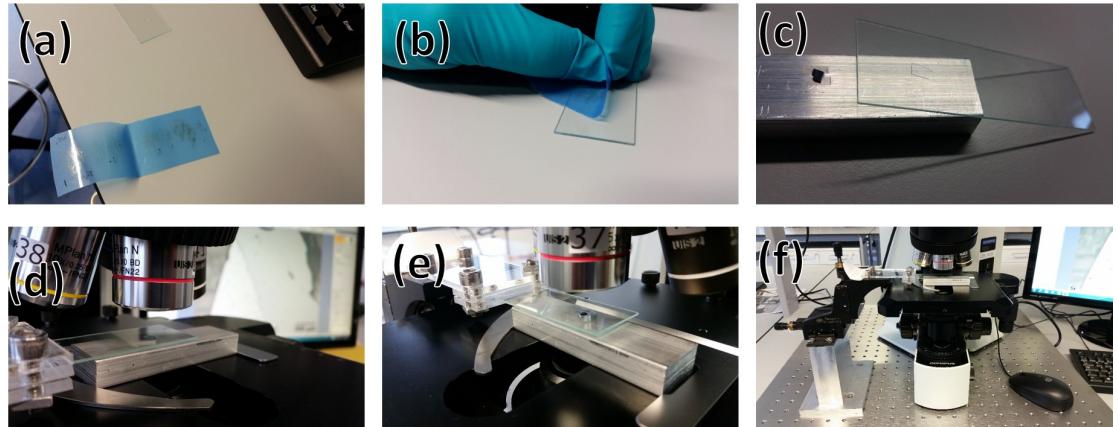


Figure A.1: (a) Exfoliation of a 2D single crystal with elastomer tape. (b) Stamping of the 2D flake onto PDMS (or another) polymer for ‘dry-stamping’. (c) Metal block with a acceptor substrate, usually a 3 mm TEM grid (here a piece of Si wafer is shown). On the right is the glass slide/polymer/2D flake stack. (d) Stamping of the 2D flake onto the acceptor substrate under the microscope, with the viscoelastic contact effect visible on the background screen. (e) Alternative view of (d). (f) Overview of the entire stamping set-up.



Figure A.2: (a) Wafer spin-coat holder designed and used during the project. (b) Thermal tap/o-ring/polymer stack after cutting and fishing the polymer from the Si/SiO₂ substrate. (c) Searching for the transferred flake under an optical microscope. This metal slide is then placed upside-down over the TEM grid and stamped.

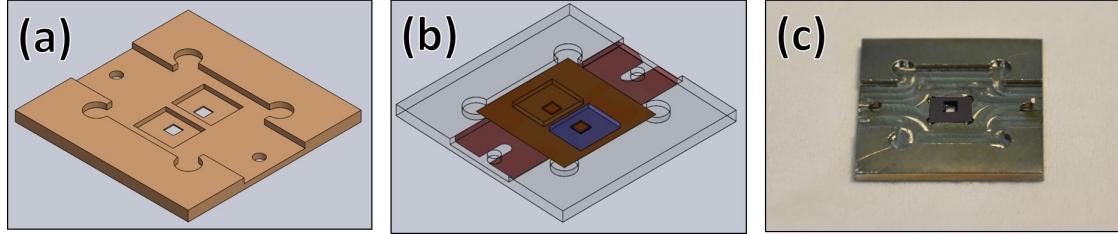


Figure A.3: (a) Solidworks view of the base-plate for a double DENS in-situ heating chip. (b) Solidworks view of the two clamp-plates with a DENS in-situ TEM grid in place (blue). (c) Machined single DENS in-situ ion implanter holder.

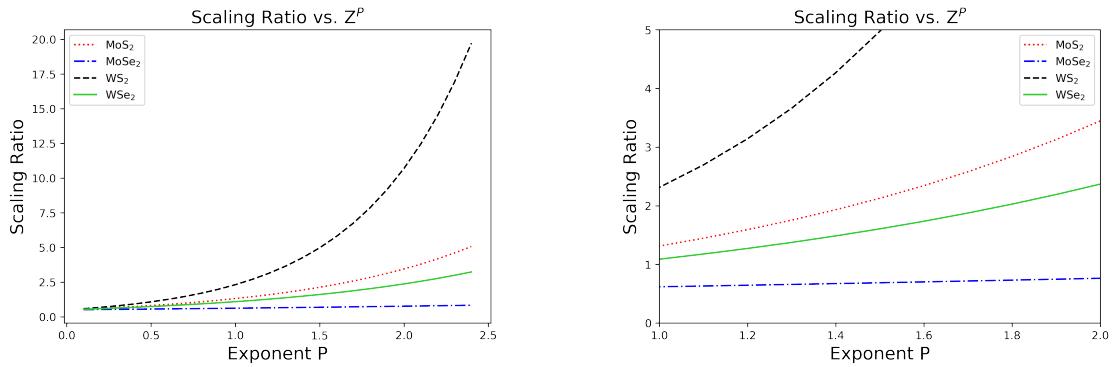


Figure A.4: Scaling ratio R vs Z-contrast scaling exponent n (written here as P) for the four most common TMDs. A zoom in of the 1-2 region in x is shown on the right.

A.3. Results 1

A.3 Results 1

Part1 Section 1 Microscopy

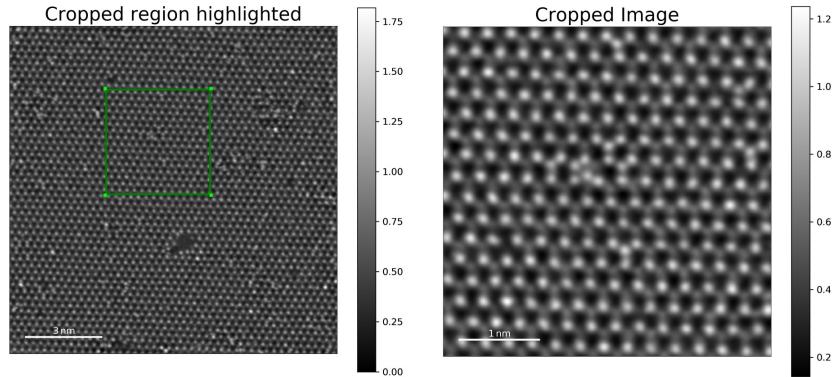


Figure A.5: Full figure of HAADF image 017 with the green rectangle marking the cropped image on the right. The Cropped Image is used in Subsection 4.1.1(Regions affected by Se Ion Implantation).

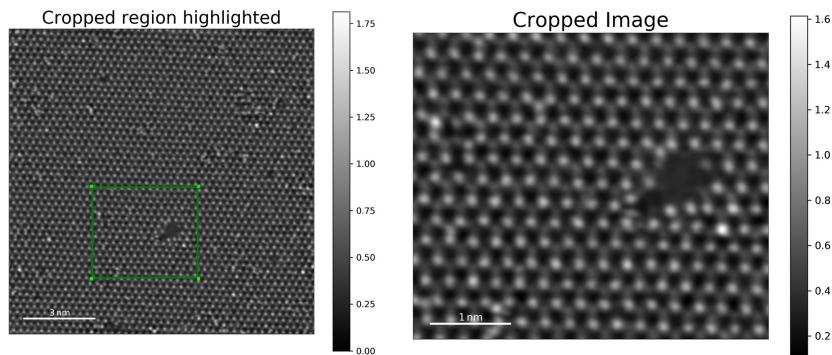


Figure A.6: Full figure of HAADF image 017 with the green rectangle marking the cropped image on the right. The Cropped Image is used in Subsection 4.1.1(Vacancies and Hole Damage).

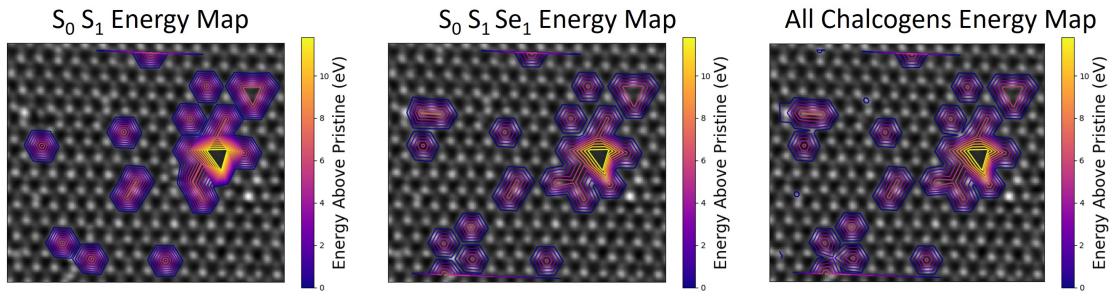


Figure A.7: Energy maps of various vacancies near the hole in Figure A.6(Cropped Image) above. The S₀ S₁ Energy Map only presents the energy above pristine MoS₂ caused by double and single sulphur vacancies. The S₀ S₁ Se₁ Energy Map presents the energy above pristine MoS₂ caused by double and single sulphur vacancies, and the single sulphur vacancies in a Se₁ column. The All Chalcogen Energy Map presents the energy above pristine MoS₂ caused any chalcogen column with a any combination of sulphur vacancies, such as: S₀, S₁, Se₁, Se₁S₁ and Se₂. The S₀ S₁ Energy Map is shown in Subsection 4.1.1(Vacancies and Hole Damage).

A.3.1 Low loss EELS

The ZLP was removed from the low loss signals via a power law subtraction fitting. This was fitted between 1.38 eV and 1.61 eV to reduce the signal before the onset of the A exciton as close to zero as possible. The bilayer region in Figure A.9 is small and therefore readings from the area are unreliable, as noted in the main text. The dispersion were 5 meV and 10 meV for the pristine and implanted spectra, respectively. This results in the implanted spectra in Figure A.9 to have a higher error.

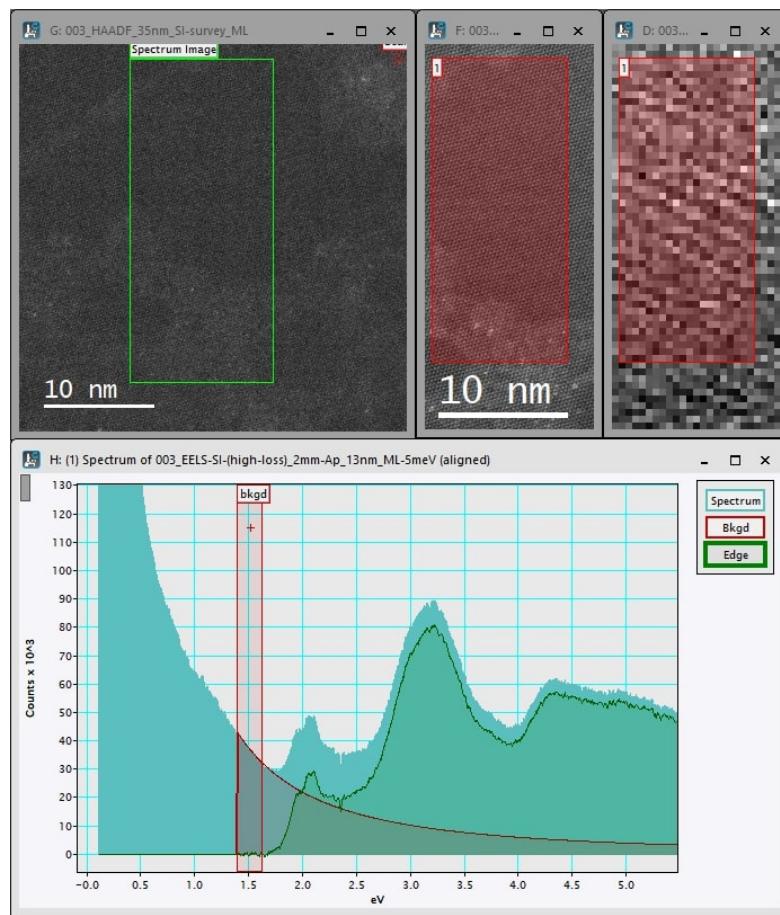


Figure A.8: (Clockwise starting in the top left) Overview ADF image of pristine monolayer MoS₂. The green region of interest is scanned in the ADF and SI on the right. The red regions of interest are summed and the spectrum shown on the bottom, where the LL EELS background is subtracted and signal shown (dark green).

A.3. Results 1

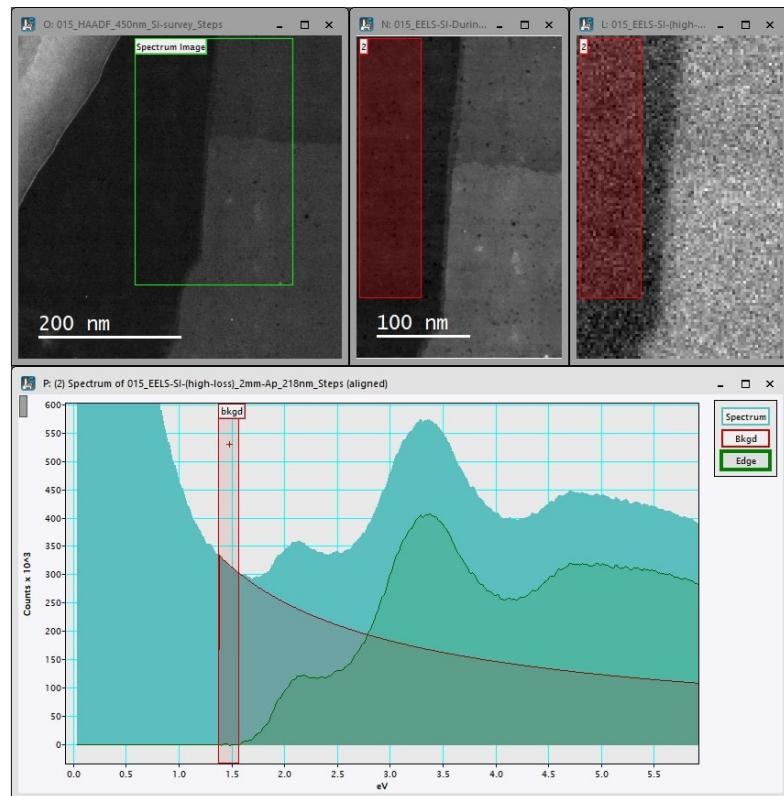


Figure A.9: (Clockwise starting in the top left) Overview ADF image of implanted monolayer MoS₂. The green region of interest is scanned in the ADF and SI on the right. The red regions of interest are summed and the spectrum shown on the bottom, where the LL EELS background is subtracted and signal shown (dark green).

A.4 Results 2

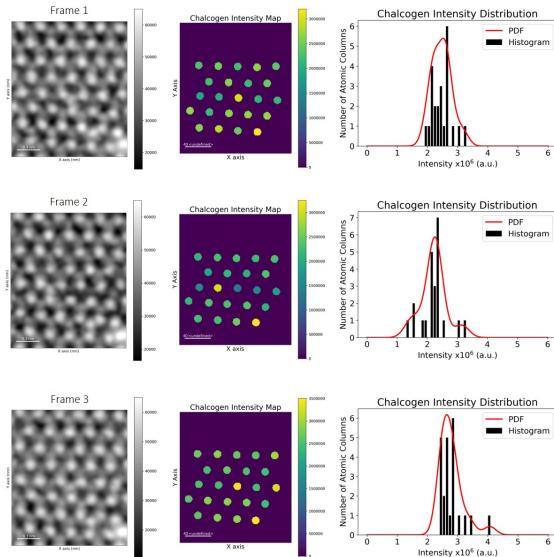


Figure A.10: Three frames of double-Gaussian filtered HAADF STEM images (column 1) of Se implanted MoS₂. The Se-vacancy swap is visualised via the integrated intensity in column 2. The histograms in column 3 show the intensities integrated in column 2.

A.4. Results 2

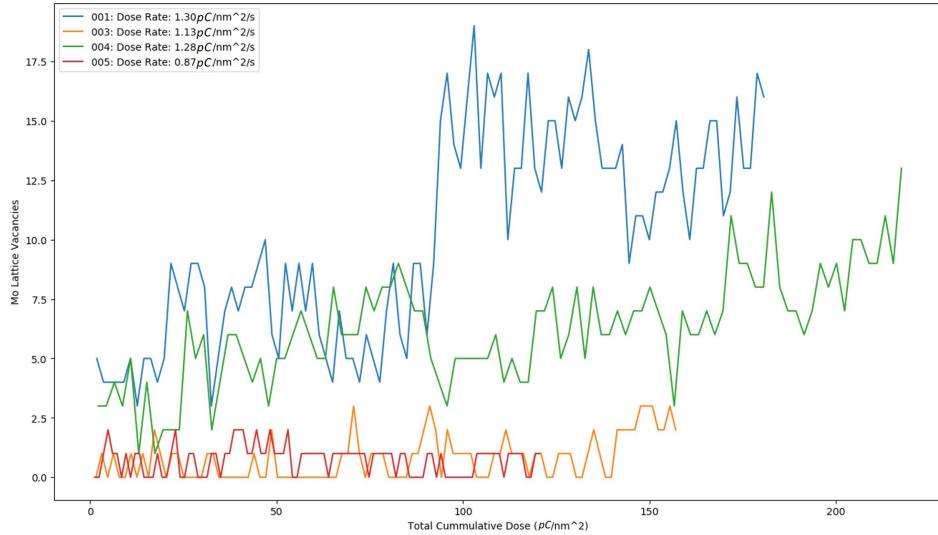


Figure A.11: Lattice Mo vacancies as a function of the total cumulative dose for movies 001, 003, 004 and 005. Movie 001 is not included in the main section due to alignment issues with; the image area was drifting over a large hole. This made quantification unreliable.

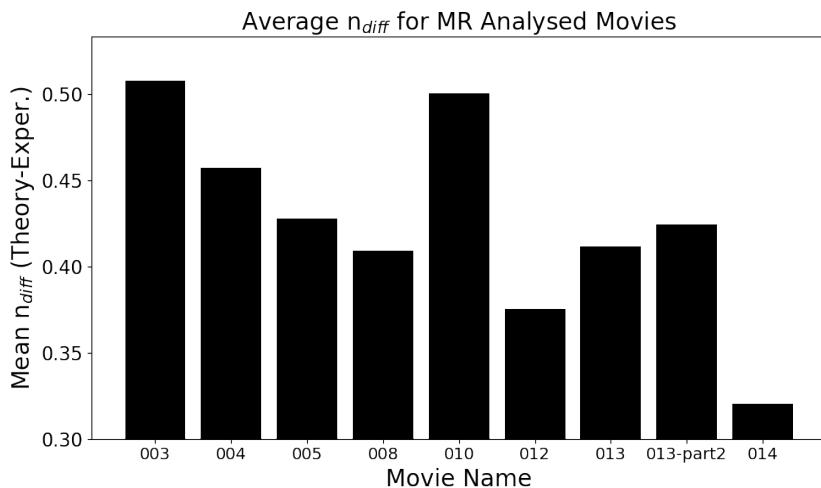


Figure A.12: Average Z-contrast exponent n_{diff} (theoretical minus experiment) for each movie analysed with the Model Refiner tool in subsection 5.2.3. Relevant to comparisons in Subsection 5.2.3.

A.4. Results 2

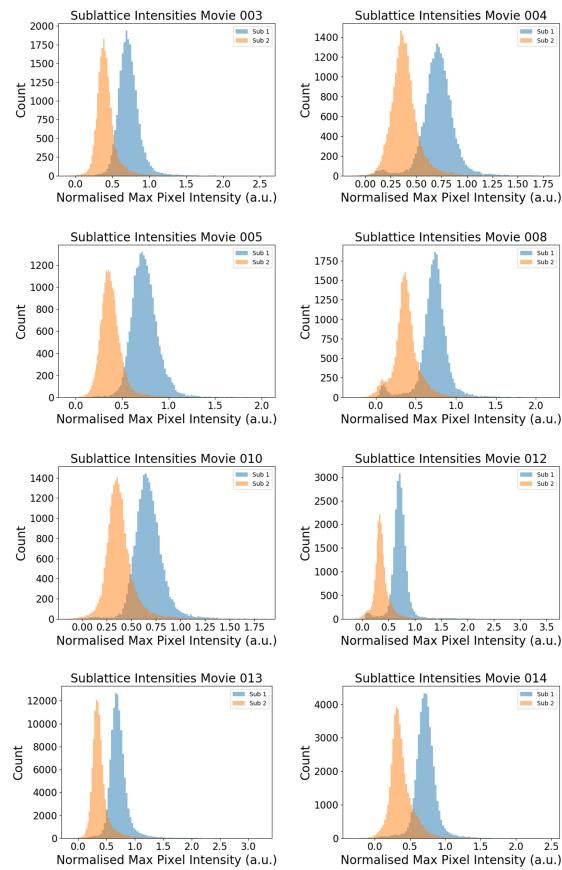


Figure A.13: Histogram intensities of the chalcogen sublattice (orange, lower intensities) and transition metal sublattice (blue) for each movie analysed in Subsection 5.2(Atom Dynamics in Implanted TMDs with the MR Tool).

A.5 Results 3

A.5.1 PTO Junction

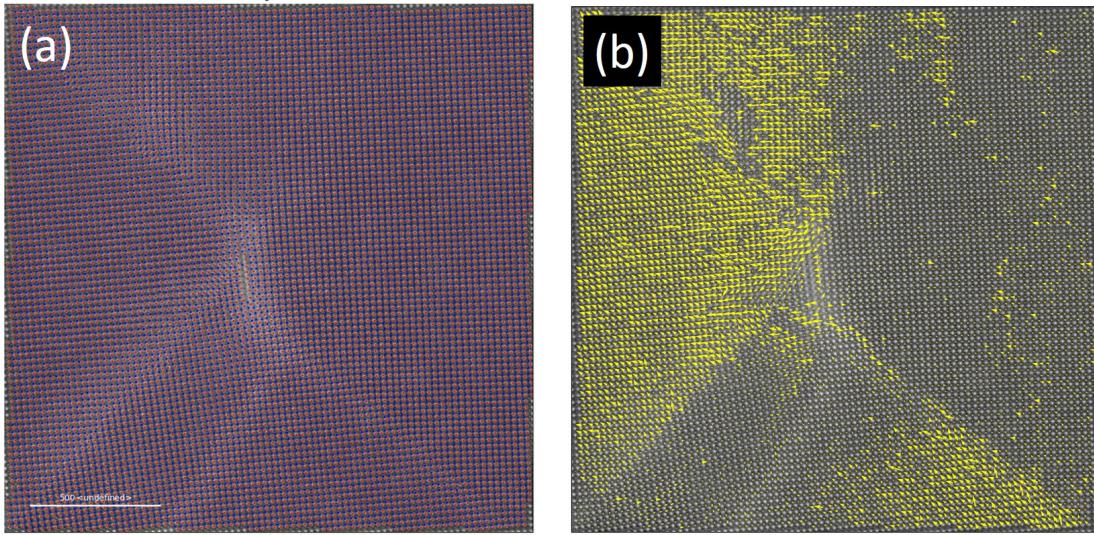


Figure A.14: (a) Atom lattice of the region near a PTO DW junction. Red dots are Pb atomic columns and blue dots are Ti atomic columns. (b) Atomic shift of the Ti sublattice. The high-magnitude vectors displayed in the left are due to off zone axis imaging.

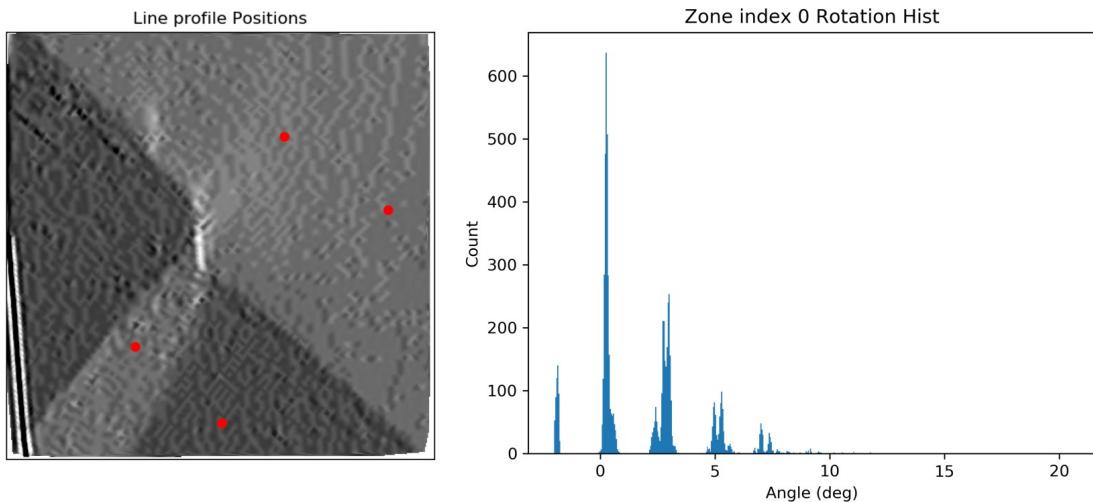


Figure A.15: (a) start and end positions of the line profiles used in subsection 6.2.1(PTO Domain Wall Junctions). (b) Histogram of the angles calculated by `rotation_of_atom_planes` for Figure 6.9. The groupings show the angles in each domain.

After position refinement of the Ti atoms with either a center of mass or Gaussian fitting algorithm, we calculate the difference from this point to their initial “ideal” position. These ideal positions are defined by their crystal structure as the center point of the nearest four Pb atoms. By plotting this, we can view the

A.5. Results 3

polarisation of the Ti atoms in the PTO structure near the junction of several domain walls, wherein several domain walls intersect. This polarisation mapping is shown in Figure A.14(b). The high-magnitude vectors displayed in the left side of Figure A.14(b) are due to off zone axis imaging. This is due to the strained nature of the material near the junction in the centre of the image. Ignoring this region, the general movement of the other domains indicate a head-to-head (H-H) domain wall as shown in the main text in subsection 6.2.1(PTO Domain Wall Junctions) and in [17].

A.6 TEMUL Code

The code used during this thesis is open-source and available for use and download in the TEMUL toolkit GitHub repository: <https://github.com/PinkShnack/TEMUL>. Some of the code for the following is included below: experimental Z-contrast calculations, the refinement algorithms, the Model Refiner and polarisation functions.

A.6.1 Z Contrast

For the full TEMUL code, see <https://github.com/PinkShnack/TEMUL>.

```
1 from temul.intensity_tools import get_sublattice_intensity
2 from temul.element_tools import split_and_sort_element
3 import atomap.api as am
4 import scipy
5 import numpy as np
6 from numpy import log
7
8
9 # For the full TEMUL code, see https://github.com/PinkShnack/TEMUL
10
11 def scaling_z_contrast(numerator_sublattice, numerator_element,
12                         denominator_sublattice, denominator_element
13                         ,
14                         intensity_type, method,
15                         remove_background_method,
16                         background_sublattice, num_points,
17                         percent_to_nn=0.40, mask_radius=None,
18                         split_symbol='_'):
19     # Make sure that the intensity_type input has been chosen.
20     Could
21     # make this more flexible, so that 'all' could be calculated
22     # in one go
23     # simple loop should do that.
24     if intensity_type == 'all':
25         TypeError
26         print('intensity_type must be "max", "mean", or "min"')
27     else:
28         pass
29
30     sublattice0 = numerator_sublattice
31     sublattice1 = denominator_sublattice
```

A.6. TEMUL Code

```
28 # use the get_sublattice_intensity() function to get the mean/
mode
29 # intensities of
30 # each sublattice
31 if type(mask_radius) is list:
32     sublattice0_intensity = get_sublattice_intensity(
33         sublattice0, intensity_type, remove_background_method,
34         background_sublattice,
35         num_points, percent_to_nn=percent_to_nn,
36         mask_radius=mask_radius[0])
37
38     sublattice1_intensity = get_sublattice_intensity(
39         sublattice1, intensity_type, remove_background_method,
40         background_sublattice,
41         num_points, percent_to_nn=percent_to_nn,
42         mask_radius=mask_radius[1])
43 else:
44     sublattice0_intensity = get_sublattice_intensity(
45         sublattice0, intensity_type, remove_background_method,
46         background_sublattice,
47         num_points, percent_to_nn=percent_to_nn,
48         mask_radius=mask_radius)
49
50     sublattice1_intensity = get_sublattice_intensity(
51         sublattice1, intensity_type, remove_background_method,
52         background_sublattice,
53         num_points, percent_to_nn=percent_to_nn,
54         mask_radius=mask_radius)
55
56 if method == 'mean':
57     sublattice0_intensity_method = np.mean(
58         sublattice0_intensity)
59     sublattice1_intensity_method = np.mean(
60         sublattice1_intensity)
61 elif method == 'mode':
62     sublattice0_intensity_method = scipy.stats.mode(
63         np.round(sublattice0_intensity, decimals=2))[0][0]
64     sublattice1_intensity_method = scipy.stats.mode(
65         np.round(sublattice1_intensity, decimals=2))[0][0]
66
67 # Calculate the scaling ratio and exponent for Z-contrast
68 # images
69 scaling_ratio = sublattice0_intensity_method /
70 sublattice1_intensity_method
71
72 numerator_element_split = split_and_sort_element(
```

A.6. TEMUL Code

```
69     element=numerator_element, split_symbol=split_symbol)
70 denominator_element_split = split_and_sort_element(
71     element=denominator_element, split_symbol=split_symbol)
72
73 if len(numerator_element_split) == 1:
74     scaling_exponent = log(
75         denominator_element_split[0][2] * scaling_ratio) / (
76             log(numerator_element_split[0][3]) -
77             log(denominator_element_split[0][3]))
78 else:
79     pass
80
81
82 return(scaling_ratio, scaling_exponent,
83 sublattice0_intensity_method,
84     sublattice1_intensity_method)
85
86
87
88 def find_middle_and_edge_intensities(sublattice,
89                                     element_list,
90                                     standard_element,
91                                     scaling_exponent,
92                                     largest_element_intensity=
93                                     None,
94                                     split_symbol=['_', '.']):
95     """
96     Create a list which represents the peak points of the
97     intensity distribution for each atom.
98
99     works for nanoparticles as well, doesn't matter what
100    scaling_exponent you use for nanoparticle. Figure this out!
101
102    If the max_element_intensity is set, then the program assumes
103    that the standard element is the largest available element
104    combination, and scales the middle and limit intensity lists
105    so that the middle_intensity_list[-1] == max_element_intensity
106
107    """
108
109    middle_intensity_list = []
110    limit_intensity_list = [0.0]
111
112    if isinstance(standard_element, str) is True:
113        standard_split = split_and_sort_element(
```

A.6. TEMUL Code

```

113         element=standard_element, split_symbol=split_symbol)
114         standard_element_value = 0.0
115         for i in range(0, len(standard_split)):
116             standard_element_value += standard_split[i][2] * \
117                 (pow(standard_split[i][3], scaling_exponent))
118     else:
119         standard_element_value = standard_element
120 # find the values for element_lists
121 for i in range(0, len(element_list)):
122     element_split = split_and_sort_element(
123         element=element_list[i], split_symbol=split_symbol)
124     element_value = 0.0
125     for p in range(0, len(element_split)):
126         element_value += element_split[p][2] * \
127             (pow(element_split[p][3], scaling_exponent))
128     atom = element_value / standard_element_value
129     middle_intensity_list.append(atom)
130
131     middle_intensity_list.sort()
132
133     for i in range(0, len(middle_intensity_list) - 1):
134         limit = (middle_intensity_list[i] + middle_intensity_list[
135             i + 1]) / 2
136         limit_intensity_list.append(limit)
137
138     if len(limit_intensity_list) <= len(middle_intensity_list):
139         max_limit = middle_intensity_list[-1] + \
140             (middle_intensity_list[-1] - limit_intensity_list[-1])
141         limit_intensity_list.append(max_limit)
142     else:
143         pass
144
145     if largest_element_intensity is not None:
146         ratio = sublattice.image.max() / largest_element_intensity
147         middle_intensity_list = [
148             middle / ratio for middle in middle_intensity_list]
149         limit_intensity_list = [
150             limit / ratio for limit in limit_intensity_list]
151
152
153
154
155 def find_middle_and_edge_intensities_for_background(
156     elements_from_sub1,
157     elements_from_sub2,

```

A.6. TEMUL Code

```
158     sub1_mode,
159     sub2_mode,
160     element_list_sub1,
161     element_list_sub2,
162     middle_intensity_list_sub1,
163     middle_intensity_list_sub2):
164
165     middle_intensity_list_background = [0.0]
166
167     # it is necessary to scale the background_sublattice
168     # intensities here
169     # already because otherwise
170     # the background_sublattice has no reference atom to base
171     # its mode
172     # intensity on. e.g. in MoS2, first sub has Mo
173     # as a standard atom, second sub has S2 as a standard
174     # reference.
175
176     for i in elements_from_sub1:
177         index = element_list_sub1.index(i)
178         middle = middle_intensity_list_sub1[index] * sub1_mode
179         middle_intensity_list_background.append(middle)
180
181     for i in elements_from_sub2:
182         index = element_list_sub2.index(i)
183         middle = middle_intensity_list_sub2[index] * sub2_mode
184         middle_intensity_list_background.append(middle)
185
186     middle_intensity_list_background.sort()
187
188     limit_intensity_list_background = [0.0]
189     for i in range(0, len(middle_intensity_list_background) - 1):
190         limit = (
191             middle_intensity_list_background[i] +
192             middle_intensity_list_background[i + 1]) / 2
193         limit_intensity_list_background.append(limit)
194
195     if len(limit_intensity_list_background) <= len(
196         middle_intensity_list_background):
197         max_limit = middle_intensity_list_background[-1] + (
198             middle_intensity_list_background[-1] -
199             limit_intensity_list_background[-1])
200         limit_intensity_list_background.append(max_limit)
201     else:
202         pass
```

A.6. TEMUL Code

```
201     return middle_intensity_list_background,
202     limit_intensity_list_background
203
204
205 def sort_sublattice_intensities(sublattice,
206                                 intensity_type='max',
207                                 element_list=[],
208                                 scalar_method='mode',
209                                 middle_intensity_list=None,
210                                 limit_intensity_list=None,
211                                 remove_background_method=None,
212                                 background_sublattice=None,
213                                 num_points=3,
214                                 intensity_list_real=False,
215                                 percent_to_nn=0.40,
216                                 mask_radius=None):
217
218     # intensity_list_real is asking whether the intensity values
219     # in your
220     # intensity_list for the current sublattice
221     # are scaled. Scaled meaning already multiplied by the mean
222     # or mode
223     # of said sublattice.
224     # Set to True for background sublattices. For more details
225     # see
226     # "find_middle_and_edge_intensities_for_background()"
227     # You can see that the outputted lists are scaled by the
228     # mean or mode,
229     # whereas in
230     # "find_middle_and_edge_intensities()", they are not.
231
232     # For testing and quickly assigning a sublattice some elements
233     .
234
235     if middle_intensity_list is None:
236         elements_of_sublattice = []
237         for i in range(0, len(sublattice.atom_list)):
238             sublattice.atom_list[i].elements = element_list[0]
239             elements_of_sublattice.append(sublattice.atom_list[i].
234
235             elements)
236
237     else:
238         sublattice_intensity = get_sublattice_intensity(
239             sublattice=sublattice,
238             intensity_type=intensity_type,
239             remove_background_method=remove_background_method,
```

A.6. TEMUL Code

```
240     background_sub=background_sublattice,
241     num_points=num_points,
242     percent_to_nn=percent_to_nn,
243     mask_radius=mask_radius)

244
245     for i in sublattice_intensity:
246         if i < 0:
247             i = 0.0000000001
248             # raise ValueError("You have negative intensity.
Bad Vibes")

249
250     if intensity_list_real is False:
251
252         if scalar_method == 'mean':
253             scalar = np.mean(sublattice_intensity)
254         elif scalar_method == 'mode':
255             scalar = scipy.stats.mode(
256                 np.round(sublattice_intensity, decimals=2))
257 [0][0]
258         elif isinstance(scalar_method, (int, float)):
259             scalar = scalar_method
260
261         if len(element_list) != len(middle_intensity_list):
262             raise ValueError(
263                 "element_list length does not equal "
264                 "middle_intensity_list length")
265         else:
266             pass
267
268         elements_of_sublattice = []
269         for p in range(0, (len(limit_intensity_list) - 1)):
270             for i in range(0, len(sublattice.atom_list)):
271                 if limit_intensity_list[p] * scalar <
272                     sublattice_intensity[i] < limit_intensity_list[p + 1] * scalar:
273                         sublattice.atom_list[i].elements =
274                             element_list[p]
275                         elements_of_sublattice.append(
276                             sublattice.atom_list[i].elements)

277
278         elif intensity_list_real is True:
279             if len(element_list) != len(middle_intensity_list):
280                 raise ValueError(
281                     "element_list length does not equal
middle_intensity_list "
282                     "length")
283             else:
```

```

281         pass
282
283     elements_of_sublattice = []
284     for p in range(0, (len(limit_intensity_list) - 1)):
285         for i in range(0, len(sublattice.atom_list)):
286             if limit_intensity_list[p] <
287                 sublattice_intensity[i] < limit_intensity_list[p + 1]:
288                     sublattice.atom_list[i].elements =
289                         element_list[p]
290                         elements_of_sublattice.append(
291                             sublattice.atom_list[i].elements)
292
293             for i in range(0, len(sublattice.atom_list)):
294                 if sublattice.atom_list[i].elements == '':
295                     if sublattice_intensity[i] > limit_intensity_list
296                         [-1]:
297                             sublattice.atom_list[i].elements =
298                         element_list[-1]
299                         else:
300                             sublattice.atom_list[i].elements =
301                         get_most_common_sublattice_element(
302                             sublattice)
303                         elements_of_sublattice.append(sublattice.atom_list
304                         [i].elements)
305                         else:
306                             pass
307
308     return(elements_of_sublattice)
309
310
311
312
313
314 def assign_z_height(sublattice, lattice_type, material):
315     for i in range(0, len(sublattice.atom_list)):
316         if material == 'mose2_one_layer':
317             if lattice_type == 'chalcogen':
318                 if len(split_and_sort_element(element=sublattice.
319                     atom_list[i].elements)) == 1 and split_and_sort_element(
320                         element=sublattice.atom_list[i].elements)[0][2] == 1:
321                     sublattice.atom_list[i].z_height = '0.758'
322                 elif len(split_and_sort_element(element=sublattice.
323                     atom_list[i].elements)) == 1 and split_and_sort_element(
324                         element=sublattice.atom_list[i].elements)[0][2] == 2:
325                     sublattice.atom_list[i].z_height = '0.242,
326                     0.758'
```

A.6. TEMUL Code

```
315         elif len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 1 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] > 2:  
            sublattice.atom_list[i].z_height = '0.242,  
0.758, 0.9'  
317         elif len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 2 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] == 1 and  
split_and_sort_element(element=sublattice.atom_list[i].elements  
) [1][2] == 1:  
            sublattice.atom_list[i].z_height = '0.242,  
0.758'  
319         elif len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 2 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] > 1:  
            sublattice.atom_list[i].z_height = '0.242,  
0.758, 0.9'  
321     else:  
        sublattice.atom_list[i].z_height = '0.758'  
323     elif lattice_type == 'transition_metal':  
        if len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 1 and split_and_sort_element(element  
=sublattice.atom_list[i].elements)[0][2] == 1:  
            sublattice.atom_list[i].z_height = '0.5'  
        elif len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 1 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] == 2:  
            sublattice.atom_list[i].z_height = '0.5, 0.95'  
        elif len(split_and_sort_element(element=sublattice.  
.atom_list[i].elements)) == 1 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] > 2:  
            sublattice.atom_list[i].z_height = '0.5, 0.95,  
1'  
        elif len(split_and_sort_element(element=sublattice.  
.atom_list[i].elements)) == 2 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] == 1 and  
split_and_sort_element(element=sublattice.atom_list[i].elements  
) [1][2] == 1:  
            sublattice.atom_list[i].z_height = '0.5, 0.95'  
        elif len(split_and_sort_element(element=sublattice.  
.atom_list[i].elements)) == 2 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] > 1:  
            sublattice.atom_list[i].z_height = '0.5, 0.95,  
1'  
        else:  
            sublattice.atom_list[i].z_height = '0.5'
```

A.6. TEMUL Code

```
336         elif lattice_type == 'background':
337             sublattice.atom_list[i].z_height = '0.95'
338
339
340     else:
341         print(
342             "You must include a suitable lattice_type.
This feature is limited")
343
344     if material == 'mos2_one_layer':
345         if lattice_type == 'chalcogen':
346             if len(split_and_sort_element(element=sublattice.
atom_list[i].elements)) == 1 and split_and_sort_element(element
=sublattice.atom_list[i].elements)[0][2] == 1:
347                 # from L Mattheis, PRB, 1973
348                 sublattice.atom_list[i].z_height = '0.757'
349
350             elif len(split_and_sort_element(element=sublattice.
atom_list[i].elements)) == 1 and split_and_sort_element(
element=sublattice.atom_list[i].elements)[0][2] == 2:
351                 sublattice.atom_list[i].z_height = '0.242,
352                 0.757'
353
354             elif len(split_and_sort_element(element=sublattice.
atom_list[i].elements)) == 1 and split_and_sort_element(
element=sublattice.atom_list[i].elements)[0][2] > 2:
355                 sublattice.atom_list[i].z_height = '0.242,
356                 0.757, 0.95'
357
358             elif len(split_and_sort_element(element=sublattice.
atom_list[i].elements)) == 2 and split_and_sort_element(
element=sublattice.atom_list[i].elements)[0][2] == 1 and
split_and_sort_element(element=sublattice.atom_list[i].elements
)[1][2] == 1:
359                 sublattice.atom_list[i].z_height = '0.242,
360                 0.757'
361
362             elif len(split_and_sort_element(element=sublattice.
atom_list[i].elements)) == 2 and split_and_sort_element(
element=sublattice.atom_list[i].elements)[0][2] > 1:
363                 sublattice.atom_list[i].z_height = '0.242,
364                 0.757, 0.95'
365
366             else:
367                 sublattice.atom_list[i].z_height = '0.757'
368
369         elif lattice_type == 'transition_metal':
370             if len(split_and_sort_element(element=sublattice.
atom_list[i].elements)) == 1 and split_and_sort_element(element
=sublattice.atom_list[i].elements)[0][2] == 1:
371                 sublattice.atom_list[i].z_height = '0.5'
```

A.6. TEMUL Code

```
362         elif len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 1 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] == 2:  
            sublattice.atom_list[i].z_height = '0.5, 0.95'  
        elif len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 1 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] > 2:  
            sublattice.atom_list[i].z_height = '0.5, 0.95,  
1'  
        elif len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 2 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] == 1 and  
split_and_sort_element(element=sublattice.atom_list[i].elements  
)[1][2] == 1:  
            sublattice.atom_list[i].z_height = '0.5, 0.95'  
        elif len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 2 and split_and_sort_element(  
element=sublattice.atom_list[i].elements)[0][2] > 1:  
            sublattice.atom_list[i].z_height = '0.5, 0.95,  
1'  
    else:  
        sublattice.atom_list[i].z_height = '0.5'  
    elif lattice_type == 'background':  
        sublattice.atom_list[i].z_height = '0.95'  
  
    else:  
        print(  
            "You must include a suitable lattice_type.  
This feature is limited")  
  
379     if material == 'mos2_two_layer': # from L Mattheis, PRB,  
1973  
380         if lattice_type == 'TM_top':  
381             if len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 3:  
                 sublattice.atom_list[i].z_height = '0.1275,  
0.3725, 0.75'  
            elif len(split_and_sort_element(element=sublattice.  
atom_list[i].elements)) == 2:  
                 sublattice.atom_list[i].z_height = '0.1275,  
0.3725, 0.75'  
            else:  
                 sublattice.atom_list[i].z_height = '0.95'  
                 # raise ValueError("z_height is limited to  
only a handful of positions")  
388         elif lattice_type == 'TM_bot':
```

A.6. TEMUL Code

```
389         if len(split_and_sort_element(element=sublattice.
atom_list[i].elements)) == 3:
390             sublattice.atom_list[i].z_height = '0.25,
391             0.6275, 0.8725'
392             elif len(split_and_sort_element(element=sublattice
.atom_list[i].elements)) == 2:
393                 sublattice.atom_list[i].z_height = '0.25,
394                 0.6275, 0.8725'
395             else:
396                 sublattice.atom_list[i].z_height = '0.95'
397                 elif lattice_type == 'background':
398                     sublattice.atom_list[i].z_height = '0.95'
399
400             else:
401                 print(
402                     "You must include a suitable lattice_type.
This feature is limited")
```

Code Segment A.1: Some of the developed Z-contrast functions of the TEMUL toolkit discussed in Chapter 3 and in Chapter 4.

A.6.2 Model Refinement Algorithms

For the full TEMUL code, see <https://github.com/PinkShnack/TEMUL>.

```
1
2 from temul.element_tools import split_and_sort_element
3 import atomap.api as am
4 from atomap.atom_finding_refining import
    get_atom_positions_in_difference_image
5 import matplotlib.pyplot as plt
6 import numpy as np
7 import hyperspy.api as hs
8 import pandas as pd
9
10 # For the full TEMUL code, see https://github.com/PinkShnack/TEMUL
11
12 def change_sublattice_atoms_via_intensity(
13     sublattice,
14     image_diff_array,
15     darker_or_brighter,
16     element_list,
17     verbose=False):
18     # get the index in sublattice from the
        image_difference_intensity() output,
```

A.6. TEMUL Code

```
19     # which is the image_diff_array input here.
20     # then, depending on whether the image_diff_array is for atoms
21     # that should
22     # be brighter or darker, set a new element to that sublattice
23     atom_position
24     ''
25
26     Change the elements in a sublattice object to a higher or
27     lower combined
28     atomic (Z) number.
29
30     The aim is to change the sublattice elements so that the
31     experimental image
32     agrees with the simulated image in a realistic manner.
33     See image_difference_intensity()
34
35     Parameters
36     -----
37
38     sublattice : Atomap Sublattice object
39         the elements of this sublattice will be changed
40     image_diff_array : Numpy 2D array
41         Contains (p, x, y, intensity) where
42         p = index of Atom_Position in sublattice
43         x = Atom_Position.pixel_x
44         y = Atom_Position.pixel_y
45         intensity = calculated intensity of atom in sublattice.
46
47     image
48     darker_or_brighter : int
49         if the element should have a lower combined atomic Z
50         number,
51         darker_or_brighter = 0.
52         if the element should have a higher combined atomic Z
53         number,
54         darker_or_brighter = 1
55         In other words, the image_diff_array will change the given
56         sublattice elements to darker or brighter spots by
57         choosing 0 and 1,
58         respectively.
59     element_list : list
60         list of element configurations
61
62     Returns
63     -----
64
65     n/a - changes sublattice elements inplace
66
67     Examples
```

A.6. TEMUL Code

```
57 -----
58 >>> from temul.model_creation import
59 change_sublattice_atoms_via_intensity
60 >>> sublattice = am.dummy_data.get_simple_cubic_sublattice()
61 >>> for i in range(0, len(sublattice.atom_list)):
62     ...     sublattice.atom_list[i].elements = 'Mo_1'
63     ...     sublattice.atom_list[i].z_height = [0.5]
64 >>> element_list = ['H_0', 'Mo_1', 'Mo_2']
65 >>> image_diff_array = np.array([[5, 2, 2, 20], [1, 2, 4, 7]])
66 >>> # This will change the 5th atom in the sublattice to a
67 lower atomic Z
68 >>> # number, i.e., 'H_0' in the given element_list
69 >>> change_sublattice_atoms_via_intensity(sublattice=
70 sublattice,
71     ...                                     image_diff_array=
72     ...                                     darker_or_brighter=0,
73     ...                                     element_list=element_list)
74 Changing some atoms
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
```

A.6. TEMUL Code

```
96     p = int(p)
97
98     elem = sublattice.atom_list[p].elements
99
100    if elem not in element_list:
101        raise ValueError("The element {}, {}, {} isn't
102        in the "
103                           "element_list".format(
104                               sublattice.name, p, elem))
105
106    atom_index = element_list.index(elem)
107
108    if darker_or_brighter == 0:
109        if '_0' in elem:
110            pass
111        else:
112            new_atom_index = atom_index - 1
113            if len(sublattice.atom_list[p].z_height) == 2:
114                z_h = sublattice.atom_list[p].z_height
115                sublattice.atom_list[p].z_height = [
116                    (z_h[0] + z_h[1]) / 2]
117            else:
118                pass
119
120            try:
121                new_atom = element_list[new_atom_index]
122            except IndexError:
123                print("No lower element configuration in
124                  element_list")
125
126            elif darker_or_brighter == 1:
127                new_atom_index = atom_index + 1
128                if len(sublattice.atom_list[p].z_height) == 2:
129                    z_h = sublattice.atom_list[p].z_height
130                    sublattice.atom_list[p].z_height = [
131                        (z_h[0] + z_h[1]) / 2]
132                else:
133                    pass
134
135            try:
136                new_atom = element_list[new_atom_index]
137            except IndexError:
138                print("No higher element configuration in
139                  element_list")
140
141            elif new_atom_index < 0:
```

A.6. TEMUL Code

```
139             raise ValueError("You don't have any smaller atoms"
140         ")
141     elif new_atom_index >= len(element_list):
142         raise ValueError("You don't have any bigger atoms"
143     )
144     new_atom = element_list[new_atom_index]
145     try:
146         new_atom
147     except NameError:
148         pass
149     else:
150         sublattice.atom_list[p].elements = new_atom
151
152     sublattice.atom_list[p].elements = new_atom
153
154 def image_difference_intensity(sublattice,
155                                 sim_image,
156                                 element_list,
157                                 filename=None,
158                                 percent_to_nn=0.40,
159                                 mask_radius=None,
160                                 change_sublattice=False,
161                                 verbose=False):
162     """
163     Find the differences in a sublattice's atom_position
164     intensities.
165     Change the elements of these atom_positions depending on this
166     difference of
167     intensities.
168
169     The aim is to change the sublattice elements so that the
170     experimental image
171     agrees with the simulated image in a realistic manner.
172
173     Parameters
174     -----
175
176     sublattice : Atomap Sublattice object
177         Elements of this sublattice will be refined
178     sim_image : HyperSpy 2D signal
179         The image you wish to refine with, usually an image
180         simulation of the
181         sublattice.image
182     element_list : list
183         list of element configurations, used for refinement
```

A.6. TEMUL Code

```
179     filename : string, default None
180         name with which the image will be saved
181     percent_to_nn : float, default 0.40
182         Determines the boundary of the area surrounding each
183         atomic
184             column, as fraction of the distance to the nearest
185             neighbour.
186             mask_radius : float, default None
187                 Radius of the mask around each atom. If this is not set,
188                 the radius will be the distance to the nearest atom in the
189                 same sublattice times the ‘percent_to_nn’ value.
190                 Note: if ‘mask_radius’ is not specified, the Atom_Position
191                 objects
192                     must have a populated nearest_neighbor_list.
193             change_sublattice : bool, default False
194                 If change_sublattice is set to True, all incorrect element
195                 assignments
196                     will be corrected inplace.
```

197
198

199 Returns
200 -----
201
202 Nothing - changes the elements within the sublattice object.

203
204

205 Example
206 -----
207
208
209 >>> sublattice = am.dummy_data.get_simple_cubic_sublattice()
210 >>> sim_image = am.dummy_data.
211 get_simple_cubic_with_vacancies_signal()
212 >>> for i in range(0, len(sublattice.atom_list)):
213 ... sublattice.atom_list[i].elements = 'Mo_1'
214 ... sublattice.atom_list[i].z_height = [0.5]
215 >>> element_list = ['H_0', 'Mo_1', 'Mo_2']
216 >>> image_difference_intensity(sublattice=sublattice,
217 ... sim_image=sim_image,
218 ... element_list=element_list)
219
220
221 with some image noise and plotting the images
222 >>> sublattice = am.dummy_data.get_simple_cubic_sublattice(
223 ... image_noise=True)
224 >>> sim_image = am.dummy_data.
225 get_simple_cubic_with_vacancies_signal()
226 >>> for i in range(0, len(sublattice.atom_list)):
227 ... sublattice.atom_list[i].elements = 'Mo_1'
228 ... sublattice.atom_list[i].z_height = [0.5]

A.6. TEMUL Code

```
219     >>> element_list = ['H_0', 'Mo_1', 'Mo_2']
220     >>> image_difference_intensity(sublattice=sublattice,
221 ...                                     sim_image=sim_image,
222 ...                                     element_list=element_list)
223
224     ''
225
226     # np.array().T needs to be taken away for newer atomap
227     # versions
228     sublattice_atom_positions = np.array(sublattice.atom_positions
229 ).T
230
231     diff_image = hs.signals.Signal2D(sublattice.image - sim_image.
232 data)
233
234     # create sublattice for the 'difference' data
235     diff_sub = am.Sublattice(
236         atom_position_list=sublattice_atom_positions, image=
237 diff_image)
238
239     if percent_to_nn is not None:
240         sublattice.find_nearest_neighbors()
241         diff_sub.find_nearest_neighbors()
242     else:
243         pass
244
245     # Get the intensities of these sublattice positions
246     diff_sub.get_atom_column_amplitude_mean_intensity(
247         percent_to_nn=percent_to_nn, mask_radius=mask_radius)
248     diff_mean_ints = np.array(
249         diff_sub.atom_amplitude_mean_intensity, ndmin=2).T
250     # diff_mean_ints = np.array(diff_mean_ints, ndmin=2).T
251
252     # combine the sublattice_atom_positions and the intensities
253     for
254         # future indexing
255         positions_intensities_list = np.append(
256             sublattice_atom_positions,
257                                         diff_mean_ints, axis=1)
258
259         # find the mean and std dev of this distribution of
260         # intensities
261         mean_ints = np.mean(diff_mean_ints)
262         std_dev_ints = np.std(diff_mean_ints)
263
264         # plot the mean and std dev on each side of intensities
265         histogram
```

A.6. TEMUL Code

```
257     std_from_mean = np.array([mean_ints - std_dev_ints, mean_ints
258                               + std_dev_ints,
259                               mean_ints - (2 * std_dev_ints),
260                               mean_ints + (2 * std_dev_ints),
261                               mean_ints - (3 * std_dev_ints),
262                               mean_ints + (3 * std_dev_ints),
263                               mean_ints - (4 * std_dev_ints),
264                               mean_ints + (4 * std_dev_ints)
265                               ], ndmin=2).T
266
267     y_axis_std = np.array([len(diff_mean_ints) / 100] * len(
268                           std_from_mean),
269                           ndmin=2).T
270
271     std_from_mean_array = np.concatenate((std_from_mean,
272                                         y_axis_std), axis=1)
273
274     std_from_mean_array = np.append(std_from_mean, y_axis_std,
275                                     axis=1)
276
277
278     # if the intensity if outside 3 sigma, give me those atom
279     # positions
280
281     # and intensities (and the index!)
282
283     outliers_bright, outliers_dark = [], []
284
285     for p in range(0, len(positions_intensities_list)):
286
287         x, y = positions_intensities_list[p,
288                                           0],
289
290         positions_intensities_list[p, 1]
291
292         intensity = positions_intensities_list[p, 2]
293
294
295         if positions_intensities_list[p, 2] > std_from_mean_array
296             [7, 0]:
297
298             outliers_bright.append([p, x, y, intensity])
299
300         elif positions_intensities_list[p, 2] <
301             std_from_mean_array[6, 0]:
302
303             outliers_dark.append([p, x, y, intensity])
304
305     # Now we have the details of the not correct atom_positions
306
307     outliers_bright = np.array(outliers_bright)
308
309     outliers_dark = np.array(outliers_dark)
310
311
312     if change_sublattice is True:
313
314         # Now make the changes to the sublattice for both bright
315         # and dark arrays
316
317         change_sublattice_atoms_via_intensity(sublattice=
318                                             sublattice,
```

A.6. TEMUL Code

```
290                               image_diff_array=
291
292                               outliers_bright,
293                               darker_or_brighter
294                               =1,
295                               element_list=
296                               element_list,
297                               verbose=verbose)
298
299                               change_sublattice_atoms_via_intensity(sublattice=
300                               sublattice,
301                               image_diff_array=
302                               outliers_dark,
303                               darker_or_brighter
304                               =0,
305                               element_list=
306                               element_list,
307                               verbose=verbose)
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
```

A.6. TEMUL Code

```
326     plt.ylabel("Count", fontsize=16)
327     plt.tight_layout()
328     plt.show()
329     plt.savefig(
330         fname="Histogram of " + sublattice.name + "
331         Intensities.png",
332         transparent=True, frameon=False, bbox_inches='tight',
333         pad_inches=None, dpi=300, labels=False)
334
335 else:
336     pass
337
338 def image_difference_position(sublattice,
339                               sim_image,
340                               pixel_threshold,
341                               comparison_sublattice_list=None,
342                               filename=None,
343                               percent_to_nn=0.40,
344                               mask_radius=None,
345                               num_peaks=5,
346                               inplace=True):
347     """
348     Find new atomic coordinates by comparing experimental to
349     simulated image.
350     Changes the sublattice inplace using
351     change_sublattice_pseudo_inplace.
352
353     The aim is to change the sublattice elements so that the
354     experimental image
355     agrees with the simulated image in a realistic manner.
356     See also image_difference_intensity function and the
357     Model_Refiner class.
358
359     Parameters
360     -----
361     sublattice : Atomap sublattice object
362     sim_image : simulated image used for comparison with
363     sublattice image
364     pixel_threshold : int
365         minimum pixel distance from current sublattice atoms. If
366         the new atomic
367             coordinates are greater than this distance, they will be
368             created.
369             Choose a pixel_threshold that will not create new atoms in
370             unrealistic
```

```
363     positions.
364     filename : string, default None
365         name with which the image will be saved
366     percent_to_nn : float, default 0.40
367         Determines the boundary of the area surrounding each
368         atomic
369         column, as fraction of the distance to the nearest
370         neighbour.
371         mask_radius : float, default None
372             Radius of the mask around each atom. If this is not set,
373             the radius will be the distance to the nearest atom in the
374             same sublattice times the ‘percent_to_nn’ value.
375             Note: if ‘mask_radius’ is not specified, the Atom_Position
376             objects
377                 must have a populated nearest_neighbor_list.
378             num_peaks : int, default 5
379                 number of new atoms to add
380             add_sublattice : bool, default False
381                 If set to True, a new sublattice will be created and
382                 returned.
383                 The reason it is set to False is so that one can check if
384                 new atoms
385                 would be added with the given parameters.
386             sublattice_name : string, default ‘sub_new’
387                 the outputted sublattice object name and sublattice.name
388                 the new
389                 sublattice will be given
390
391             Returns
392             -----
393             Atomap Sublattice object, with added positions if found.
394
395             Examples
396             -----
397
398             >>> from temul.model_creation import (
399                 image_difference_position,
400                 ...
401                 change_sublattice_pseudo_inplace)
402             >>> import atomap.api as am
403             >>> sublattice = am.dummy_data.
404                 get_simple_cubic_with_vacancies_sublattice(
405                     ...
406                     image_noise=
407                     True)
408             >>> sim_image = am.dummy_data.get_simple_cubic_signal()
409             >>> for i in range(0, len(sublattice.atom_list)):
```

A.6. TEMUL Code

```
399     ...         sublattice.atom_list[i].elements = 'Mo_1'
400     ...         sublattice.atom_list[i].z_height = '0.5'
401 >>> len(sublattice.atom_list)
402 397
403
404 >>> sublattice = image_difference_position(
405     sublattice_to_refine=sublattice,
406     ...
407     ...
408     ...
409     ...
410     ...
411 3 new atoms found! Adding new atom positions.
412
413 >>> len(sublattice.atom_list)
414 400
415
416 One can now sort these atom positions into elements.
417
418 Returns
419 -----
420 Atomap Sublattice object
421 ,,
422
423 image_for_sublattice = sublattice.signal
424 diff_image = hs.signals.Signal2D(
425     image_for_sublattice.data - sim_image.data)
426 diff_image_inverse = hs.signals.Signal2D(
427     sim_image.data - image_for_sublattice.data)
428
429 # below function edit of get_atom_positions. Just allows
430 num_peaks from
431 # sklearn>find_local_maximum
432 atom_positions_diff_image =
433 get_atom_positions_in_difference_image(
434     diff_image, num_peaks=num_peaks)
435 atom_positions_diff_image_inverse =
436 get_atom_positions_in_difference_image(
437     diff_image_inverse, num_peaks=num_peaks)
438
439 diff_image_sub = am.Sublattice(atom_positions_diff_image,
440 diff_image)
441 # diff_image_sub.find_nearest_neighbors()
442 # diff_image_sub.refine_atom_positions_using_center_of_mass(
443 #     percent_to_nn=percent_to_nn, mask_radius=mask_radius)
```

A.6. TEMUL Code

```
440 # diff_image_sub.refine_atom_positions_using_2d_gaussian(
441 #     percent_to_nn=percent_to_nn, mask_radius=mask_radius)
442 atom_positions_sub_diff = np.array(diff_image_sub.
443 atom_positions).T
444
445
446 diff_image_sub_inverse = am.Sublattice(
447 atom_positions_diff_image_inverse,
448                                         diff_image_inverse)
449 # diff_image_sub_inverse.find_nearest_neighbors()
450 # diff_image_sub_inverse.
451 refine_atom_positions_using_center_of_mass(
452 #     percent_to_nn=percent_to_nn, mask_radius=mask_radius)
453 # diff_image_sub_inverse.
454 refine_atom_positions_using_2d_gaussian(
455 #     percent_to_nn=percent_to_nn, mask_radius=mask_radius)
456 atom_positions_sub_diff_inverse = np.array(
457     diff_image_sub_inverse.atom_positions).T
458
459 # these should be inputs for the image_list below
460
461 atom_positions_diff_all = np.concatenate(
462     (atom_positions_sub_diff, atom_positions_sub_diff_inverse))
463
464 atom_positions_sub_new = []
465
466 # for sublattice in sublattice_list:
467 # SHOULD i HAVE ALL SUBLATTICE POSITIONS HERE?
468
469 if comparison_sublattice_list is None:
470     comparison_sublattice_list = [sublattice]
471
472 for atom in range(0, len(atom_positions_diff_all)):
473
474     new_atom_distance_list = []
475
476     for comp_sub in comparison_sublattice_list:
477         comp_sub_atom_pos = np.array(comp_sub.atom_positions).
478 T
479
480         for p in range(0, len(comp_sub_atom_pos)):
481
482             xy_distances = atom_positions_diff_all[atom] - \
483                         comp_sub_atom_pos[p]
```

A.6. TEMUL Code

```
480
481     # put all distances in this array with this loop
482     vector_array = []
483     vector = np.sqrt((xy_distances[0]**2) +
484                       (xy_distances[1]**2))
485     vector_array.append(vector)
486
487     new_atom_distance_list.append(
488         [vector_array, atom_positions_diff_all[atom],
489          sublattice])
490
491     # use list comprehension to get the distances on their own
492     # the [0] is
493     # changing the list of lists to a list of floats
494     new_atom_distance_sublist = [sublist[:1][0] for sublist in
495                                   new_atom_distance_list]
496     new_atom_distance_min = min(new_atom_distance_sublist)
497
498     new_atom_distance_min_index = new_atom_distance_sublist.
499     index(
500         new_atom_distance_min)
501
502     new_atom_index = new_atom_distance_list[
503     new_atom_distance_min_index]
504
505     if new_atom_index[0][0] > pixel_threshold: # greater than
506     10 pixels
507
508     if len(atom_positions_sub_new) == 0:
509         atom_positions_sub_new = [np.ndarray.tolist(
510             new_atom_index[1])]
511     else:
512         atom_positions_sub_new.extend(
513             [np.ndarray.tolist(new_atom_index[1])])
514     else:
515         pass
516
517     if len(atom_positions_sub_new) == 0:
518         print("No New Atoms")
519
520     elif len(atom_positions_sub_new) != 0 and inplace is True:
521         print("{} new atoms found! Adding new atom positions.".
522               format(
523                 len(atom_positions_sub_new)))
524
525     sublattice = change_sublattice_pseudo_inplace(
```

A.6. TEMUL Code

```
520         new_atom_positions=atom_positions_sub_new,
521         old_sublattice=sublattice)

522     #     sub_new.refine_atom_positions_using_center_of_mass(
523     #         percent_to_nn=percent_to_nn, mask_radius=mask_radius)
524     #     sub_new.refine_atom_positions_using_2d_gaussian(
525     #         percent_to_nn=percent_to_nn, mask_radius=mask_radius)

527     # try:
528     #     sub_new
529     # except NameError:
530     #     sub_new_exists = False
532     # else:
533     #     sub_new_exists = True

534
535     if filename is not None:
536         '',
537         diff_image.plot()
538         diff_image_sub.plot()
539         diff_image_inverse.plot()
540         diff_image_sub_inverse.plot()
541         ''

542
543     plt.figure()
544     plt.suptitle('Image Difference Position' + filename)

545
546     plt.subplot(1, 2, 1)
547     plt.imshow(diff_image.data)
548     plt.title('diff')
549     plt.axis("off")

550
551     plt.subplot(1, 2, 2)
552     plt.imshow(diff_image_inverse.data)
553     plt.title('diff_inv')
554     plt.axis("off")
555     plt.show()

556
557     plt.savefig(fname='pos_diff_' + filename + '.png',
558                 transparent=True, frameon=False, bbox_inches='tight',
559                 pad_inches=None, dpi=300, labels=False)

560
561     sublattice.plot()
562     plt.title(sublattice.name + filename, fontsize=20)
563     plt.gca().axes.get_xaxis().set_visible(False)
564     plt.gca().axes.get_yaxis().set_visible(False)
```

A.6. TEMUL Code

```
565     plt.tight_layout()
566     plt.savefig(fname='pos_diff_' + sublattice.name + filename
567 + '.png',
568                 transparent=True, frameon=False, bbox_inches='
569 tight',
570                 pad_inches=None, dpi=300, labels=False)
571
572     return(sublattice)
```

Code Segment A.2: Some of the developed algorithm functions of the TEMUL toolkit discussed in Chapter 5.

A.6.3 Model Refiner

For the full TEMUL code, see <https://github.com/PinkShnack/TEMUL>.

```
1
2 from temul.model_creation import (
3     count_atoms_in_sublattice_list,
4     compare_count_atoms_in_sublattice_list,
5     image_difference_intensity,
6     image_difference_position,
7     count_all_individual_elements,
8     get_positions_from_sublattices,
9     get_most_common_sublattice_element
10 )
11
12 from temul.element_tools import (
13     get_individual_elements_from_element_list,
14     combine_element_lists,
15     split_and_sort_element,
16     atomic_radii_in_pixels
17 )
18
19 from temul.io import create_dataframe_for_xyz
20 from temul.signal_processing import (
21     compare_two_image_and_create_filtered_image,
22     calibrate_intensity_distance_with_sublattice_roi,
23     measure_image_errors
24 )
25
26 from temul.simulations import (
27     simulate_with_prismatic,
28     load_prismatic_mrc_with_hyperspy
29 )
```

A.6. TEMUL Code

```
30 from atomap.initial_position_finding import add_atoms_with_gui
31 import matplotlib.pyplot as plt
32 import copy
33
34 # For the full TEMUL code, see https://github.com/PinkShnack/TEMUL
35
36 class Model_Refiner():
37     def __init__(self, sublattice_and_elements_dict,
38                  comparison_image=None, sampling=None,
39                  thickness=10, name=''):
40         '''
41             Object which is used to refine the elements in a list of
42             Atomap sublattices. There are currently two refinement
43             methods:
44                 1. Refine with position ‘
45                 image_difference_position_model_refiner‘
46                 2. Refine with intensity ‘
47                 image_difference_intensity_model_refiner‘.
48
49             Parameters
50             -----
51             sublattice_and_elements_dict : dictionary
52                 A dictionary of the structure ‘{sublattice:
53                 element_list,}’ where each
54                     sublattice is an Atomap sublattice object and each
55                     element_list is
56                         a list of elements of the form given in the examples
57                         of
58                         ‘get_individual_elements_from_element_list’ in the
59                         element_tools.py
60                         module.
61                         comparison_image : Hyperspy Signal2D, default None
62                             This is the image with which the first ‘sublattice’
63                             image will be
64                             compared to refine the list of sublattices’ elements.
65                             If None is given when the Model Refiner is created, or
66                             if the
67                                 ‘comparison_image’ is not the same shape as the first
68                                 sublattice image,
69                                 a warning will be returned.
70                                 To set a ‘comparison_image’, you can either use
71                                 ‘model_refiner.create_simulation’ or by setting the
72                                 ‘model_refiner.comparison_image’ to an image.
73                         sampling : float, default None
74                             The real space sampling of the ‘sublattice’ images in
75                             Angstrom. The
```

A.6. TEMUL Code

```
65         sampling is defined as: sampling = angstrom/pixel.
This sampling will
66             be identical for the 'comparison_image'. If it is set
to None, the
67                 sampling will automatically be set by the 'sublattice.
signal' object.
68             thickness : float, default 10
69                 Physical thickness of the sample in Angstrom. This
will be used for the
70                     simulation.
71             name : string, default ''
72                 Name of the Model Refiner.

73
74
75     Attributes
76     -----
77     sublattice_and_elements_dict: dictionary
78     sublattice_list: list
79     name: string
80     element_list: list of strings, or lists of lists of
strings
81     flattened_element_list: list of strings
82     _element_count: Counter
83     element_count_history_list: list of Counters
84     refinement_history: list of strings
85     reference_image: Hyperspy Signal2D
86     calibration_area: 2D array-like; list of 2 lists
87     calibration_separation: int
88     sampling: float
89     thickness: float
90     image_xyz_sizes: list
91     previous_refiner_instance: Model Refiner object
92     error_between_comparison_and_reference_image: list of
floats
93         error_between_images_history: list of lists

94
95     Examples
96     -----
97     For now, see 'image_difference_intensity_model_refiner'
and
98     'image_difference_position_model_refiner' below.
99     '',
100
101     self.sublattice_and_elements_dict =
sublattice_and_elements_dict
```

A.6. TEMUL Code

```
102         self.sublattice_list = list(sublattice_and_elements_dict.
103                                     keys())
104         self.element_list = list(sublattice_and_elements_dict.
105                                   values())
106         self.flattened_element_list = combine_element_lists(
107             self.element_list)
108         self._comparison_image_init(comparison_image)
109         self.name = name
110         self._element_count = count_atoms_in_sublattice_list(
111             self.sublattice_list, filename=None)
112         self.element_count_history_list = []
113         if len(self.element_count_history_list) == 0:
114             self.element_count_history_list.append(self.
115             _element_count)
116
117         self.refinement_history = []
118         if len(self.refinement_history) == 0:
119             self.refinement_history.append("Initial State")
120
121         self.reference_image = self.sublattice_list[0].signal
122         self.calibration_area = [[1, 1],
123                                 [self.reference_image.data.shape
124                                  [-1] - 1,
125                                  self.reference_image.data.shape
126                                  [-2] - 1]]
127         self.calibration_separation = 12
128         # maybe have a _sampling_init function
129         if sampling is None:
130             self.sampling = self.reference_image.axes_manager[-1].
131             scale
132         else:
133             self.sampling = sampling
134         if self.reference_image.axes_manager[-1].units == 'nm':
135             self.sampling = self.sampling * 10
136         self._reference_image_init()
137
138         self._auto_mask_radii_init()
139
140         self.thickness = thickness
141         self.image_xyz_sizes = [
142             self.sampling * self.reference_image.data.shape[-1],
143             self.sampling * self.reference_image.data.shape[-2],
144             self.thickness]
```

A.6. TEMUL Code

```
142
143     self.error_between_comparison_and_reference_image = []
144     self.error_between_images_history = []
145     if len(self.error_between_comparison_and_reference_image)
146         != 0:
147         self.error_between_images_history.append(
148             self.error_between_comparison_and_reference_image)
149
150     def _reference_image_init(self):
151         axes = self.reference_image.axes_manager
152         axes[-1].scale = axes[-2].scale = self.sampling
153         axes[-1].units = axes[-2].units = 'Å'
154
155     def _comparison_image_init(self, comparison_image):
156
157         if comparison_image is None:
158             print("Warning: "
159                  "comparison_image is set to None. You will not
160                  be able to "
161                  "refine the model until a comparison_image is
162                  set. You can "
163                  "do this via Model_refiner.create_simulation()
164 or by "
165                  "setting the Model_Refiner.comparison_image to
166                  an image.")
167         else:
168             if not isinstance(comparison_image,
169                               hyperspy._signals.signal2d.Signal2D):
170                 :
171                     raise ValueError(
172                         "comparison_image must be a 2D Hyperspy signal
173                         of type "
174                         "hyperspy._signals.signal2d.Signal2D. The
175                         current incorrect "
176                         "type is {}".format(str(type(comparison_image))
177 ))
178
179         for sublattice in self.sublattice_list:
180             if not comparison_image.data.shape == sublattice.
181             image.shape:
182                 print("Warning: "
183                     "comparison_image must have the same
184                     shape as each "
185                     "sublattice image. comparison_image
186                     shape is {}, "
```

A.6. TEMUL Code

```
175                         "while sublattice '()' is {}. This will
176                         stop you from "
177                         "refining your model.".format(
178                             comparison_image.data.shape,
179                             sublattice.name,
180                             sublattice.image.data))
181
182             comparison_image.axes_manager = self.reference_image.
183             axes_manager
184
185             self.comparison_image = comparison_image
186
187     def _comparison_image_warning(self, error_message=['None', 'wrong_size']):
188         if 'None' in error_message:
189             if self.comparison_image is None:
190                 raise ValueError(
191                     "The comparison_image attribute has not been "
192                     "set. You will not be able to "
193                     "refine the model until a comparison_image is "
194                     "set. You can "
195                     "do this via Model_refiner.create_simulation()"
196                     "or by "
197                     "setting the Model_Refiner.comparison_image to "
198                     "an image.")
199
200         if 'wrong_size' in error_message:
201             for sublattice in self.sublattice_list:
202                 if not self.comparison_image.data.shape ==
203                     sublattice.image.shape:
204                     raise ValueError(
205                         "comparison_image must have the same shape "
206                         "as each "
207                         "sublattice image. comparison_image shape "
208                         "is {}, ".format(
209                             "while sublattice '()' is {}".format(
210                                 self.comparison_image.data.shape,
211                                 sublattice.name,
212                                 sublattice.image.data)))
213
214     def __repr__(self):
215         return '<%s, %s (sublattices:%s,element_list:%s)>' % (
216             self.__class____name__,
217             self.name,
218             len(self.sublattice_list),
219             len(self.element_list),
```

A.6. TEMUL Code

```
212     )
213
214     def _auto_mask_radii_init(self):
215         '''
216             Automatically set the mask_radius for each sublattice.
217             Use mask_radius='auto' when calling a Model Refiner method
218
219             Ideally you could have the mask_radius/percent_to_nn
220             change depending
221                 on the atom, but for now depending on the sublattice is
222             okay.
223             '''
224
225             mask_radii = []
226             for sublattice in self.sublattice_list:
227                 element_config = get_most_common_sublattice_element(
228                     sublattice, info='element')
229                 chemical = split_and_sort_element(element_config)
230
231                 [0][1]
232                 radius = atomic_radii_in_pixels(self.sampling,
233                     chemical)
234
235                 mask_radii.append(radius)
236
237
238             self.auto_mask_radius = mask_radii
239
240
241             @property
242             def element_count(self):
243                 self._element_count = count_atoms_in_sublattice_list(
244                     self.sublattice_list, filename=None)
245
246                 return self._element_count
247
248
249             def update_element_count_history(self):
250                 self.element_count
251
252                 self.element_count_history_list.append(self._element_count
253             )
254
255
256             def update_refinement_history(self, refinement_method):
257                 self.refinement_history.append(refinement_method)
258
259
260             def update_element_count_and_refinement_history(self,
261                 refinement_method):
262
263                 self.update_element_count_history()
264
265                 self.update_refinement_history(refinement_method)
266
267
268             def compare_latest_element_counts(self):
```

A.6. TEMUL Code

```
251     if len(self.element_count_history_list) < 2:
252         return False
253         # raise ValueError("element_count_history must have at
254         # least two ")
255         #                                     "element_counts for comparison")
256     else:
257         return(compare_count_atoms_in_sublattice_list(
258             self.element_count_history_list[-2:]))
259
260
261     def get_element_count_as_dataframe(self):
262
263         elements_ = [i for sublist in self.element_list for i in
264         sublist]
265         elements_ = list(set(elements_))
266         elements_.sort()
267
268         df = pd.DataFrame(columns=elements_)
269         for element_in_history in self.element_count_history_list:
270             df = df.append(element_in_history, ignore_index=True).
271             fillna(0)
272             for i, refinement_name in enumerate(self.
273             refinement_history):
274                 df.rename(index={i: str(i) + " " + refinement_name},
275                 inplace=True)
276
277             return df
278
279
280     def get_individual_elements_as_dataframe(self, split_symbol=[
281         '_', '.']):
282
283         df_all = self.get_element_count_as_dataframe()
284         indiv_element_list =
285         get_individual_elements_from_element_list(
286             self.element_list, split_symbol=split_symbol)
287         indiv_element_counts = count_all_individual_elements(
288             indiv_element_list, df_all)
289         df = pd.DataFrame.from_dict(indiv_element_counts)
290         return df
291
292
293     def combine_individual_and_element_counts_as_dataframe(
294         self, split_symbol=[ '_', '.']):
295
296         df_configs = self.get_element_count_as_dataframe()
297         df_indiv = self.get_individual_elements_as_dataframe(
298             split_symbol=[ '_', '.'])
```

A.6. TEMUL Code

```
290     df_combined = pd.concat([df_configs, df_indiv], axis=1)
291     return df_combined
292
293     def plot_element_count_as_bar_chart(self, element_configs=0,
294                                         flip_colrows=True,
295                                         title="Refinement of
296                                         Elements",
297                                         fontsize=16, split_symbol
298                                         =[',', ',']):
299         if element_configs == 0: # only element configs
300             df = self.get_element_count_as_dataframe()
301         elif element_configs == 1: # only individual elements
302             df = self.get_individual_elements_as_dataframe(
303                 split_symbol=[',', ','])
304         elif element_configs == 2: # both element configs +
305             individual
306             df = self.
307             combine_individual_and_element_counts_as_dataframe(
308                 split_symbol=[',', ','])
309         else:
310             raise ValueError(
311                 "element_configs can only be 0, 1, or 2. "
312                 "0 returns only the element configurations given
313                 in "
314                 "self.element_list ({}). "
315                 "1 returns the individual elements ({}). "
316                 "2 returns a combination of 1 and 2.".format(
317                     self.element_list,
318                     get_individual_elements_from_element_list(
319                         self.element_list,
320                         split_symbol=split_symbol)))
321
322         if flip_colrows:
323             df = df.T
324
325         df.plot.bar(fontsize=fontsize)
326         plt.title(title, fontsize=fontsize + 4)
327         plt.ylabel('Element Count', fontsize=fontsize)
328         plt.legend(loc=0, fontsize=fontsize - 4)
329         # plt.gca().axes.get_xaxis().set_visible(False)
330         plt.tight_layout()
331
332     def set_calibration_area(self, manual_list=None):
333         ''
334         Area that will be used to calibrate a simulation. The
335         pixel separation
```

A.6. TEMUL Code

```
329         can be set with set_calibration_separation. The average
330         intensity of
331             the atoms chosen by this separation within the area will
332             be set to 1.
333             The idea is to calibrate the experimental and simulated
334             images with a
335                 known intensity (e.g., single Mo atom in MoS2 is
336                 relatively
337                     consistant).
338
339             reference_image can be changed via the Model_refiner.
340             reference_image
341                 attribute.
342
343             manual_list must be a list of two lists, each of length
344             two.
345             For example: [ [0,0], [50, 50] ]
346             '',
347
348             if isinstance(manual_list, list):
349                 self.calibration_area = manual_list
350             else:
351                 self.calibration_area = add_atoms_with_gui(self.
352             reference_image)
353
354             def set_calibration_separation(self, pixel_separation):
355                 self.calibration_separation = pixel_separation
356
357             def update_error_between_comparison_and_reference_image(self):
358
359                 mse_number, ssm_number = measure_image_errors(
360                     imageA=self.reference_image.data,
361                     imageB=self.comparison_image.data,
362                     filename=None)
363
364                 self.error_between_comparison_and_reference_image = [
365                     mse_number, ssm_number]
366                 self.error_between_images_history.append(
367                     self.error_between_comparison_and_reference_image)
368
369             def plot_error_between_comparison_and_reference_image(self,
370                     style='plot'):
371
372                 errors = self.error_between_images_history
373
374                 mse = [i[0] for i in errors]
```

A.6. TEMUL Code

```
367     ssm = [i[1] for i in errors]
368     x = range(0, len(errors))
369
370     plt.figure()
371     if style == 'plot':
372         plt.plot(x, mse, color='b', marker='o',
373                   linestyle='-', label='Mean Square Error')
374         plt.plot(x, ssm, color='r', marker='^',
375                   linestyle='--', label='Struc. Sim. Index')
376     elif style == 'scatter':
377         plt.scatter(x=x, y=mse, color='b', marker='o',
378                     label='Mean Square Error')
379         plt.scatter(x=x, y=ssm, color='r', marker='^',
380                     label='Struc. Sim. Index')
381     plt.title("Reference and Comparison Image Diff.", fontsize
382 =16)
383     plt.xlabel("Simulation Order")
384     plt.ylabel("MSE and SSM Value")
385     plt.legend()
386
387     def update_sublattices_positions(self, sublattice_list):
388
389         positions = get_positions_from_sublattices(sublattice_list
390     )
391         self._sublattices_positions = positions
392
393     def set_thickness(self, thickness):
394         self.thickness = thickness
395
396     def set_image_xyz_sizes(self, xyz_list):
397         self.image_xyz_sizes = xyz_list
398
399     # tools for reverting to the previous version of the refiner
400     def _save_refiner_instance(self):
401
402         self.previous_refiner_instance = copy.deepcopy(self)
403
404     def revert_to_previous_refiner_instance(self):
405         print("This doesn't seem to work as intended."
406              "If you wish to revert to the previous version of
407              the refiner, "
408              "you can create a new object. For example: "
409              "refiner_2 = refiner_1.previous_refiner_instance")
410         self = self.previous_refiner_instance
411
412     # tools for using refinement algorithm
```

A.6. TEMUL Code

```
410     def image_difference_intensity_model_refiner(
411         self,
412         sublattices='all',
413         comparison_image='default',
414         change_sublattice=True,
415         filename=None,
416         verbose=False,
417         refinement_method="Intensity"):
418     """
419         Change the elements for sublattice atom positions that don
420         't match with
421         comparison_image.
422
423         See image_difference_intensity for details.
424
425         Parameters
426         -----
427         See image_difference_intensity for parameter information.
428         refinement_method : string, default "Intensity"
429             Name passed to self.refinement_history for tracking
430             purposes.
431
432         Examples
433         -----
434         >>> from temul.dummy_data import (
435             get_model_refiner_with_12_vacancies_refined)
436         >>> refiner = get_model_refiner_with_12_vacancies_refined(
437             ...           image_noise=True)
438         Changing some atoms
439         Changing some atoms
440         >>> history = refiner.element_count_history_list
441         >>> refiner.
442         combine_individual_and_element_counts_as_dataframe()
443
444             Ti_0   Ti_1   Ti_2   Ti_3   Ti_4   Ti_5
445             Ti
446             0 Initial State   0.0   0.0   400.0   0.0   0.0   0.0
447             800.0
448             1 Intensity      0.0   12.0   388.0   0.0   0.0   0.0
449             788.0
450             2 Intensity      12.0   0.0   388.0   0.0   0.0   0.0
451             776.0
452             3 Intensity      12.0   0.0   388.0   0.0   0.0   0.0
453             776.0
454             >>> refiner.plot_element_count_as_bar_chart(
455                 ...           element_configs=2, flip_colrows=True, fontsize=24)
456             >>> refiner.plot_element_count_as_bar_chart(
```

A.6. TEMUL Code

```
448     ...      element_configs=2, flip_colrows=False, fontsize  
449     =24)  
450     >>> refiner.sublattice_list[0].plot()  
451     >>> refiner.comparison_image.plot()  
452     , , ,  
453  
454     self._comparison_image_warning()  
455     self._save_refiner_instance()  
456  
457     # define variables for refinement  
458     if 'all' in sublattices:  
459         sublattice_list = self.sublattice_list  
460         element_list = self.element_list  
461         mask_radius_list = self.auto_mask_radius  
462     elif isinstance(sublattices, list):  
463         sublattice_list = [self.sublattice_list[i] for i in  
sublattices]  
464         element_list = [self.element_list[i] for i in  
sublattices]  
465         mask_radius_list = [self.auto_mask_radius[i] for i in  
sublattices]  
466  
467     if 'default' in comparison_image:  
468         comparison_image = self.comparison_image  
469  
470     for sublattice, element_list_i, mask_radius in zip(  
471             sublattice_list, element_list, mask_radius_list):  
472  
473         image_difference_intensity(  
474             sublattice=sublattice,  
475             sim_image=comparison_image,  
476             element_list=element_list_i,  
477             filename=filename,  
478             percent_to_nn=None,  
479             mask_radius=mask_radius,  
480             change_sublattice=change_sublattice,  
481             verbose=verbose)  
482  
483         self.update_element_count_and_refinement_history(  
refinement_method)  
484  
485     def repeating_intensity_refinement(self, n=5,  
486                                         sublattices='all',  
487                                         comparison_image='default',  
change_sublattice=True,
```

A.6. TEMUL Code

```

489                                         filename=None ,
490                                         verbose=False ,
491
492     ignore_element_count_comparison=False):
493
494         self._comparison_image_warning()
495         self._save_refiner_instance()
496
497         for i in range(n):
498             self.image_difference_intensity_model_refiner(
499                 sublattices=sublattices ,
500                 comparison_image=comparison_image ,
501                 change_sublattice=change_sublattice ,
502                 filename=filename ,
503                 verbose=verbose)
504
505
506         if not ignore_element_count_comparison:
507
508             if self.compare_latest_element_counts():
509                 print("The latest refinement did not change
the model. "
510
511                 "Exiting the refinement after {} loops.
To ignore "
512
513                 "this, set
514 ignore_element_count_comparison=True."
515                 .format(i + 1))
516
517             break
518
519
520     def image_difference_position_model_refiner(
521         self ,
522         chosen_sublattice ,
523         sublattices='all' ,
524         comparison_sublattice_list='auto' ,
525         comparison_image='default' ,
526         pixel_threshold=5,
527         num_peaks=5,
528         inplace=True ,
529         filename=None ,
530         refinement_method="Position"):
531
532         """
533
534             Find new atom positions that were perhaps missed by the
initial
535
536             position finding steps. Each sublattice will be updated
with new atom
537
538             positions if new atom positions are found.

```

A.6. TEMUL Code

```
529     See image_difference_position for details.
530
531     Parameters
532     -----
533     See image_difference_position for parameter information.
534     refinement_method : string, default "Position"
535         Name passed to self.refinement_history for tracking
536         purposes.
537
538     Examples
539     -----
540
541     >>> from temul.dummy_data import (
542     ...      get_model_refiner_one_sublattice_3_vacancies)
543     >>> refiner = get_model_refiner_one_sublattice_3_vacancies
544     ()
545
546     >>> refiner.sublattice_list[0].plot()
547     >>> refiner.comparison_image.plot()
548     >>> refiner.image_difference_position_model_refiner(
549     ...      pixel_threshold=10)
550     3 new atoms found! Adding new atom positions.
551     >>> refiner.sublattice_list[0].plot()
552
553
554     Combination of Refinements (cont.)
555
556
557     >>> refiner.image_difference_intensity_model_refiner()
558     Changing some atoms
559     >>> refiner.image_difference_intensity_model_refiner()
560     Changing some atoms
561     >>> history_df = refiner.get_element_count_as_dataframe()
562     >>> refiner.plot_element_count_as_bar_chart(
563     ...      element_configs=2, flip_colrows=True, fontsize=24)
564
565     ...
566
567
568     self._comparison_image_warning()
569     self._save_refiner_instance()
570
571
572     # define variables for refinement
573     if 'all' in sublattices:
574         sublattice_list = self.sublattice_list
575         mask_radius_list = self.auto_mask_radius
576     elif isinstance(sublattices, list):
577         sublattice_list = [self.sublattice_list[i] for i in
578                           sublattices]
579         print(
```

A.6. TEMUL Code

```
571         "Warning: Setting 'sublattices' to a list can
572 cause overwrite"
573         "errors. Best use 'sublattice='all'' until this is
574 fixed")
575         mask_radius_list = [self.auto_mask_radius[i] for i in
576 sublattices]
577
578     # elif isinstance(positions_from_sublattices, list):
579     #     positions_from_sublattices = [
580     #         positions_from_sublattices[i] for i in
581     #         positions_from_sublattices] # not correct, would
582 be nice to
583     #             have
584
585     if 'default' in comparison_image:
586         comparison_image = self.comparison_image
587
588     # if you set sublattices=[1], you get an error that the
589 line
590     # self.sublattice_list[i] = sublattice will overwrite the
591 first
592     # sublattice with the sublattice_list[1] sublattice! could
593 just not
594     # have sublattice as an option! see print warning above!
595     for i, (sublattice, mask_radius) in enumerate(zip(
596         sublattice_list, mask_radius_list)):
597
598         if sublattice == chosen_sublattice:
599
600             # update the positions_from_sublattices before
601 running the next
602             # sublattice if chosen
603             if comparison_sublattice_list == 'auto':
604
605                 comparison_sublattice_list = self.
606 sublattice_list
607
608             elif comparison_sublattice_list == 'each':
609                 comparison_sublattice_list = []
610                 comparison_sublattice_list.append(self.
611 sublattice_list[i])
612
613                 sublattice = image_difference_position(
614                     sublattice=sublattice,
615                     sim_image=comparison_image,
616                     pixel_threshold=pixel_threshold,
```

A.6. TEMUL Code

```
607         comparison_sublattice_list=
608         comparison_sublattice_list,
609             filename=filename,
610             percent_to_nn=None,
611             mask_radius=mask_radius,
612             num_peaks=num_peaks,
613             inplace=inplace)
614
615         self.sublattice_list[i] = sublattice
616
617     self.update_element_count_and_refinement_history(
618         refinement_method)
619
620     def create_simulation(
621         self,
622         sublattices='all',
623         filter_image=False,
624         calibrate_image=True,
625         xyz_sizes=None,
626         header_comment='example',
627         filename='refiner_simulation',
628         reference_image='auto',
629         probeStep=1.0,
630         E0=60e3,
631         integrationAngleMin=0.085,
632         integrationAngleMax=0.186,
633         detectorAngleStep=0.001,
634         interpolationFactor=16,
635         realspacePixelSize=0.0654,
636         numFP=1,
637         cellDimXYZ=None,
638         tileXYZ=None,
639         probeSemiangle=0.030,
640         alphaBeamMax=0.032,
641         scanWindowMin=0.0,
642         scanWindowMax=1.0,
643         numThreads=2,
644         algorithm="prism",
645         delta_image_filter=0.5,
646         max_sigma=6,
647         mask_radius='auto',
648         percent_to_nn=None,
649         refine=True,
650         sigma_offset=0):
651         """
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635
2636
2637
2638
2639
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680
2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2
```

A.6. TEMUL Code

```
650     Create and simulate a .xyz file from the sublattice
651     information.
652
653     Uses the pyprismatic prism algorithm by default.
654
655     Note: The error resulting from filter=True with
656     percent_to_nn set can
657     be fixed by setting mask_radius instead of percent_to_nn.
658     Error is: "TypeError: 'NoneType' object is not iterable"
659
660     Parameters
661     -----
662     sublattices : Atomap Sublattices, default 'all'
663         If set to 'all', sublattices that exist in the 'Model_Refiner' will
664         all be used. The 'sublattice' indexes can be specified
665         in a list.
666         For example [0, 2] will select the first and third
667         sublattices.
668         A list of 'sublattice' objects can instead be used.
669         filter_image : Bool, default False
670             Choose whether to filter the simulation with a
671             Gaussian to match
672             the 'reference_image'.
673             calibrate_image : Bool, default True
674                 Choose whether to calibrate the simulation with a set
675                 'calibration_area' and 'calibration_separation'.
676             xyz_sizes : list, default None
677                 List of the x, y, z physical size of the
678                 reference_image in
679                 angstrom. If None is set, the sizes will be
680                 automatically taken
681                 from the 'image_xyz_sizes' attribute.
682             header_comment : string, default 'example'
683                 The first line comment for the .xyz file.
684             filename : string, default 'refiner_simulation'
685                 filename with which the .xyz file and simulated .mrc
686                 file will be
687                 saved.
688             delta_image_filter : float, default 0.5
689                 The change in sigma for the 'filter' Gaussian filter.
690                 Small values
691                     will slow the 'filter' down.
692             max_sigma : float, default 6
693                 The maximum sigma used for the 'filter' Gaussian
694                 filter. Large
695                     values will slow the 'filter' down.
```

```
685     percent_to_nn : float, default None
686         The percentage distance to the nearest neighbour atom
687         used for
688             atomap atom position refinement.
689         mask_radius : float, default 'auto'
690             The pixel distance to the nearest neighbour atom used
691             for atomap
692                 atom position refinement
693                 refine : Bool, default True
694                     Whether to refine the atom positions for the 'calibrate' parameter.
695
696             Note: The error resulting from filter=True with
697             percent_to_nn set can
698                 be fixed by setting mask_radius instead of percent_to_nn.
699
700             Examples
701             -----
702
703             Returns
704             -----
705             Updates the comparison_image attribute with the newly
706             simulated model.
707             """
708
709             if 'all' in sublattices:
710                 sublattice_list = self.sublattice_list
711                 element_list = self.flattened_element_list
712                 elif isinstance(sublattices, list):
713                     sublattice_list = [self.sublattice_list[i] for i in
714                     sublattices]
715                     element_list = [self.element_list[i] for i in
716                     sublattices]
717                     element_list = combine_element_lists(element_list)
718
719                     if xyz_sizes is None:
720                         x_size = self.image_xyz_sizes[0]
721                         y_size = self.image_xyz_sizes[1]
722                         z_size = self.image_xyz_sizes[2]
723
724                         create_dataframe_for_xyz(
725                             sublattice_list=sublattice_list,
726                             element_list=element_list,
727                             x_size=x_size,
728                             y_size=y_size,
```

A.6. TEMUL Code

```
724     z_size=z_size,
725     filename=filename + '_xyz_file',
726     header_comment=header_comment)
727
728     if reference_image == 'auto':
729         reference_image = self.reference_image
730
731     simulate_with_prismatic(
732         xyz_filename=filename + '_xyz_file.xyz',
733         filename=filename + '_mrc_file',
734         reference_image=reference_image,
735         probeStep=probeStep,
736         E0=E0,
737         integrationAngleMin=integrationAngleMin,
738         integrationAngleMax=integrationAngleMax,
739         detectorAngleStep=detectorAngleStep,
740         interpolationFactor=interpolationFactor,
741         realspacePixelSize=realspacePixelSize,
742         numFP=numFP,
743         cellDimXYZ=cellDimXYZ,
744         tileXYZ=tileXYZ,
745         probeSemiangle=probeSemiangle,
746         alphaBeamMax=alphaBeamMax,
747         scanWindowMin=scanWindowMin,
748         scanWindowMax=scanWindowMax,
749         algorithm=algorithm,
750         numThreads=numThreads)
751
752     simulation = load_prismatic_mrc_with_hyperspy(
753         'prism_2Doutput_' + filename + '_mrc_file.mrc',
754         save_name=None)
755
756     if mask_radius == 'auto':
757         mask_radius = np.mean(self.auto_mask_radius)
758
759     if filter_image:
760         # set filename as not None - good to output the mse
761         ssm plot
762         simulation =
763         compare_two_image_and_create_filtered_image(
764             image_to_filter=simulation,
765             reference_image=self.reference_image,
766             delta_image_filter=delta_image_filter,
767             cropping_area=self.calibration_area,
768             separation=self.calibration_separation,
769             filename=None,
```

A.6. TEMUL Code

```
767         max_sigma=max_sigma ,
768         percent_to_nn=percent_to_nn ,
769         mask_radius=mask_radius ,
770         refine=False ,
771         sigma_offset=sigma_offset)

772

773     if calibrate_image:
774         calibrate_intensity_distance_with_sublattice_roi(
775             image=simulation ,
776             cropping_area=self.calibration_area ,
777             separation=self.calibration_separation ,
778             filename=None ,
779             reference_image=self.reference_image ,
780             percent_to_nn=percent_to_nn ,
781             mask_radius=mask_radius ,
782             refine=refine ,
783             scalebar_true=True)

784

785     self._save_refiner_instance()
786     self._comparison_image_init(simulation)
787     self.update_error_between_comparison_and_reference_image()

788

789 def calibrate_comparison_image(self ,
790                                 filename=None ,
791                                 percent_to_nn=None ,
792                                 mask_radius='auto' ,
793                                 refine=True):

794

795     self._comparison_image_warning()

796

797     if mask_radius == 'auto':
798         mask_radius = np.mean(self.auto_mask_radius)

799

800     calibrate_intensity_distance_with_sublattice_roi(
801         image=self.comparison_image ,
802         cropping_area=self.calibration_area ,
803         separation=self.calibration_separation ,
804         filename=filename ,
805         reference_image=self.reference_image ,
806         percent_to_nn=percent_to_nn ,
807         mask_radius=mask_radius ,
808         refine=refine ,
809         scalebar_true=True)

810

811 def plot_reference_and_comparison_images(
```

A.6. TEMUL Code

```
812         self, title_reference='Reference', title_comparison='
813         Comparison'):
814     _, (ax1, ax2) = plt.subplots(1, 2)
815     ax1.imshow(self.reference_image.data)
816     ax2.imshow(self.comparison_image.data)
817     ax1.set_title(title_reference)
818     ax2.set_title(title_comparison)
819     plt.show()
```

Code Segment A.3: The developed Model Refiner class of the TEMUL toolkit discussed in Chapter 5.

A.6.4 Some functions of the Polarisation Module

For the full TEMUL code, see <https://github.com/PinkShnack/TEMUL>.

```
1
2 import numpy as np
3 import scipy
4 import matplotlib.pyplot as plt
5 import matplotlib.colors as colors
6 from matplotlib.cm import ScalarMappable
7 from decimal import Decimal
8
9 # For the full TEMUL code, see https://github.com/PinkShnack/TEMUL
10
11 def plot_polarisation_vectors(x, y, u, v, image,
12                                 sampling=None, units='pix',
13                                 plot_style='vector',
14                                 overlay=True, normalise=False,
15                                 save='polarisation_image', title="",
16                                 color='yellow', cmap=None,
17                                 monitor_dpi=96,
18                                 pivot='middle', angles='xy',
19                                 scale_units='xy',
20                                 scale=None, headwidth=3.0,
21                                 headlength=5.0,
22                                 headaxislength=4.5, no_axis_info=
23                                 True):
24     """
25     Plot the polarisation vectors.
```

Parameters

See `matplotlib's quiver()` function for more details.

A.6. TEMUL Code

```
26
27     x, y : list or 1D NumPy array
28         xy coordinates on the image
29     u, v : list or 1D NumPy array
30         uv vector components
31     image : 2D NumPy array
32     sampling : float, default None
33         Pixel sampling of the image for calibration.
34     units : string, default "pix"
35         Units used to display the magnitude of the vectors.
36     plot_style : string, default "vector"
37         Options are "vector", "colormap", "contour"
38     overlay : Bool, default True
39         If set to True, the 'image' will be plotting behind the
arrows
40     normalise : Bool, default False
41         Normalise the vectors to unit vectors for plotting
purposes.
42         Magnitude will still be displayed correctly.
43     save : string, default "polarisation_image"
44         If set to 'save=None', the array will not be saved.
45     title : string, default ""
46         Title of the plot.
47     color : string, default "r"
48         Color of the arrows when 'plot_style="vector" or "contour
".
49     cmap : matplotlib colormap, default "viridis"
50     monitor_dpi : int, default 96
51         The DPI of the monitor, generally 96 pixels. Used to scale
the image
52         so that large images render correctly. Use a smaller value
or
53         'monitor_dpi=None' to enlarge too-small images.

54
55 See matplotlib's quiver function for the remaining parameters.

56
57 Examples
58 -----
59
60 >>> import matplotlib.pyplot as plt
61 >>> import numpy as np
62 >>> import atomap.api as am
63 >>> from temul.polarisation import (
64 ...     combine_atom_deviations_from_zone_axes,
...     plot_polarisation_vectors)
65 >>> atom_lattice = am.dummy_data.
get_polarization_film_atom_lattice()
```

A.6. TEMUL Code

```
66     >>> sublatticeA = atom_lattice.sublattice_list[0]
67     >>> sublatticeA.find_nearest_neighbors()
68     >>> sublatticeA.refine_atom_positions_using_center_of_mass()
69     >>> sublatticeA.construct_zone_axes()
70     >>> x, y, u, v = combine_atom_deviations_from_zone_axes(
71             sublatticeA,
72             ...           save=None)
73
74             vector plot with red arrows:
75
76     >>> plot_polarisation_vectors(x, y, u, v, image=sublatticeA.
77             image,
78             ...           normalise=False, save=None,
79             ...           plot_style='vector', color='r',
80             ...           overlay=False, title='Vector
81             Arrows',
82             ...           monitor_dpi=50)
83
84             vector plot with red arrows overlaid on the image:
85
86     >>> plot_polarisation_vectors(x, y, u, v, image=sublatticeA.
87             image,
88             ...           normalise=False, save=None,
89             ...           plot_style='vector', color='r',
90             ...           overlay=True, monitor_dpi=50)
91
92             vector plot with colormap viridis:
93
94     >>> plot_polarisation_vectors(x, y, u, v, image=sublatticeA.
95             image,
96             ...           normalise=False, save=None,
97             ...           plot_style='colormap',
98             ...           overlay=False, cmap='viridis')
99
100            colormap arrows with sampling applied:
101
102     >>> plot_polarisation_vectors(x, y, u, v, image=sublatticeA.
103             image,
104             ...           sampling=3.0321, units='pm',
105             ...           normalise=False, plot_style='
106             colormap',
107             ...           overlay=False, save=None, cmap='
108             viridis')
109
110            vector plot with colormap viridis and unit vectors:
```

A.6. TEMUL Code

```

104     >>> plot_polarisation_vectors(x, y, u, v, image=sublatticeA.
105     image,
106         ...                               normalise=True, save=None,
107         ...                               plot_style='colormap', color='r
108     ,
109         ...                               overlay=False, cmap='viridis')
110
111     normalised vectors on a contourf map:
112
113     >>> plot_polarisation_vectors(x, y, u, v, image=sublatticeA.
114     image,
115         ...                               normalise=True, plot_style='
116     contour',
117         ...                               overlay=False, pivot='middle',
118         ...                               color='darkgray', cmap='viridis
119     ,
120         save=None)
121
122     , ,
123
124     u, v = np.array(u), np.array(v)
125
126     if sampling is not None:
127         u, v = u * sampling, v * sampling
128
129     # for ax.quiver optional C parameter, we need to set this to
130     # something. None doesn't work!
131     vector_mags = get_vector_magnitudes(u, v)
132
133     if normalise:
134         # Normalise the data for uniform arrow size
135         u_norm = u / np.sqrt(u ** 2.0 + v ** 2.0)
136         v_norm = v / np.sqrt(u ** 2.0 + v ** 2.0)
137         u = u_norm
138         v = v_norm
139
140     if monitor_dpi is not None:
141         _, ax = plt.subplots(figsize=[image.shape[1] / monitor_dpi
142         ,
143                               image.shape[0] / monitor_dpi
144     ])
145
146     else:
147         _, ax = plt.subplots()
148
149     ax.set_title(title, loc='left', fontsize=20)
150
151     # plot_style options
152     if plot_style == "vector":

```

A.6. TEMUL Code

```
143     Q = ax.quiver(
144         x, y, u, v, color=color, pivot=pivot, angles=angles,
145         scale_units=scale_units, scale=scale, headwidth=
146         headwidth,
147         headlength=headlength, headaxislength=headaxislength)
148         length = np.max(np.hypot(u, v)) / 2
149         ax.quiverkey(Q, 0.9, 1.025, length,
150             label='{:0E} {}'.format(Decimal(length),
151             units),
152             labelpos='E', coordinates='axes')
153
154     elif plot_style == "colormap":
155         if cmap is None:
156             cmap = 'viridis'
157         ax.quiver(
158             x, y, u, v, vector_mags, color=color, cmap=cmap,
159             pivot=pivot, angles=angles, scale_units=scale_units,
160             scale=scale, headwidth=headwidth,
161             headlength=headlength, headaxislength=headaxislength)
162
163         norm = colors.Normalize(vmin=np.min(vector_mags),
164                             vmax=np.max(vector_mags))
165         sm = plt.cm.ScalarMappable(cmap=cmap, norm=norm)
166         sm.set_array([])
167         cbar = plt.colorbar(mappable=sm)
168         cbar.ax.set_ylabel('Vector Magnitude {}'.format(units))
169
170     elif plot_style == 'contour':
171         if cmap is None:
172             cmap = 'viridis'
173
174         contour_map = plt.tricontourf(x, y, vector_mags, cmap=cmap)
175
176         ax.quiver(
177             x, y, u, v, color=color, pivot=pivot,
178             angles=angles, scale_units=scale_units,
179             scale=scale, headwidth=headwidth,
180             headlength=headlength, headaxislength=headaxislength)
181
182         ax.set(aspect='equal')
183         ax.set_xlim(0, image.shape[1])
184         ax.set_ylim(image.shape[0], 0)
185
186     if plot_style == 'contour':
```

A.6. TEMUL Code

```
185     cbar = plt.colorbar(mappable=contour_map, fraction=0.046,
186     pad=0.04)
187     cbar.ax.tick_params(labelsize=14)
188     cbar.ax.set_ylabel('Vector Magnitude {}'.format(units),
189                         fontsize=14)
190     if overlay:
191         plt.imshow(image)
192
193     if no_axis_info:
194         plt.gca().axes.get_xaxis().set_visible(False)
195         plt.gca().axes.get_yaxis().set_visible(False)
196
197     # plt.tight_layout()
198     if save is not None:
199         plt.savefig(fname=save + '_' + plot_style + '.png',
200                     transparent=True, frameon=False, bbox_inches='tight',
201                     pad_inches=None, dpi=300, labels=False)
202
203
204 def atom_deviation_from_straight_line_fit(sublattice,
205                                         axis_number: int = 0,
206                                         save: str = ''):
207     """
208         delete atom_planes from a zone axis. Can choose whether to
209         delete
210         every second, third etc., and the offset from the zero index.
211
212         Parameters
213         -----
214         sublattice : Atomap Sublattice object
215         axis_number : int, default 0
216             The index of the zone axis (translation symmetry) found by
217             the Atomap
218             function 'construct_zone_axes()' .
219         save : string, default ''
220             If set to 'save=None', the array will not be saved.
221
222         Returns
223         -----
224         Four lists: x, y, u, and v where x,y are the original atom
225         position
226         coordinates (simply sublattice.x_position, sublattice.
227         y_position) and
```

A.6. TEMUL Code

```
224     u,v are the polarisation vector components pointing to the new
225     coordinate.
226
227     These can be input to 'plot_polarisation_vectors()' for
228     visualisation.
229
230
231     Examples
232     -----
233
234     >>> import atomap.api as am
235     >>> from temul.polarisation import
236     atom_deviation_from_straight_line_fit
237     >>> atom_lattice = am.dummy_data.
238     get_polarization_film_atom_lattice()
239     >>> sublatticeA = atom_lattice.sublattice_list[0]
240     >>> sublatticeA.find_nearest_neighbors()
241     >>> sublatticeA.refine_atom_positions_using_center_of_mass()
242     >>> sublatticeA.construct_zone_axes()
243     >>> x,y,u,v = atom_deviation_from_straight_line_fit(
244     sublatticeA, save=None)
245
246
247     This polarisation can then be visualised in
248     plot_polarisation_vectors()
249
250     '',
251
252
253     if sublattice.zones_axis_average_distances is None:
254         raise Exception(
255             "zones_axis_average_distances is empty. "
256             "Has sublattice.construct_zone_axes() been run?")
257     else:
258         zon_vec_needed = sublattice.zones_axis_average_distances[
259             axis_number]
260
261         original_atom_pos_list = []
262         new_atom_pos_list = []
263         new_atom_diff_list = []
264
265
266         # this loop creates two arrays.
267         # the original array contains all the original atom positions
268         # the new array contains all the xy positions on the fitted
269         straight
270         # lines the new array positions are the point at which the
271         original
272         # position is perpendicular to the fitted line.
273         for i, atom_plane in enumerate(sublattice.atom_plane_list):
274
275             if sublattice.atom_plane_list[i].zone_vector ==
276                 zon_vec_needed:
```

A.6. TEMUL Code

```
260     original_atoms_list = []
261     for atom_pos in sublattice.atom_plane_list[i].
262         atom_list:
263             original_atoms_list.append(
264                 [atom_pos.pixel_x, atom_pos.pixel_y])
265
266     original_atoms_array = np.array(original_atoms_list)
267
268     slope, intercept = scipy.polyfit(
269         original_atoms_array[:, 0], original_atoms_array
270         [:, 1], 1)
271
272     slope_neg_inv = -(1 / slope)
273     angle = np.arctan(slope_neg_inv) # * (180/np.pi)
274
275     x1 = atom_plane.start_atom.pixel_x
276     y1 = slope * x1 + intercept
277     x2 = atom_plane.end_atom.pixel_x
278     y2 = slope * x2 + intercept
279
280     p1 = np.array((x1, y1), ndmin=2)
281     # end xy coord for straight line fit
282     p2 = np.array((x2, y2), ndmin=2)
283
284     atoms_on_plane_list = []
285     atom_dist_diff_list = []
286     # original_atom position, point an arrow towards it by
287     using
288     # original_atom_pos_array and new_atom_diff_array,
289     # or away using new_atom_pos_array and -
290     new_atom_diff_array
291     for original_atom in original_atoms_array:
292
293         distance = np.cross(p2 - p1, original_atom -
294             p1) / np.linalg.norm(p2 - p1)
295         distance = float(distance)
296         x_diff = distance * np.cos(angle)
297         y_diff = distance * np.sin(angle)
298
299         x_on_plane = original_atom[0] + x_diff
300         y_on_plane = original_atom[1] + y_diff
301
302         atoms_on_plane_list.append([x_on_plane, y_on_plane])
303     )
304     atom_dist_diff_list.append([x_diff, y_diff])
305     # atoms_not_on_plane_list.append([original_atom])
```

A.6. TEMUL Code

```
301
302     original_atom_pos_list.extend(original_atoms_list)
303     new_atom_pos_list.extend(atoms_on_plane_list)
304     new_atom_diff_list.extend(atom_dist_diff_list)
305
306     original_atom_pos_array = np.array(original_atom_pos_list)
307     new_atom_pos_array = np.array(new_atom_pos_list)
308     distance_diff_array = np.array(new_atom_diff_list)
309
310     if save is not None:
311         np.save(save + '_sublattice_xy', original_atom_pos_array)
312         np.save(save + '_new_atom_positions_xy',
313             new_atom_pos_array)
314         np.save(save + '_vector_uv', distance_diff_array)
315
316     # this is the difference between the original position and the
317     # point on
318     # the fitted atom plane line. To get the actual shift
319     # direction, just
320     # use -new_atom_diff_array. (negative of it!)
321
322     x = [row[0] for row in original_atom_pos_list]
323     y = [row[1] for row in original_atom_pos_list]
324     u = [row[0] for row in new_atom_diff_list]
325     v = [row[1] for row in new_atom_diff_list]
326
327
328 def combine_atom_deviations_from_zone_axes(
329     sublattice, image=None, axes=None, sampling=None, units='pix',
330     plot_style='vector', overlay=True, normalise=False,
331     save='atom_deviation', title="", color='yellow', cmap=None
332     ,
333     pivot='middle', angles='xy', scale_units='xy', scale=None,
334     headwidth=3.0, headlength=5.0, headaxislength=4.5,
335     monitor_dpi=96):
336     """
337     Combine the atom deviations of each atom for all zone axes.
338     Good for plotting vortexes and seeing the total deviation from
339     a
340     perfect structure.
```

Parameters

A.6. TEMUL Code

```

340
341     sublattice : Atomap Sublattice object
342     For the remaining parameters see 'plot_polarisation_vectors()'
343     '.
344
345     Returns
346     -----
347     Four lists: x, y, u, and v where x,y are the original atom
348     position
349     coordinates (simply sublattice.x_position, sublattice.
350     y_position) and
351     u,v are the polarisation vector components pointing to the new
352     coordinate.
353     These can be input to 'plot_polarisation_vectors()' for
354     visualisation.
355
356     Examples
357     -----
358
359     >>> import atomap.api as am
360     >>> from temul.polarisation import (plot_polarisation_vectors,
361     ...     combine_atom_deviations_from_zone_axes)
362     >>> atom_lattice = am.dummy_data.
363     get_polarization_film_atom_lattice()
364     >>> sublatticeA = atom_lattice.sublattice_list[0]
365     >>> sublatticeA.find_nearest_neighbors()
366     >>> sublatticeA.refine_atom_positions_using_center_of_mass()
367     >>> sublatticeA.construct_zone_axes()
368     >>> x,y,u,v = combine_atom_deviations_from_zone_axes(
369     ...     sublatticeA, save=None)
370     >>> plot_polarisation_vectors(u, v, x, y, save=None,
371     ...     image=sublatticeA.image)
372
373
374     # You can also choose the axes:
375
376     # >>> x,y,u,v = combine_atom_deviations_from_zone_axes(
377     # ...     sublatticeA, axes=[0,3], save=None)
378     # >>> plot_polarisation_vectors(u, v, x, y, save=None,
379     # ...     image=sublatticeA.image)
380
381
382     ...
383
384
385     all_atoms_in_planes_xy = []
386     all_atoms_in_planes_uv = []
387
388
389     if sublattice.zones_axis_average_distances is None:

```

A.6. TEMUL Code

```
380     raise Exception(
381         "zones_axis_average_distances is empty. "
382         "Has sublattice.construct_zone_axes() been run?")
383 else:
384     pass
385
386 if axes is None:
387     axes_list = range(len(sublattice.
388 zones_axis_average_distances))
389 else:
390     axes_list = axes
391
392 for axis_number in axes_list:
393
394     x, y, u, v = atom_deviation_from_straight_line_fit(
395         sublattice=sublattice, axis_number=axis_number,
396         save=None)
397
398     xy_array = np.array([x, y]).T
399     atoms_in_plane_xy = [list(i) for i in xy_array]
400     uv_array = np.array([u, v]).T
401     atoms_in_plane_uv = [list(i) for i in uv_array]
402
403     all_atoms_in_planes_xy.extend(atoms_in_plane_xy)
404     all_atoms_in_planes_uv.extend(atoms_in_plane_uv)
405
406     sublattice_xy = []
407     for atom in sublattice.atom_list:
408         sublattice_xy.append([atom.pixel_x, atom.pixel_y])
409
410     atoms_not_found = []
411     combined_vectors = []
412     for atom_xy in sublattice_xy:
413         individual_vectors = []
414         for atom_along_plane_xy, atom_along_plane_uv in zip(
415             all_atoms_in_planes_xy, all_atoms_in_planes_uv):
416
417             if atom_xy == atom_along_plane_xy:
418                 individual_vectors.append(atom_along_plane_uv)
419
420             if len(individual_vectors) != 0:
421                 calc_combined_vectors = list(sum(np.array(
422                     individual_vectors)))
423                 combined_vectors.append(calc_combined_vectors)
424             else:
425                 atoms_not_found.append(atom_xy)
```

A.6. TEMUL Code

```
424
425     if len(atoms_not_found) != 0:
426         print("This sublattice_xy atom isn't included in the "
427               "axes given, removing atoms: {}".format(
428                   atoms_not_found))
429     for atom in atoms_not_found:
430         sublattice_xy.remove(atom)
431
432     if len(sublattice_xy) != len(combined_vectors):
433         raise ValueError("len(sublattice_xy) != len(
434             combined_vectors)")
435
436     x = [row[0] for row in sublattice_xy]
437     y = [row[1] for row in sublattice_xy]
438     u = [row[0] for row in combined_vectors]
439     v = [row[1] for row in combined_vectors]
440
441     if save is not None:
442         np.save(save + '_sublattice_xy', np.array(sublattice_xy))
443         np.save(save + '_vector_uv', np.array(combined_vectors))
444
445     if image is None:
446         image = sublattice.image
447
448         plot_polarisation_vectors(
449             u=u, v=v, x=x, y=y, image=image, sampling=sampling,
450             units=units,
451             plot_style=plot_style, overlay=overlay, normalise=
452             normalise,
453             save=save, title=title, color=color, cmap=cmap, pivot=
454             pivot,
455             angles=angles, scale_units=scale_units, scale=scale,
456             headwidth=headwidth, headlength=headlength,
457             headaxislength=headaxislength, monitor_dpi=monitor_dpi
458         )
459
460     return(x, y, u, v)
461
462
463
464 def get_average_polarisation_in_regions(x, y, u, v, image,
465 divide_into=8):
466     """
467     This function splits the image into a certain number of
468     regions and
469     averages and plots the polarisation vectors in those regions.
470 
```

A.6. TEMUL Code

```
463     Parameters
464     -----
465     x, y : list or 1D NumPy array
466         xy coordinates on the image
467     u, v : list or 1D NumPy array
468         uv vector components
469     image : 2D NumPy array
470     divide_into : int, default 8
471         The number used to divide the image up. If 8, then the
472         image will be
473             split into an 8x8 grid.
474
475     Returns
476     -----
477     Four lists: x_new, y_new, u_new, v_new.
478     x_new and y_new are the certain coordinates of the regions.
479     u_new and v_new are the averaged polarisation vectors.
480
481     Examples
482     -----
483     >>> import matplotlib.pyplot as plt
484     >>> import numpy as np
485     >>> import atomap.api as am
486     >>> from temul.polarisation import (
487     ...     combine_atom_deviations_from_zone_axes,
488     ...     plot_polarisation_vectors,
489     get_average_polarisation_in_regions)
490     >>> atom_lattice = am.dummy_data.
491     get_polarization_film_atom_lattice()
492     >>> sublatticeA = atom_lattice.sublattice_list[0]
493     >>> sublatticeA.find_nearest_neighbors()
494     >>> sublatticeA.refine_atom_positions_using_center_of_mass()
495     >>> sublatticeA.construct_zone_axes()
496
497     Get and plot the original polarisation vectors:
498
499     >>> x, y, u, v = combine_atom_deviations_from_zone_axes(
500     sublatticeA,
501     ...     save=None)
502     >>> plot_polarisation_vectors(x, y, u, v, image=sublatticeA.
503     image,
504     ...                     normalise=False, save=None,
505     ...                     plot_style='vector', color='r',
506     ...                     overlay=False, title='Actual Vector
507     Arrows',
508     ...                     monitor_dpi=50)
```

A.6. TEMUL Code

```
503     Get and plot the new, averaged polarisation vectors
504
505     >>> x_new, y_new, u_new, v_new =
506         get_average_polarisation_in_regions(
507             ...      x, y, u, v, image=sublatticeA.image, divide_into=8)
508     >>> plot_polarisation_vectors(x_new, y_new, u_new, v_new,
509             ...                  image=sublatticeA.image,
510             ...                  normalise=False, save=None,
511             ...                  plot_style='vector', color='r',
512             ...                  overlay=False, title='Averaged Vector
513             Arrows',
514             ...                  monitor_dpi=50)
515
516     '',
517
518     if divide_into >= np.sqrt(len(x)):
519         raise ValueError(
520             "divide_into ({}) cannot be greater than the number of
521             "
522             "vector coordinates in each dimension ({})".format(
523                 divide_into, np.sqrt(len(x))))
524
525     # divide the image into sections
526     image_x_max, image_y_max = image.shape[-1], image.shape[-2]
527     x_region_length = image_x_max // divide_into
528     y_region_length = image_y_max // divide_into
529
530     all_x_region_lengths = []
531     all_y_region_lengths = []
532     for i in range(divide_into):
533         all_x_region_lengths.append(
534             [i * x_region_length, (i + 1) * x_region_length])
535         all_y_region_lengths.append(
536             [i * y_region_length, (i + 1) * y_region_length])
537
538     # get the new x, y coords
539     x_new, y_new = [], []
540     for x_length in all_x_region_lengths:
541         for y_length in all_y_region_lengths:
542             x_new.append(x_length[1] - ((x_length[1] - x_length
543             [0]) / 2))
544             y_new.append(y_length[1] - ((y_length[1] - y_length
545             [0]) / 2))
546
547     # get the new averaged u, v components
548     u_new, v_new = [], []
```

A.6. TEMUL Code

```
544     for x_length in all_x_region_lengths:
545         for y_length in all_y_region_lengths:
546             u_area, v_area = [], []
547             for (x_i, y_i, u_i, v_i) in zip(x, y, u, v):
548                 if x_length[0] < x_i < x_length[1] and \
549                     y_length[0] < y_i < y_length[1]:
550
551                 u_area.append(u_i)
552                 v_area.append(v_i)
553
554             u_mean = np.mean(np.array(u_area))
555             u_new.append(u_mean)
556             v_mean = np.mean(np.array(v_area))
557             v_new.append(v_mean)
558
559     return(x_new, y_new, u_new, v_new)
560
561
562
563 def get_strain_map(sublattice, zone_axis_index, theoretical_value,
564                    sampling=1, units='pix', vmin=None, vmax=None,
565                    cmap='inferno',
566                    title='Strain Map', filename=None, **kwargs):
567
568     zone_vector_index_list = sublattice.
569     _get_zone_vector_index_list(
570         zone_vector_list=None)
571     zone_index, zone_vector = zone_vector_index_list[
572         zone_axis_index]
573     zone_data = sublattice.
574     get_monolayer_distance_list_from_zone_vector(
575         zone_vector)
576     x_position, y_position, xy_distance = zone_data
577
578     xy_distance = [i*sampling for i in xy_distance]
579     xy_distance = [(i-theoretical_value)/theoretical_value for
580                     i in xy_distance]
581     xy_distance = [i*100 for i in xy_distance]
582
583     strain_signal = sublattice.get_property_map(
584         x_position, y_position, xy_distance, upscale_map=1)
585     strain_signal.axes_manager[0].scale = sampling
586     strain_signal.axes_manager[1].scale = sampling
587     strain_signal.axes_manager[0].units = units
588     strain_signal.axes_manager[1].units = units
```

A.6. TEMUL Code

```
586     if vmax == 'max':
587         vmax = np.max(xy_distance)
588     if vmin == 'min':
589         vmin = np.min(xy_distance)
590
591     strain_signal.plot(vmin=vmin, vmax=vmax, cmap=cmap,
592                         colorbar=False, **kwargs)
593     plt.gca().axes.get_xaxis().set_visible(False)
594     plt.gca().axes.get_yaxis().set_visible(False)
595     plt.title("{}_{}".format(title, zone_axis_index))
596     cbar = ScalarMappable(cmap=cmap)
597     cbar.set_array(xy_distance)
598     cbar.set_clim(vmin, vmax)
599     plt.colorbar(cbar, fraction=0.046, pad=0.04, label="Strain (%)"
600                  above "{} {}".format(
601                      theoretical_value, units))
602     plt.tight_layout()
603
604     if filename is not None:
605         plt.savefig(fname="{}_{}_{}.png".format(filename, title,
606                                            zone_axis_index),
607                     transparent=True, frameon=False, bbox_inches='tight',
608                     pad_inches=None, dpi=300, labels=False)
609         strain_signal.save("{}_{}_{}.hspy".format(filename, title,
610                                              zone_axis_index))
611
612     return(strain_signal)
```

Code Segment A.4: Some of the developed polarisation functions of the TEMUL toolkit discussed in Chapter 6.