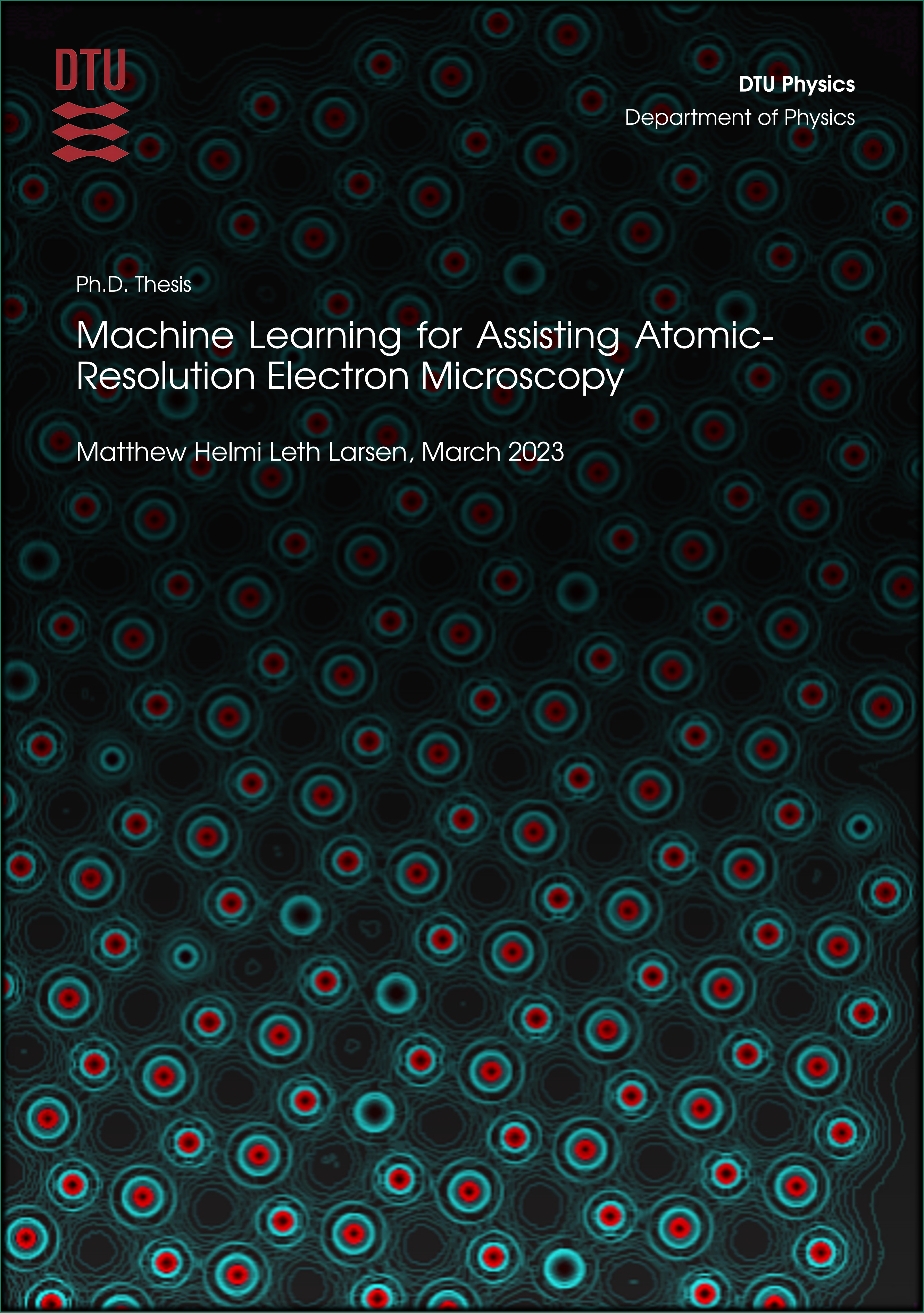


DTU Physics
Department of Physics

Ph.D. Thesis

Machine Learning for Assisting Atomic-Resolution Electron Microscopy

Matthew Helmi Leth Larsen, March 2023

DTU Physics
Department of Physics

Ph.D. Thesis

Machine Learning for Assisting Atomic-Resolution Electron Microscopy

Matthew Helmi Leth Larsen

Kongens Lyngby, 2023

Machine Learning for Assisting Atomic-Resolution Electron Microscopy

Ph.D. Thesis
Matthew Helmi Leth Larsen
March 1st, 2023

Cover: A graphically modified imaginary part of the exit wave function for a 2H phase monolayer MoS_2 atomic system with randomly placed Sulphur vacancy defects.

Technical University of Denmark
DTU Physics
Department of Physics

Fysikvej 311
2800 Kongens Lyngby, Denmark
Phone: +45 4525 3344
info@fysik.dtu.dk

www.fysik.dtu.dk

Preface

This thesis is submitted in candidacy for a Ph.D. degree in Physics from the Technical University of Denmark (DTU). The Ph.D project was carried out at the Center for Atomic-scale Materials Design (CAMD) at the DTU Department of Physics between December 2019 and March 2023. The project was supervised by Main Supervisor Professor Jakob Schiøtz and Co-Supervisor Professor Thomas Willum Hansen.

Matthew Helmi Leth Larsen

Kongens Lyngby, March 1st, 2023

Abstract

This thesis investigates how deep learning neural networks can solve two significant problems that currently hinder the field of atomic resolution transmission electron microscopy. The first problem is the large-scale analysis of data. Reliable information for materials design must be quantitative and statistically significant. While modern electron microscopes can produce large data sets of high resolution images, the data is typically analysed manually by an electron microscopist, which is a tedious and time-consuming task.

The second issue is the problem of electron beam influence on the material samples. The high-energy electrons in a microscope will inevitably influence the sample, both by directly damaging it and by inducing diffusion and chemical reactions.

To tackle these issues, the thesis presents a software pipeline that utilises simulated high-resolution transmission electron microscopy (HR-TEM) images to train robust neural networks. The software pipeline generates thousands of varied atomic systems of oxide supported metallic nanoparticles and 2-Dimensional monolayer nanoflakes. From these atomic systems, the user can generate thousands of HR-TEM images with assorted microscope conditions, by varying the contrast transfer function, presence of noise and the modulation transfer function, which enables image simulations for different electron detectors. The generated HR-TEM images are then paired with a ground truth label that defines the task the neural network should solve. The user can select from three labels: Mask labels, Disk labels, and Exitwave labels.

A study was made training neural networks with Exitwave labels paired with images of MoS_2 nanoflakes (2H phase) to perform exit wave reconstructions. The study presents that neural networks can perform exit wave reconstructions with smaller focal series and no information of the specific CTF parameters that are comparable to traditional algorithms. The exit waves reconstructed by the neural networks permit structural determination of 1Mo, 2S, and 1S atomic columns *via* Argand plots, but were limited in differentiating 1S atomic columns that exist in the upper or lower sulphur layer with respect to the optical axis. The study also shows that the neural networks perform best when trained on data sets of a single type of atomic system with minimal complexity.

Neural networks trained with Mask labels can perform binary segmentations of Au nanoparticles on CeO_2. The binary segmentation maps can be separated *via* a watershed algorithm, producing a multi-valued map that separates every instance of a nanoparticle in the image. Tools are implemented in a graphical user interface based software to track these instances across all frames. These tools facilitate large-scale data analysis by allowing for properties of each instance to be contained across frames and extracted by the user with the click of a button. This work shows that the segmentations can be leveraged to provide Fourier transforms of each isolated nanoparticle across all frames with a high enough resolution that crystal planes corresponding to different regions of a twinned nanoparticle can be easily distinguished. These tools allow for a statistically significant quantification of various dynamic properties.

The thesis presents a study that gauges the ability of the U-net and MSD-net neural network architectures included in the software pipeline to perform mask segmentations at low signal-to-noise ratios. The study identifies a lower limit for a reliable neural network segmentation of a CeO_2 supported Au nanoparticle at 200 $^{e^-}/Å^2$. The MSD-net presents an enhanced ability to differentiate between signal and noise and perform reasonable segmentations below the lower limit of the training data, which shows a strong generalisability. The study also highlights the importance of modelling the modulation transfer function to optimise the segmentations by the neural networks in all electron dose regimes. This provides intuition into the minimal electron beam conditions that still allow for reliable neural network data extraction.

Disk labels are shown to train neural networks to perform multi-class segmentations of individual atomic columns in monolayer MoS_2 nanoflakes (2H phase). Multi-class segmentation consists of both identifying the 1Mo, 2S, and 1S atomic columns and classifying them. The U-net, U-net++, and MSD-net neural network architectures in the software pipeline all display a powerful ability to perform the multi-class segmentation when provided a simulated focal series of at least 3 HR-TEM images over a large range of microscope conditions. This presents the ability for neural networks to interpret the intricate intensity variations in HR-TEM images and classify various atomic columns to identify local structure and defects in monolayer MoS_2.

This work shows that deep learning neural networks provide a powerful tool for analysing atomic-resolution image sequences captured by transmission electron microscopes. The thesis presents a software pipeline that allows for the generation of large and diverse data sets, the pairing of simulated images with ground truth labels, the training of multiple neural network architectures, and tools to apply the trained neural networks for various tasks. The work also highlights the limitations of neural networks and the importance of modelling noise parameters to optimise results. Overall, this work contributes to the field of electron microscopy and paves the way for future automated analysis of atomic-resolution transmission electron microscopy data.

Resumé

Denne afhandling undersøger, hvordan deep learning neurale netværk kan løse to væsentlige problemer, der i øjeblikket holder transmissions elektronmikroskopi med atomar opløsning tilbage. Det første problem er datamængden. Pålidelig information til materialedesign skal være kvantitativ og statistisk signifikant. Mens moderne elektronmikroskoper kan producere store datasæt af billeder i høj opløsning, analyseres dataene typisk manuelt af en elektronmikroskopist, hvilket er en besværlig og tidskrævende opgave.

Det andet problem er elektronstrålens påvirkning af materialeprøverne. Elektronerne i et elektronmikroskop har meget høj energi, og vil uundgåeligt påvirke prøven, både ved direkte at beskadige den og ved at inducere diffusion og kemiske reaktioner.

For at imødegå disse problemer præsenterer denne afhandling en metodlogi og softwarepakke, der anvender simulerede højopløsnings transmissions elektronmikroskopi (HR-TEM) billeder til at træne robuste neurale netværk. Softwarepakken genererer tusindvis af forskellige atomare systemer af oxid-understøttede metalliske nanopartikler og 2-Dimensionel monolags nanoflager. Ud fra disse atomsystemer kan brugeren generere tusindvis af HR-TEM-billeder med forskellige mikroskopforhold ved at variere kontrast-overførselsfunktionen, tilstedeværelsen af støj og modulations-overførselsfunktionen. Sidstnævnte giver mulighed for billedsimuleringer for forskellige elektrondetektorer. De genererede HR-TEM-billeder tilskrives derefter en ground-truth-label, der definerer den opgave, som det neurale netværk skal løse. Brugeren kan vælge mellem tre labels: Mask-labels, Disk-labels og Exitwave-labels.

Der blev gennemført en undersøgelse om træning af neurale netværk med Exitwave-labels parret med billeder af MoS_2 nanoflakes (2H-fase) for at udføre udgangsbølge-rekonstruktioner. Undersøgelsen viser, at neurale netværk kan udføre udgangsbølge-rekonstruktioner med mindre fokusserier og ingen information om de specifikke CTF-parametre, der kan konkurrere med traditionelle algoritmer. Udgangsbølgerne rekonstrueret af de neurale netværk tillader strukturel bestemmelse af 1Mo-, 2S- og 1S-atomsøjler via Argand-plot, men var ikke i stand til at differentiere enkelte svovlatommer, der optræder i det øverste eller nederste svovllag. Undersøgelsen viser også, at de neurale netværk fungerer bedst, når de trænes på datasæt af en enkelt type atomsystem med minimal kompleksitet.

Neurale netværk trænet med Mask-labels kan udføre binære segmenteringer af Au-nanopartikler på CeO_2. De binære segmenteringsbilleder kan adskilles *via* en watershed-algoritme, der producerer et billede med flere værdier, der adskiller hver nanopartikel i billedet. Værktøjer er implementeret i en grafisk bruger-interface baseret software til at spore nanopartiklerne på tværs af alle frames. Disse værktøjer muliggøre storskala

dataanalyse ved at tillade egenskaber for hver instans at blive sporet på tværs af billeder og udtrukket af brugeren med et klik på en knap. Det vises, at segmenteringerne kan udnyttes til at finde Fourier-transformationer af hver isoleret nanopartikel på tværs af alle billeder med en høj nok opløsning til, at krystalplaner, der svarer til forskellige regioner af en tvilling nanopartikel, let kan skelnes. Disse værktøjer giver mulighed for en statistisk signifikant kvantificering af forskellige dynamiske egenskaber.

Afhandlingen præsenterer en undersøgelse, der måler evnen af de to forskellige neurale netværksarkitekturer (U-net og MSD-net) inkluderet i softwarepakken til at udføre maskesegmenteringer ved lave signal-til-støj-forhold. Undersøgelsen identificerer en nedre grænse for pålidelig neural netværkssegmentering af en CeO_2-understøttet Au-nanopartikel ved 200 $^{e^-}/\text{Å}^2$. MSD-nettet præsenterer en forbedret evne til at skelne mellem signal og støj og til at udføre rimelige segmenteringer under den nedre grænse for træningsdataene, hvilket viser en stærk generaliserbarhed. Undersøgelsen fremhæver også vigtigheden af at modellere modulations-overførselsfunktionen for at optimere segmenteringerne af de neurale netværk i alle elektrondosisregimer. Dette giver indsigt i de minimale elektronstråleforhold, der stadig giver mulighed for pålidelig udtrækning af data med neurale netværk.

Disk-labels præsenteres som en måde at træne neurale netværk til at udføre multiklasse segmenteringer af individuelle atomare kolonner i monolag MoS_2 nanoflakes (2H fase). Multi-klasse segmentering består af både identifikation af 1Mo, 2S og 1S atomare kolonner samt deres klassificering. U-net, U-net++ og MSD-net neurale netværksarkitekturerne i softwarepakken viser alle en god evne til at udføre multi-klasse segmentering over et stort område af mikroskopforhold, når der leveres en simuleret fokalserie på mindst 3 HR-TEM billeder. Dette illusterer neurale netværks evne til at fortolke de indviklede intensitetsvariationer i HR-TEM-billeder og til at klassificere forskellige atomsøjler for at identificere lokal struktur og defekter, for eksempel i monolag MoS_2.

Dette afhandling viser, at deep learning neurale netværk er et kraftfuldt værktøj til at analysere billedsekvenser med atomar opløsning optaget af transmissions elektronmikroskoper. Afhandlingen præsenterer en softwarepakke, der giver mulighed for at generere store og forskellige datasæt, parring af simulerede billeder med ground-truth-labels, træning af flere neurale netværksarkitekturer og værktøjer til at anvende de trænede neurale netværk til forskellige opgaver. Afhandlingen fremhæver også begrænsningerne af neurale netværk og vigtigheden af at modellere støjparametre for at optimere resultaterne. Samlet set bidrager afhandlingen til elektronmikroskopiforskningen og baner vejen for fremtidig automatiseret analyse af data fra transmissions elektronmikroskopi med atomar opløsning.

Acknowledgements

The main goal of any Ph.D. project is to navigate the unknown, which results in numerous dead ends, failures, and redirections. Success is achieved by brushing your shoulder off and overcoming the obstacles along the way. Rarely is this achieved alone, and I have lots of gratitude to pass around.

Firstly, I would like to thank my supervisors Professor Jakob Schiøtz and Professor Thomas Willum Hansen for allowing me the opportunity to pursue this project. It has been an incredibly enriching learning experience. Thank you for helping me choose the path that this project has navigated.

Following my supervisors, I would like to thank my experimental collaborators at DTU Nanolab. Thank you William Bang Lomholdt, Dr. Pei Liu, Dr. Daniel Kelly, Professor Stig Helveg, and Professor Jakob B. Wagner, for your experimental data and insights, that have to a large extent shaped this project to what it is today. Theory without a real life application is hard to distinguish from philosophy.

I would also like to thank Ole Holm Nielsen, Lone Jensen Coe, and Bettina Norling Raae, for managing all of the computing, financial, and administrative resources that have made my work possible.

Next, I would like to thank our collaborators at Object Research Systems in Montréal, Canada. Thank you Jocya Hamelin for your cooperation and efforts in organising my visit. Thank you Nicolas Piché and Benjamin Provencher for materialising my vision for large-scale data analysis in high-resolution electron microscopy during my stay. Thank you Jeremy Simons for your extended hospitality during my stay and when inviting me for dinner with the team in Portland, Oregon.

It was important to me that this thesis is comprehensive and logically sound to all of my peers on both the theoretical and experimental sides. Thank you to Dr. Mark Kamper Svendsen and Stefan Kei Akizawa for reading my theory section on electron microscopy. Thank you Peder Lyngby, Hadeel Moustafa, and Abdulstar Kousa for reading my theory section on deep learning. Thank you Mads Søndergaard Larsen for reading my chapter on the implementation of my software pipeline. Thank you Dr. Thorbjørn Skovhus for reading my chapter on the summary of my results. Thank you Dr. Pei Liu for reading my chapter on the software implemented in Dragonfly from Object Research Systems. Thank you Dr. Fabian Bertoldo, Cuauhtemoc Nuñez Valencia and Mukesh Narendran for reading several chapters throughout this thesis. Thank you to my brother, Jeremy Johan Leth Larsen (Junior Copywriter), for editing my introduction, conclusion, and synopsis. Thank you to Michael Moth (CCO - Graphic Designer) for advice on graphical design and layout.

The COVID-19 lockdown was a unique challenge for society as a whole. Unfortunately, it occupied a majority of the timeline of this project. I would like to thank the group of Computational Atomic-scale Materials Design (CAMD) for their efforts in keeping together and maintaining the work culture. Thank you Dr. Morten Niklas Gjerding, Dr. Daniele Torelli, Dr. Anders Christian Riis-Jensen, Dr. Line Jelver, Dr. Simone Manti, Dr. Asbjørn Rasmussen, Dr. Estefanía Garijo del Río, Dr. Luca Vannucci, Dr. Alireza Taghizadeh, Dr. Thorbjørn Skovhus, Dr. Mark Kamper Svendsen, Dr. Fabian Bertoldo, Dr. Nikolaj Knøsgaard, Dr. Sajid Ali, Mads Kruse, Stefano Americo, Dr. Jiban Kangsabanik, Casper Larsen, Hadeel Moustafa, Joachim Sødequist, Dr. Sahar Pakdel, Dr. Tara Maria Boland, Dr. Haiyuan Wang, Martin Ovesen, Dr. Joel Davidsson, Dr. Mikael Kuisma, Dr. Fredrik Andreas Nilsson, Dr. Ask Hjorth Larsen, and Varun Rajeev Pavizhakumari. Thank you to my office mates Cuauhtemoc Nuñez Valencia, Dr. Julian Heske, Dr. Sami Juhani Kaappa, Dr. Urko Petralanda Holguin, and Dr. Peter Mahler Larsen, for the good mornings, good evenings, numerous conversations, YouTube videos, and TikTok and Instagram reels.

Last but not least, I would like to thank my family for their emotional support and patience with my absence. Thank you Mum, Dad, Jeremy Johan, Benjamin Norman, and Katja Naasunnguaq.

Dedicated to Mum and Dad.

List of Publications

[I] M. H. Leth Larsen, F. Dahl, L. P. Hansen, B. Barton, C. Kisielowski, S. Helveg, O. Winther, T. W. Hansen, J. Schiøtz,
"Reconstructing the exit wave of 2D materials in high-resolution transmission electron microscopy using machine learning,"

Ultramicroscopy Volume 243, January 2023, 113641.

Printed in copy from page 92.

[II] M. H. Leth Larsen, W. B. Lomholdt, C. N. Valencia, T. W. Hansen, J. Schiøtz,
"Quantifying Noise Limitations of Neural Network Segmentations in High-Resolution Transmission Electron Microscopy,"

submitted to Ultramicroscopy. arXiv:2302.12629.

Printed in copy from page 104.

[III] M. H. Leth Larsen, C. N. Valencia, J. Schiøtz,
"Identifying Atomic Positions in MoS2 with Neural Networks using Focal Series from High-Resolution Transmission Electron Microscopy,"

in preparation.

Printed in copy from page 118.

Contents

Introduction

Materials science is crucial for society as it plays a vital role in shaping our daily lives and drives technological advancements. A booming field within materials science is the development of catalysts and semiconductor technology. Catalysts are materials that increase the rate of chemical reactions, and are used in a wide range of applications, including fuel cells, refining and petrochemical processing, and environmental control technologies [1, 2]. Semiconductors are used in a variety of electronic devices, including computer chips, solar cells, and light-emitting diodes [3]. The development of new catalysts and semiconductors requires a deep understanding of the materials' properties and behaviour, as well as their structure and composition at the atomic level.

Scientists have an intimate relationship with their tools that allow them to see beyond the limitations of the human eye. Telescopes open doors to the distant universe beyond our planet and optical microscopes allow them to see microorganisms that are present all around us. Materials scientists use a range of tools and techniques, including electron microscopy and image processing, to study and understand the properties and behaviour of materials at the atomic scale. **Electron microscopy** is a technique that uses a beam of electrons to create an image of a sample. The electrons are accelerated to high energies with velocities in the relativistic regime. The wavelengths achieved allow scientists to observe very small structures, including those that are too small to be seen with an optical microscope. To be more specific, electron microscopes allow for up to 50 million times magnification, which vastly exceeds the optical microscope of about 1000 times magnification. The large magnification achieves **atomic-resolution** imaging, which is at the nanometer scale. There are several types of electron microscopes, including the transmission electron microscope (TEM) and the scanning electron microscope (SEM). In TEM, the electron beam is transmitted through the sample, in contrast to SEM, where the electron beam is focused and scanned over the surface of the sample. Microscopes today are designed to operate in either TEM or SEM modes, and the SEM mode of such a microscope is referred to as scanning transmission electron microscopy (STEM).

Machine learning, and in particular **deep learning** [4], have revolutionised fields such as image analysis [5, 6], speech recognition [7], self-driving cars [8], among many others. The field of bio-medicine was one of the first to adopt deep learning models, known as **neural networks**, for analysis on data from various scientific imaging methods [9, 10]. Given the success in bio-medical imaging, it is no surprise that materials scientists can equip similar methods for their image analysis. One specific area where deep learning is arguably necessary is **high-resolution transmission electron microscopy** (HR-TEM). HR-TEM is of central importance for modern materials design and development, as it is the main method for directly observing the atomic-scale structure of materials. HR-TEM offers atomic scale imaging with a time resolution that exceeds STEM, allowing for a more detailed analysis of atom dynamics in materials samples. The pixel intensities in a HR-TEM image make it difficult to apply traditional methods, such as thresholding methods, to separate and analyse regions of the image. Probing the positions of atoms and extracting information is exceedingly difficult, but deep learning based processing tools have proven themselves to be efficient in doing exactly that [11, 12], alongside gathering statistically significant data on the shapes, sizes, and orientation of nanoparticles [13], segmenting and classifying nanoparticles [14], retrieving phase information [15], and more [16].

Deep learning can help solve two specific and critical problems that hold back the field. The first is the problem of **large-scale data analysis**, and it is well established that deep learning is solidifying its presence in standard workflows for analysing data from HR-TEM [17, 18]. For information gathered to provide reliable value as input for materials design, it must be quantitative and statistically significant. This means that large amounts of high-resolution data must be acquired and analysed. While modern electron microscopes can produce rapid image sequences containing gigabytes or terabytes of data, the data is typically analysed manually by an electron microscopist looking at each image to determine what is being observed, for example by locating the atoms. Automated or computer-aided analysis techniques open a whole new way of using electron microscopy to gather statistically significant data sets, while limiting operator bias. Despite the efforts already made, the field is still young and there is ongoing research on the reliability of and optimal conditions for deep learning based methods for data extraction [19, 20].

The second issue is the problem of **beam damage and beam effects**. The high-energy electrons in a microscope will invariably influence the sample, both by directly damaging the sample, and by inducing diffusion and chemical reactions [21–27]. For example, it is slowly being realised that many of the dynamical effects observed in catalytic metallic nanoparticles may be induced by the beam [28]. To obtain a HR-TEM image with a barely acceptable signal-to-noise (S/N) ratio and atomic resolution, around 100 electrons pass by each atom in the sample; high quality images require at least ten times that electron dose. Operating the microscope in STEM mode further increases this dose by at least another order of magnitude.

This thesis investigates the extent to which deep learning can address the two highlighted problems. An analysis will be made of the limitations to deep learning based analysis of images and image sequences with a very low S/N ratio, and whether it can permit imaging under less damaging conditions. Secondly, an investigation into what deep learning can provide with regards to automated analysis tools enabling users to process large amounts of data. Deep learning models, like any machine learning model, are data driven and the accumulation of sufficient data sets by experimental means is a major limitation. This thesis focuses on the training of deep learning models with simulated data and the generalisability of these models to experimental data. Simulated HR-TEM images of various atomic systems will provide large and diverse data sets for deep learning without any material consumption or manual labour. This project primarily contributes to the United Nation's **Sustainability and Development Goal** (SDG) to ensure sustainable consumption and production patterns. This is a direct result of the fact that the investigations will provide knowledge to what extent TEM users can reduce material damage and consequently material waste, which makes for a more efficient resource-use in the materials characterisation and research industry. This ensures an efficient use of natural resources, environmentally sound management of waste, and reduce waste by prevention of material destruction.

The main result of this work is a software pipeline that can be easily equipped by users to generate vast data sets and train neural networks for various tasks. These tasks include the segmentation of metallic nanoparticles, the identification and classification by segmentation of atomic columns in 2-Dimensional (2D) materials with multiple chemical species, and exit wave reconstructions of 2D materials. Exit wave reconstructions retrieve information of the chemical composition and 3-Dimensional (3D) positions of atomic columns. From the retrieved exit wave, a scientist can manually identify and classify atomic columns. Exit wave reconstruction serves as another route with the same goal as the neural network segmentation of atomic columns, but leverages the neural network in a different way. The strengths and limitations of each task are highlighted and documented. As an addition to the software pipeline, tools have been implemented to apply the trained deep learning models for large-scale data analysis.

Thesis Outline

This work is highly inter-disciplinary, merging the fields of **computational atomic-scale simulations**, **HR-TEM**, and **deep learning**. The thesis will begin by introducing the relevant information regarding HR-TEM and deep learning that should be sufficient to grasp the investigations presented in the thesis. The outline is as follows:

- **Chapter 1**: This chapter presents the fundamental concepts behind the theoretical modelling of HR-TEM. The electron beam is treated as plane waves incident on the material sample, which traverse through and interact with the material potential. The electron wave function present on the other side of the material is altered by the imperfections present in the microscope before the final image is formed. An understanding of the steps leading up to the image formation, will be crucial for the reader to understand the simulations and treatment of HR-TEM images in the software pipeline.
- **Chapter 2**: This work leverages the power of deep learning to map complex features and patterns in the HR-TEM images to outputs that are applicable for information extraction and more interpretable by a human operator. This chapter describes the inner workings behind **neural networks**, one of the main deep learning based models. Deep learning models adapt weights that are applied to the input images to transform the images to the desired output. This chapter is crucial to understanding how the neural networks are trained on the simulated data and prepared for analysis tools in the software pipeline.
- **Chapter 3**: The main result of this work is a software pipeline that allows the user to prepare deep learning neural networks for various tasks applicable to

images of various materials. This chapter describes the workflow and implementation of the software pipeline, which consists of data generation, neural network training, and applied analysis tools for the user. This chapter is important for the reader to understand the decisions made behind the diversity of the simulated data sets, neural network constructions and training procedures, and how to equip the neural networks for data extraction.

- **Chapter 4**: This chapter summarises the main results achieved with the software pipeline. These results have been documented into three separate publications, each tackling the applications and limitations of exit wave reconstructions, nanoparticle segmentations, and atomic column identification and classification. This chapter is a highlight of the main results from each publication, where **Chapter 8** contains full reprints of each publication.
- **Chapter 5**: Chen *et al.* [29] argue that atomic vibrations are heterogenous and largely induced by the electron beam. The inclusion of these vibrations were crucial to a proper analysis of the exit wave function retrieved from HRTEM images *via* traditional algorithms. This chapter provides a novel method to introducing the heterogeneity of the atomic vibrations into simulated images, which are inherently static snapshots of an atomic configuration. This method serves as an extension to the software pipeline.
- **Chapter 6**: An external collaboration was made with the group behind the image analysis software, *Dragonfly*. Together with them tools were implemented in their software so that the neural networks trained in the software pipeline can be deployed for large-scale analysis with a graphical user interface, rendering the tools more accessible to all users. This chapter presents the tools as an extension to the software pipeline and presents real life examples of the possibilities that come with the tools.
- **Chapter 7**: This chapter finalises the thesis with a summary of the above chapters along with final remarks and suggestions on how to proceed with the completed investigations and findings.
- **Chapter 8**: This chapter compiles all the publications related with this thesis. **Chapters 1-3** provide the background knowledge to understand all details behind every publications. See **Chapter 4** for a summary of the main results.

CHAPTER 1

Theory: Modelling The Electron Microscope

A theoretical model of the electron microscope operating in TEM mode is presented in Figure 1.1. From top to bottom *i.e.* along the optical axis, electrons traverse as plane waves until they meet the material sample, otherwise referred to as the specimen[1] [30].

The electrons interact with the electrostatic potential of this specimen and the transmitted wave is known as the exit wave function or simply the "exit wave". This wave contains all the information of the material constituents. The exit wave is, however, subject to alterations by the lenses and other components of the microscope, which

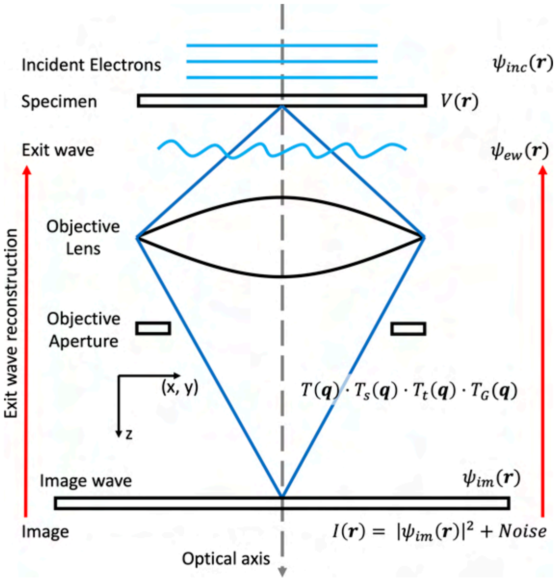


Figure 1.1: A theoretical model of the electron microscope operating in TEM mode. This illustration depicts the steps between the incident plane wave electron beam to the final image obtained, shaped by the specimen potential and elements of the microscope. Inspired by Ref. [30].

---

[1] This is different to operating in STEM mode where the electrons are focused into a spot on the specimen.

project the exit wave to a final image. This image is then further disrupted by noise due to the detector.

One of the existing data extraction methods is known as an "exit wave reconstruction", which consists of transforming the final image back into the transmitted exit wave and interpreting this complex wave function to probe the 3-Dimensional positions of the atoms in the sample [29, 31].

To provide insight into the simulations of HR-TEM images, this section will present the theory behind modelling each of the steps in Figure 1.1, from top to bottom, starting with the interactions between the specimen and the incident plane wave electrons.

1.1 The Multislice Method

"The principal method of image simulation has come to be known as simply *the multislice method*" as said by Earl J. Kirkland [32]. This method handles the dynamical scattering processes occurring in material samples as thin as 10-50 Å. A full derivation of the multislice algorithm is presented in Ref. [33]. This derivation begins with the time independent Schrödinger equation,

$$\left[-\frac{\hbar^2}{2m}\nabla^2 - eV(\mathbf{r},z)\right]\Psi(\mathbf{r},z) = E\Psi(\mathbf{r},z), \tag{1.1}$$

where $\Psi(\mathbf{r},z)$ is the full electron wave function, $\mathbf{r}$ is defined in the (x,y)-plane perpendicular to the z-axis (optical axis), $m = \gamma m_0$ is the relativistic electron mass, and E is the kinetic energy of the electron[2]. Electrons are accelerated in the microscope to energies between 60-1000 keV, which are far greater than the potential of the material sample; Any energy gain or loss of the electrons due to the specimen potential, $eV(\mathbf{r},z)$, will be far lesser than their kinetic energy. This motivates the high energy or fast electron approximation, which involves treating the rapidly varying plane wave along the z-axis separately from the slowly varying part, $\psi(\mathbf{r},z)$,

$$\Psi(\mathbf{r},z) = \psi(\mathbf{r},z)\cdot e^{2\pi iz/\lambda}. \tag{1.2}$$

Solving for the first and second derivatives cancels out the rapidly varying part, described by the exponential term, and the assumption that

$$\left|\frac{\partial^2\psi}{\partial z^2}\right| << \left|\frac{1}{\lambda}\frac{\partial\psi}{\partial z}\right| \tag{1.3}$$

due to ψ being the slowly varying part in z and λ being very small, simplifies the expression to first order derivatives in z. This is known as the high-energy approximation, which is sometimes interpreted as ignoring back-scattered electrons [34]. This is an appropriate assumption for the high energy electrons present in HR-TEM and

---

[2] Strictly speaking, the Dirac equation would be the appropriate description of relativistic electrons, however is significantly more difficult to work with. The Schrödinger equation with a corrected relativistic wavelength and mass is therefore preferred and has proved to be accurate enough for most relevant energies in HR-TEM [30].

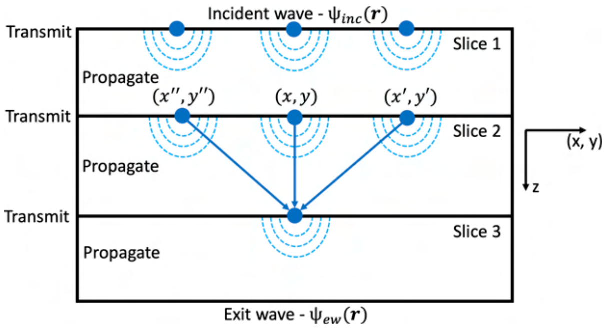


Figure 1.2: The multislice algorithm depicted as a Fresnel diffraction following Huygens' principle. Every point in slice n acts as a point source and the total interactions of each point source constructs the next slice. This is defined as a convolution of propagator function, $p(\mathbf{r}, \Delta z)$. Inspired by Ref. [33].

simplifies the Schrödinger equation to

$$\frac{\partial \psi(\mathbf{r}, z)}{\partial z} = \left[\frac{i\lambda}{4\pi}\nabla^2_{\mathbf{r}} + i\sigma_e V(\mathbf{r}, z)\right]\psi(\mathbf{r}, z) \quad ; \quad \sigma_e = \frac{2\pi m e \lambda}{h^2}, \tag{1.4}$$

where σ_e is the interaction parameter, which scales the interactive term of the electron and specimen potential depending on the relativistic electron mass and wavelength. The interaction parameter is almost constant for energies above 300 keV, and increases rapidly for energies below this [35].

Solving Eq. (1.4) can be approached in multiple ways, but regardless will involve integrating over small slices $(z + \Delta z)$ of the specimen potential. This results in

$$\psi(\mathbf{r}, z + \Delta z) = e^{\frac{i\lambda\Delta z}{4\pi}\nabla^2_{\mathbf{r}}} \cdot t(\mathbf{r}, z) \cdot \psi(\mathbf{r}, z) + \mathcal{O}(\Delta z^2) \tag{1.5}$$

where $t(\mathbf{r}, z)$ is the transmission function of each slice defined as

$$t(\mathbf{r}, z) = e^{i\sigma_e \int_z^{z+\Delta z} V(\mathbf{r}, z')dz'}. \tag{1.6}$$

The transmission function treats each slice of a thick specimen as a weak phase object; An object that contributes only with a phase change to the incident wave, which is an approximation made for thin specimens [35, 36]. Eq. (1.5) can be interpreted as a convolution,

$$\psi(\mathbf{r}, z + \Delta z) = \left(p(\mathbf{r}, \Delta z) * [t(\mathbf{r}, z) \cdot \psi(\mathbf{r}, z)]\right)_{\mathbf{r}} + \mathcal{O}(\Delta z^2), \tag{1.7}$$

where $(*)_{\mathbf{r}}$ denotes a real space convolution in the (x,y)-plane. This is the governing equation of the multislice method that propagates the wave function from one slice of the specimen potential to the next *via* a convolution of $p(\mathbf{r}, \Delta z)$ known as the propagator function. The incident wave, $\psi_{inc}(\mathbf{r})$, is propagated through slices of the specimen potential labelled by n as

$$\psi_{n+1}(\mathbf{r}) = t_n(\mathbf{r}) \cdot \left(p_n(\mathbf{r}, \Delta z_n) * \psi_n(\mathbf{r})\right)_{\mathbf{r}} + \mathcal{O}(\Delta z^2), \tag{1.8}$$

which ultimately produces the exit wave, $\psi_{ew}(\mathbf{r})$. This procedure is depicted in Figure 1.2 where every slice contributes a phase shift to the latest wave function defined by the transmission function, and every point in a slice n contributes to a point in the slice $n+1$, due to the convolution. A physical perspective of this can be described by Huygen's principle that each point in a wavefront acts as a spherical point source, and the next wavefront is the interaction of all the spherical wave fronts from each point in the previous wavefront [33].

1.2 The Imperfections of the Microscope

In practice the microscope is not perfect. The imperfections include lens aberrations that cause the wavefronts to deviate from that of a perfect point source and partial coherence. The final image wave is described as

$$\psi_{im}(\mathbf{r}) = (\psi_{ew} * T)_{\mathbf{r}} \tag{1.9}$$

where T($\mathbf{r}$) is known as the contrast transfer function (CTF). Performing a Fourier transform allows for the direct multiplication

$$\psi_{im}(\mathbf{q}) = \psi_{ew}(\mathbf{q}) \cdot T(\mathbf{q}), \tag{1.10}$$

which presents the CTF as a spatial frequency filter applied to the exit wave. The CTF describes the microscope aperture and coherent lens aberrations as a phase error, $\chi(\mathbf{q})$. The formal definition is

$$T(\mathbf{q}) = A(\mathbf{q}) \cdot e^{-i\chi(\mathbf{q})} \tag{1.11}$$

where

$$A(\mathbf{q}) = \begin{cases} 1, & \text{for } |\mathbf{q}| \leq q_A \\ 0, & \text{for } |\mathbf{q}| > q_A \end{cases} \tag{1.12}$$

is the aperture which cuts off beams scattered above a certain angle, defining a spatial frequency limit, q_A (see Figure 1.3). The phase error is commonly described as a series expansion [30, 37, 38], which for a well-aligned[3] microscope can be approximated as

$$\chi(\mathbf{q}) \approx \frac{1}{2}\pi C_s \lambda^3 \mathbf{q}^4 + \pi \Delta f \lambda \mathbf{q}^2. \tag{1.13}$$

This definition states that the phase error is controlled by the spherical aberration, C$_s$, the defocus, Δf, and the wavelength, λ. In Krivanek notation [39] the spherical aberration and defocus are known as $C_s \equiv C_{3,0}$ and $\Delta f \equiv -C_{1,0}$. In certain cases higher order aberrations can be important. For details on the full expansion and higher order coefficients see Refs. [30, 40, 41].

---

[3] This refers to the adjustments of the illumination, various lenses, aperture, *etc.* of the microscope.

One can speculate that the ideal CTF is unity across all spatial frequencies, such that it does not alter the original exit wave, however this is not the case. If one allows for the assumption that the material sample acts purely as a phase object such that

$$\psi_{ew}(\mathbf{r}) = \psi_{inc}(\mathbf{r}) \cdot e^{i\phi(V(\mathbf{r}))} \tag{1.14}$$

with a phase ϕ dependent on V($\mathbf{r}$), the local potential of the sample. Without the effects of the CTF means that the obtained image is

$$|\psi_{im}(\mathbf{r})|^2 = |\psi_{ew}(\mathbf{r})|^2 = 1 \tag{1.15}$$

since $\psi_{inc}(\mathbf{r})$ is a plane wave with magnitude 1. Under the weak phase approximation [36], where $\phi << 1$ the resulting image with a CTF is

$$|\psi_{im}(\mathbf{r})|^2 \approx 1 + 2(\phi(V) * Im(T))_{\mathbf{r}}. \tag{1.16}$$

The role of the CTF is that it introduces contrast into the image, where the image intensities are related to the phase.

In practice one can attempt to tune the defocus to obtain an optimal CTF and one way to do so is to image at Scherzer defocus [36, 42]. The Scherzer defocus attempts to tune the defocus such that it counter acts the $\mathrm{C}_s\mathbf{q}^4$ term of $\chi(\mathbf{q})$ and obtains as wide of a flat band region as possible. The (extended) Scherzer defocus is defined as

$$\Delta f_{Scherzer} = -1.2(\lambda C_s)^{1/2}. \tag{1.17}$$

In Figure 1.3 the “zero point crossing” (or “point resolution”) is presented, which is the first intersect of the imaginary part of the CTF with the zero axis. This is maximised by the Scherzer defocus to obtain the wide flat band region [43].

Beyond tuning the CTF a combination of high spatial and temporal coherence is essential. Spatial coherence refers to the degree of correlation between the phase of points along a wavefront of the incident electron beam. A highly spatially coherent beam is essential for HR-TEM because it allows the electrons to interfere constructively at the sample, creating high-resolution images. A low spatial coherence beam would produce low contrast and blurry images. The distribution of angles in the beam direction across the wavefront is quantified by a parameter β and the distribution of directions $\mathbf{q}_\beta$ is modelled into $T(\mathbf{q})$ under the thin sample or quasi-coherent approximation [36, 38]. This follows Eq. (1.10) as

$$\psi_{im}(\mathbf{q}) = \psi_{ew}(\mathbf{q}) \cdot \int T(\mathbf{q} + \mathbf{q}_\beta) \cdot p(\mathbf{q}_\beta)\, d\mathbf{q}_\beta \tag{1.18}$$

where

$$p(\mathbf{q}_\beta) = \frac{1}{\pi q_s^2} e^{-q_\beta^2/q_s^2} \tag{1.19}$$

is a Gaussian distribution describing the spread in the beam directions with a $1/\mathrm{e}$ width of $q_s = \beta/\lambda$. The result of evaluating Eq. (1.18) is an envelope function defined as

$$T_s(\mathbf{q}) = e^{-\frac{\beta^2}{4\lambda^2}\left|\frac{\partial}{\partial \mathbf{q}}\chi(\mathbf{q})\right|^2}. \tag{1.20}$$

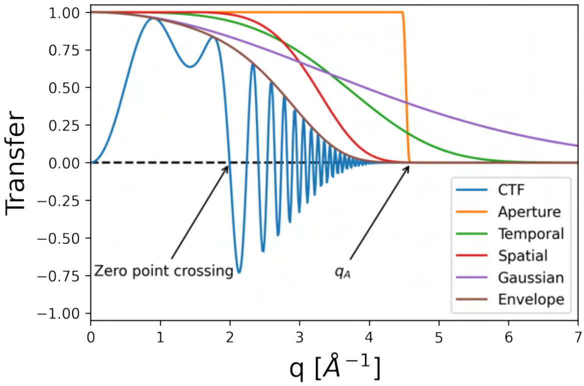


Figure 1.3: Total transfer function due to microscope imperfections with an acceleration voltage of 300 keV and a $C_s = 5\ \mu$m, which gives a Scherzer defocus of 38.43 Å.

Temporal coherence refers to the degree to which the phase of points on separate wavefronts of the electron beam correlate in time (in the direction of propagation). Any temporal incoherence can be due to minor spreads of energy which produce a more significant spread in focal length of the objective lens. Similar to the integral over the distribution directions $\mathbf{q}_\beta$ in Eq. (1.18), here an integral over a spread of defocus values Δf is performed, which leads to an additional envelope function,

$$T_t(\mathbf{q}) = e^{-\frac{\Delta^2}{4}\left[\frac{\partial}{\partial \Delta f}\chi(\mathbf{q})\right]^2}. \tag{1.21}$$

where Δ is the defocus (focal) spread. The aperture defined in Eq. (1.12) is typically referred to as the semi-angle cut-off, while the spatial coherence, T_s, and temporal coherence, T_t, are typically referred to as the angular spread and focal spread, respectively. Additional blurring in the images can derive from vibrations, drifts in the stage, magnetic lens noises *etc.*, which are modelled to follow a Gaussian distribution that provides the final envelope function,

$$T_G(\mathbf{q}) = e^{-\frac{\sigma^2 q^2}{2}}, \tag{1.22}$$

where σ is the $1/e$ amplitude of image deflections due to the mentioned noise sources. The total image wave is finally defined as

$$\psi_{im}(\mathbf{q}) = \psi_{ew}(\mathbf{q}) \cdot T(\mathbf{q}) \cdot T_s(\mathbf{q}) \cdot T_t(\mathbf{q}) \cdot T_G(\mathbf{q}) \tag{1.23}$$

under the quasi-coherent approximation, which allows only for small amounts of incoherence, and will therefore breakdown in cases with large incoherence. This models the complete transformation from the exit wave propagating through the microscope to the image plane.

1.3 Modelling Detector Noise

Once the propagating electrons have interacted with the material sample and been focused through the microscope they are imaged on a detector. There are several types of detectors that are commonly used in HR-TEM. In the modern day these primarily include charge-coupled device (CCD) and direct detectors. The former utilises a scintillator screen, based on phosphor/fibre-optics, which converts the electrons into light that is then captured by a camera. The latter directly counts electrons by a microchannel plate detector such as the monolithic active plate detector [44]. Each detector has its own advantages and disadvantages, and the choice of detector will depend on the specific requirements of the experiment. But regardless of the choice the presence of noise is inevitable.

Noise in direct detectors can be approximated by shot noise [45–49], which describes the electrons incident on the detector by a Poisson distribution, such that that the image intensities for every real space pixel $(x, y) \in \mathbf{r}$ is

$$|\psi_{im}(\mathbf{r})|^2 = I^{shot}_{x,y} = N_D \pm \sqrt{N_D}, \tag{1.24}$$

where N_D is the number of electrons detected per pixel.

For scintillating material based detectors, such as the Gatan OneView [50], the approximation of pure Poissonian noise breaks down, since the spectral profile of the noise is altered by the modulation transfer function (MTF) [51, 52]. This is an intrinsic property of the scintillating material describing how the energy supplied by the electron is spread over a given area of the scintillating material, effectively spreading the electron's signal across multiple pixels [44, 53]. This results in a loss of spatial resolution. A popular model of the MTF is the parametric form from Lee *et al.* [45],

$$MTF(\tilde{q}) = (1 - C) \cdot \frac{1}{1 + (\frac{\tilde{q}}{c_0})^{c_3}} + C, \tag{1.25}$$

where the spatial frequencies are normalised by the Nyquist frequency, which is related to the sampling, s of the detector by $\tilde{q} = q/q_N = 2 \cdot q \cdot s$. As this is a property of the detector, which discretises the image wave, it will always be Nyquist frequency limited. The limits of the function are 1 for $\tilde{q} \to 0$ due to a normalisation and C for $q \to \infty$. The value of C represents a noise floor, that will scale depending on the dominating noise source [46]. This function is commonly extracted from the power spectral density of an image of the detector (no material sample) *i.e.* $|FT[I(\mathbf{r})]|^2$ [53]. The MTF acts as a spatial frequency filter and is applied to the image in an identical way to the CTF described in the previous section, namely as a product in Fourier space,

$$I^{shot}_f(\mathbf{q}) = I^{shot}(\mathbf{q}) \cdot MTF(\mathbf{q}), \tag{1.26}$$

where $I(\mathbf{q}) = FT[I_{x,y}]$, with FT being the Fourier transform. This is depicted in 1-Dimension in Figure 1.4.

There is a fraction of the noise that is not affected by the MTF and appears after the scintillating material. This fraction of noise is likely from multiple sources, but

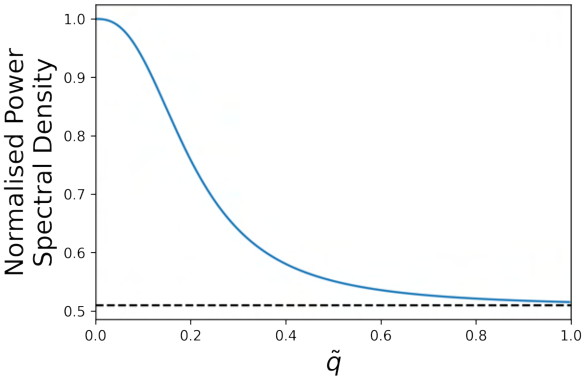


Figure 1.4: A modulation transfer function from Eq. (1.25) with $c_0 = 0.2$, $c_3 = 2.57$, and $C = 0.51$. The x-axis are the spatial frequencies normalised by the Nyquist frequency. The dashed black line depicts the noise floor defined by C.

can be approximated as the readout noise [46]. The contributions of the readout noise and shot noise can be modelled separately. The $q \to \infty$ limit, or the C value, of the MTF (Eq. (1.25)) will be controlled by the fractional contribution of the readout noise to the total noise. It acts as a noise floor that can be dominated by either source at different electron dose regimes. It is expected that at higher frame dose the readout noise is washed out by shot noise, but at lower dose the noise source is dominated by the readout noise [46]. The readout noise is also modelled as a Poisson distribution, $P(\lambda = N_0)$, where N_0 is a constant noise floor. The total noise is a sum of the two Poisson distributions

$$P(\lambda = N_D) + P(\lambda = N_0) = P(\lambda = (N_D + N_0)) \tag{1.27}$$

meaning that each pixel intensity, $I_{x,y}$ in the final image is within the distribution

$$I_{x,y}^{shot+readout} = N_D + N_0 \pm \sqrt{N_D + N_0}, \tag{1.28}$$

where N_0 is a constant that is intrinsic to the detector conditions, and N_D is tuneable by the user, such that it will dominate the other term when $N_D >> N_0$. A high electron dose, however, also means that the MTF will have a stronger influence on the spectral profile of the noise.

1.4 Exit Wave Reconstruction and Structural Determination

The aim of an exit wave reconstruction is to reverse the effects of the microscope and map back to the exit wave from the detected image. This involves deconvoluting the total transfer function due to microscope imperfections, *i.e.*

$$\psi_{ew}^{recon}(\mathbf{q}) = \psi_{im}(\mathbf{q}) \cdot [T(\mathbf{q}) \cdot T_s(\mathbf{q}) \cdot T_t(\mathbf{q}) \cdot T_G(\mathbf{q})]^{-1}\,. \tag{1.29}$$

Several methods exist to perform this deconvolution, such as the algorithm by Allen et al. [38] and the Gerchberg-Saxton algorithm [54]. These algorithms require obtaining a “focal series” - several images of the same sample with a varying defocus [29, 55]. The idea is to use the defocus as a varying parameter to extract phase and amplitude information given the parameters describing the CTF, such as the C_s value [42]. Doing so allows for an approximate reconstruction of the exit wave from the focal series images.

The exit wave contains maximal information about the atomic structure of the material sample. In ideal cases the optical axis is aligned along an orientation where the material sample exhibits columns of atoms aligned along the optical axis *i.e.* a “zone axis”. Each atomic column can contain varying chemical species and/or total atomic mass. A way to interpret the exit wave of each atomic column is *via* channelling theory [56]. Channelling theory, like the multislice method from Section 1.1, is another way to described the propagation of electrons through the material. The difference is that this theory focuses on the propagation of a single electron and describes this electron being bound by the potential of the atomic column, allowing it to “channel” along an atomic column. An electron channelling through an atomic column along the optical axis is depicted in Figure 1.5(a). In this picture the material sample can be thought of as an assembly of potential wells, one for each atomic column, that are preferably well separated in the (x,y)-plane. The electron wave function from Eq. (1.4) can be written as an expansion over the Eigen functions for each atomic column, and for thin samples the primary bound state is the 1S state of the atomic column [57]. Each atomic column in the exit wave (with a subtraction of the entry wave) can be described as

$$\psi_{ew}^{column}(\mathbf{r}, z) - 1 = 2ic_{1S}sin\left(-\pi\frac{E_{1S}}{E_0}\frac{k}{2}z\right) \cdot \psi_{1S}(\mathbf{r})e^{-i\pi\frac{E_{1S}}{E_0}\frac{k}{2}z}, \tag{1.30}$$

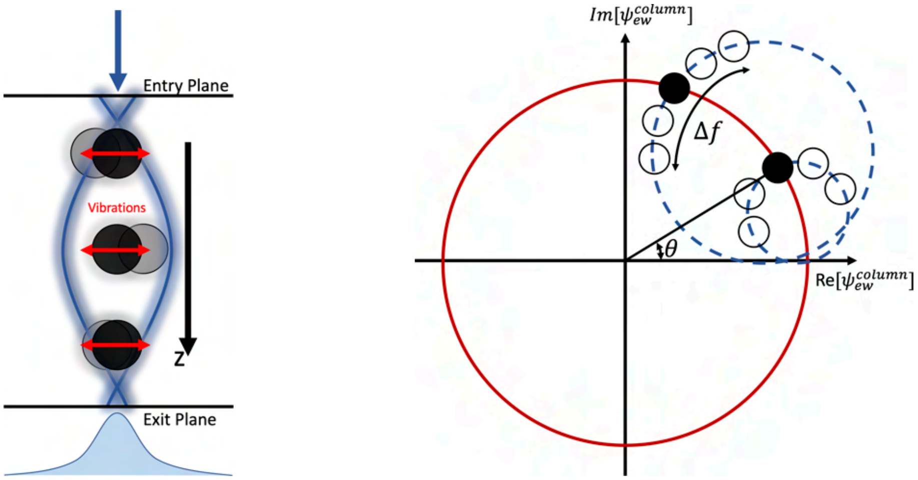
where c_{1S}, $\psi_{1S}(\mathbf{r})$ and E_{1S} are the excitation coeffecient, Eigen state and Eigen energy of the projected column, respectively, z is the depth along the optical axis, and $\mathrm{E}_0 = {}^{h^2k^2}\!/_{2m}$ is the incident electron energy [42, 56–59]. The amplitude and phase of Eq. (1.30) is

$$abs\left(\psi_{ew}^{column}(\mathbf{r}, z) - 1\right) = \left|2c_{1S}sin(-\pi\frac{E_{1S}}{E_0}\frac{k}{2}z) \cdot \psi_{1S}(\mathbf{r})\right| \quad \text{and} \tag{1.31}$$

$$arg\left(\psi_{ew}^{column}(\mathbf{r}, z) - 1\right) = -\pi\left(\frac{E_{1S}}{E_0}\frac{k}{2}z - \frac{1}{2}\right), \tag{1.32}$$

respectively. The highlight of this is that the amplitude peaks at the atomic column position in $\mathbf{r}$ and varies periodically with the depth, z, which means the amplitude can be used to determine the positions of the atomic columns. The phase on the other hand is constant in the (x,y)-plane over the atomic column. The phase is also linearly dependent with depth and E_{1S}, which is proportional with the mass density of the atomic column. The phase is, as a result, proportional to the atomic number of the atoms and can be used to determine the composition of the atomic columns.

The highlighted features regarding the amplitude and phase of Eq. (1.30) is commonly utilised by plotting the (x,y)-pixels of the exit wave covering the atomic column on



(a) (b)

Figure 1.5: (a) An illustration of how an electron “channels” through a column of atoms. The electrons can be thought of as “trapped” by the potential of the atomic column. These ideal columns are sensitive to atom vibrations and tilt. The electron enters at the entry plane of the atomic column and a conceptual exit wave is shown at the exit plane of the atomic column. Inspired by Ref. [29]. (b) An illustration of the mass circle and defocus circle present on an Argand plot of the atomic column exit wave, which is used to determine the structure (constituents and positions) of each atomic column. Inspired by Ref. [31].

an Argand plot [29, 31, 42, 57–59]. See Figure 1.5(b), which presents the “mass circle” with a radius defined by the amplitude of the atomic column and a phase, θ, which is, as mentioned, dependent on the atomic number of the atomic column. This provides a means to identify the atomic composition of each atomic column. The atomic constituents and the number of atoms, *i.e* the heights, of the various atomic columns will not necessarily be flat. The defocus value with respect to the exit plane of each column will vary. The interpretation of the defocus variation across atomic columns is treated in Refs. [29, 57]. The result is that the defocus variation can be interpreted by the “defocus circle” shown as a dashed circle in Figure 1.5(b), which will present the relative heights of the atomic columns’ exit plane. The Argand plot analysis can be used to identify both the varying atomic numbers of each atomic column and the relative position of the exit plane to determine placement of the atomic column along the optical axis. Channelling theory is sensitive to atom vibrations and tilting of the material sample that both deviate the situation from well-aligned atomic columns [29]. The issue of atomic vibrations will be further explored in Chapter 5.

In summary, this chapter described the modelling of the electron microscope in TEM mode. This includes the transformation of the incident plane wave electrons to the final image, which involves the inclusion of the imperfections in coherence and aberrations from the microscope, and noise due to the detector. An emphasis was placed on how the final image can be processed to reconstruct the exit wave by deconvolution of the CTF. This exit wave is useful for structural determination of material samples. The upcoming chapter will describe how deep learning can be equipped to process and extract information from the final image.

CHAPTER 2

Theory: Deep Learning

The electron microscope has opened a window into the atomic world, allowing users to see atoms and their dynamic behaviour under various environmental perturbations. The images obtained can be thought of as an array of data containing information that has to be extracted by the user. This data extraction can be done in many ways and this thesis is a study on what deep learning can provide to this objective.

This chapter will provide an understanding of the fundamental concepts of deep learning models known as neural networks and what allows a neural network to detect features in the input space and map it to an output.

2.1 Feed Forward Neural Networks

The feed forward neural network is one of the simplest, yet powerful, forms of artificial neural networks [60]. The "neural" in neural network stems from the idea of modelling networks to mimic the brain, which is operated by the activation of billions of interconnected neurons. Each neuron that is activated, activates a number of other neurons and this chain effect defines the decision-making of any brain. A neural network's equivalent of the neuron can be exemplified by the perceptron [61]. Consider a neuron that is activated by the following rule:

$$y(x) = \begin{cases} 1, & \text{for } x \geq 0 \\ 0, & \text{for } x < 0 \end{cases}. \tag{2.1}$$

One could apply this activation to make a binary decision based on a simple condition of the input variable x.

The fundamental idea of a neural network is to approximate a function f that maps an input x to an output y, where the mapping will typically exceed the complexity of a step function shown in Eq. (2.1). In the case where the true mapping $y = f(x)$ is not known, the neural networks approximate mapping can be defined as $\tilde{y} = \tilde{f}(x)$.

An example of a feed forward neural network is shown in Figure 2.1 for an arbitrary input vector with 3 elements that should be mapped to an arbitrary output of 2 elements. It is shown that the mapping is achieved by a given set of weights applied at each neuron in a so-called "hidden" layer, which consist of multiple neurons. The hidden layer transforms the input data based on the weights at every neuron to map it to the desired output. A feed forward neural network is constructed in layers, in this case: An input layer, a hidden layer, and an output layer.

Figure 2.1: An illustration of a single hidden layer feed forward neural network. At each neuron in the hidden and output layer, a weight is applied to transform the output. These weights map a given input to a desired output.

In a fully-connected network every point in the input layer is connected to every neuron in a hidden layer, every neuron in that hidden layer is connected to every neuron in the next hidden layer, and finally every neuron in the last hidden layer is connected to every point in the output layer. The output of each neuron affects many others and the connections are only in the forward direction (input to output), hence the given name - “feed forward”.

The output of a neuron in the hidden layer, z_j, for an input vector, $\mathbf{x}$, and weights vector, $\mathbf{w}$, is given by

$$z_j^{linear}(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{N} x_i w_i + b_0, \tag{2.2}$$

where N denotes the number of points in the input layer, w is a tuneable weight for each connection from a point in the input layer to a neuron in the hidden layer, and b_0 is a tuneable bias associated with the layer. The issue at this stage is that the mapping in Eq. (2.2) is only linear, which will limit the applications to only linear solutions. To introduce non-linearity this output is passed through an activation function, f_A as

$$z_j^{non-linear}(\mathbf{x}, \mathbf{w}) = f_A\left(\sum_{i=1}^{N} x_i w_i + b_0\right). \tag{2.3}$$

The first example of an activation function was the perceptron activation in Eq. (2.1), and more commonly applied activation functions will be presented in Section 2.3. Similar to the principle of the brain, it is the activations of the neurons that will effectively map the input to the output, and the activation will be controlled by the weights, biases, and the activation function.

The reason these models are referred to as networks is because there are typically multiple hidden layers where the resulting mapping function is represented by a composition of the functions of each layer [60]. Neglecting the weights and number of neurons, a simple representation of this composite function is

$$\tilde{f}(\mathbf{x}) = f^{(H)}(\ ...\ f^{(3)}(f^{(2)}(f^{(1)}(\mathbf{x})))), \tag{2.4}$$

where the number of hidden layers, H, defines the "depth" of the network. For many tasks a large depth has proven to be crucial, which is why this field is referred to as "Deep" learning [60].

2.2 Loss Functions and Back Propagation

In supervised learning (as opposed to unsupervised learning, which is beyond the scope of this thesis) [4], every input x has a paired label y *i.e.* the data set, $\mathcal{D}$, is defined as

$$\mathcal{D} = \{(x_i, y_i) | i = 1, ..., N\} \tag{2.5}$$

for N data examples. After the forward passing of an input x_i through the neural network, an approximate $\tilde{y}_i$ is inferred by the network and this can be compared to the label or ground truth y_i by means of an error function or loss function, $\mathcal{L}(\mathbf{y}, \tilde{\mathbf{y}})$.

A common loss function for regression tasks is the mean squared error averaged over the data set defined as

$$\mathcal{L}_{MSE}(\mathbf{y}, \tilde{\mathbf{y}}) = \frac{1}{N} \sum_{i}^{N} ||\mathbf{y}_i - \tilde{\mathbf{y}}_i||^2, \tag{2.6}$$

which is most suitable when the neural network should take an input and output a value that is not categorised into discrete classes [62]. On the contrary, in classification problems $\mathbf{y}$ and $\tilde{\mathbf{y}}$ represent probabilities for each data point belonging to a given category or class, hence the cross entropy is introduced as a measure of the difference between the true and predicted distribution. The cross-entropy loss, also known as log loss, is used as a measure of how well the neural networks predictions match the true values. It measures the dissimilarity between the predicted probability distribution, which is obtained from the output of the neural network, and the true distribution. The categorical cross entropy loss is defined as

$$\mathcal{L}_{CCE}(\mathbf{y}, \tilde{\mathbf{y}}) = -\frac{1}{N} \sum_{i}^{N} \sum_{j}^{C} y_{i,j} \cdot log(\tilde{y}_{i,j}), \tag{2.7}$$

which presents a sum over categories or classes, C [63]. It can be seen that for a given data example and class, the loss function is defined by $-log(\tilde{y})$ which is 0 if the predicted probability is correct *i.e.* $\tilde{y} = 1$, and increases with decreasing $\tilde{y}$. This function is calculated for each example in the data set and then averaged over the entire data set. In the case where C=2, this simplifies to the binary cross entropy loss,

$$\mathcal{L}_{BCE}(\mathbf{y}, \tilde{\mathbf{y}}) = \frac{1}{N}\sum_{i}^{N} -(y_i \cdot log(\tilde{y}_i) + (1 - y_i) \cdot log(1 - \tilde{y}_i)), \tag{2.8}$$

where y_i, can be interpreted as the probability that the output belongs to class 1, rendering $1 - y_i$ the probability that the output belongs to the class 0 [64].

As mentioned, the neural network is attempting to learn the mapping between $\mathbf{x}$ and $\mathbf{y}$, defined as $f(\mathbf{x}, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ contains all the tuneable parameters of the model *i.e.* $\boldsymbol{\theta} = (\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, ...)$, where $\mathbf{W}$ is a matrix for each hidden layer containing the vector weights for each neuron. To do so the neural network must minimise the chosen loss function and identify the optimal parameters associated with this minimum. Since the loss function is typically not linear or parabolic, evaluating the minimum of the derivative is not trivial [65]. To solve this the gradient descent method is utilised, which iteratively updates the parameters towards the minimum of the loss function [66]. This involves computing the partial derivatives of the loss function with respect to all the parameters and updating them by

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \eta\frac{\partial\mathcal{L}}{\partial\boldsymbol{\theta}}, \tag{2.9}$$

where η is known as the “learning rate”, which controls the magnitude of the change in response to the gradient, and i denotes the update iteration [67].

Many gradient descent algorithms exist such as the Root Mean Square Propagation (RMSprop) and Adaptive Moment Estimation (ADAM) algorithm that differ in their approach to applying the gradient to update each step [68]. These algorithms are referred to as “optimisers” and selecting an optimiser along with a learning rate is a part of identifying the best possible training procedure for the neural network *i.e.* one that minimises the loss function and converges within a sensible number of iterations.

The entire process of computing the gradients and updating the weights is referred to as the “back propagation”, which will always succeed a feed forward step. Training neural networks, like other machine learning algorithms, is a cyclic process of repeated feed forward and back propagation steps, until the loss function converges, or a user defined maximum number of cycles is reached. These cycles are referred to as “training epochs”.

2.3 Activation Functions

The role of the activation function is to add non-linearity to the activations of the neurons. There are many activation functions to choose from and the choice is often an experimental science and a large part of the optimisation process. In fact, neural networks will typically utilise different activation functions at different layers. The reasons to choose one over the other is problem specific and at this point in time many have become standard for certain types of problem. This section will present the relevant activation functions for the work and provide some of the general advantages and disadvantages [69]. These activation functions are presented in Figure 2.2.

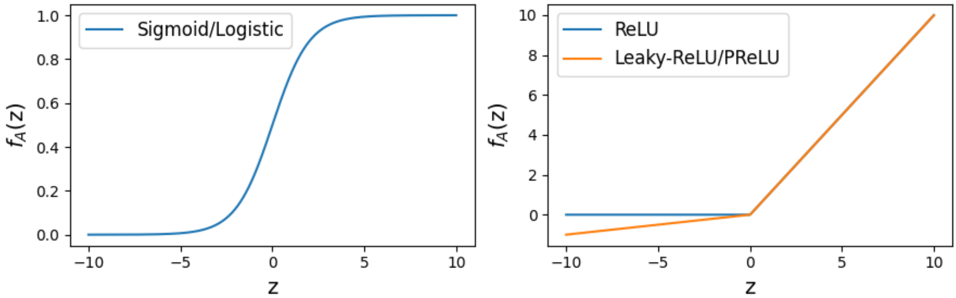


Figure 2.2: Common activation functions that add non-linearity to the output of each neuron in a neural network.

Sigmoid/Logistic

The sigmoid activation function is defined as

$$f_A(z) = \frac{1}{1+e^{-z}}. \tag{2.10}$$

The advantages of using this activation function include:

- The output of the sigmoid function is always between 0 and 1, which makes it a good choice for binary classification problems, since the output can be interpreted as a probability.
- The sigmoid function is fully differentiable, which allows the use of a gradient descent optimisation algorithm in the back propagation described in the previous section.

The disadvantages of using this activation function include:

- The sigmoid function can cause the “vanishing gradients problem” [70] when the input values are very large or small. This is due to the values saturating to 0 or 1 in the lower and upper limits, causing the gradient to go to 0 very quickly. This can make it difficult for the neural network to learn.

Softmax

Consider the sigmoid function in a multi-class classification problem. It will produce a value between 0-1 for each class, however nothing ensures that the sum of the output values over all classes gives 1. The softmax activation function is defined to provide a probability of each class that sums to 1. The softmax activation function is defined as

$$f_A(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}, \tag{2.11}$$

which is in fact a combination of multiple sigmoids.

The advantages of using this activation function include:

- The softmax function normalises the output of the network to a probability distribution, which makes it a good choice for multi-class classification problems.
- The softmax function is fully differentiable, which allows the use of a gradient descent optimisation algorithm in the back propagation described in the previous section.

The disadvantages of using this activation function include:

- Similar to the sigmoid, the softmax function can cause the vanishing gradients problem.
- The softmax function can be computationally expensive when the number of classes is very large.

Rectified Linear Unit

The Rectified Linear Unit (ReLU) activation function is defined as

$$f_A(z) = max(0, z). \tag{2.12}$$

The function maps any input value x to either z if z is positive, or 0 if z is negative.

The advantages of using this activation function include:

- The ReLU function is computationally efficient because it only requires a simple comparison and a subsequent multiplication by a scalar value.
- The ReLU function is non-saturating, which means that the gradient does not saturate as the input grows large.

The disadvantages of using this activation function include:

- The ReLU function is not defined for negative values, which means that the gradient is 0 for negative input values. This can cause the so-called “dead neurons” problem [71, 72], where certain neurons in the network never activate and never update their weights during training.
- The ReLU function is not differentiable at $z = 0$, which can cause some issues with gradient-based optimisation algorithms [72].

Due to the computational efficiency of the ReLU, it is very desirable to utilise it in deep neural networks, with many neurons. As an improvement to the original ReLU, the leaky ReLU was introduced to reduce the dead neurons problem by introducing a slope in the negative region of the function. Succeeding this modification, the slope was made tuneable in the parametric ReLU (PRelu), which is defined as

$$f_A(z) = max(az, z), \tag{2.13}$$

where a is now a tuneable parameter, allowing an optimal slope or optimal gradient to be trained.

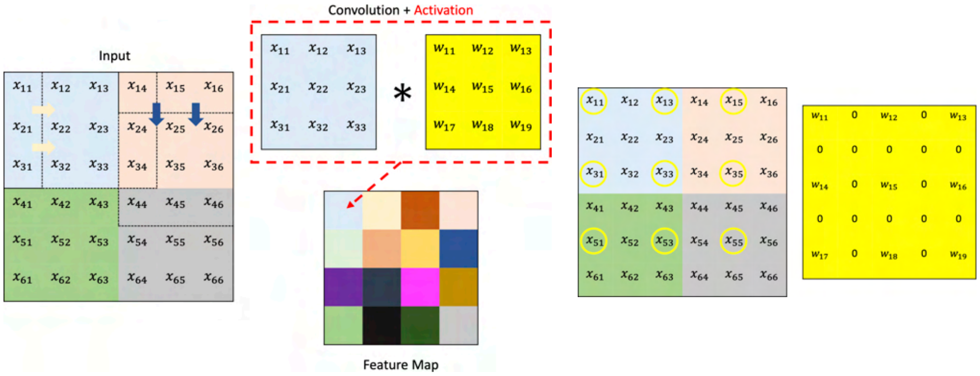
2.4 Convolutional Neural Networks

Consider the feed forward neural network presented in Figure 2.1 where the input data is an image, flattened so that it is a 1-Dimensional vector rather than a 2-Dimensional matrix. For an image of 512 x 512 pixels, this results in a vector of 262,144 input points. Now define a hidden layer with 1000 neurons and the number of weights becomes 262,144,000, which is a tremendous amount. Image dimensions can easily be larger and so can the number of neurons, making fully-connected feed forward neural networks very impractical for images or other arrays of large sizes.

Convolutional neural networks approach this problem by combining the activation of a set of data points into a single value in the succeeding layer. As hinted at by the name, the method of combining the data points or (in the context of images) pixels is by a convolution. Defining a kernel, K, that has k_x x k_y dimensions, a convolution between the kernel and the image is performed. For an image, I, with pixels, (i,j), this is defined as

$$S_{i,j} = (I * K)_{i,j} = \sum_{m} \sum_{n} I_{i+m,j+n} K_{m,n}, \tag{2.14}$$

where the output, S, is commonly referred to as a feature map. An illustration of performing a convolution with a kernel on an input image is presented in Figure 2.3(a), where it is shown that a 3 x 3 region in the input image is reduce to a single value that is passed through an activation function, forming a single pixel in a feature map. The feature map has a reduced dimension compared to the input image, so the input image is typically padded (values are added around the border to increase its dimensions) by various methods [73].



(a)

(b)

Figure 2.3: (a) Illustration of the convolution between a kernel and a local region of the image. The kernel scans across the image in strides of 1 and produces a feature map. (b) Illustration of dilating the kernel. Using the same input image as in (a), the weights of the kernel are spread across a large spatial region, with a dilation value of 1.

The kernel is a primary component in defining the performance of convolution neural networks. The kernel has three key features: sparse interactions, parameter sharing, and translational equivariance [74].

The kernel dimensions are typically defined to be significantly smaller than the input image dimension. The convolution is therefore performed between the kernel and a local region of the image. The kernel is translated along the image space in strides, where a convolution is performed after every stride. The connectivity between a pixel in the input space and a pixel in the feature map is said to be sparse, since the pixel in the input space does not contribute to every pixel in the output, but only to a local region of pixels in the output space.

The kernel contains the trainable weights of the model and these weights are reused at multiple, possibly every, point in the input space. This idea is referred to as parameter sharing; Each weight is tuned by multiple points in the input space. This proves to be very useful for images that typically contain features, which exist within smaller local regions of the image and can exist at various spatial regions in the image.

The output is also equivariant under translations, since the translation of a feature in the input space will produce the same translation in the output space. Consider for example a kernel that detects an edge (a boundary between high and low contrast in an image). This kernel would generate a feature map that highlights all the edges present in the image. Translating the image pixels, would translate the feature map pixels corresponding to the edges. This is an efficient use of the trainable weights.

The kernels are tuned by defining the dimensions, the stride length, and a dilation parameter. Dilation involves enlarging the dimensions of the kernel by adding zeros between the trainable weights and as a result spreading the weights across a larger spatial region. This concept is illustrated in Figure 2.3(b), and presents a way for the kernel to detect features in the input space that exist across larger areas without increasing the number of trainable weights in the kernel.

2.5 A Practical Note on Deep Learning

In practice, constructing and training neural networks is very much an experimental science with many technical details one will experience along the way. What activation function and what kernel size and dilation, for example, are a few of many parameters that will have to be optimised for each problem by what is referred to as "hyperparameter tuning".

A single convolutional layer in a neural network consists of the convolution and an activation for each kernel. Multiple kernels can be applied within a convolutional layer, each producing different feature maps, which is typically referred to as the output channels or "filters" of a convolutional layer. Each feature map extracts a certain feature from the input image, which is passed on to succeeding layers to build upon those features. Most convolutional neural networks consist of many successive convolutional layers, each generating many features maps, which are combined in various ways to transfer information in the feed forward direction of the neural network. Other layer types exist that perform pooling operations to down-sample the feature maps or interpolation operations to up-sample feature maps, as well as combine feature maps by element-wise addition or concatenations, referred to as “skip-connections”. The way these many layers are constructed and connected is referred to as the “architecture” of the neural network, and at this point of time there are many existing architectures. Some architectures will be introduced later, in Section 3.3.

Once a neural network is constructed it must be trained on a data set. The goal of many machine learning methods is to train a model that provides reliable enough inferences to predict information outside of the given data set. To gauge the ability of the model, or neural network, to generalise, the data set is commonly split into a “training”, “validation”, and “test” data set. For an illustration of data splitting see Figure 2.4. Typically most of the data is contained in the training data set, which is the portion of the data that contributes to the feed forward and back propagation steps, or training epochs, described in Section 2.2. At the end of every training epoch, the loss function is computed for the training data set, as well as the validation data set, to ensure that the updated weights of the neural network do not only map the training data set space but are applicable outside. Cases where the neural network converges the loss function for the training data set but not the validation data set is referred to as “overfitting”, which results in an unreliable neural network that only maps the training data space. When a neural network has, after several training epochs, converged the loss function for both the training and validation data set, this neural network can be applied to the test data set as a final sanity check to ensure generalisability (More on this in Chapter 3).

Neural networks are computationally heavy, and in many cases it is not possible to load the entire data set, along with the neural network on the computer memory. The data set, $\mathcal{D}$, is typically divided into batches of size B such that

$$\mathcal{D}_B = \{(x_i, y_i) | i = 1, ..., B\}. \tag{2.15}$$

In practice the loss function and gradients of a given batch is computed, instead of the entire data set, and is used as an approximate gradient to the entire space. This is referred to as mini-batching and has proven to much more computationally efficient and a good approximate to minimising the entire data set space [67].

After being exposed to a training data set for a sufficient number of training epochs, the neural network’s many neuron activations can map an input image to some other output image, or classify pixels belonging to a given class, which is known as segmentation. This highlights the motivation to apply convolutional neural networks as a tool to processing HR-TEM images.

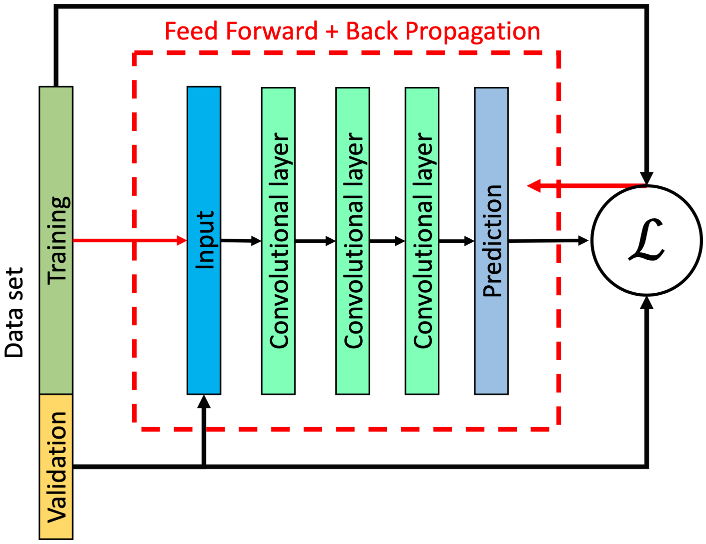


Figure 2.4: An illustration of the training procedures for a simple convolutional neural network architecture. Each convolutional layer can apply any number of convolutional kernels to generate multiple feature maps. This figure depicts how the data set is split into a training portion and validation portion. The training portion contributes to the feed forward and back propagation stages of neural network training, and after each training epoch the loss function is computed for both the training and validation data set, where the training loss is used for back propagation *i.e.* updating the neural network weights. The loss function should converge to a minimum for both portions of the data set to ensure that the neural network can generalise to data outside the training data set. The final test data set is neglected here.

CHAPTER 3

Software: Neural Network Assisted HR-TEM Pipeline

A software pipeline[1] has been implemented to facilitate data generation, neural network training, and analysis tools that utilise trained neural networks for various tasks in HR-TEM. The pipeline is summarised in Figure 3.1, where (from top to bottom) the first step is data generation.

Data generation involves the preparation of hundreds to thousands of diverse atomic systems. These atomic systems are constructed in the *Atomic Simulation Environment* (ASE) [75]. A HR-TEM image of each system is simulated with a varying degree of the microscope imperfections and noise described in Chapter 1. The image simulations are executed using *abTEM* (transmission electron microscopy from first-principles) [76] - an image simulation software based on the multislice method presented in Section 1.1. The software pipeline provides an abundance of simulated HR-TEM images for training neural networks. Each HR-TEM image has a paired output referred to as the "ground truth" or "label". Atomic positions from the simulated atomic systems are utilised to generate accurate labels, beyond the accuracy of manually segmented experimental data [77]. The neural network will map the HR-TEM image (input) to the label (output) and so labels are generated for various purposes, involving both segmentation (pixel-wise classification) and regression tasks.

In the next step, neural networks are constructed using *TensorFlow+Keras* [78], which provides an application programming interface (API) to set up the input, output and convolutional layers, as well as set up activation functions for each layer, define the loss function, and more, all introduced in Chapter 2. Training scripts are developed for numerous neural network architectures and are automated to flag a loss function and final output layer associated with the labels provided (see Section 2.2 and Section 2.3).

Last but not least, tools have been implemented to apply the neural networks to both simulated and experimental data in order to extract information from the HR-TEM images. These tools include gauging the performance of the trained neural networks by various metrics and handling of experimental HR-TEM images with the associated metadata. The tools further include methods to apply neural networks to both simulate and experimental HR-TEM images and to utilise neural network segmentations to extract information such as the area, position, orientation and the chemical composition

---

[1] For sake of clarity, "in software engineering, a pipeline consists of a chain of processing elements (processes, threads, coroutines, functions, etc.), arranged so that the output of each element is the input of the next" - From Wikipedia, the free encyclopedia.

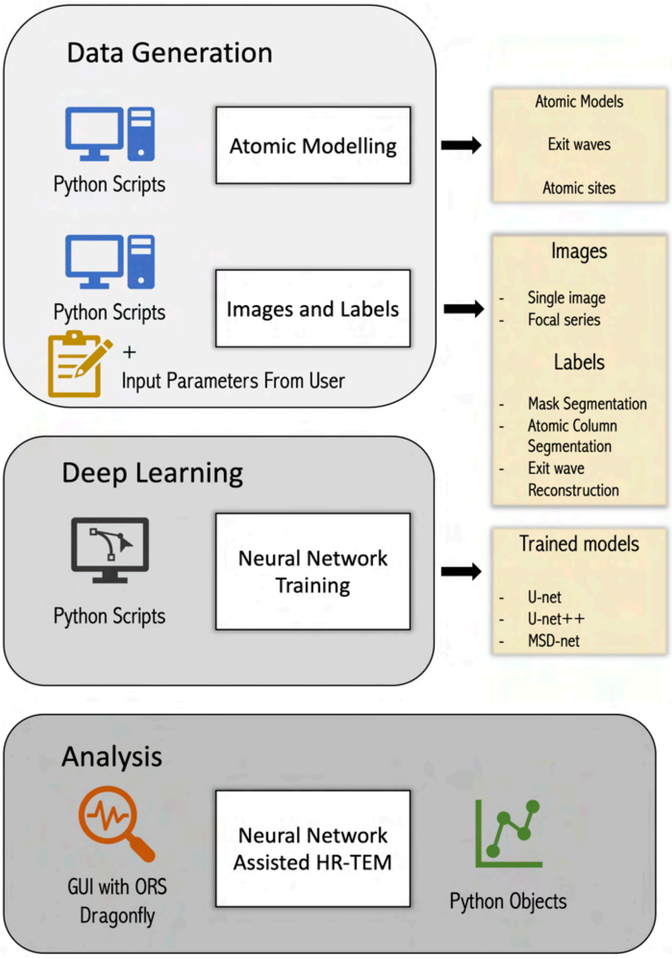


Figure 3.1: A diagram of the software pipeline implemented as a central part of this thesis. The pipeline allows for numerous diverse atomic systems to be constructed and imaged with varying microscope conditions. The atomic systems allow for accurate labels (ground truths) to be computed and paired with each HR-TEM image associated to a given atomic structure. The paired images and labels allow for supervised training of multiple neural network architectures for a number of given tasks, including both segmentation and regression tasks. The final segment of the pipeline involves tools implemented to utilise the trained neural networks for data extraction.

in a consistent and large-scale manner to achieve statistically significant information.

The following sections will delve into the specific technical aspects of each part of the pipeline. Section 3.1 will provide a detailed description of how atomic models are constructed in order to expose the neural networks to large variations of atomic systems, whilst still being realistic in order to match experimental data. Section 3.2 will leverage the concepts and theory from Chapter 1 to simulate HR-TEM images. The application of the CTF and MTF described in Sections 1.2 and 1.3 will be controlled *via* an input file, which will be a central part of the entire pipeline. In Section 3.5 and Section 3.4 a presentation will be made of several *Python* objects implemented to handle HR-TEM data files and to apply the neural networks. The focus is to open the door to large-scale analysis with neural networks. An emphasis will be placed on how the *Python* objects improve accessibility to handling HR-TEM images and to utilising the neural networks to a user with limited programming experience. The pipeline in Figure 3.1 presents a graphical user interface (GUI) part under the analysis segment, which is handled later in Chapter 6.

The entire software pipeline is available as a GitLab repository for open access use and development under the MIT license[2]. This software pipeline is prepared for high performance computing (HPC) clusters and is therefore designed to be accessible through a *Unix* terminal, whilst the analysis tools are designed for use in *Jupyter notebooks*.

3.1 Atomic Modelling

For the majority of this work two classes of atomic systems were considered: face centred cubic (FCC) crystals, and monolayer 2D materials. More specifically, Au (FCC) nanoparticles and MoS_2 monolayer nanoflakes have been extensively studied, due to the experimental data sets provided for this thesis. In this section the considerations behind the construction of these atomic systems are described. To train robust neural networks that are applicable with minimal bias it is important to expose the neural network to as many atomic systems as possible, which is feasible through simulations.

The software pipeline supplies *Python* scripts that are executable with command-line arguments. The purpose of these scripts is to generate a data set of diverse atomic systems and exit waves through *ASE* and *abTEM*. The atomic systems and exit waves generated are stored in a folder as they will be the input to the next component of the pipeline - the generation of images and labels (Section 3.2). The output of these scripts are summarised in Table 3.1.

For every atomic system, information of the atomic columns are stored, such as its position in space and categorical class. Atoms that are displaced along the optical (z-)axis but fall within a certain small distance in the (x,y)-plane (*e.g.* a radius of 0.3 Å - this value should be well below the lattice constant of the atomic system, but large enough to allow for minor perturbations in the atomic positions), are clustered into an atomic column positioned at the centre-of-mass of all the atoms belonging to that column. This is referred to as a “site”, seen in Table 3.1. An example of such a

---

[2] Code availability: `https://gitlab.com/matthewhelmi/NeuralNetwork_HRTEM`.

Table 3.1: Output from atomic modelling *Python* scripts.

Models	*format*: *.cfg* or *.traj* Stores the atomic system, with information of the atomic elements.
Waves	*format*: *.h5* (HDF5) Stores the exit wavefunction as an *abTEM* Wave object.
Points	*format*: *.npz* (*Python* dictionary) Stores the (x,y) coordinates of atomic columns under "sites" and their categorised class under "classes".

column is shown in Figure 1.5(a). This could be a column of Au atoms along the [110] zone axis of a Au nanoparticle, or a 2S column in monolayer MoS_2 (2H phase). Each atomic site can be classified for either binary or multi-class segmentation (*e.g.* 1Mo, 2S, 1S sites), which is stored as "classes" in Table 3.1.

For each atomic structure an exit wave is computed *via* the following code snippet, which is a direct application of the multislice algorithm described in Section 1.1 with parameterised potentials from Kirkland [79]:

```python
from abtem.potentials import Potential
from abtem.waves import PlaneWave

# A potential is defined for a given atomic system, using a
# parametrisation such as the Kirkland parametrised potentials
potential = Potential(atoms, ...)
# Set up a plane wave with a given energy (acceleration voltage)
wave = PlaneWave(energy)
# Apply the multislice algorithm to generate the exit wave
exitwave = wave.multislice(potential)
```

This concludes the general concepts behind all of the atomic modelling in the software pipeline and the following will provide more specific details behind the two aforementioned classes of materials, starting with Au (FCC) nanoparticles.

Oxide Supported FCC Nanoparticles

Oxide supported FCC metallic nanoparticles are studied at large for catalytic purposes [80–82]. Often there is a need to gather statistics on morphological properties such as the orientation and size of the nanoparticles, or to identify the mobility of atoms/atomic columns under different conditions to gain a better understanding of the catalytic properties [14, 83–85].

FCC nanoparticles, for a given chemical species, can be generated by specifying the (h,k,l) miller indices of the desired layers and the number of planes for each layer, which will construct a pristine FCC crystal. Here is an example of constructing a Au nanoparticle with 7, 8 and 9, (100), (110) and (111) planes, respectively:

```
from ase.cluster.cubic import FaceCenteredCubic

surfaces = [(1, 0, 0), (1, 1, 0), (1, 1, 1)] # Miller indices
layers = [7, 8, 9] # Number of layers
nanoparticle = FaceCenteredCubic('Au', surfaces, layers)
```

The software pipeline generates FCC nanoparticles where the size of a nanoparticle is controlled by varying the number of (100), (110), and (111) planes that make up the nanoparticle. Between 7 and 15 number of planes for each layer constructs realistic sizes of nanoparticles [86]. Each nanoparticle is rotated so that the [110] crystal zone axis is aligned with the optical (z-)axis (see Figure 1.1). A slight random tilt off zone axis between 0 and a maximum value provided by the user (see Listing 3.3) is applied since it is not certain that the nanoparticle's zone axis will always be perfectly aligned with the optical axis. At a random layer from the centre of the nanoparticle a slice is made perpendicular to the [111] direction, effectively exposing a cross section of a (111) facet, which is the expected interface to a supporting oxide material [86].

The oxide support (in this case CeO_2) is generated by first constructing the unit cell. This unit cell is then used to build a slab for a given facet, in this case the (111) surface using the lattice parameters from the Materials Project [87, 88]. Here is an example:

```
from ase.build import bulk, surface

ceria = bulk('CeO2', crystalstructure='fluorite',
             a=5.47, b=5.47, c=5.47, alpha=90) # unit cell
slab = surface(ceria, indices=(1,1,1), layers = 20, vacuum = 0)
rep = 20
slab *= (rep,rep,1) # (x,y) repetitions
```

The repetitions are set to fill a large area. The exposed (111) facet of the FCC nanoparticle is attached to the (111) plane of a CeO_2 slab, whose [110] zone axis is also aligned with the optical (z-)axis.

The CeO_2 supported Au systems are randomly rotated about the optical (z-)axis, so that the support and interface will not always be located at the same region of the HR-TEM image. An example is shown in Figure 3.2 along with the degrees of freedom at which the structures are randomised. The random degrees of freedom includes:

- The size of the nanoparticle.
- The Z thickness of the cell, effectively altering the thickness of the CeO_2 support.
- The zone axis tilt, tilting the [110] zone axis slightly off the optical (z-)axis.
- The nanoparticle (NP) placement in the y-direction along the surface of the CeO_2 support.
- A rotation between $0°$ and $360°$ about the optical (z-)axis.
- The placement of the atomic system along the x-axis.

All atoms that may appear outside the simulation cell due to the numerous translations, rotations, and sizing, are removed with at least 2 Å vacuum to every edge to avoid

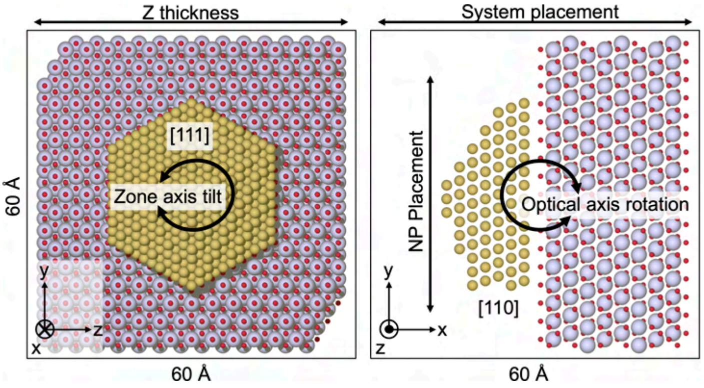


Figure 3.2: Visualising the degrees of freedom for generating numerous CeO_2 supported FCC nanoparticles. This is a CeO_2 supported Au nanoparticle in a simulation box of 60 x 60 x 60 Å. The nanoparticle can be rotated about the x-axis to shift the [110] zone axis slightly off axis and the entire system is rotated about the optical (z-)axis. The nanoparticle is also translated along the y-axis to situate it differently along the CeO_2 support, and the CeO_2 support thickness is varied by varying the cell thickness in the z-direction.

periodic boundary effects in *abTEM*. The interface spacing between the nanoparticle and surface is set to 1 Å. Any interactions in the potential at or near the interface is not treated by *abTEM*. The CeO_2 supported Au systems are simulated with a constant beam energy of 300 keV.

The pipeline provides a *Python* script that takes in minimal user input and executes the construction steps described above to generate realistic CeO_2 supported nanoparticles, shown in Listing 3.3. The mandatory arguments include: the name of the folder to store the items listed in Table 3.1, the chemical species of the nanoparticle, the maximum off zone axis tilt of the nanoparticle, and the number of desired systems.

```
$ python make_fcc_cluster_supported.py [Folder name] [Chemical Species]
  [Maximum zone axis tilt] [Number of systems]
  Optional: --sampling [Sampling in Å/pixel of the exit wave]
  Optional: --resolution [Number of x, y pixels of the exit wave]
  Optional: -s/--start [Starting number]
  Optional: -n/--numproc [Number of cores for parallelisation]
  Optional: --seed [Seed random number generator reproducibility]
  Optional: --train [Store as training data set (Default)]
  Optional: --test [Store as validation/test data set]
```

Listing 3.3: An instruction on the *Unix* command-line usage of the *Python* script for generating numerous CeO_2 supported metallic nanoparticles with varying sizes.

2D Monolayer Nanoflakes

2D materials such as MoS_2 are studied and applied in industry for sulphur extraction from crude oil and gas sensing amongst other applications [1, 89]. Effects on the material sample induced by the electron beam are increasingly important in 2D materials due to the reduced dimensionality, which also supplies an increased density of catalyt-

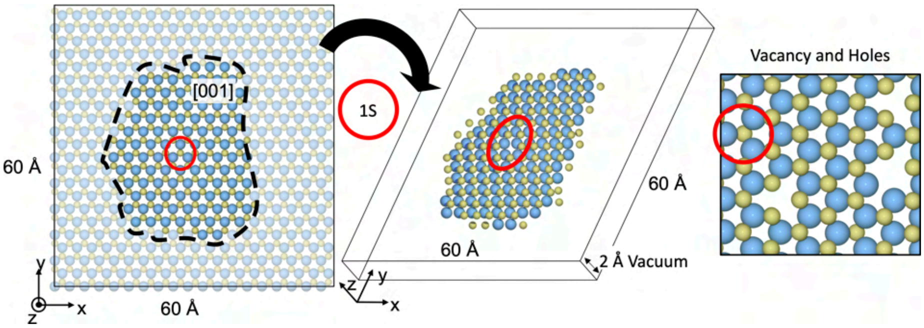


Figure 3.3: MoS_2 nanoflakes are generated by selecting random atoms from a larger sheet and computing the convex hull of these atoms to define a boundary between which atoms to keep and which to discard. S vacancies, or 1S atomic columns, are generated by removing random S atoms, and holes can be generated by removing entire 2S or 1Mo atomic columns at random.

tically active sites [90]. The pipeline supplies *Python* scripts to simulate 2D material structures with defects. More specifically this work tackles monolayer MoS_2 in the 2H phase in order to study the abilities of neural networks to differentiate between 1Mo, 2S, and 1S atomic columns - otherwise known as vacancy defects, which are common defects occurring due to the electron beam [90–92].

Here is an example of defining a MoS_2 unit cell:

```
from ase.build import mx2

MoS2 = mx2(formula='MoS2')
```

The unit cell is then repeated to fill a given cell size forming a large sheet of MoS_2. Atoms are selected at random within the sheet and the convex hull of those atoms is used to cut out a randomly shaped nanoflake, which is visualised in Figure 3.3. The randomly shaped cut-outs will avoid the neural network from biasing specific shapes of the nanoflakes and specific edge terminations [93]. Within the nanoflake, S vacancies, or 1S atomic columns, are created by removing randomly selected S atoms. The same can be done to create holes (empty atomic columns) by removing entire 2S or 1Mo columns. Similar rotations and translations as presented in Figure 3.2 for the Au nanoparticles are applied to the MoS_2 nanoflakes to introduce more diversity in the spatial placing and orientation in the (x,y)-plane.

Listing 3.5 presents a *Python* script for generating numerous monolayer MoS_2 nanoflakes, where the only required input is the folder name to place the outputs summarised in Table 3.1 and the number of systems. The code currently applies a default beam energy of 50 keV, but another value can be supplied by the user. The user can also apply perturbations to the atomic positions by a Gaussian defined by the `distort` value in unit Å, and even a Debye-Waller smearing of the atomic potential, which will be described later in Chapter 5.

```
$ python make_MoS2.py [Folder name] [Number of systems]
  Optional: --beamenergy [Energy of the electron beam in eV]
  Optional: --debyewaller [Gaussian smearing of the potential in Å]
  Optional: --distort [Gaussian perturbation of atomic positions in Å]
  Optional: --sampling [Sampling of the exit wave in Å/pixel]
  Optional: --arraysize [Number of x, y pixels of the exit wave]
  Optional: -s/--start [Starting number (Default: 0)]
  Optional: -n/--numproc [Number of cores for parallelisation]
  Optional: --seed [Seed random number generator reproducibility]
  Optional: --train [Store as training data set (Default)]
  Optional: --test [Store as validation/test data set]
```

Listing 3.5: An instruction on the *Unix* command-line usage of the *Python* script for generating numerous monolayer MoS_2 nanoflakes with varying shapes and defects.

3.2 Image and Label Generation

The previous section presented how thousands of randomly configured atomic systems are generated. For each atomic system an exit wave is computed *via* the multislice simulation method which is the most computationally expensive part of the data generation segment of the pipeline. From each exit wave a HR-TEM image can be computed by applying a CTF, optional noise along with an optional MTF, and other microscope imperfections from Chapter 1. All of these factors are collectively referred to as the "microscope parameters" and are all set by the user. Applying the microscope imperfections to the image is a relatively cheap operation, which facilitates the generation of multiple images from each exit wave with different microscope parameters. This concept is referred to as an "image epoch". As shown in Figure 3.4, with 3 image epochs, each exit wave produces 3 images that project the same atomic system with different microscope parameters.

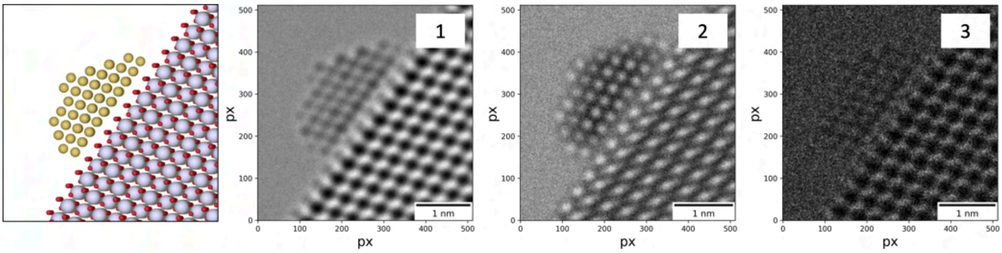


Figure 3.4: Illustration of "image epochs", which are cycles (in this case 3 cycles) of images of the same atomic system (*i.e.* generated from the same exit wave) with varying degrees of microscope imperfections from Chapter 1. This includes the CTF, electron dose, MTF *etc.*, which are collectively referred to as the "microscope parameters". This illustration shows the same atomic system projected under varying degrees of noise, contrast, and blur, which can promote the ability of a neural network to be robust to these variations - more in Section 3.3.

There are a multitude of choices to be made by the user to ensure that the images and labels are sensible. The choices include the range of each microscope parameter, the type of label that should be paired with the image, the number of image epochs, and more. These choices are defined and passed into the pipeline *via* an input file, which will be referred to as the ”parameters file”. The choices of parameters are presented in Table 3.2. These parameters are handled as a *Python* dictionary and stored as a `json` file for reproducibility and book-keeping. This is a key element to allow the user to control the automation of the remaining parts of the pipeline. The user can for example choose to simulate images matching experimental images obtained with either a direct electron detector or with a CCD and scintillator based detector, as described in Section 1.3. The user can generate datasets with various MTF and CTF parameter ranges, varying number of image epochs, and more. All of the microscope parameters should be defined as a range *e.g.* (L, U), where L and U is the lower and upper bound, respectively. See a full example of a parameters file in Listing A.1. A seed value is also definable to allow the user to reproduce exact copies of datasets, by seeding random generators that define the various parameters for each image from the given ranges.

Table 3.2: Input parameters that should be defined by the user. This file defines the ranges of the microscope parameters and will control the automation of the remaining pipeline by initialising the neural network training parameters accordingly.

`Label`	Type: *string* | Options: Mask, Disk, Exitwave Selects the label to be created. Selecting Mask or Disk will generate binary images that segment the entire atomic structure (see Figure 3.5), or individual atomic columns (see Figure 3.6), respectively. Selecting Exitwave will store the real and imaginary part of the exit wave of the system (see Figure 3.7).
`Image epochs`	Type: *int* Number of images (at different microscope parameters) per exit wave/atomic system. Only applies to training data. It will be forced to 1 for validation/test data. See Figure 3.4.
`Image size`	Type: *int tuple* Number of pixels in (x, y). This will crop or zero pad the exit wave if necessary.
`Noise`	Type: *string* or None | Options: poisson The stochastic noise model to apply for shot noise. Can be None for no noise. See Eq. (1.24).
`MTF`	Type: *string* or None | Options: parametric Apply an MTF. See Eq. (1.25).
`Number of classes`	Type: *int* The number of classes. Effectively the number of output channels *i.e.* Output shape = N_x x N_y x $\mathrm{N}_{\text{channels}}$. For the exit wave this must be 2 to store the real and imaginary part.

Continued on next page

Table 3.2 – continued from previous page

`Null class`	Type: *bool* Whether or not to include a background class. This channel is the sum of all other channels and inverted.
`Spot size`	Type: *float* Width of the circular discs used to segment atomic columns in unit Å. See option “label: ’Disk’ ”.
`Multifocus`	Type: *float tuple* or None `tuple`(Number of images in focal series, defocus step in Å, uncertainty of defocus step in Å) - *e.g.* `tuple`(3, 50, 1) will generate a focal series of 3 images with (50 ± 1) Å in between.
`Normalisation Distance`	Type: *float* Gaussian width in Å for image standardisation. See Section 3.3.
`Debug`	Type: *int* Number of images and labels to save into a debug folder for verification.
`Seed`	Type: *int* Seed for reproducing microscope parameters below.
Microscope Parameters All parameters below are type *float tuple* to specify the lower and upper bound.	
`Sampling`	Sampling range in unit Å/pixel. This should match experimental data.
`Dose`	Range of electron dose in $e^{-}/$Å on a log scale. See Eq. (1.28).
`Blur`	Range for Gaussian width of blur in Å. See Section 1.2.
`Focal spread`	Range of focal spread in Å. See Section 1.2.
`Defocus`	Defocus range in Å. See Section 1.2.
`CTF`	Range for c_{30}, c_{12}, c_{21}, c_{23}, c_{32}, c_{34}, c_{45} individually. See Section 1.2.
`MTF`	Range for $c_1 \equiv C$, $c_2 \equiv c_0$, $c_3 \equiv c_3$ individually. See Eq. (1.25).
`Readout`	Range for N_0. See Eq. (1.28).

The pipeline supplies a *Python* script for generating a great number of images and paired labels. Only two inputs are necessary from the user. These inputs include the parameters file, which is passed as an argument along with the path to the atomic systems generated prior to this *i.e.* the path where Table 3.1 is stored. This is documented in Listing 3.6.

```
$ python make_image_data.py [Folder with atomic models+waves+points]
  [Input parameters]
  Optional: --train # Training data?
  Optional: --test  # Validation/test data?
  Optional: -n/--numproc [Number of cores for parallelisation]
  Optional: --seed [seed value for reproducibility]
```

Listing 3.6: An instruction on the *Unix* command-line usage of the *Python* script for generating images and labels.

Notice that Listings 3.3, 3.5 and 3.6 all provide options to select between training and validation/test data. Referring back to Section 2.5, the user will be training neural networks with a training data set, but will need to verify the neural network's generalisability on separate data sets. At the very least the user must generate a training data set and validation data set to verify the neural network training, but numerous test data sets can be generated with different ranges of microscope parameters to test the neural network on varying conditions post-training.

Section 2.2 introduced the idea of supervised learning. In short, supervised learning involves training the neural network to map the input to the paired label (output). This means that the label will define the task for the neural network, which for this thesis consists of multiple segmentation tasks and the exit wave reconstruction, which is a regression task. Three types of labels have been the focus of this work: Mask, Disk, and Exitwave labels, each of which will be defined and described, starting with Mask labels.

Mask Labels

> Description: A binary label that separates pixels belonging to a spatial region of interest of the input image (Binary segmentation).

An example of this type of label is shown in Figure 3.5. During the atomic modelling and exit wave simulation stage explained in Section 3.1, sites are stored containing the positions of the atomic columns. The convex hull of these sites are used to generate an outlining mask of the atomic system. This binary mask will have the same resolution and size as the image and effectively assigns each pixel to either a class belonging to the atomic system of interest or a background class.

These labels are useful for identifying regions of interest in large samples consisting of many nanoparticles. The regions of interest are masked with pixel-wise resolution, which is a probability map of each pixel belonging to either one of the classes. From the masked region the user can utilise the mask to extract morphological properties, such as the size of the nanoparticle, and to crop spatial regions and compute Fourier transforms to identify crystallographic properties and orientation [14].

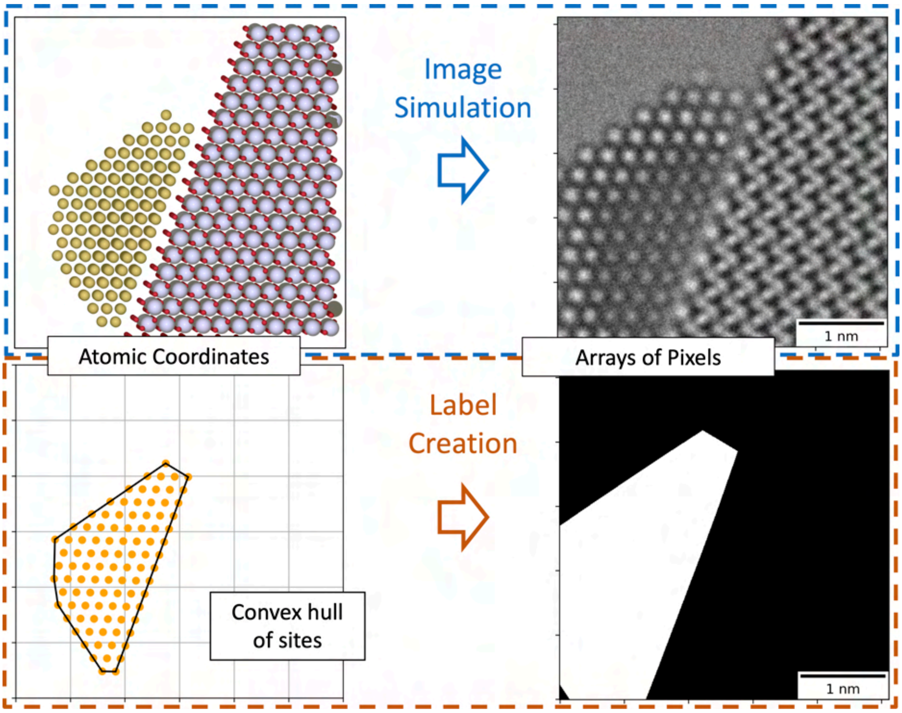


Figure 3.5: Example of a Mask label for a CeO_2 supported Au nanoparticle. Top: Presents the simulate HR-TEM image for a CeO_2 supported Au nanoparticle, with an image size of 512 x 512 pixels. Bottom: Presents the convex hull of the stored sites (atomic column positions) being used to generate an outlining mask. The binary mask has the same resolution and size as the image above and separates the pixels belonging to the Au nanoparticle from the rest of the image by assigning each pixel to a class.

Disk Labels

> Description: Binary labels that separate pixels belonging to various atomic columns for a given atomic system. The label separates the atomic columns by one-hot encoding of multiple classes (Multi-class segmentation).

In MoS_2 multiple classes of atomic columns exist, namely 1Mo, 2S, and 1S atomic columns, which is presented in Figure 3.6. The atomic sites that were stored during atomic modelling and exit wave generation, were categorised into their respective classes depending on the chemical composition of the atomic site. For each class a binary mask is generated where a binary disc is placed at each site corresponding to that class. The width of the disc is defined by the user (see “spotsize” in Table 3.2). Each class is a separate channel in that output. Each channel is a binary label for the given class representing the probability of the pixels belonging to each class, which is referred to as one-hot encoding.

These labels are useful for identifying and classifying atomic columns to probe atom dynamics under various environmental conditions, forming strain maps, and identifying defects [11, 94]. In the illustration a focal series of 3 images is presented, which can be toggled by the “multifocus” parameter in Table 3.2.

Exitwave Labels

> Description: The stored exit wave resampled to match the user defined sampling of the input image. The label will have 2 channels to store the real and imaginary part of the exit wave (Regression).

An example is shown in Figure 3.7 with a focal series of three images. The image is computed from the exit wave, which was generated and stored from the atomic modelling segment of the pipeline in Section 3.1. The exit wave corresponding to the image is resampled to match the image, and the real and imaginary part is stored as a 2-channel label. This label differs from the previous two, as it is no longer binary. A neural network that has to reconstruct an exit wave from images will solve a regression task, rather than the segmentation task assigned by the Mask and Disk labels. This requires different settings in the training process, which will be discussed in the Section 3.3.

As described in Section 1.4, exit wave reconstruction is a common practice in HR-TEM analysis and is usually computed from focal series consisting of 20-50 images [29, 38]. It is a popular method to facilitate structural determination of atomic systems.

This concludes the data generation segment of the pipeline. Rich datasets of simulated HR-TEM images of diverse atomic systems are supplied with pixel-wise accurate labels for various tasks. The next segment will involve training neural networks to map between the images and given labels.

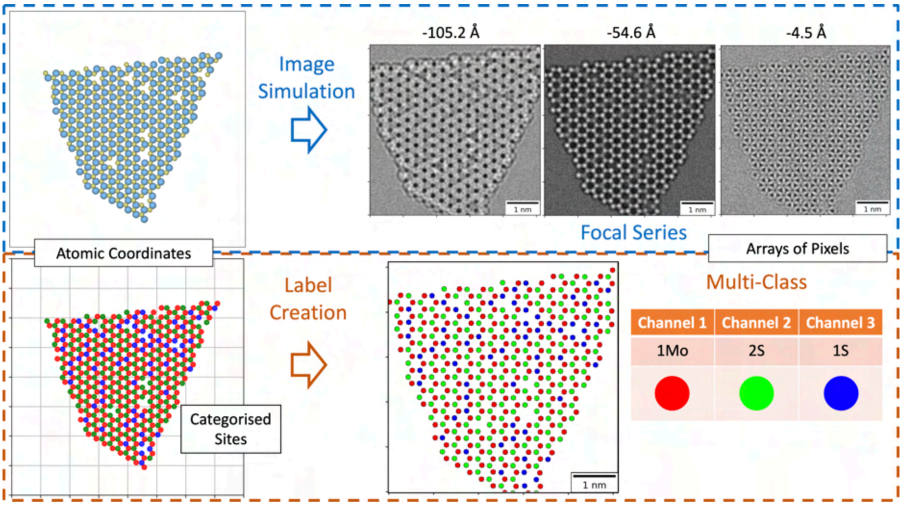


Figure 3.6: Example of a Disk label for monolayer MoS_2 (2H phase). Top: Presents a focal series of 3 images of the MoS_2 sample, which is generated by toggling the "multifocus" parameter in Table 3.2. The images have a fixed size of 512 x 512 pixels. Bottom: Presents the categorised sites and a corresponding "Disk" label as a 3-channel binary array (RGB) classifying the pixels belonging to atomic columns of 1Mo, 2S, or 1S.

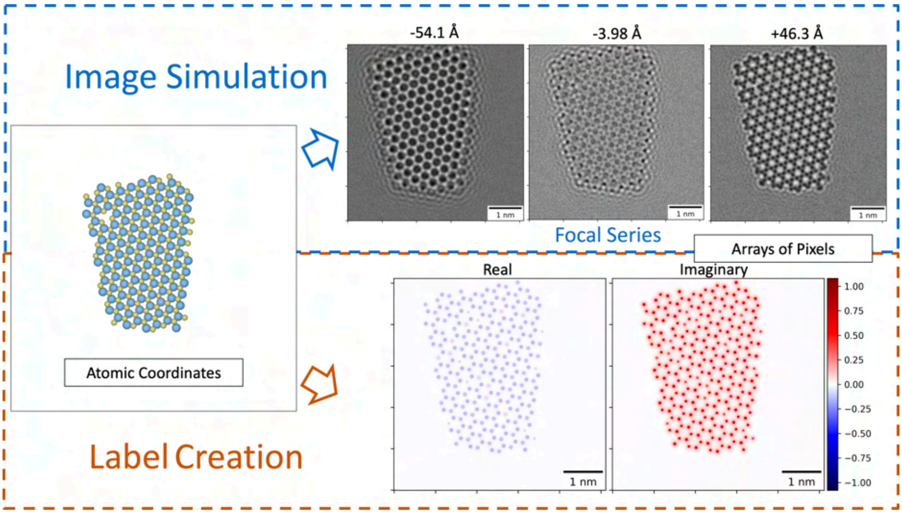


Figure 3.7: Example of an Exitwave label for monolayer MoS_2 (2H phase). Top: Presents a focal series of 3 images of the MoS_2 sample, which is generated by toggling the "multifocus" parameter in Table 3.2. The images have a fixed size of 512 x 512 pixels. Bottom: The corresponding Exitwave label that is stored as a 2-channel array with the real and imaginary part. The exit wave is resampled to match the sampling of the image. This sampling is provided by the user - see Table 3.2.

3.3 Neural Network Training

The previous sections describe how datasets of hundreds to thousands of randomly yet realistically configured atomic systems can be generated with ease. From each model, HR-TEM images can be simulated along with corresponding labels that will assign the network a specific task to learn. These tasks consist of segmentation tasks by classifying pixels into nanoparticle regions or atomic columns of a given category, or the regression task of performing an exit wave reconstruction.

The next step is to feed this data into a neural network to train it for whichever task required. Neural network training is split into three parts: Data Loading, Neural Network Initialisation, and Feed Forward and Back Propagation.

Part 1: Data Loading

There must be a consistent way to feed data into the neural network for training. One of the most vital aspects of this is to ensure that all images contain pixels in the same range of intensities. This is most commonly referred to as normalisation or standardisation, where the former is defined as shifting the pixel intensity range between 0 and 1, and the latter is defined as shifting the pixel intensity range to have a mean of 0 and standard deviation of 1. Standardisation is only applicable under the assumption that the pixel distribution follows a normal distribution. This work applies a standardisation method referred to as a “local standardisation”, which is defined as

$$\hat{I}_{xyc} = \frac{I_{xyc} - \frac{1}{N_c}\sum_c (I_c * G)_{xy}}{\frac{1}{N_c}\sum_c \sqrt{(I_c^2 * G)_{xy}}} \quad ; \quad G_{xy} = e^{-\frac{x^2+y^2}{2\sigma^2}} \,, \tag{3.1}$$

for an image I with real space pixels (x,y) and c denotes the channel for an input with N_c-channels. In the case of a focal series, with more than one input channel, the subtracting term and normalisation term are both averaged over the number of channels. The σ is defined by the “Normalisation Distance” in Table 3.2, which specifies the “locality” of the standardisation and it is recommended that this distance is larger than the defining features in the image. This method of standardising is robust to empty or over-saturated pixels and varying illumination in the (x,y)-plane that may occur in HR-TEM data.

Data augmentation is a key method to introduce sensible variations to the data in order to avoid overfitting and train a more robust network. Common data augmentations for images are adjusting the brightness (Eq. (3.2)), contrast (Eq. (3.3)), and gamma (Eq. (3.4)) of the image. The operations are

$$\hat{I}_{xyc} = I_{xyc} + \mathcal{U}[a,b] \tag{3.2}$$

$$\hat{I}_{xyc} = (I_{xyc} - \bar{I}_{xyc}) \cdot \mathcal{U}[a,b] + \bar{I}_{xy} \tag{3.3}$$

$$\hat{I}_{xyc} = (I_{xyc} - min[I_{xyc}]) \cdot \mathcal{U}[a,b] + min[I_{xyc}] \tag{3.4}$$

where $\mathcal{U}[a,b]$ defines a random number drawn from a uniform distribution between a and b, and $\bar{I}_{xyc}$ denotes the mean of the image for a given channel. Other common data augmentations involve flipping the images and labels both horisontally and vertically at random, which is also implemented. In this work an additional data augmentation is applied, namely the image epochs described in Section 3.2. The images are loaded one image epoch at a time, so that each training epoch utilises the same atomic systems imaged at varying microscope parameters. When the training epochs are larger than the image epochs, the training will continue and cycle through the image epochs as many times as necessary. The image epochs, along with the brightness, contrast and gamma augmentations, promote the robustness of the neural networks against varying microscope conditions.

Part 2: Neural Network Initialisation

With the data prepared the next step is to initialise the neural networks. Neural network initialisation involves defining the convolutional and other various layers, activation functions, and optimiser, all described in Chapter 2. In this work three existing neural network architectures have been implemented: The U-net [95] (modified with inspiration from the FusionNet [96]), U-net++ [97] (also modified with inspiration from the FusionNet), and MSD-net [98]. All three neural network architectures have been applied to various problems in general microscopy [11, 14, 98].

The MSD-net and U-net/U-net++ take different approaches in identifying features across varying length scales. The U-net has its name from the characteristic U-shape due to the series of down-sampling layers to spread features across a larger length scale, followed by an equal number of up-sampling layers to return to the original resolution. The down-sampling is achieved by a max-pooling, which takes the maximum of a set of pixels as the value for a pixel in the down-sampled image. The up-sampling is achieved by bi-linear interpolation. The stages of down-sampling and up-sampling are referred to as “levels” of the U-net architecture. While the layer resolution decreases across layers, the channels (number of feature maps - see Section 2.4) are increased.

The MSD-net in contrast retains the same resolution throughout all layers and the same number of channels, but dilates the kernel to spread its weights over a larger region (see Figure 2.3(b)). The dilation value iterates over some given range repeatedly.

The MSD-net belongs to the family of dense-nets, as every layer in the neural network is connected to previous layers by concatenations following every convolution. These concatenations are the skip connections introduce in Section 2.5 and share feature information between different layers in the neural networks. The U-net applies skip connections between outputs of the same resolution. As an extension to the U-net, the U-net++ takes the U-net architecture and includes more skip connections across outputs of layers at different resolutions, resulting in a more densely connected network.

An illustration of all three neural network architectures is presented in Figure 3.8 along with a legend defining the various layers (convolutional, max-pooling, up-sampling *etc.*) constructed using the *TensorFlow+Keras* API [78]. The input channels and output channels are automatically detected from the parameters file (refer to Table 3.2). The output activation function will vary depending on the given label or problem defined by the user. For segmentation problems, the output must apply a sigmoid activation for binary segmentation and softmax activation for multi-class segmentation (see Section 2.3 for reasoning). The sigmoid/softmax activation is applied on a

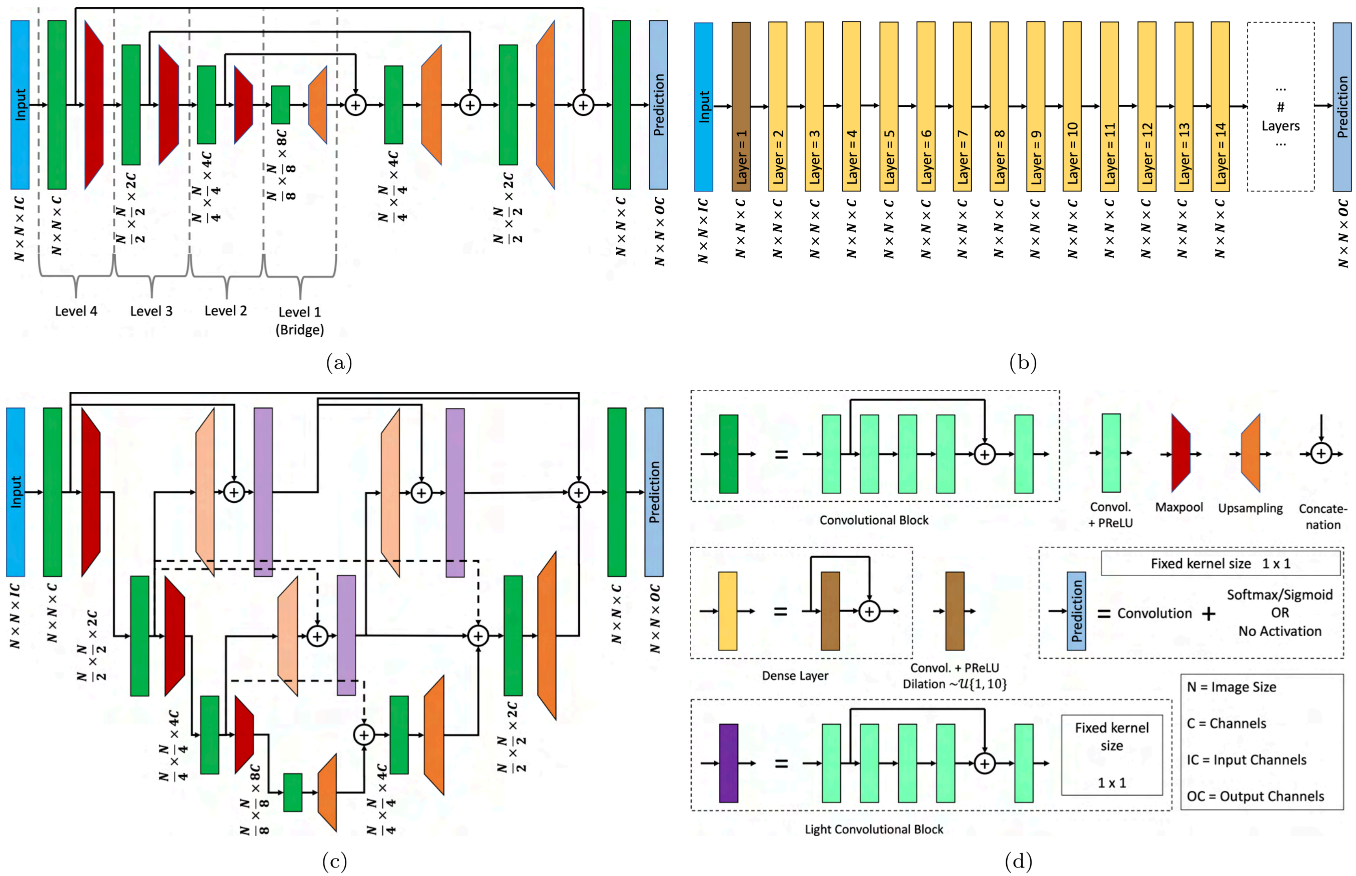


Figure 3.8: Illustration of this works implementation of the (a) U-net, (b) MSD-net, and (c) U-net++. Refer to the legend in (d) for definitions of the various layers. All networks apply a Parametric Rectified Linear Unit (PReLU) activation function (see Section 2.3) for all layers besides the output layer.

pixel-wise level by defining the output layer as a convolutional layer with a kernel size of 1, effectively producing a probability for each pixel belonging to a given class. For regression tasks the final output layer is a pixel-wise convolutional layer with no activation function applied to allow for the pixel-wise values of, for example, the real and imaginary part of the exit wave to be the output. The script automatically initialises the output activation based on the parameters file from Table 3.2. All networks apply a PReLU activation function in hidden layers between the input and output for computational efficiency, as introduced in Section 2.3. The user also has the choice between an RMSProp and ADAM optimiser (see Section 2.2), with the ADAM + AMSGrad optimiser as a default [68].

Part 3: Feed Forward and Back Propagation

The final step is to train the neural network that has been initialised. The training procedure, described in Chapter 2, consists of repeated feed forward and back propagation steps *i.e.* training epochs to minimise the loss function between the label and predicted output. Based on the parameters file (refer to Table 3.2) the loss function will be automatically defined depending on the problem. Binary segmentation will apply the binary cross-entropy loss function, and multi-class segmentation will use the multi-class equivalent (see Eq. (2.7)). If the task is a regression, the loss function will be a mean squared error loss function (see Eq. (2.6)). A visual of the training procedure is shown in Figure 3.9 for the MoS_2 multi-class segmentation problem of identifying pixels belonging to 1Mo, 2S, 1S atomic columns, or the background. In this case a focal series of 3 images is applied.

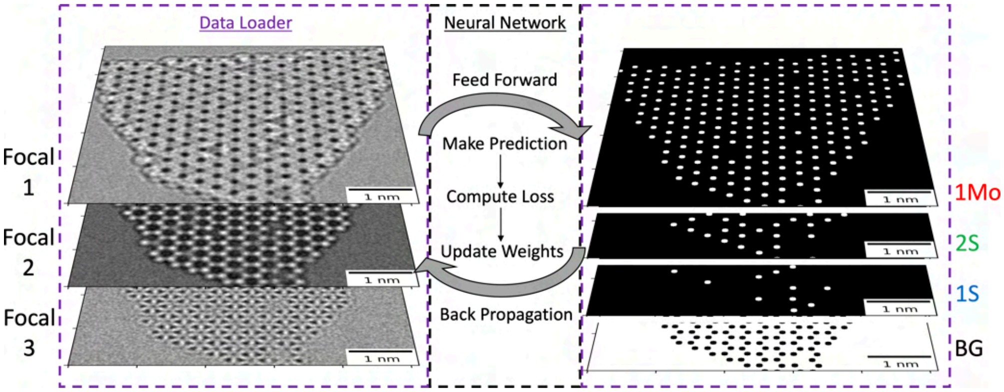


Figure 3.9: Illustration of training procedure for the MoS_2 multi-class segmentation problem of identifying pixels belonging to 1Mo, 2S, 1S atomic columns, or the background (BG). In this case a focal series of 3 images is applied. The number of input channels is 3 and output channels is 4.

The pipeline supplies *Python* scripts for running a training procedure for each of the neural networks in Figure 3.8. Documentation for the MSD-net *Python* script is shown in Listing 3.7, along with a U-net and U-net++ equivalent in Listings A.2 and A.3, respectively. The necessary inputs include the number of training epochs, the number of convolutional layers (specific to the MSD-net), the path to the training data and validation data from the previous segments in the pipeline, and a folder name for the output where the trained neural network is stored along with the averages loss function values after each training epoch. The user must have access to these values to ensure that the loss function is minimised by the end of the requested number of training epochs for both the training and validation data set.

```
$ python train_imageepochs.py [Epochs] [Layers]
  [Training Data] [Validation Data] [Output Folder]
  Optional: --filters [Number of channels]
  Optional: --kernel [Kernel size]
  Optional: --restart [Restart from epoch number]
  Optional: --epochsave [Epoch interval for saving network]
  Optional: --adam [Adam optimiser passed as dict with parameters]
  Optional: --rmsprop [RMSProp learning rate]
  Optional: --limitdata [Truncate training data at a given number]
```

Listing 3.7: An instruction on the *Unix* command-line usage of the *Python* script for training an MSD-net.

At this point a user has the knowledge to run the data generation and neural network training segments of the pipeline with only a few command in a *Unix* terminal. Within an hour the user can generate over 1000 atomic models and over 10,000 images and labels for training and validation. Training times can vary between hours to days depending on the user defined options. All the scripts are therefore compatible with HPC clusters in any case that it should be necessary. The simplicity of this is presented in Listing 3.8 and Listing A.4 for two different examples.

```
# Data Generation (Atomic Models, Exit waves, Sites)
$ python make_MoS2.py example_data 1000
$ python make_MoS2.py example_data 500 --test

# Data Generation (Images and Labels)
$ python make_image_data.py example_data example_input_parameters.json
$ python make_image_data.py example_data example_input_parameters.json
  --test

# Neural Network Training (MSD-net)
$ python train_imageepochs.py 100 50 example_data example_data-test
  example_network
```

Listing 3.8: Example of running the data generation and neural network segment of the pipeline, where a training and validation/test data set of MoS_2 nanoflakes are generated with images and labels defined by an input parameters file, and an MSD-net is trained.

The final segment involves deploying the trained neural network to validate their performance on simulated data and apply them to experimental data for quantitative data extraction.

3.4 Python Objects for Data Handling

Following the previous sections, the user should have a simple interface to apply the trained neural networks to both simulated and experimental images. For this purpose two *Python* objects have been constructed: The `Simulated_Data`-object and The `Experimental_Data`-object.

The Simulated Data Object

The basic idea of the `Simulated_Data`-object is to allow the user to intialise an object in *Python* with the path to the image and label file, the path to the file containing specific microscope parameters of that image (stored together with the simulated images), and the path to the parameters file defined by the user (see Table 3.2). The initialised `Simulated_Data`-object enable the application of a trained neural network on a HR-TEM image, visualisation of the result, and verification of the neural network output by some metric. The neural network output is referred to as a "prediction" or "inference". Various metrics exist to gauge the performance of the prediction.

For regression tasks, such as the exit wave reconstruction, the mean squared error, Eq. (2.6), is suitable as a score between the predicted exit wave and the ground truth exit wave. The square root of Eq. (2.6) is also applicable and is known as the root mean squared error.

When the task, however, is segmentation, the prediction (by the sigmoid or softmax output activation) is a probability map of each pixel belonging to a certain class. By applying a threshold probability, this prediction can be converted to a binary array, classifying each pixel to a class by one-hot encoding. When comparing the binarised prediction to the ground truth (which is also binary), each pixel in the prediction can be considered as either a true positive (TP), false positive (FP), true negative (TN), or false negative (FN) classification. A metric for segmentation tasks based on these values is known as the F1-Score, which is defined as

$$\mathrm{F1} = \frac{TP}{TP + \frac{1}{2}(FN + FP)} = 2 \cdot \frac{P \cdot R}{P + R}, \tag{3.5}$$

where P and R are the precision and recall, respectively. The F1-Score is a harmonic mean between the precision and recall. The precision is a measure of the neural network's ability to correctly classify pixels for a given class and is defined as

$$P = \frac{TP}{TP + FP}. \tag{3.6}$$

The recall on the other hand is a measure of the neural network's ability to identify a given class and is defined as

$$R = \frac{TP}{TP + FN}. \tag{3.7}$$

Ultimately, both a good precision and recall is desired, which motivates the use of the F1-Score. This score ranges between 0 and 1, where 1 represents a perfect segmentation.

An example of using the `Simulated_Data`-object is presented in Listing 3.9. This example is an extension of Listing 3.8 and shows how the output of the pipeline is leveraged by the `Simulated_Data`-object to contain the image and label, apply a standardisation, infer the prediction of the neural network, and compute an F1-Score between the prediction and label. See Table A.1 for an explanation of the individual method attributes of the `Simulated_Data`-object.

```
import glob
# Select dataset
fn = 'example_data-test'

# Grab all images and labels, specific image parameters,
# and the global input parameters file
image_parameters_file = fn + '/parameters.json'
image_label_files = sorted(glob.glob(fn + '/images_labels/*.npz'))
tem_parameters_files = sorted(glob.glob(fn + '/tem_params/*'))

from temnn.analysis.SimulatedAnalysis import Simulated_Data
# Initiialise a Simulated_Data instance
simulated_data = Simulated_Data(image_label_files[0],
                                image_parameters_file,
                                tem_parameters_files[0])
# Load the image, label, and parameters
simulated_data.load()
# Apply local standardisation
simulated_data.cp_local_standardise()

import tensorflow.keras as keras
# Load the final trained model
network_file = 'example_network/model-0'
mod = keras.models.load_model(network_file)

# Apply the network, compute the F1-Score,
# and plot the image along with the result
simulated_data.infer(mod)
f1 = simulated_data.get_f1_score()
simulated_data.plot() # plots the image and prediction side by side
                      # with a scale bar
```

Listing 3.9: Code example presenting the usage of the `Simulated_Data`-object.

The Experimental Data Object

The same ease is provided for experimental data as presented in Listing 3.10 and is achieved using Hyperspy [99] to manage the `dm3`/`dm4` or `Tiff` files that contain the experimentally obtained image and metadata. Usually many lines of code are required to load the image and extract specific metadata. The `Experimental_Data`-object removes a lot of that complexity. The object is initialised with just a path to the `dm3`/`dm4` or `Tiff` file. The `Experimental_Data`-object will contain various metadata as attributes. Utilising the class methods, the user can compute the dose, apply a neural network, visualise the results, extract the Fourier transform, all with one line of code.

An example is shown in Listing 3.10, which is also an extension of Listing 3.8, where the trained neural network is supplied to the initialised instance of the `Experimental_Data`-object. Working with experimental data means that there is no ground truth, disallowing the use of a metric to gauge the performance as in the simulated data case. See Table A.1 for an explanation of the individual methods and attributes of the `Experimental_Data`-object.

```
import glob
# Select dataset
fn = 'exp_example_data'
files = glob.glob(fn + '/Hour_00/Minute_*/Second_*/*.dm4')

# Initialise an Experimental_Data instance
from temnn.analysis.ExperimentalAnalysis import Experimental_Data
experimental_data = Experimental_Data(files[0])

# Load the image and metadata
experimental_data.load()
# Apply local standardisation
standardised_image = experimental_data.cp_local_standardise()

import tensorflow.keras as keras
# Load the final trained model
network_file = 'example_network/model-0'
mod = keras.models.load_model(network_file)

# Apply the network and plot the image along with the result
inference = experimental_data.infer(mod)
dose = experimental_data.compute_dose()
experimental_data.plot() # plots the image and prediction side by side
                         # with a scale bar
```

Listing 3.10: Code example presenting the use of the `Experimental_Data` object.

These two previous code listings (Listings 3.9 and 3.10) present how the user can continue the pipeline and easily apply the trained neural networks and extract the inferences, whether it be a mask segmentation, atomic column segmentation, or exit wave reconstruction.

3.5 Instance Segmentation Tracking for Large-Scale Analysis

Last but not least the user should have access to tools that apply these neural networks in consistent ways to extract information. This section focuses on the mask segmentation of oxide supported nanoparticles. These segmentations are a powerful tool to automate region of interest (ROI) selection with pixel-wise precision and to extract properties such as the position and shape of each region (in this case nanoparticles). The aim is to do this in a large-scale sense with minimal user interference to extract statistically signification data on various properties, similar to Refs. [9, 13].

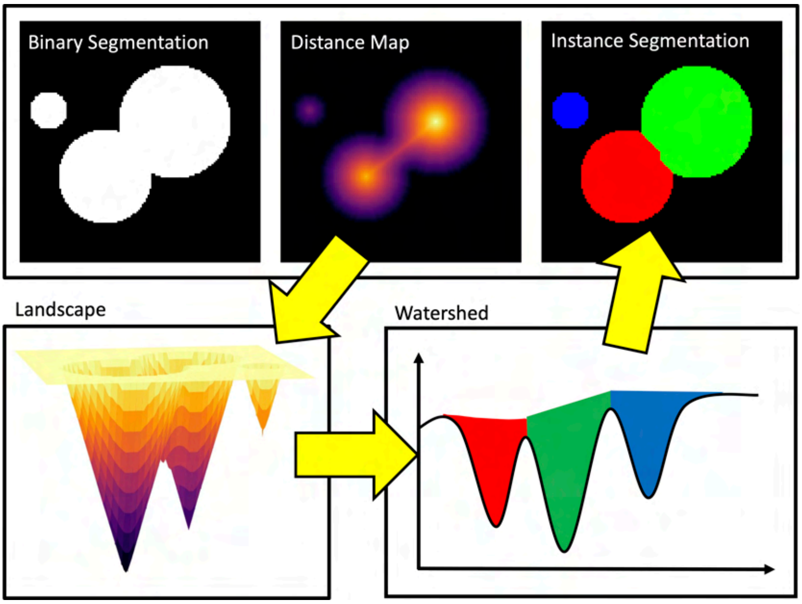


Figure 3.10: Illustration of the pseudo-instance segmentation implemented to separate segmented regions into individual instances. This is done by performing a watershed on the landscape of the distance map of the binary segmentation. Separating instance can is the first step in facilitating large-scale analysis.

HR-TEM images are segmented frame by frame. Each segmentation is a map over the probability of each pixel belonging to a certain class. After applying a threshold, a binary map is produced highlighting the nanoparticle regions, as exemplified in Figure 3.5. To analyse the nanoparticles individually the regions must be separated, where each region should cover a single nanoparticle. To do so the segmentation must be separated into individual instances, this is known as instance segmentation [6]. Obtaining individual instances is achieved by taking the binary mask and computing a distance map - a map of the distance from every pixel with value 1 to the nearest pixel with value 0. An example is shown in Figure 3.10, where segmented regions are even overlapping. The distance map forms peaks at the centre of the individual segmented regions. These peaks are passed as markers to a watershed algorithm which fills a unique value into each peak up to the nearest boundary [100]. This performs a pseudo-instance segmentation, where the separation of instances is done post-segmentation.

The pseudo-instance segmentation approach converts the binary array into an array with multiple unique values for each ROI. The unique value is used as an identification number (ID) to tag the ROI. An `Instance2D`-object is defined that stores the ID along with properties of the ROI, such as the area computed from the segmentation and the centre-of-mass. A `Slice`-object is defined to contain the entire segmented frame, which can hold multiple instances of `Instance2D`-objects *i.e.* an image can contain multiple nanoparticles where each nanoparticle is separately accessible. To track properties of each `ROI`-instance across frames, individual `ROI`-instances are connected between con-

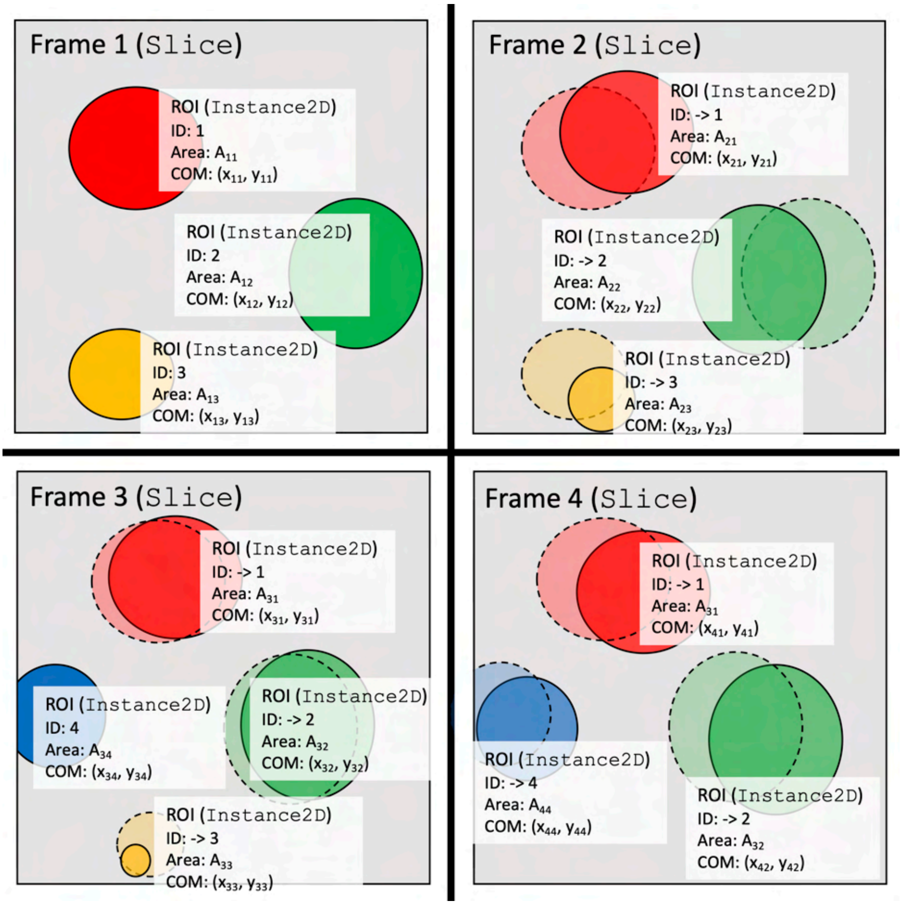


Figure 3.11: Illustration of how individual instances are tracked across frames. Every instance in a frame that is near an instance in the previous frame is selected and the overlap of the selected instance with the instance from the previous frame is taken. The selected instance with the largest overlap, above a threshold percentage, is assigned the same ID. Tracking the instances allow for the extraction of properties such as the area, and centre-of-mass, of each instance across all frames.

secutive frames. If the area of overlap of two ROIs in two consecutive frames is over a threshold percentage, then the latter `ROI`-instance ID is matched with the former `ROI`-instance ID, as shown in Figure 3.11. If multiple ROIs have a significant overlap, then the one with the highest overlap percentage is taken. This connects all `ROI`-instances across all frames, so that properties can be extracted across frames *i.e.* as a function of time. The threshold percentage controls the amount of movement there may be for a given ROI between consecutive frames. This is currently set to 50%.

Table 3.3 summarises the method attributes of the `Instance2D`-object and the `Slice`-object. The former stores the ROI array along with the ID and properties, and the latter stores the segmented frame and multiple instances of `Instance2D`. It is beneficial to construct it this way as it is straight forward to implement new properties into the `Instance2D`-object, making the code easily expandable.

Table 3.3: Overview of classes handling the the segmented frames and segmentation instances, to extract properties of each nanoparticle across all frames..

Class `Instance2D`

Contains a `ROI`-instance together with its id, properties *e.g.* area, centre of mass *etc.*

Attributes	Descriptions
`get_id()/set_id()`	get/set the id for the instance. This is used to set the id of an instance in a frame to match an instance in the previous frame with significant overlap.
`get_area()/set_area()`	get/set the area of the `ROI`-instance. This is set using the number of pixels covered by the segmentation multiplied by the sampling.
`get_com()/set_com()`	get/set the centre of mass (COM) of the `ROI`-instance.

Class `Slice`

Contains the frame segmentation and `ROI`-instances.

Attributes	Descriptions
`get_instances_2d()`	returns all `ROI`-instances.
`add_instance_2d()`	adds a `ROI`-instance.

Instance tracking across many frames of data facilitates large-scale data analysis and will serve as a core concept behind the analysis tools presented in Chapter 6. The *Python* objects in Section 3.4 are available as part of the pipeline and can be utilised in conjunction with the `Slice`- and `Instance2D`-objects to analyse many frames of data containing many ROI. All the objects are prepared to be used with *Jupyter notebooks.* Instead of presenting examples of large-scale analysis with *Jupyter notebooks*, however, Chapter 6 presents how these tools have been implemented with a graphical user interface.

CHAPTER 4

Results: Software Pipeline

Chapter 3 presented a software pipeline to generate large and diverse data sets of HR-TEM images with paired labels, and to train neural networks for various tasks. The label paired with the image defines the task and three main types of labels and tasks presented in Section 3.2 were:

- Exitwave labels for training a neural network to regress between the HR-TEM image and the real and imaginary part of the exit wave,
- Mask labels for training a neural network to segment regions of the HR-TEM image that belong to a region of interest, for example a nanoparticle,
- Disk labels for training a neural network to segment and classify individual atomic columns in material samples.

The following chapter will present results obtained for each task. The results have been compiled into publications available in Chapter 8. The presentation in this chapter will be a summary of the main highlights from each publication. For full details of all the results and more specific details regarding the methods and discussions refer to the the publications in Chapter 8.

While being a summary of the main results of this work, this chapter is also a showcase of the applications of the software pipeline for neural network assisted HR-TEM from Chapter 3. All training and validation/test data generated and neural network training in the upcoming sections is achieved by executing the lines in Listing 3.8 for the various contexts. Each data set is constructed based on the atomic modelling scripts from Section 3.1 and a parameters file (defined in Table 3.2 with an example in Listing A.1) that defines the label along with the range of microscope parameters and noise to set the diversity of the HR-TEM images in the data set. The specific parameters for each publication are available in Chapter 8. A large degree of the visualisations and computed metrics are obtained using the `Simulated_Data`- and `Experimental_Data`-object presented in Section 3.4.

Chapter 2 introduced convolutional neural network parameters such as the convolutional kernel size, learning rate, optimiser, and more, all of which have a default in the software pipeline, but are customisable by the user, as described in Section 3.3. In all of the following publications the optimal parameters were determined by hyperparameter tuning, which consists of running the training procedure multiple times for a different set of parameters, and manually determining which had the best convergence of the loss function. The "best" convergence involves converging to a minimum within a reasonable number of training epochs, and obtaining the lowest minimum of the loss function for both the training and validation data set. The convergence of the loss functions over training epochs, otherwise known as learning curves, are available as supplementary material for each publication in Chapter 8.

4.1 Publication I: Reconstructing the Exit Wave of 2D Materials in High-Resolution Transmission Electron Microscopy Using Machine Learning

In Publication [I], the aim was to present the applicability of neural networks for the exit wave reconstruction task described in Section 1.4. The current state of the art for exit wave reconstructions are algorithms such as the Gerchberg-Saxton algorithm [54] and only a few attempts have been made to utilise deep learning for phase retrieval in TEM [15] and 4D-STEM [101]. This publication is the first to deploy a neural network to regress between a focal series of HR-TEM images and both the real and imaginary part of the exit wave. The main aim was for the neural network to learn the deconvolution of the CTF defined in Eq. (1.29), and to do it with less information than a traditional algorithm.

Exit wave reconstruction algorithms require detailed knowledge of several CTF parameters and large focal series in order to perform the complex deconvolution [29]. Figure 4.1 shows a predicted exit wave by a U-net given a focal series of 3 images and no information of the specific CTF parameters. The U-net was trained on a data set of 1000 monolayer MoS_2 systems, where the atomic positions are randomly perturbed based on a Gaussian with a spread of 0.01 Å and the atomic potential is smeared by a Gaussian with a spread of 0.0030 Å (see the `distort` and `debyewaller` setting in Listing 3.5). 5% of the atomic columns are randomly selected for vacancy creation (1S columns) and another 5% for holes (empty columns), additionally all systems are randomly tilted up to 10° off the [001] zone axis. For each atomic system a focal series of 3 images is generated from a range of CTF and MTF parameters, varying degrees of noise (shown in Publication [I]), and 10 image epochs (see Figure 3.4), providing a total of 10,000 diverse focal series. Each focal series is a set of 3 images with (50 ± 1) Å in between. The prediction shows an excellent exit wave reconstruction that reveals the atomic structure of the MoS_2 nanoflake and the randomly placed defects (Mo-, single S-, and double S-vacancies), with a root mean squared error (RMSE) score of 0.0062 compared to the ground truth exit wave. This prediction is the median RMSE score from the validation data set of 1000 monolayer MoS_2 systems and focal series. The region with the largest error, which is zoomed-in in the inset of Figure 4.1, shows a misplacement of the atomic column in the exit wave. This misplacement corresponds to a single pixel, or 9.7 pm, which is below realistic resolutions of HR-TEM images.

Monolayer samples are arguably as simple as it gets. Real samples will often involve the presence of other components, such as a substrate. The construction method of monolayer MoS_2 nanoflakes in Section 3.1 was extended to included graphene substrates as a way to introduce more complexity to the problem. Monolayer graphene flakes are constructed identically to the way the monolayer MoS_2 nanoflakes are con-

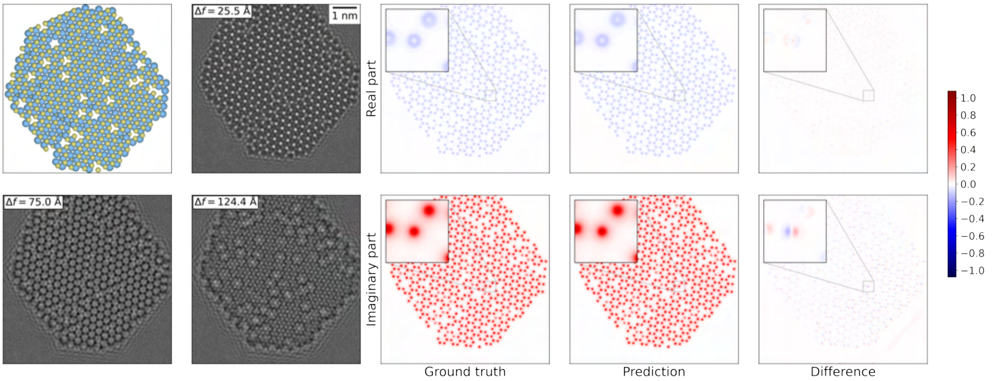


Figure 4.1: Exit wave reconstruction of a monolayer MoS_2 nanoflake by a U-net. The first two columns present the simulated atomic structure and the associated simulated focal series of 3 images as input to the U-net. The third column shows the real and imaginary part of the ground truth exit wave. The fourth columns shows the real and imaginary part of the exit wave prediction by the U-net. The last columns shows the difference in the real and imaginary part between the ground truth and prediction. The RMSE of the prediction is 0.0062. The same colourmap applies to all figures. Reproduced from Publication [I].

structed. The graphene flakes are then placed below the MoS_2 nanoflake along the [001] zone axis. The MoS_2 and graphene flakes are aligned in different ways and placed at varying distances to introduce a large diversity in the overlaps of atomic columns (more info in Publication [I]). In some regions there is even no overlap at all, so the neural network will see some monolayer MoS_2 in the data set.

Figure 4.2 shows the predicted exit wave by a U-net trained on the MoS_2@graphene data set with the same number of atomic systems and focal series as the MoS_2 data set, which again shows excellent performance with an RMSE of 0.0122 and that the largest regions of error are again present due to a single pixel misplacement of atomic columns. The U-net is able to reveal the atomic structure of both the MoS_2 and underlying graphene, along with vacancy defects present in both nanoflakes.

The distribution of the RMSE over a validation set of 1000 focal series of 1000 different atomic systems is shown in Figure 4.3. The difference in the RMSE distribution between the two data sets highlight that the difficulty increases when including the graphene substrate. The decrease in performance when the graphene substrate is included is sensible since the neural network aims to reconstruct both the atomic structure of the MoS_2 nanoflake and the graphene substrate, along with handling the larger variation of intensities present in the images. An additional difficulty was added in the paper by training the U-net on a data set of thousands of 2D materials from the Computational 2D Materials Database (C2DB) [102, 103]. Based on the RMSE distribution of this data set being significantly worse (shown in Publication [I]), it was concluded that the neural networks should be trained on specific atomic systems for

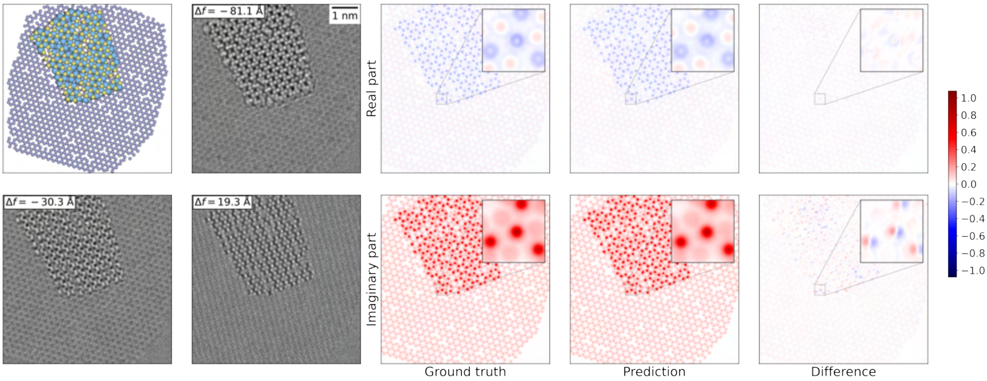


Figure 4.2: Exit wave reconstruction of a graphene supported monolayer $\mathrm{MoS_2}$ nanoflake by a U-net. The columns are the same as in Figure 4.1. The prediction shows that atoms are located both in the $\mathrm{MoS_2}$ nanoflake and the underlying graphene support. The RMSE of the prediction is 0.0122. The same colour map applies to all figures. Reproduced from Publication [I].

more reliable performance. The fact that the neural network performs better when trained on specific atomic systems reveals that it in fact utilises information of the structure of the specific atomic system to generate the exit wave and does not solely perform a deconvolution of the CTF. This conclusion is also backed by a few selected atomic systems from the C2DB that exhibit a small deviation in the symmetry of the structure, where the neural network fails to predict that break in symmetry or places nonphysical defects to satisfy a more symmetric structure (shown in Publication [I]). The U-net had seen a larger number of symmetric structures and ultimately preferred a presence of symmetry. It could be that training on the specific structures with less symmetry would allow the U-net to provide more reliable exit waves of that structure.

It is not sufficient that the neural network merely predicts an exit wave that is visually and quantitatively similar to the ground truth, but also predicts an exit wave that is useful for structural determination. The method for utilising exit waves for structural determination was presented in Section 1.4, which is achievable *via* channelling theory and analysis of the wave function at each atomic column. The mass and defocus circle (see Figure 1.5(b)) is achievable by plotting the complex values of the exit wave at the atomic columns on an Argand plot. The atomic columns are identified in the change in the wave function, *i.e.* $|\psi_{ew}^{column}(\mathbf{r}, z) - 1|$ (see Eq. (1.31)), by traditional peak finding and are then separated into either an Mo or S class based on their separating distance. Each peak corresponds to a complex value in the exit wave that can be plotted for analysis. The Argand plots for the predicted and ground truth exit wave are presented in Figure 4.4 for the same median example in Figure 4.1. The predicted exit wave shows a clear separation along the mass circle, separating the 1Mo, 2S, and 1S atomic columns. The variation along the optical (z-)axis is, however, missing in the prediction.

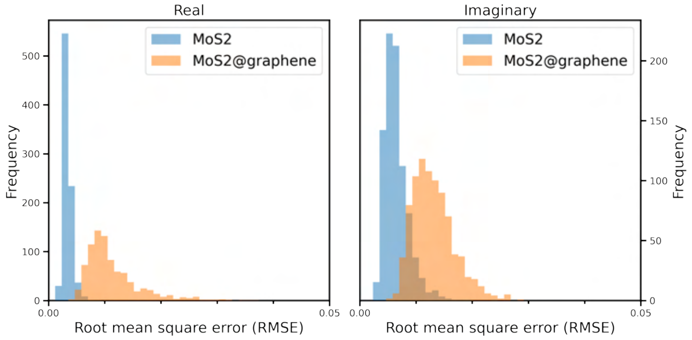


Figure 4.3: Histograms of the RMSE of the focal series in the validation data sets. Blue is the neural network trained on the monolayer MoS_2 nanoflakes, and orange is the graphene supported monolayer MoS_2 nanoflakes. It is clear that the increased complexity from unsupported to supported systems introduces a larger error. Modified from Publication [I].

There is no spread along the defocus circle, as there is in the ground truth exit wave, so the predicted exit waves cannot be used to differentiate between defects in the upper or lower layer of Sulphur from the 1S atomic columns.

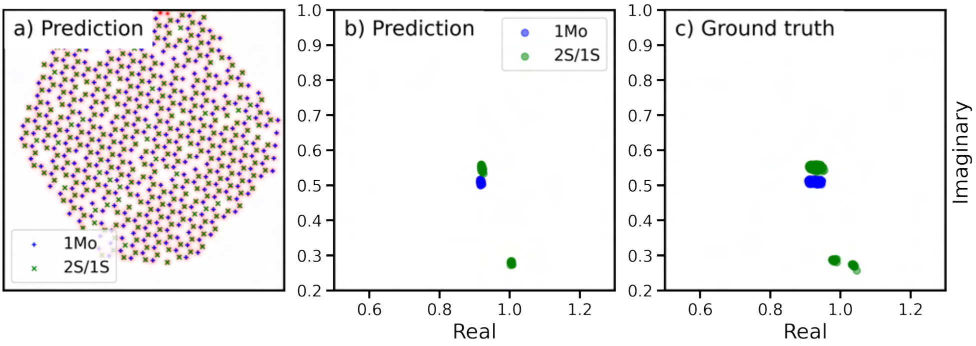


Figure 4.4: Argand plots of the complex value of the exit wave function at the local maximum of $|\psi_{ew}^{column}(\mathbf{r}, z) - 1|$ (see Section 1.4) for the same system as in Figure 4.1. The 1Mo peaks are marked with blue pluses, and 2S/1S peaks with green crosses. The Argand plots of the predicted exit wave present the separation of 1Mo - 2S - 1S atomic columns, however lack the z-position difference between 1S columns as present in the ground truth. Reproduced from Publication [I].

A U-net trained on the MoS_2@graphene data set was applied on an experimental focal series from Ref. [29], where the prediction along with the focal series is presented in Figure 4.5. As a disclaimer, it should be noted that the experimental data set contains a graphite support (*i.e.* multi-layer graphene - instead of the monolayer graphene as in the simulated training data), as well as Co atoms. Regardless of the extra complexities, the neural network is clearly able to identify atomic columns in the MoS_2 nanoflake. The neural network also provides a prediction on the atomic columns of carbon atoms in the support, however these cannot be considered reliable for any structural analysis due to the significant mismatch between the simulated training data and experimental data.

The Argand plots for any system with a substrate present, including the experimental focal series and the simulated MoS_2@graphene data set, are not directly interpretable. There is a large variation in intensities, due to the varying overlap of atomic columns in the material of interest and the underlying substrate. This variation causes a significant smearing of values along the mass circle of the Argand plot (shown in Publication [I]). To circumvent this issue, the substrate is commonly removed from the image by Fourier filtering, permitting an analysis with Argand plots [29]. The problem of analysing Argand plots for supported materials does not affect the method of reconstructing the exit wave presented in this work and was therefore not treated here. A comment will be made on how neural networks can assist treatment of the substrate in Section 7.1.

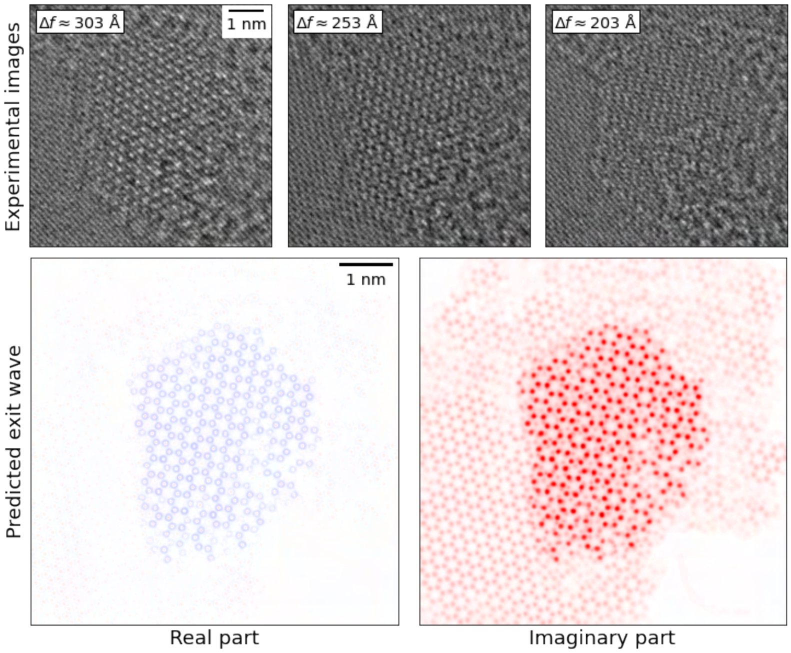


Figure 4.5: Applying the neural network to three experimental images (top row) results in the exit wave function depicted in the bottom row. The defocus values are as reported by MacTempas (overfocus), which uses the opposite sign convention from abTEM and this work. The colours are the same as in Figure 4.1 and Figure 4.2. Reproduced from Publication [I].

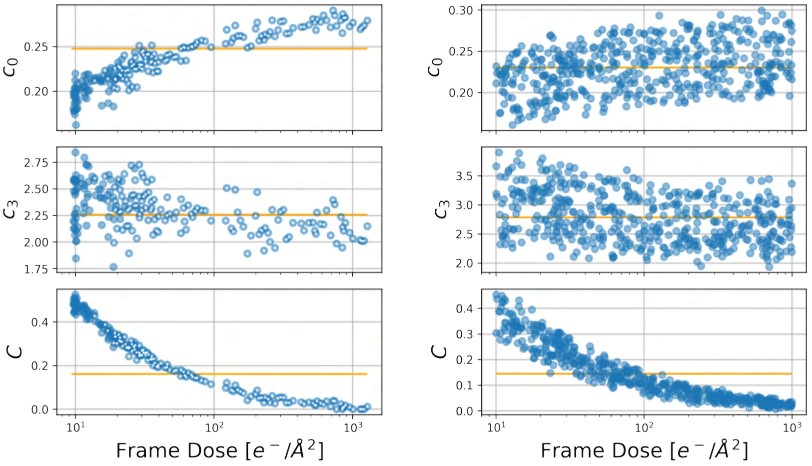
In conclusion, neural networks have been tailored to regress between a focal series to the exit wave function. The neural networks prefer to be trained on specific atomic systems and were shown in Publication [I] to produce exit waves comparable to those reconstructed by traditional algorithms. The neural networks do not require any CTF values as input, as opposed to the traditional algorithms, and require far fewer images in the focal series. This is a novel method that prove neural networks serve as an efficient tool to obtain approximate exit waves that can be used to determine atomic scale structure to a limited degree (missing defocus dependency), before selecting images for a full exit wave analysis by traditional methods to obtain the full 3D positions.

4.2 Publication II - Quantifying Noise Limitations of Neural Network Segmentations in High-Resolution Transmission Electron Microscopy

In Publication [II], the aim was to identify the optimal noise conditions for neural network segmentations. The human interpretation of HR-TEM images is largely incapacitated by the presence of noise and there is a general interest in the community to identify noise robust models for data extraction [48, 49]. Multiple images are typically averaged to wash out noise and enhance features of the image [104], even for existing deep learn applications [105]. Averaging multiple images, however, reduces the time resolution of the interpretable result. The frame electron dose (or simply frame dose) is one of the determining factors of the degree of noise present in the image [45]. On one hand, a low frame dose means a high degree of noise. On the other hand, the electron beam can induce irreversible effects to the material sample and so low frame dose imaging is desired. The aim of this work is to explore the reliability of neural network segmentations at low frame dose, in order to extract the same quality of information at less damaging conditions and with a high temporal resolution.

An experimental data set was acquired by imaging a CeO_2 supported Au nanoparticle with a continuously increasing frame dose from 10^1 to above 10^3 $^{e^-}/_{Å^2}$. The images were produced on a Gatan OneView, which is a scintillator based detector and so the present noise is not purely Poissonian. A full description of how noise can be modelled is provided in Section 1.3. In short, there is a shot noise contribution from the electrons incident on the scintillating material, whose spectral profile is altered by the MTF, and after scintillation there is, amongst other sources, an additional Poisson distributed readout noise. While the readout noise is constant, the shot noise is controlled by the frame dose. As an approximation, the total noise distribution is modelled as a sum of two Poisson distributions, Eq. (1.27), one source from the shot noise and the other from readout noise, although in reality there are other sources present.

The parametric form of the MTF in Eq. (1.25) was fitted to the power spectral density (*i.e.* the norm squared of the Fourier transform) of a vacuum region in every image frame of the experimental data set. Figure 4.6(a) presents the distribution of each parameter as a function of frame dose. The C value is the high spatial frequency limit of MTF and can be interpreted as a noise floor (see Figure 1.4), which shows a clear dependency with the frame dose. This is speculated to be due to the dominating shot noise at high frame dose when $N_D >> N_0$ in Eq. (1.27), which forces the C value to 0 due to the smearing of each electron detection at the scintillator. The readout noise appears after the scintillating material and is therefore not affected by the MTF, which lifts the C value to a finite value. The value of C represents the fractional contributions of the shot noise and readout noise to the total noise. Figure 4.6(b) shows the distribution of the MTF parameters for a simulated replica of the experimental data set. The C dependency, or the readout to shot noise ratio, as a function of frame dose is reproduced. This result highlights that there are two primary noise sources at play. The user of the software pipeline can specify the readout noise contribution *via* the N_0 parameter in Table 3.2. In Publication [II], the N_0 was fitted from its fractional contribution to the C value as a function of electrons per pixel. The fitted value was $N_0 = 0.01$ and applied to the simulated series used in Figure 4.6(b) by a uniform distribution of $0.01 \pm 50\%$. The c_0 and c_3 parameters are varied within a uniform distribution, where the limits are supplied by the user in the input parameters file from Table 3.2. An intuitive understanding for their dependency is missing, however they both vary on a smaller scale than the C parameter. The c_0 parameter primarily controls the spatial frequencies maintained in the image by tuning the half-maximum of the MTF along the spatial frequency axis. When this value approaches 1, the MTF approaches a flat distribution across spatial frequencies below the Nyquist frequency, which would resemble the noise profile of a direct electron detector [52]. The values of c_0 in the fitted range of Figure 4.6(a) represent a strong presence of the MTF.



(a) (b)

Figure 4.6: The fitted parameters of Eq. (1.25) for each frame in (a) the experimental data series and (b) a simulated set of images to replicate the experimental data series. Fits with $R^2 \geq 0.98$ in the experimental images are marked by a white dot. The data points for the simulated images all had fits with $R^2 \geq 0.98$. The orange line represents the mean value. The C dependency on the frame dose is replicated in simulations *via* modelling of the noise contributions through a modulated electron dose dependent shot noise and constant readout noise contribution. The c_0 and c_3 parameters are simulated within a uniform distribution. Reproduced from Publication [II].

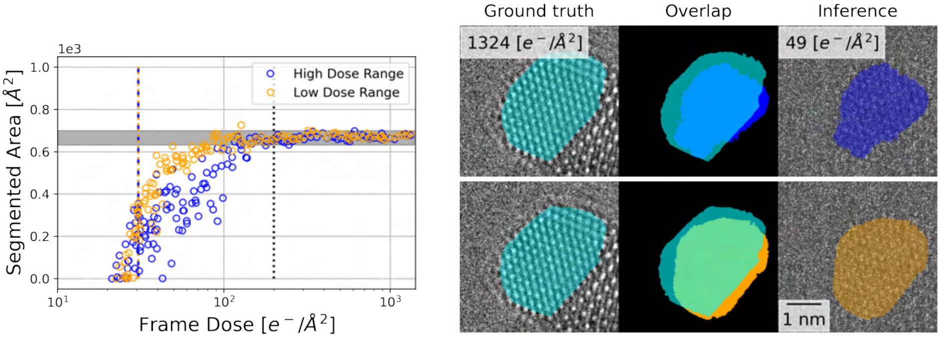
The final frame of the acquired experimental series is considered to be the most "interpretable" image, since it has the highest frame dose *i.e.* the lowest degree of noise, and the segmentation of this frame by a neural network is assigned as the ground truth segmentation. The nanoparticle is in vacuum at 200 °C and so significant morphological changes are not expected throughout the series. The segmentation of frames up to the last can be compared to the last, ground truth segmentation, as a way to gauge the performance. The "area of segmentation", which is defined as the number of pixels segmented with a probability above 90% times the area of each pixel based on the sampling, is equipped as a metric. The idea is that the frame dose is eventually so low that the area of segmentation drops to zero. This experiment allows for the lower frame dose limit of neural network segmentations by the models in the software pipeline to be identified. If the aim of the experiment is to track the area of the nanoparticle as a function of time, this study will provide an idea of the lowest frame dose possible where the area segmented by a neural network is the true area.

Table 4.1: Frame dose ranges for simulated images. L and U denote the lower and upper range, respectively. The value of the frame dose controls N_D of Eq. (1.28) and is defined by the user in the input parameters file from Table 3.2. From Publication [II].

Parameter	L	U	Unit
High frame dose range	10^2	10^6	$e^-/\text{Å}^2$
Low frame dose range	10^1	10^4	$e^-/\text{Å}^2$

A comparison of two neural networks in the software pipeline is made in Publication [II], namely the U-net and MSD-net, from Section 3.3. Two data sets were generated with different ranges of frame dose shown in Table 4.1 and used to train two separate neural networks of each model. Both data sets contain the same 1000 CeO_2 supported Au nanoparticle systems, that are tilted up to 3° off the [110] zone axis (see Listing 3.3), and applies 10 image epochs, resulting in 10,000 training images. As presented in Figure 4.7, the MSD-net trained with the high frame dose range shows a promising ability to segment experimental frames below the lower limit of the high frame dose range training data set (see U-net results in Publication [II]). This means that the MSD-net learns to distinguish signal from noise well enough to generalise outside the training data. The MSD-net trained on the low dose range provides more significant segmentations at lower frame doses, which is expected. The area of segmentation is safely converged for both models at approximately 200 $e^-/\text{Å}^2$, which is the first reported lower limit of a reliable full neural network segmentation. Segmentations below this limit should only be equipped for object detection purposes until approximately 30 $e^-/\text{Å}^2$, where 50% of the nanoparticle is segmented, which would be useful for simply detecting the presence of nanoparticles.

The added complexity from the MTF for scintillator based detectors, as described in Section 1.3, is also studied. The importance of including an MTF for reliable results has been established in previous works [11], but there is a lack of understanding on how to properly model the MTF. This work explores the effect of tuning the MTF by providing different ranges of parameters in Eq. (1.25). Multiple training sets of identical atomic systems, but varying MTFs, are created by writing multiple input parameter



(a) (b)

Figure 4.7: Comparison of the MSD-net trained with simulated images within the high frame dose range to a separate MSD-net trained with simulated images within the low frame dose range (see Table 4.1). (a) Segmented area of each model as a function of frame dose. Black dashed line represents the convergence of the area of segmentation, symbolising where the network segmented the entire nanoparticle. The minimum and maximum area beyond this point form the shaded grey bar as a visual aid for the target area of segmentation. Colour coded dashed lines for each model represent the frame dose at which the model achieves 50% segmentation of the nanoparticle. (b) Colour coded examples of the segmentation at 49 $e^-/\text{Å}^2$ overlapped with the segmentation of the final frame (ground truth). This highlights the ability to achieve low dose segmentation. Reproduced from Publication [II].

files (Table 3.2), one for each MTF model, and generating images based on each set of parameters for the same set of atomic systems. An optimal model of the MTF was identified by studying the area of segmentation curve by neural networks trained with each MTF model. The best MTF model should avoid over-segmentation, where vacuum regions of noise triggers the segmentations, and achieve significant segmentations in experimental images with low frame dose. Overall, the MTF model should assist the neural network in achieving reliable segmentations across the entire frame dose range of the experimental data set (see Publication [II]). The optimal model contained a range of MTF parameters that both are in the fitted range of Figure 4.6(a) and maintained spatial frequencies up to the half-Nyquist frequency (*i.e.* a c_0 up to and slightly above 0.5). It is an interesting result that the MSD-net seemingly prefers a larger range of MTF parameters and not just values within the fitted range of the experimental data set. It seems that preserving larger spatial frequencies provides richer information for the MSD-net to distinguish between signal and noise, but that parameters within the fitted range of the experimental data set are also still required to ensure that the neural network is optimised for that data set.

In conclusion, this work reported a quantitative limit based on the frame electron dose for neural network segmentations by the models implemented in the software pipeline, providing users with the knowledge on how to design suitable experiments that should be analysed by these neural networks. A closer look at modelling the noise contributions and the MTF was presented to optimise simulations for training.

Information regarding the electron dose limit ultimately provides knowledge of the degree to which the neural networks in this software pipeline can assist microscopists in extracting data from HR-TEM images obtained with less damaging electron beam conditions. This knowledge also serves as a benchmark for other groups to compare the robustness of their models as a function of frame dose, hopefully resulting in the advancement of models for extracting reliable information at low frame dose.

4.3 Publication III - Identifying Atomic Positions in MoS2 with Neural Networks using Focal Series from High-Resolution Transmission Electron Microscopy

In Publication [III], the aim was to gauge the ability of all three neural network architectures (U-net, U-net++, and MSD-net) from Section 3.3 in identifying the various atomic columns in monolayer MoS_2 (2H phase) from HR-TEM images. The atomic columns present are 1Mo, 2S, and 1S columns, which were introduced in Section 3.2 as a multi-class segmentation task. Several works have implemented deep learning frameworks to identify primarily defects in 2D monolayer materials, however most of this work is for STEM images [94, 105]. The intensities in STEM are closely related to the mass of the atomic columns, making it comparatively easier to differentiate between the various atomic columns present in the STEM image. TEM provides a higher time resolution for defect studies and lower electron doses, but atomic columns are difficult to interpret. This difficulty is depicted in Figure 4.8(c) with the line scan of the intensities over 2S, 1Mo, and 1S atomic columns in simulated HR-TEM images at various defocus values. At none of the defocus values is it as clear where each of the atomic column classes would be placed along the line scans compared to STEM intensities [91, 94]. In TEM other techniques, such as Fourier filtering, are often applied to identify vacancy defect atomic columns [92].

The three neural networks were trained using the software pipeline from Chapter 3 to segment the various atomic columns from a focal series of 3 images. The data set for training consisted of 1000 monolayer MoS_2 systems with randomly placed vacancy defects. The defect percentages for each atomic system ranged between 0-20% of atomic columns with a S vacancy (1S columns) and 0-5% of atomic columns for Mo vacancies (empty columns). All atomic positions are perturbed based on a Gaussian with a spread of 0.01 Å (see the `distort` setting in Listing 3.5). The atomic systems are tilted up to 1° off the [001] zone axis and are imaged with 10 image epochs, resulting in 10,000 focal series for training. The focal series images are separated by a defocus of (50 ± 1) Å.

In order to have a more accurate measure of the performance a new F1-Score is introduced - the atomic F1-Score. The ordinary pixel-wise F1-Score in Eq. (3.5) measures both the ability for the neural networks to identify and correctly classify every atomic column, as well as place the atomic columns with pixel-wise precision. Given samplings where tens of picometres represents a pixel, this measure is sensitive to misplaced atoms at a resolution that is negligible. In order to place more emphasis on the identification and classification of the atomic columns, the ground truth peaks are located, and peaks within the predictions are matched within a 10 pixel radius. True positives (TP) are defined as atoms identified in the predictions that are placed within 10 pixels from a ground truth atom. With a sampling of 0.1 $^{\text{Å}}/_{pixel}$, this corresponds to predicting the atom within 1 Å from the correct position. The false positives (FP) and false negatives (FN) are then defined as

$$\text{FP} = \text{No. of predicted atoms - TP}, \tag{4.1}$$

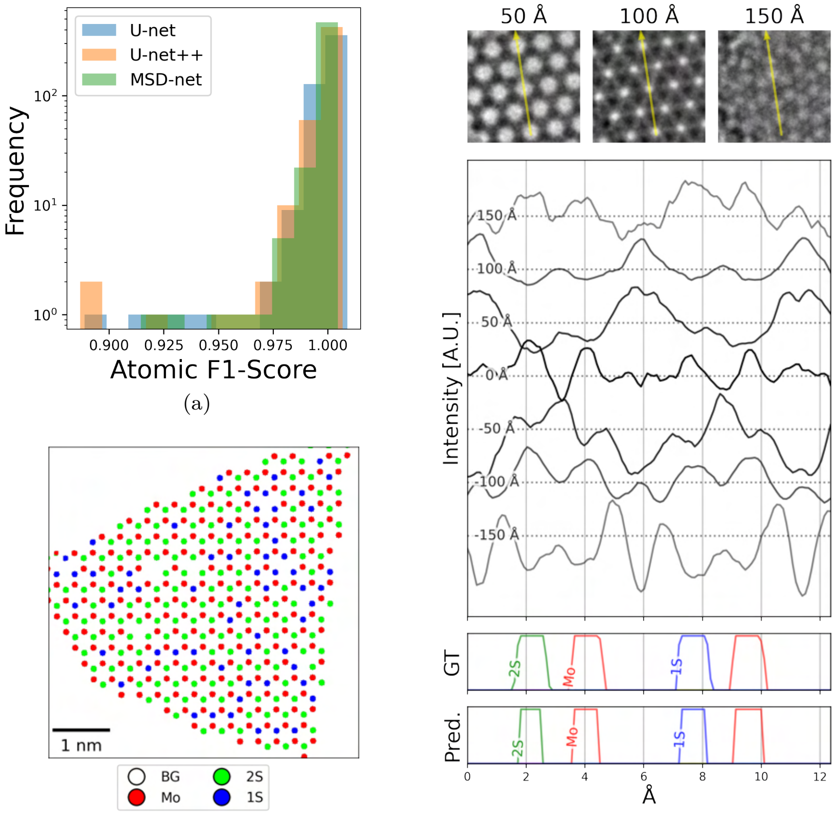
$$\text{FN} = \text{No. of ground truth atoms - TP} . \tag{4.2}$$

Figure 4.8(a) shows a distribution of the atomic F1-Score over a data set of 500 focal series for 500 MoS_2 monolayer systems, covering a large range of microscope parameters (shown in Publication [III]). The distribution highlights a perfect atomic column identification and classification for over half of the data set. The best example of the MSD-net is shown in Figure 4.8(b) that visualises the number of atomic columns that are being classified, which would be an extraordinary task for a human due to the difficulties in interpreting the intensities and the time it would take to perform this. The bottom rows of Figure 4.8(c) show the neural network prediction against the ground truth atomic columns of the focal series shown in the top row. This shows that the neural network achieves precise classification of the atomic columns regardless of the complicated intensities present.

The next step was to verify whether the network required a focal series at all. Figure 4.9 presents the performance of the same neural networks trained with a data set of the same atomic systems with a single image, a focal series of two images, and a focal series of three images, with a constant defocus difference of (50 ± 1) Å. There is a considerable performance boost from a single image to a focal series of 2 images, where only the U-net++ seems to be struggling. Moving to the 3 image focal series brings all networks to their peak performance. In Publication [III], only the C_s and defocus were varied, which are the dominant terms for well-aligned micrscopes, however involving higher order CTF parameters could alter the number of required images in the focal series.

An emphasis was placed in Publication [III] on the amount of data required and how large the neural networks have to be, in terms of the number of trainable parameters. The performance of all models were shown to converge at 100 focal series (100 atomic systems and 10 image epochs) in the training data set (shown in Publication [III]). The number of trainable parameters is one way to fairly compare neural networks. During training the neural network is attempting to tune these parameters in order to minimise a high dimensional loss function, as described in Section 2.2. It can be argued that the optimal neural network is one that minimises this loss function with the fewest number of parameters. Typically, increasing the number of parameters also increases training and inference times, which justifies another reason to identify the fewest parameters possibles to solve the task. All neural networks had a maximum average atomic F1-Score at around 10^6 trainable parameters, but did not suffer significantly below this (shown in Publication [III]).

As a final test a separate validation/test data set was generated containing defocus

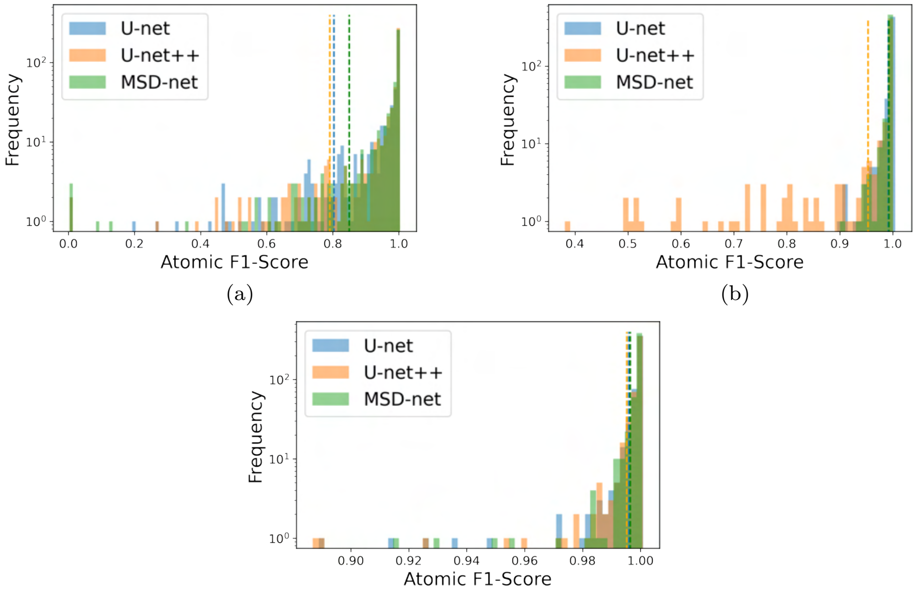


(a)

(b)

(c)

Figure 4.8: (a) Distribution of atomic F1-Scores for each network. All three present very strong detection of Mo, 2S, and 1S atomic columns. (b) Best MSD-net segmentation from the validation data set, visualising the background (BG) and all the identified atomic columns of varying chemical composition. (c) Simulated focal series where a line scan is placed over two Mo columns, a single 2S column, and a single 1S column. The intensities of the image line scans in the middle present the complexity of deducing atomic columns in HR-TEM images, however feeding a three image focal series (an example is shown at the top) to a neural network provides perfect atomic column identification, as shown at the bottom comparing the ground truth (GT) and prediction (Pred.) line scans. Reproduced from Publication [III].



(a) (b)

(c)

Figure 4.9: The ability, based on the atomic F1-Score, of all three networks to classify 1Mo, 2S, or 1S columns given (a) a single image, (b) a focal series of 2 images, or (c) a focal series of 3 images. Dashed lines represent the 10^{th} percentile. A focal series of 3 images marks the lowest number of images required for peak performance from all neural networks. Reproduced from Publication [III].

values outside of the range defined in the training data set. This gauges whether the neural networks utilise the focal series well enough to generalise outside of the range of the training data. Figure 4.10 shows that this is not the case, as the performance is considerably worse in the defocus range not included in the training data set. The neural networks were able to identify 1Mo and 2S columns without much difficulty, but extreme weaknesses were present in the identification of 1S columns (shown in Publication [III]). There may be a more complicated contrast relationship for 1S atomic columns compared to the 1Mo and 2S atomic column.

In conclusion, all neural network architectures from Section 3.3 exhibit a pronounce ability to differentiate between 1Mo, 2S, and 1S columns in HR-TEM focal series, despite the complexities of the intensity variations. This serves as a powerful tool to probe atomic column positions in materials with multiple chemical species to study dynamics and defect formations under different environmental conditions. Tracking defect formation and lattice deformations at the atomic scale can assist studies aiming to understand beam induced effects in materials with a higher time resolution and lower electron dose.

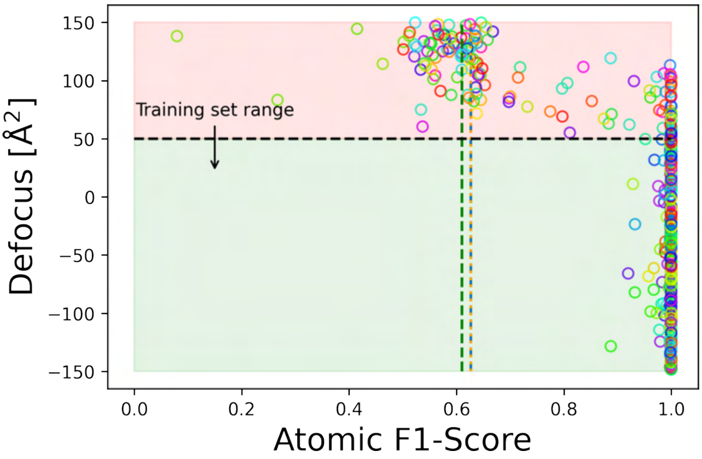


Figure 4.10: Atomic F1-Score versus the minimum defocus value in the focal series. This data set contained defocus values outside the training range, and the figure shows that the neural networks do not generalise well outside the range of the training data. Reproduced from Publication [III].

CHAPTER 5

Software: An Extension to Model Realistic Atomic Vibrations

The many atomic configurations generated in the software pipeline (Chapter 3) are all static snapshots of atomic systems, however reality is dynamic; Real atoms are moving and each image is an integration of several snapshots of the dynamic atomic system over the exposure time. Atomic vibrations are occurring on a time scale that are considerably faster than the exposure times of HR-TEM microscopy and if the vibrations are significant this leads to the smearing or blurring of atomic columns [29]. This section will explore the importance of including dynamical behaviour into the images to further bridge the gap between simulation and experiment.

It was already shown in Section 3.1 that the atoms in the system can be perturbed by an amount sampled from a Gaussian distribution to displace atoms out of their pristine positions. This introduces some disorder into the system. The effects of dynamic blurring can be implemented in two ways:

- The smearing of the atomic potential by a Gaussian,
- Averaging over multiple exit waves/images, known as the frozen phonon configuration method.

The former operates with an effective potential that is the convolution between the static potential of the atoms, V($\mathbf{r}$) and a thermal blurring, B($\mathbf{r}$),

$$V_{eff}(\mathbf{r}) = (V * B)_{\mathbf{r}}, \tag{5.1}$$

where $\mathbf{r}$ is a real space 2-Dimensional vector in the lattice plane, perpendicular to the optical axis [29].

The thermal blurring can be modelled as a Gaussian with a defined width. This width is typically referred to as the Debye-Waller factor, which is formally defined as a temperature related probability for elastic scattering [106], but has been applied in HR-TEM as a blurring parameter due to thermal vibrations in the sample [29]. The Debye-Waller factor in this context is defined as the mean squared displacement of an atom j, $\langle u_j^2 \rangle$, and can be calculated as

$$\langle u_j^2 \rangle = \frac{3\hbar^2}{2m_j k_B \Theta_D} \cdot \left(\frac{1}{4} + \frac{T}{\Theta_D} \right) \cdot \frac{2}{3}, \tag{5.2}$$

where m is the atomic mass, T is the temperature, k_B is the Boltzmann constant, and Θ_D is the Debye temperature [107]. This number is an input parameter in Listing 3.5 allowing the user to apply a potential smearing based on a given Debye-Waller factor.

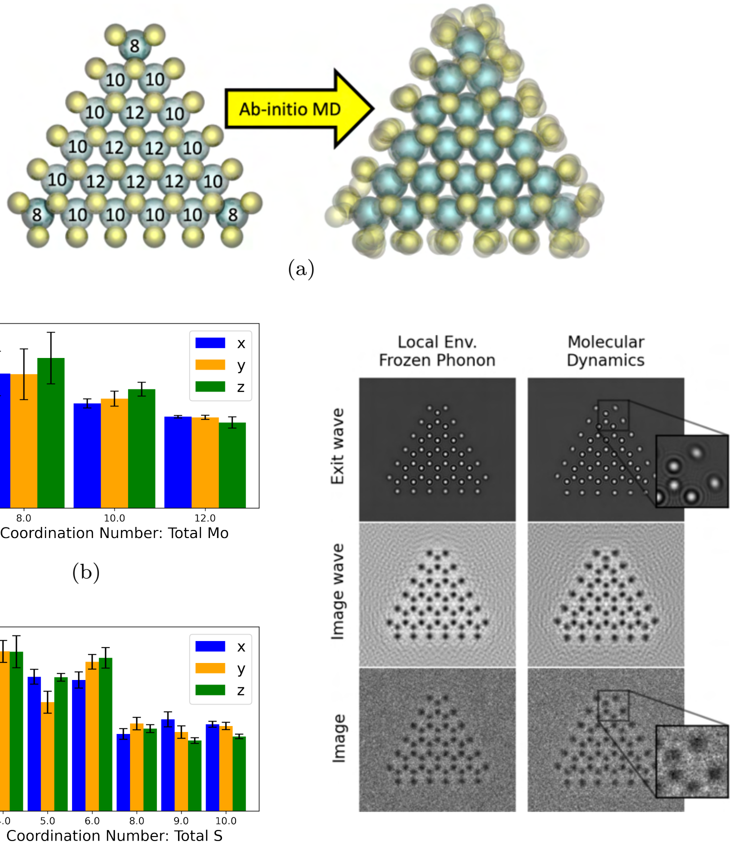
The Debye-Waller smearing applies a homogenous smearing across the atomic potential. Chen *et al.* [29] modelled blurring as a Gaussian with three components: electron scattering, atomic vibrations, and lens abberations. They proved that, for Co-Mo-S atomic nanocrystals, the vibrational radius of individual atomic columns was heterogeneous and that the heterogeneity is due to the atomic vibrations. The vibrational radius increases from the centre of a nanocrystal and towards the edge. In fact, the amplitudes of the atomic vibrations alone exceed typical Debye-Waller factors and unfortunately comprimises the ability to determine the stoichiometry of many atomic columns due to overlapping column potentials caused by the vibrations. Refer back to Figure 1.5(a) and imagine that the atoms are displaced significantly in the horisontal directions. Their conclusion points to speculation that the vibrations are largely due to the electron beam supplying energy to the vibrations of the atomic system. Their work highlights a proper treatment of the heterogeneity of the thermal vibrations in atomic systems for proper exit wave reconstructions when the aim is to determine the chemical composition of each atomic column.

The latter method to applying vibrations is already handled by *abTEM*, where a trajectory file containing multiple configurations at each time step of a molecular dynamics (MD) simulation, for example, can be provided as input, instead of a single atomic system. The molecular vibrations and exposure times for each image is larger than the time it takes for a single high energy electron to traverse through the specimen. Each electron effectively sees a frozen snapshot of the atomic configuration, which differs to the snapshot seen by other electrons. The average image over all the configurations can then be computed to include the vibrational blurring. This can also be achieved by generating multiple configurations from simply perturbing the atoms of some system multiple times from a Gaussian distribution. This would be a more crude model but would provide atomic column smearing in the image nonetheless. Perturbing every atom based on the same Gaussian distribution will still only apply a homogeneous smearing across the atomic system. See Figure 5.1(a) for an illustration of the heterogeneous blurring that can occur from an MD simulation.

To take a step towards more realistic vibrations an alternative frozen phonon method is proposed, named the “Local Environment Frozen Phonon” method. This method is an attempt to address the findings of Chen *et al.* [29] and consists of extracting a mean standard deviation of atomic positions dependent on their coordination number and chemical element from a series of MD simulations. These values are then used to perturb atoms correctly by a simple translation of the positions based on a Gaussian tailored for each coordination number. This is the frozen phonon approximation, but with vibrational amplitudes depending on the local structure obtained from MD simulations. This includes heterogeneous thermal vibrational effects into the simulated images at a low computational cost. This is especially useful when MD simulations require *ab-initio* methods, such as density functional theory (DFT), which is the case for MoS_2. MoS_2 will therefore be the focus of this chapter.

A MD simulation was carried out for the basic MoS_2 system shown in Figure 5.1(a).

A DFT calculator was set up using *GPAW*, a projector-augmented wave based DFT code [108], in LCAO mode, with a double zeta polarised basis set, and the PBE functional. The structure was first relaxed using the QuasiNewton method. The dynamics were computed *via* the NVTBerendsen algorithm with a time step of 0.5 fs and a time constant of 250 fs. The NVTBerendsen algorithm provides reliable constant temperatures for the system through a coupling with a thermal bath that has a well defined constant temperature [109]. The structures are restricted to zero linear and angular momentum. The structure is allowed to thermalise to 300 K from an initial excitation of 500 K.



(a)

(b)

(c)

(d)

Figure 5.1: (a) An illustration of a MoS_2 nanoflake with the Mo total coordination numbers and the heterogeneous effects of atomic column blurring due to thermal vibration produced by an MD simulation. The vibrational amplitudes from an MD simulation at 300 K sorted by the total coordination for (b) Mo and (c) S atoms. (d) The resulting images from averaging images from the MD simulation and the local environment frozen phonon method.

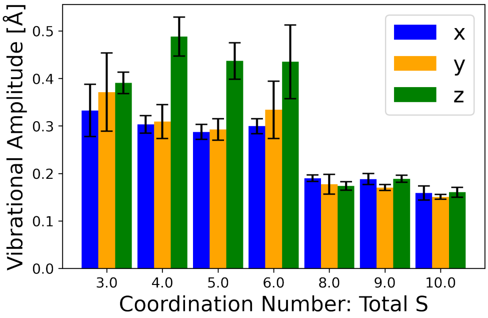
The atomic configuration is stored each time step. The atoms are indexed, which makes it simple to extract the standard deviation of the positions, *i.e* the vibrational amplitude, of each atom in the system along with a standard error. The atoms are then sorted by their total coordination number. In this case there are two chemical elements, Mo and S. See Figure 5.1(b) and Figure 5.1(c) for the distribution of vibrational amplitudes for Mo and S, respectively, based on their total coordination number. It is clear, especially for S atoms, that the lower the coordination number, the higher the vibrational amplitudes. In all cases the amplitudes at this temperature are very small. The atoms of a given system can be perturbed a number of times by a unique Gaussian defined for each chemical element and coordination number. Doing so, displaces the lower coordinated atoms more than the higher coordinated atoms. The results of the averaging of the exit wave, image wave (exit wave + CTF), and the image (image wave + noise) are shown in Figure 5.1(d) (Refer to Figure 1.1 for the definitions of each), where the vibrations are seemingly negligible in all cases.

To enhance the study, the MD simulation was repeated for a temperature of 1200 K, in an attempt to increase the effects for educational purposes, and as a possible approximation to the local heating by the electron beam [29]. The vibrational amplitudes for S atoms in Figure 5.2(a) are evidently much larger. The smearing affect due to the averaging is also much more noticeable in Figure 5.2(b). Here it is shown that the local environment frozen phonon method is capturing the same spatial distribution of vibrational smearing as the MD simulation.

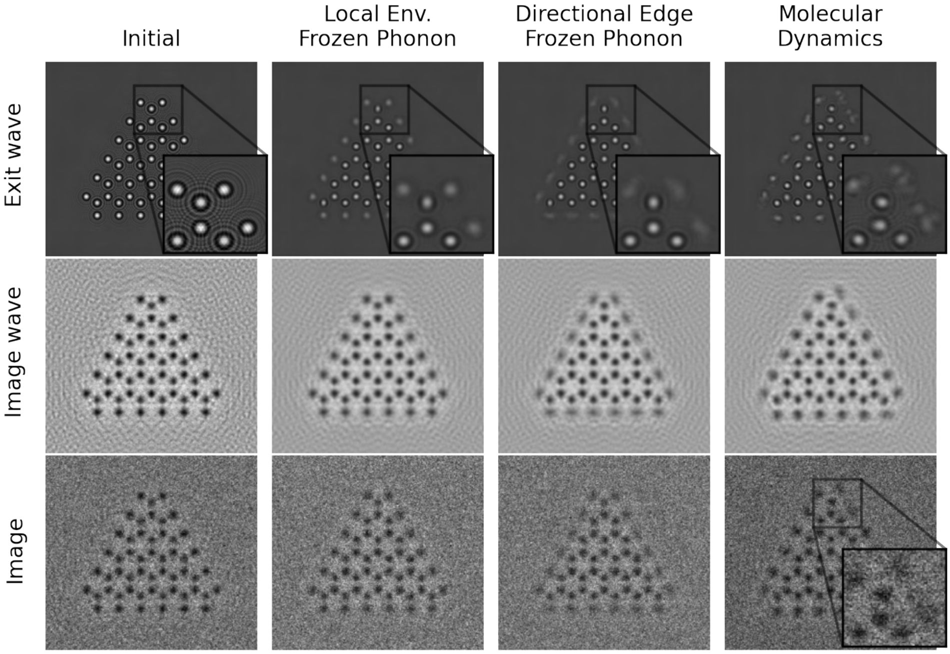
The S atoms along the edge vibrate more parallel to the edge of the MoS_2 nanoflake in the MD simulation compared to the local environment frozen phonon method. One could argue that the local environment of the S atoms located along the edges differ not only by coordination but also in their degrees of freedom, which is not captured by the local environment frozen phonon method. A separate treatment was performed on these edge atoms to determine in which direction these atoms vibrate most with respect to the edge of the system. For each S atom along the edge a y-axis is defined perpendicular to, and an x-axis parallel to, the edge of the MoS_2 nanoflake, using the nearest Mo atom. Each S atom's axis is then rotated by an angle θ and translated by a distance d to align all S edge atoms on a global axis. This is depicted in Figure 5.3(a). Vibrations on the global axis revealed an almost 50% larger vibrational amplitude parallel to the edge of the MoS_2 flake than perpendicular to (see Figure 5.3(b)). This is taken into account in the additional "Directional Edge Frozen Phonon" method, which in Figure 5.2(b), shows that the directionality of the vibrations along the edge are captured.

The vibrational smearing in Figure 5.2(b) show that the effects are most prominent in the exit wave compared to the image. The applied noise seems to drown out the blurring from the vibrations. Modelling the heterogeneous vibrations in the simulated training data may not affect on the atomic column segmentation task of MoS_2 presented in Section 4.3. In contrast, including heterogeneous vibration could have a significant affect on the exit wave reconstruction by neural networks from Section 4.1 and better prepare neural networks for reconstructing the exit wave of experimental images.

In summary the functions presented in Table 5.1 allow for the extraction of vibrational amplitudes from a few MD simulations and then the application of vibrational amplitudes to hundreds/thousands of atomic structures at relatively low computational

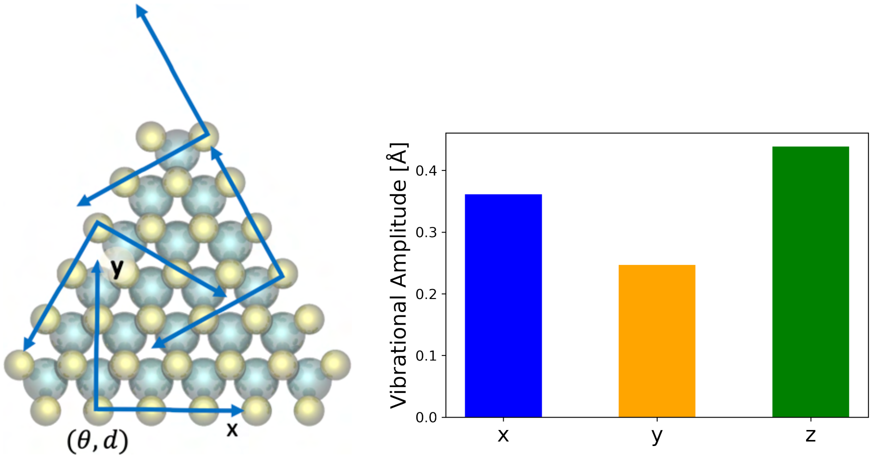


(a)



(b)

Figure 5.2: (a) The vibrational amplitudes from an MD simulation at 1200 K sorted by the total coordination for S atoms. (b) Averaged intensity (norm squared) of the exit wave, image wave, and image formed from the local environment frozen phonon, directional edge frozen phonon, and the MD systems. The local environment method captures the same heterogeneity of vibrational smearing, and the directional edge method captures the directional vibration of the edge atoms. The vibrational effects are minimal in the final image when noise is applied, but prominent in the exit wave.



(a) (b)

Figure 5.3: An illustration of the “Directional Edge Frozen Phonon” method. (a) S atoms located along the edge are treated separately by defining an axis for each atom where the y-axis points to the nearest Mo atom. By a rotation and translation all the S atoms along the edge are fixed to a global axis. (b) the vibrational amplitude distribution on the global axis described in (a). This reveals that the S atoms have a larger vibrational amplitude along the edge of the MoS_2 nanoflake.

cost. The vibrations are coordination number and chemical element dependent and will therefore mimic the vibrational heterogeneity highlighted by Chen et al [29]. The MD vibrations are only thermal vibrations and do not include well described beam induced effects, however the vibrations can easily be tuned by any factor when applied. Doing so may provide more realistic training data and improve performance on experimental data, especially when performing exit wave reconstructions.

Table 5.1: Functions for the Local Environment and Directional Edge Frozen Phonon methods.

Extraction	
Functions	Descriptions
`get_vibrational_amplitudes()`	Takes a trajectory file with multiple configurations and will return a `Frozendict` of the mean of the standard deviation of positions in each Cartesian direction for each unique coordination number of an element to a neighbouring element. Return - *va*: `Frozendict` of the vibrational amplitudes sorted by coordination number
`get_edge_vibrations()`	Takes a trajectory file with multiple configurations and handles edge effects. Edge atoms are identified by lowest coordination number, eg. an S with 1 Mo neighbour. All edge atoms are mapped onto a global (x,y)-axis where a mean and standard deviation of the positions are extracted. Return - *sigma_xyz*: Mean over all trajectory steps of the standard deviation of atomic positions of all edge atoms *mu_xyz*: Mean over all trajectory steps of the average atomic positions of all edge atoms
Application	
Functions	Descriptions
`perturb_atoms()`	Takes the `Frozendict` from `get_vibrational_amplitudes` and an atomic system and will return atoms perturbed by a Gaussian distribution for each coordination number and chemical element based on the given vibrational amplitudes. Return - *patoms*: `Atoms` The perturbed atoms configuration
`perturb_edge_atoms()`	Takes the mean and standard deviation from `get_edge_vibrations` and perturbs the edge atoms using the standard deviation and mean in cartesian directions. The edge atoms will be mapped into the global coordinate system, before perturbing, and then mapped back. Return - *patoms*: `Atoms` The perturbed atoms configuration

CHAPTER 6

Software: Graphical User Interface with Dragonfly

Dragonfly is a general image analysis software provided by *Object Research Systems*[1]. The developers are experts in computer vision and various techniques to analyse images, including deep learning. *Dragonfly* has an easy to use *Python* development module to develop user-specific tools. This has been used to implement the instance property tracking described in Figure 3.11 as a menu item that can be easily selected and executed by the user on any stack of binary segmentations. Refer to *Dragonfly*'s tutorial page on how to load images, train neural networks or load trained neural networks (for example the neural network saved at the end of Listing 3.8), and apply them to obtain the segmentation mask[2]. Menu items have been implemented to extract information from multiple instances (see Figure 3.11) that allows easy analysis of many frames of data.

6.1 Implementation of Tools as Menu Items

This section will present the menu items that have been implemented with a description of each supported by a screenshot directly from the *Dragonfly* interface. As an example dataset, hexagonal Boron Nitride (hBN) supported Au nanoparticles have been imaged along the [111] zone axis. The images are large 4K resolution images of multiple nanoparticles. *Dragonfly* refers to an image as a `Channel`-object, which can contain all the frames in the dataset. The segmentation of an image is called a `ROI`-object when it is a binary segmentation and a `multiROI`-object when the number of classes are beyond binary. These names will be adopted for the remainder of this chapter, as menu items are created for specific objects in the program.

Menu Item 1: Label by 2+1D Connectivity

This menu item is available for a `ROI`-object in *Dragonfly*. An example of a binary segmentation of the hBN supported Au nanoparticles is presented in Figure 6.1. The `ROI`-object contains the binary segmentation of every frame provided. Selecting the "Label by 2+1D Connectivity" menu item will prompt the user to specify a minimum

---

[1] Dragonfly 2020.2 [Computer software]. Object Research Systems (ORS) Inc, Montreal, Canada, 2020; software available at `http://www.theobjects.com/dragonfly`.

[2] Dragonfly tutorials: `https://www.theobjects.com/dragonfly/tutorials.html`.

separation in unit nm. This will serve two purposes. Firstly, when the landscape for the watershed algorithm is more complex than the representation in Figure 3.10 there may be an overwhelming number of peaks or valleys. The minimum separation will force the valleys to be separated by at least that given value, which is especially useful for when there are overlapping elements in the image that should be separated. Secondly, segmentations are not always perfect and small incorrectly segmented areas may appear due to noise, amongst other artefacts, in the image. The square of the minimum separation will serve as a minimum area for each ROI instance and remove every ROI instance that is smaller. This is done for every frame in the `ROI`-object, where the ROI instance tracking presented in Figure 3.11 is executed to connect the instances across all frames. The output is a `multiROI`-object in Dragonfly, which contains the now multi-valued segmentation of every frame, where the values for each instance are matched across frames. This has been named "2+1D Connectivity", since each image is 2-Dimensional where instances are separated and then connected along the additional time axis *i.e.* across all the frames.

Figure 6.1: A screenshot of the "Label by 2+1D Connectivity"-menu item, which is available for binary segmentations and will separate regions into individual instances and connect the instances across all frames provided. The only input required is a minimum distance between instances in unit nm.

The result of selecting the "Label by 2+1D Connectivity"-menu item is presented in Figure 6.2, which proves that each nanoparticle is separated into individual instances and connected across frames. The lower right even presents two overlapping nanoparticles that were divided into two separate ROI instances that are consistently tracked across frames. It should also be noted that this can be done for any number of frames and any number of instances or, in this case nanoparticles, present in each frame, which

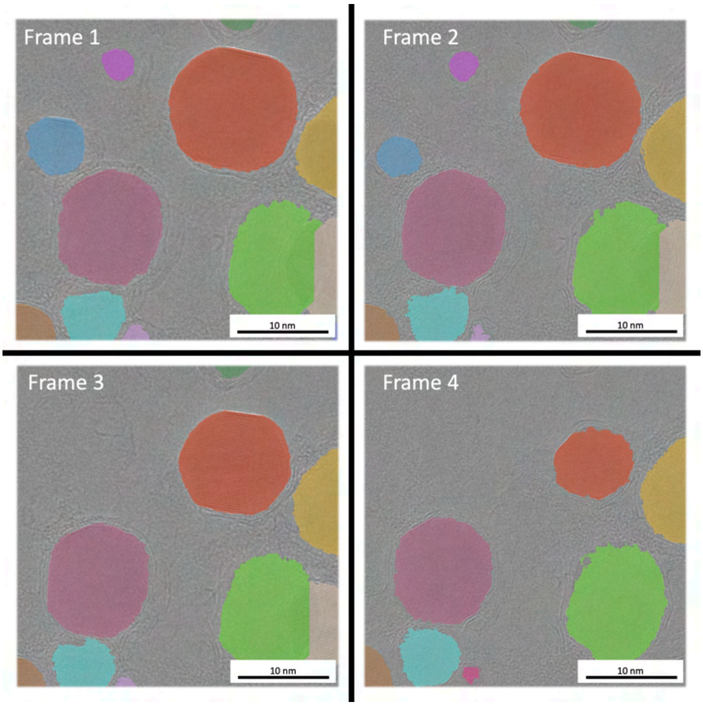


Figure 6.2: Result of the "Label by 2+1D Connectivity"-menu item. Every `ROI`-instance, in this case nanoparticle, is defined frame by frame and connected across frames. Even overlapping nanoparticles were treated. This is a real-life example of the illustration in Figure 3.11.

highlights the applicability to large scale data analysis.

Menu Item 2 & 3: Get FFT along time axis & Get ROI properties

Now that all the ROI instances are tracked, data can be extracted for each instance across all the frames. The user can select an instance or nanoparticle from the `multiROI`-object and generate a separate `ROI`-object for that instance. Both menu items are shown in Figure 6.3 for a selected nanoparticle (or `ROI`-object).

The first is a menu item to extract the Fourier transform of the given `ROI`-object across all frames. This utilises the `ROI`-object's segmentation to crop the instance out of the image, apply a Gaussian apodisation to smoothen the boundary edges, and compute the Fourier transform. The Fourier transform is a useful tool to identify orientation or crystallographic properties of the region, for example a nanoparticle. This menu item allows for a Fourier space visualisation of the entire nanoparticle isolated from any surroundings, even if the nanoparticle is in motion, as long as the displacement is not too large between consecutive frames. The extracted Fourier transforms are stored as a separate `Channel`-object that can be viewed side-by-side and analysed separately.

The second data extraction menu item currently returns the area of the instance across frames and provides a plot. This can be selected for multiple `ROI`-objects or multiple nanoparticles to form a plot of the area of each nanoparticle against time to study morphological changes due to environmental conditions, such as temperature changes, gas concentration changes, or varying electron beam doses.

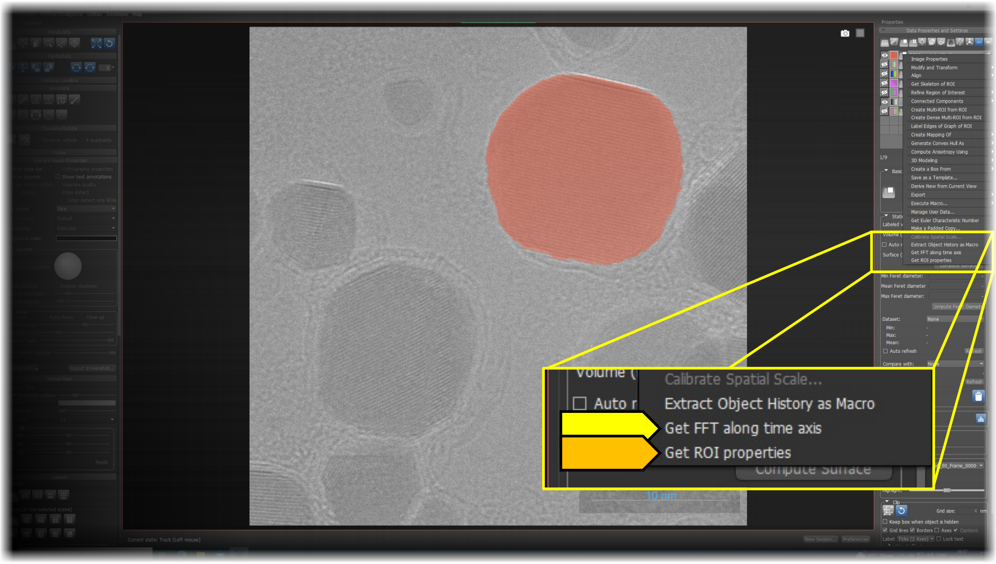


Figure 6.3: A screenshot presenting the two data extraction menu items implemented in Dragonfly. “Get FFT along time axis” extracts the Fourier transform of the isolated region for every frame and is stored as a separate `Channel`-object. “Get ROI properties” currently extracts the area of multiple selected `ROI`-instances across all frames and plots it.

This concludes the presentation of the tools available as menu items in *Dragonfly*. To emphasise the value of the menu items, the following section will present real life examples, applying the tools to extract information.

6.2 Real Life Examples of the Tools

The main goal of the tools is to open the door to large-scale data analysis in HR-TEM. This means the analysis of many regions of interest in many frames of data. The tools should alleviate tedious manual labour. To showcase the Fourier transform tracking, CeO_2 supported Au nanoparticles were imaged in vacuum at room temperature with a doserate of 35 x 10^3 $^{e^-}/_{Å^2 s}$ and exposure time of 0.4 s per frame. A neural network trained using the pipeline from Chapter 3 was loaded into *Dragonfly* to segment all the nanoparticles in the image. The “Label by 2+1D Connectivity”-menu item was selected to separate all the nanoparticles and a single nanoparticle was chosen because it shifted its position throughout the frames. The Fourier transform of the nanoparticle was extracted for every frame. The nanoparticle in mention and its Fourier transform is presented in Figure 6.4, where the displacement between frames is emphasised by the dashed lines. Despite this displacement the nanoparticle is tracked across frames and a local Fourier transform can be extracted.

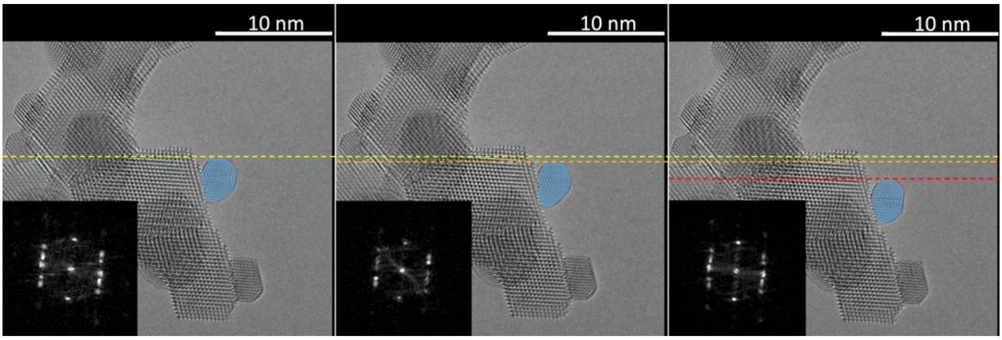
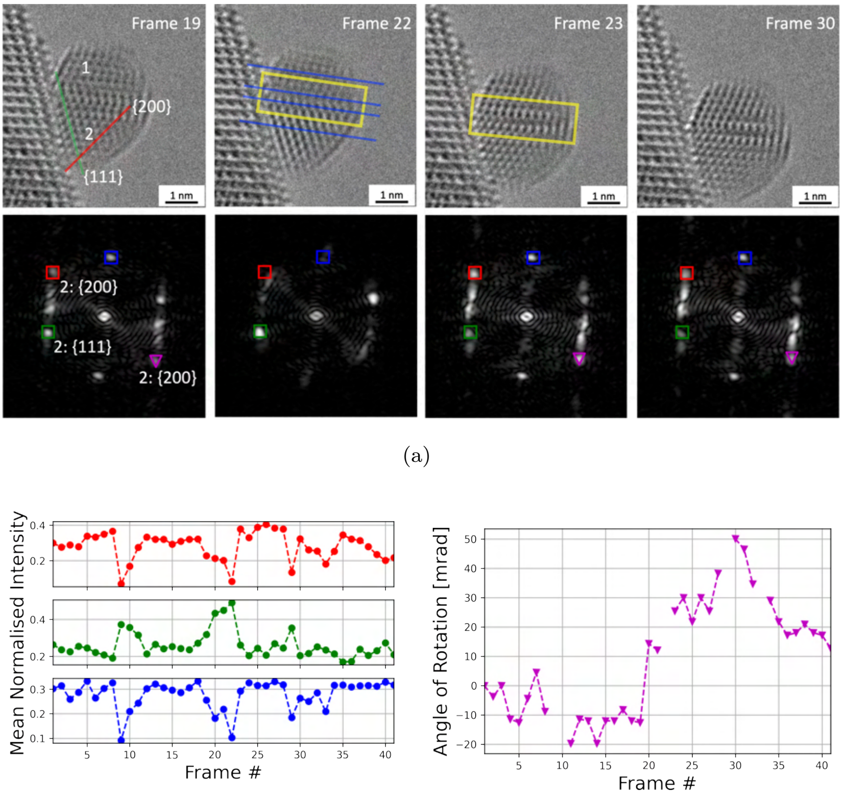


Figure 6.4: An example of extracting the local Fourier transform of a nanoparticle, despite its motion in real space. This is done using the "get FFT along time axis"-menu item presents in Figure 6.3. The dashed lines emphasise the motion of the nanoparticle across frames.

The Fourier transforms can be extracted from *Dragonfly* as image files of various formats and analysed to extract crystallographic information. A closer look at the selected nanoparticle and its corresponding local Fourier transform is presented in Figure 6.6. It is evident both in real space and Fourier space that there is a twin boundary in the nanoparticle. Two regions are identified that are labelled region 1 and 2.

The Fourier transform can help quantify out-of-plane rotations or morphological changes, that are less evident in the real space image. Tracking the intensity of the peaks can be related to out-of-plane rotations, phase changes, and other morphological changes [82], as well as aberration affects on local regions of the image that reduce the contrast. Each Fourier transform is normalised so that the intensities can be thought of as a fraction of the maximum intensity present in the Fourier transform. The mean intensity within a 30 x 30 pixel square around each peak is used. In this case the {200} and {111} peaks of region 2 in the nanoparticle are tracked and show an inverse relationship in Figure 6.5(b). The {200} intensity (red) dips as the {111} intensity (green) peaks. This relationship can be due to an aberration effect, morphological change, or out of plane rotation, reducing the visible periodicity of the {200} planes. The peak corresponding to the planes associated with the twin boundary interface is also tracked (blue). The intensity of the blue peak can quantify the relative contrast shifts between region 1 and 2, which seem to follow the same trend as the {200} intensity of region 2. The contrast shift is highlighted by the yellow marked region of frame 22 and frame 23, where there is a stark contrast difference between the two sides of the twin boundary of frame 22. The effects here are speculated to be beam related due to the conditions of the experiment.

Tracking the angle of a peak can be related to in-plane rotations. The angle of the {200} peak of region 2 shows minor in-plane rotations. These rotations are most significant between frames where the nanoparticle exhibits a large displacement, such as between frame 22 and 23. These frames are also where there are the most significant changes in the intensity of the Fourier peaks. This suggests that the nanoparticle exhibits some morphological changes or motion blurring during displacement and relaxes again to a similar state.



(a)

(b)

(c)

Figure 6.5: Image crops of the nanoparticle in motion for selected frames are shown. (a) Presents real space images (top) and corresponding local Fourier transforms extracted using the segmentation (bottom). Colour coded crystal planes are marked in both real space and Fourier space. The nanoparticle is separated into region 1 and 2 on either side of the twin boundary. (b) The intensities of {200} (red) and {111} (green) peaks for region 2, reveal a loss in the visible periodicity of {200} planes. The reduced {200} visibility correspond with a contrast difference between region 1 and 2, quantifiable by planes in the twin boundary direction (blue). (c) The angle of rotation of the region 2 {200} peak that shows very minor in plane rotations of the nanoparticle. The most significant rotations occurred near a displacement of the nanoparticle, such as between frame 22 and 23.

The above analysis serves as an example of tracking dynamics in the orientation and morphology of a nanoparticle. The "get FFT along time axis"-menu item assists users in extracting high resolution local Fourier transforms by providing it with the click of a button. This tool is robust to motion of the nanoparticle allowing for studies to correlate motion of nanoparticles and structural changes at ease.

Another experiment was conducted with the hBN supported Au nanoparticles introduced in Section 6.1, which consisted of imaging the many nanoparticles at a constant temperature of 1100 °C, a beam energy of 300 keV. In this experiment many of the nanoparticles vaporised and the experiment was to determine whether the vaporisation was due to the temperature supplied by the stage or local heating by the electron beam. To achieve this the electron beam was left on for some time and switched off for a period. This was repeated twice. The electron dose rate was 4849 ${}^{e^-}\!/\text{Å}^2 s$, 4921 ${}^{e^-}\!/\text{Å}^2 s$, and 4917 ${}^{e^-}\!/\text{Å}^2 s$ during the first, second, and third active beam intervals, respectively. A neural network was trained using *Dragonfly*'s own manual labelling and training tools to achieve a segmentation of every frame in the dataset[3]. Selecting the "Label by 2+1D Connectivity"-menu item separates and tracks every nanoparticle as shown in Figure 6.2. Multiple `ROI`-objects, each containing a nanoparticle, are highlighted and the "get ROI properties"-menu item is selected resulting in the area versus time of each highlighted nanoparticle shown in Figure 6.6. From the result it can be deduced that in this experiment the temperature supplied by the stage is the main factor causing the vaporisation of the nanoparticle. This is seen by the constant slope in the area of a nanoparticle seen across frames with no electron beam incident on the sample. Again it is emphasised that the plot in Figure 6.6 was obtainable with a single click. The large-scale tracking of properties for statistically significant analyses is achievable with these menu items and each menu item is easily expandable for other properties than what is shown here.

In this chapter, the value of the implemented tools has been highlighted with a couple of model experiments. With a single click the user can extract and quantify dynamic properties of and region of interest in the images as a function of time, facilitating large-scale data analysis for HR-TEM.

---

[3] The workflow from Chapter 3 does not include hBN supported Au nanoparticle in the [111] zone axis. Only 4 manually labelled frames were required to train a sufficient neural network in *Dragonfly*.

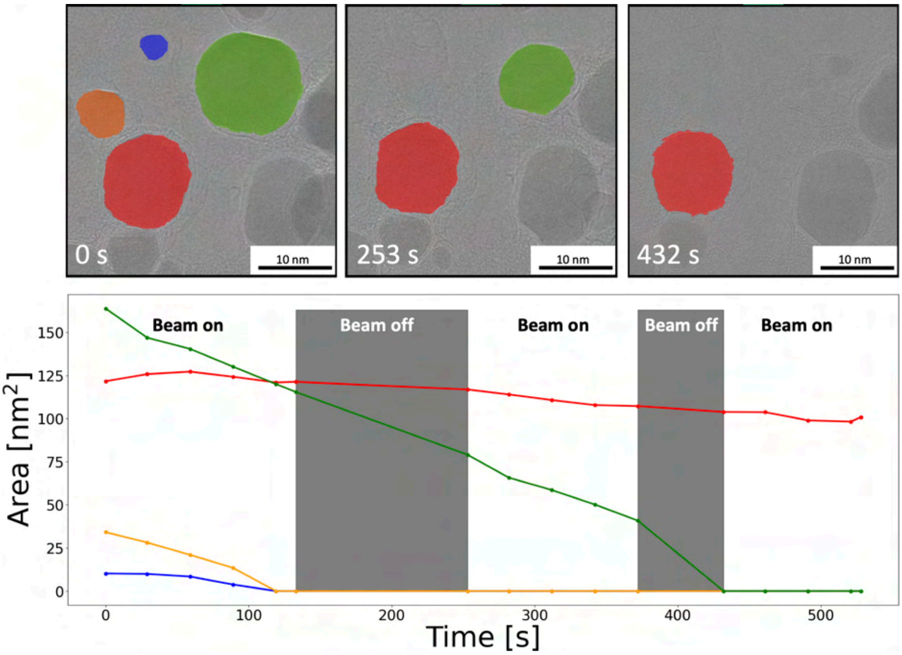


Figure 6.6: An example presenting the "get ROI properties"-menu item which allows for the area of multiple nanoparticles to be tracked. In this experiment the electron beam was switched on and off for varying periods to deduce whether the vaporisation of the Au nanoparticle were due to temperature supplied by the stage or local heating by the electron beam. Due to the constant slope in the area of each nanoparticle across periods with the electron beam switched off, it can be concluded that the dominating factor here is the temperature from the stage.

CHAPTER 7

Conclusion

Modern electron microscopes are capable of producing rapid image sequences capturing the dynamic atomic scale nature of materials. These image sequences contain gigabytes or terabytes of data that are typically analysed manually by an electron microscopist by, for example, determining the atomic columns present in the image frame by frame. This is a cumbersome task and methods to alleviate present difficulties has been introduced in the context of HR-TEM.

This work has leveraged the power of deep learning neural networks to provide tools for microscopists to analyse their data. Neural networks are data driven and require large amounts of data to sufficiently learn a given task and that data must be sufficiently diverse to avoid any bias and force the neural network to generalise to other data sets. A software pipeline has been implemented to allow users to generate thousands of varied atomic systems of CeO_2 supported metallic nanoparticles and 2H phase MoS_2 monolayer nanoflakes. From the thousands of atomic systems the user can then generate thousands of HR-TEM images with assorted microscope conditions. The assortment of the HR-TEM images is very much controlled by the user *via* the input parameters file from Table 3.2, where the user should specify, amongst other things, the range of the CTF and other microscope-related parameters. The user can toggle the presence of noise and the presence of an MTF, which would simulate images from different electron detectors. The software pipeline provides a large degree of flexibility for generating training and validation data sets, along with training neural networks. To add to this flexibility, multiple neural network architectures were implemented for the user to benchmark. The neural network architectures include: The U-net, U-net++, and MSD-net all displayed in Figure 3.8. The data generation and neural network training segment of the pipeline is executable with just 4 lines of commands in a *Unix* terminal summarised in Listing 3.8. The software pipeline is compatible with HPC clusters which allow for a neural network to be prepared within a day, depending on the set up.

Each simulated image is paired with a ground truth label that defines the task that the network should solve and this is also selected by the user. Three labels were defined in this work: Mask labels, Disk labels, and Exitwave labels. The user does not require any knowledge of specific loss or activation functions introduced in Chapter 2, as the neural network and training procedure is automatically initialised based on the type of label specified by the user in the input parameters file.

An example was made of training neural networks with Mask labels for CeO_2 supported Au nanoparticle to perform binary segmentations of the Au nanoparticles, separating them from their surroundings in the image. These segmentations can be performed on large images with many nanoparticles present to highlight all the regions of interest. The binary segmentation map can be separated *via* a watershed algorithm, producing a multi-valued map separating every instance of a nanoparticle in the image. These instances are compared across frames and tagged so that every instance is connected throughout all the frames. This concept is visualised in Figure 3.11. Tracking the instances across all frames facilitated large-scale data analysis, since properties of each instance can be contained across frames and extracted. This was exemplified in *Dragonfly*, where high-resolution Fourier transforms and properties of selected nanoparticles can be extracted with the click of a button. This work showed that the Fourier transforms were of high enough resolution to easily distinguish between crystal planes corresponding to two different regions of a twinned nanoparticle. Tracking these peaks allowed for a quantification of various dynamic properties. This work also showed that the area of many nanoparticles could be tracked across many frames to determine the cause of vaporisation, whether it be the temperature supplied by the stage or beam induced local heating.

It is important to understand the limitations of neural networks before designing experiments that should be analysed by them. The ability of the U-net and MSD-net in performing mask segmentations at low signal to noise ratio was gauged as a function of frame dose. A lower limit for a full reliable neural network segmentation of a CeO_2 supported Au nanoparticle was identified at 200 $^{e^-}/Å^2$. The MSD-net presented an enhanced ability to differentiate between signal and noise and perform reasonable segmentations below the lower limit of the training data, which shows a strong generalisability. The MSD-net was able to segment regions of the nanoparticle down to 30 $^{e^-}/Å^2$, which is useful for object detection and tracking purposes. Close attention to modelling the MTF parameters was highlighted to optimise the segmentations by the neural networks in all frame dose regimes. This work provides the user of the software pipeline knowledge of the extent to which they can limit the beam damage and beam effects in the sample and still extract necessary information from the images.

Exitwave labels paired with images of MoS_2 nanoflakes (2H phase) can train neural networks to perform exit wave reconstructions with smaller focal series (down to 3 images) that are comparable to traditional algorithms. The neural networks, unlike traditional algorithms, did not require detailed knowledge of the CTF, but were unable to reconstruct the z-axis dependency of the atomic columns, which ultimately meant that 1S atomic columns that exist in the upper or low sulphur layer along the optical (z-)axis could not be differentiated. The initial desire was for the neural network to learn the deconvolution from Eq. (1.29) and perform a general regression between any given focal series and the corresponding exit wave. The neural networks, however performed worse when going from MoS_2 nanoflakes, to graphene supported MoS_2 nanoflakes, to a data set of thousands of 2D materials from the C2DB. It turned out that the the neural networks performed best when trained on data sets of a single type of atomic system, with minimal complexity, which reveals that the networks are utilising information of the given atomic system. These neural networks provide a tool to provide an approximate description of the chemical composition and 3-Dimensional positions of the atomic columns during image acquisition to help determine significant areas of interest.

To tackle a similar problem as the exit wave reconstructions, Disk labels were paired with images of MoS_2 nanoflakes (2H phase), to perform a multi-class segmentation of atomic columns, belong to a 1Mo, 2S, or 1S class. Despite the complexity of the intensities present in the focal series images of MoS_2 nanoflakes, the U-net, U-net++, and MSD-net were all able to score a perfect atomic F1-Score of 1.0 for over 80% a data set of 500 simulated focal series over a large range of microscope parameters. The neural networks preferred a focal series of at least 3 images, as for the exit wave reconstruction. The performance of each network was gauged as a function of the number of trainable parameters in the neural network and as a function of data points in the data set. This analysis proved there was no need for more than 100 data points in the training data set and no more than 10^6 trainable parameters. The neural networks could, however not generalise to defocus values outside the training set. Neural networks show a promising ability to locate and classify many atomic columns in order to track defect formation, dynamics of atomic columns, generate strain maps and more.

Deep learning is paving its way to becoming a standard tool for large-scale analysis, especially in cases that are exceedingly difficult for human interpreters. This work has presented a baseline for the possibilities and also limitations of these tools.

7.1 Outlook

Research is an endless quest for knowledge and it goes without saying that there are many routes to pursue to extend the work of this thesis. Similarly, any good software is subject to updates that add incremental improvements ideally based on user requests. The following will present some of the possible improvements to the software pipeline and what could be done to extend research possibilities with these tools.

The deep learning models implemented in this work are only able to perform a single task, ideally for a single atomic system. Any case where an attempt was made to perform a single task over a number of atomic systems showed increased signs of failure. Chapter 3 presented a software pipeline where the first step was to generate a diverse data set of a given atomic system. An improvement at this stage would be to expand the number of atomic systems available. Such an improvement was already exemplified in the exit wave reconstruction study made in Section 4.1, where the MoS_2 nanoflake construction was extended to include graphene substrates. The *Python* script in Listing 3.5 can be improved to generate any MX_2 atomic system, where the user can specify the M and X chemical species. Listing 3.3, showed that any metallic nanoparticle chemical species could be provided by the user, but this could be further extended to allow for various substrates to be specified by the user. The overall idea is to provide more atomic systems to extend the applicability of the software pipeline and it is achievable with relatively few lines of code, since the foundation is already there.

The neural network training segment of the software pipeline can be further improved by implementing more loss functions and optimisers, as several works show that this can significantly improve performance [68]. It is convenient to define a default so that users with less knowledge can run the entire pipeline, but providing optional flexibility allows the more advanced user to tune the training process of the neural networks for an improved result. With time it may also be useful to provide more neural network architectures, with recommended architectures for specific problems.

It would be interesting to fully implement the heterogeneous vibrational amplitudes from Chapter 5 into the software pipeline and gauge the performance of the neural network exit wave reconstructions from Section 4.1. Chen *et al.* [29] found that a realistic model of the heterogeneous atomic vibrations was significant in properly interpreting the chemical compositions of the atomic columns. Involving these vibrations in the training data may improve neural network performance on experimental data.

Another issue in probing atom positions through exit wave reconstruction was the presence of a substrate. This is a global issue and is usually tackled by Fourier filtering the substrate out of the exit wave. What if a neural network was trained to reconstruct the exit wave and perform the Fourier filtering, so that the substrate could automatically be filtered out. This could be achieve in two ways. The first way could be training the neural network on images with a substrate present paired with exit waves of the same system system without the substrate. This would treat the substrate as another degree of noise and train the neural network to filter it out automatically. The other way could be to do the same input to output pairing as the former, but Fourier filter the substrate out of the image to assist the network in neglecting it as noise. These methods were already investigated and showed promising results. If the neural network can perform the filtering and provide the exit wave, this would serve as an even more valuable tool for microscopists.

A next step in the atomic column classification in Section 4.3, could be to add an extra class differentiating between upper and lower layer 1S columns, to see if this segmentation can probe the optical (z-)axis positions of the 1S columns. Doing so would assist defect formation studies in determining the electron beam related threshold energies for knocking out S atoms above or below the monolayer sample.

A fact of reality is that any atomic column related analysis requires rather ideal conditions. The resolution and interpretability of the image must meet a certain standard and the atomic system has to be well aligned along a zone axis. These conditions are not always met and microscopists typically search through large data sets seeking images with these conditions for the analysis. Groschner *et al.* [14] presented a classification model, that classified nanoparticles based on a feature vector containing the mean and standard deviation of both the real and Fourier space image of the nanoparticle. Hand labelled feature vectors were used to train a random forest classifier to classify the nanoparticles into several classes including whether the nanoparticle was well aligned along a zone axis or slightly tilted displaying planar fringes rather than atomic columns. A more sophisticated model such as a neural network may be able to perform these classification based on only the Fourier transform and be equipped together with the large-scale Fourier transform extraction tool presented in Chapter 6. The many Fourier transforms can be extracted and classified with a click, identifying frames where nanoparticles are suitable for atomic column analysis. A simulated data set consisting of nanoparticles aligned along a zone axis and nanoparticles tilted off zone axis can be constructed and labelled in order to train a sufficient model. A prototype was constructed and proved to be applicable to Fourier transforms from experimental images.

Similar to the study made in determining the frame dose limitations of the neural network segmentation, the limitations to various CTF parameters could also be investigated. Experiments could be made where a frame is assigned as the ground truth frame and various CTF parameters could be controlled to identify when the neural networks lose sight of all atomic columns for example.

As a final note, it would be beneficial to implement the entire software pipeline into a graphical user interface software, such as *Dragonfly*. This would remove any need for knowledge in running scripts in a *Unix* command line and *Python* programming. Simulating the data sets and preparing the neural networks would all be available at the click of a button, similar to the large-scale analysis tools implemented in *Dragonfly* from Chapter 6. For live analysis tools the *Gatan Microscopy Suite*[1] would be a perfect platform to implement, for example, the neural network exit wave reconstructions. Making these tools highly accessible will provide a significant boost in the extraction and analysis of atomic scale data in high-resolution transmission electron microscopy.

---

[1] The industry standard software for (scanning) transmission electron microscope ((S)TEM) experimental control and analysis: `https://www.gatan.com/products/tem-analysis/digitalmicrograph-software`.

CHAPTER 8

Publications

The proceeding pages are reprints of the publications associated with this work. This chapter comes at the end since it would obstruct the general flow of the thesis to be placed in between. A summary of the main highlights of each publication is made in Chapter 4, which connects the work produced in each publication to the thesis in a more natural manner. The summary, however, neglected some specific details and discussions present in each publication, and so the reader is urged to read these publications as part of this thesis. All of the information provided by the thesis, provides an extensive insight into the work behind each publication, so that the reader can focus on the outcomes and conclusions.

Thank you for reading.

Matthew Helmi Leth Larsen
March 2023

8.1 Publication I: Reconstructing the Exit Wave of 2D Materials in High-Resolution Transmission Electron Microscopy Using Machine Learning.

Matthew Helmi Leth Larsen
Frederik Dahl
Lars P. Hansen
Bastian Barton
Christian Kisielowski
Stig Helveg
Ole Winther
Thomas W. Hansen
Jakob Schiøtz

Reprinted from Elsevier Ultramicroscopy Volume 243, January 2023, 113641.

Ultramicroscopy 243 (2023) 113641

Contents lists available at ScienceDirect

Ultramicroscopy

journal homepage: www.elsevier.com/locate/ultramic

Reconstructing the exit wave of 2D materials in high-resolution transmission electron microscopy using machine learning

Matthew Helmi Leth Larsen [a], Frederik Dahl [a], Lars P. Hansen [b], Bastian Barton [c], Christian Kisielowski [c], Stig Helveg [d], Ole Winther [e], Thomas W. Hansen [f], Jakob Schiøtz [a,*]

[a] Computational Atomic-scale Materials Design (CAMD), Department of Physics, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
[b] Topsoe A/S, Haldor Topsøes Allé 1, DK-2800 Kgs. Lyngby, Denmark
[c] The Molecular Foundry, Lawrence Berkeley National Laboratory, One Cyclotron Road, CA 94720, Berkeley, USA
[d] Center for Visualizing Catalytic Processes (VISION), Department of Physics, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
[e] Department of Applied Mathematics and Computer Science, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark
[f] National Center for Nano Fabrication and Characterization, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

ARTICLE INFO

Dataset link: doi:10.11583/DTU.15263655

Keywords:
Exit wave reconstruction
HRTEM
2D materials
Machine learning

ABSTRACT

Reconstruction of the exit wave function is an important route to interpreting high-resolution transmission electron microscopy (HRTEM) images. Here we demonstrate that convolutional neural networks can be used to reconstruct the exit wave from a short focal series of HRTEM images, with a fidelity comparable to conventional exit wave reconstruction. We use a fully convolutional neural network based on the U-Net architecture, and demonstrate that we can train it on simulated exit waves and simulated HRTEM images of graphene-supported molybdenum disulphide (an industrial desulfurization catalyst). We then apply the trained network to analyse experimentally obtained images from similar samples, and obtain exit waves that clearly show the atomically resolved structure of both the MoS_2 nanoparticles and the graphene support. We also show that it is possible to successfully train the neural networks to reconstruct exit waves for 3400 different two-dimensional materials taken from the Computational 2D Materials Database of known and proposed two-dimensional materials.

1. Introduction

Machine learning has become a powerful tool for analysing images. In fact, machine learning is a nascent tool in electron microscopy that is envisioned to have a large potential for quantitative image analysis [1,2]. In electron microscopy, applications of machine learning have up to now included segmentation of medical images [3], grain and phase identification [4–6], noise filtering [7,8] and in-plane location of atoms [9–11]. Moreover, Ede et al. showed recently that the imaginary part of the exit wave function can be reconstructed from the real part using a convolutional neural network [12] and Meyer showed that off-axis holograms, where phase information is recorded directly into the image, can be reconstructed using neural networks [13]. In this work we suggest that neural networks could potentially solve the classical phase problem and thus retrieve the entire electron wave function exiting the specimen in a transmission electron microscopy experiment.

Aberration-corrected high-resolution transmission electron microscopy (HRTEM) is one of the important experimental techniques to study the structure of materials at the atomic scale. The maximal amount of information about the sample is present in the exit wave, i.e. the wavefunction of the electrons exiting the sample. As the image is formed, some of this information is blurred or lost, both due to aberration in the lenses, and because the camera detects the intensity of the wave, not its phase.

It is well established that the full exit wave can be reconstructed from a focal series of images [14–17]. A series of typically around 20–50 images with varying defocus is used to numerically reconstruct the most likely wave function of the electron beam as it exits the sample. This can then be used to further reconstruct information about the chemical composition and 3D structure of the sample [18–20]. For beam-sensitive samples [21], exit wave reconstruction has the advantage of being averaging in nature such that information from many images with very low signal-to-noise ratio is combined in a single exit wave image of superior signal-to-noise ratio [20]. Several numerical algorithms are available for reconstructing the exit wave [15–17].

Here we examine a Convolutional Neural Network (CNN) as an alternative way to reconstruct the exit wave. This reconstruction is possible from a low number of HRTEM images, and with the advantage that the detailed knowledge of the aberration parameters of the

---

[*] Corresponding author.
E-mail address: schiotz@fysik.dtu.dk (J. Schiøtz).

https://doi.org/10.1016/j.ultramic.2022.113641
Received 27 December 2021; Received in revised form 23 September 2022; Accepted 30 October 2022

microscope is not needed. We envision that this can be developed into a tool for on-the-fly exit wave reconstruction while taking data on the microscope, perhaps supplemented with more traditional exit wave reconstruction as post processing. In the present case, the images were convoluted with the effects of defocus, first order astigmatism, coma, and blurring including focal spread. In this case a focal series of two to three simulated HRTEM images were sufficient to reconstruct the exit wave with sufficient accuracy in order to extract quantitative information about the sample. In principle, it should be straightforward to extend the present method to situations with low signal-to-noise ratio and more unknown aberrations, in which case it is likely that a larger focal series will be needed.

Recently, atomically thin two-dimensional (2D) materials have been an active topic of research, with applications ranging from electronics to energy storage and catalysis [22,23]. For example, molybdenum disulphide (MoS_2) is the preferred catalyst for removing sulphur from crude oil destillates, and is one of the reasons that acid rain is no longer one of the most pressing environmental problems [24]. In this paper, we focus on exit wave reconstruction for the rapidly growing class of 2D materials, although the methods should be generally applicable. We show that neural networks can reconstruct the exit wave both when trained to a single material, and to a database of thousands of proposed 2D materials. The reconstruction is of sufficient quality to permit analysis of the image peaks associated with the atomic columns e.g. by using Argand plots to identify the type and number of elements in the material [19].

We also show that it is possible to train the neural network purely on simulated data, and apply it successfully to experimental images of non-trivial complexity, in this case a model catalyst based on molybdenum disulphide.

2. Methods

The neural network architecture is a Unet [25] / FusionNet [3] architecture, very close to the one used by Madsen et al. [9], with the main modification that concatenation is used instead of elementwise addition for the skip connections. A linear activation function is applied in the output layer, as exit wave reconstruction is a regression problem rather than a classification/segmentation problem. Details of the architecture can be found in the Supplementary Online Information (SOI Sec. S1). The neural network is implemented and trained using the Keras interface [26] to Tensorflow version 2.5 [27]. We train using simulated images only. We computer-generate a training set and a corresponding validation set of atomic structures, using the Atomic Simulation Environment (ASE) [28].

Three data sets of increasing complexity were created. The first consists of nanoparticles (nanoflakes) of molybdenum disulphide (MoS_2). In this data set we ignore that the nanoparticles will typically be supported on another material in the microscope. Nevertheless this data set will be relevant for e.g. edges of MoS_2 films on a TEM grid, where no support is visible in the region of interest.

The second dataset is MoS_2 supported on a graphene substrate. A nanoflake of graphene and one of MoS_2 are generated in the computer, and are placed with a random distance between 3.3 and 7.0 Å. One quarter of the cases are placed with the lattice vectors of the two layers in the same directions, another quarter with a rotation of 15°, one quarter with a rotation of 30°, and the rest with a random rotation. In both of these datasets 1000 samples are created for training, and 1000 for validation.

The third dataset consists of nanoflakes of materials from the Computational 2D-materials Database (C2DB) [29] in the latest version dated 2021/06/24. This version of the database contains 4056 known or proposed 2D materials, but a significant number of these have very complex structures where the quasi-2D material contains a large number of atomic layers. We filtered the database so we only keep structures with at most eight atoms in the unit cell. That left us with 3393 materials. Two samples are created of each material. Materials are randomly assigned to the training or validation set with a probability of 2:1, but in such a way that all materials containing the same set of elements are assigned to the same set.

For all three datasets, vacancies and holes are introduced in the systems. A vacancy is introduced by selecting a random atom and removing it; holes are made by selecting a random atom and then removing the entire atomic column. In the case of MoS_2, if a sulphur atom is selected then a vacancy would be removing just that atom, whereas creating a hole would be removing an S_2 dimer. If a molybdenum atom is selected there will be no difference. We select 5% of the atoms for vacancy creation, then 5% for hole creation. All atomic positions are then perturbed by adding a Gaussian with mean of 0 and spread of 0.01 Å to all atomic positions. Finally, all samples are tilted by a random angle up to 10° in a random direction.

Exit waves are then calculated using the multislice algorithm [30, 31], using the abTEM software [32]. The lateral sampling of the wave function is 0.05 Å, and the slice thickness is 0.2 Å, see the SOI Sec. S2. As a simple model of atomic vibrations, the potential of the atoms is smeared by a Gaussian with

$$\left\langle u^2 \right\rangle = \frac{3\hbar^2}{2\, mk_B\theta_D} \cdot \left(\frac{1}{4} + \frac{T}{\theta_D} \right) \tag{1}$$

where m is the atomic mass and θ is the Debye temperature [33, supplementary online information]. As the same value must be used for all atoms, we use the atomic mass of Sulphur. With $\theta_D = 580$ K for bulk MoS_2 [33], this gives a value of $\left\langle u^2 \right\rangle = 0.0030\text{Å}^2$ at 300 K. Our own *ab initio* molecular dynamics simulations of MoS_2 gives a somewhat larger value, which is expected as molecular dynamics ignores the quantization of the phonons which is important below the Debye temperature. As an approximation, we also use this value of $\left\langle u^2 \right\rangle$ for the materials in the C2DB. If the reconstructed exit wave is to be used to gain information about the vibrational amplitudes of different kinds of atoms, as is done in Ref. [20], the phonons need to be modelled with a more sophisticated method, such as the frozen phonon method, at a significant cost in computational burden (up to two orders of magnitude).

After generating the exit waves, the abTEM software is used to generate typically three images of the sample by applying a Contrast Transfer Function (CTF), Poisson noise in the detector, and a Modulation Transfer Function (MTF) introducing correlations in the noise. This is described in detail elsewhere [9]. The parameters of the CTF and the MTF (collectively referred to as the "microscope parameters") are drawn from distributions given in Table 1. The three images have the same microscopy parameters except that the defocus is changed by 5 ± 0.1 nm between the images. If a different number of images is used, the total variation in defocus remains at 10 nm. The minimal value of the focal spread is sufficient to ensure that the CTF has gone to zero well below the Nyquist frequency corresponding to the spatial resolution, thus avoiding aliasing effects in the image simulations.

The expensive part of the image simulation is the multislice algorithm calculating the interaction between the electron beam and the sample. The action of the CTF and the MTF are computationally cheap, and for that reason it is convenient to generate multiple images of the same sample with varying microscope parameters. Depending on the computational setup, it may be most convenient to generate images on-the-fly during training, such that the network sees different images of the same samples in each training epoch, or it is possible to pre-generate and store the images. In this work we pre-generated ten epochs of images for the training set, and one for the validation set. We then cycled through the pre-generated epochs for the actual training, which were up to 200 epochs (leading to each image being reused 20 times).

The neural network is trained using the mean square error (MSE) as the loss function, with the RMSprop training algorithm as implemented in Keras, and a learning rate of 5×10^{-4}. We also tried using the

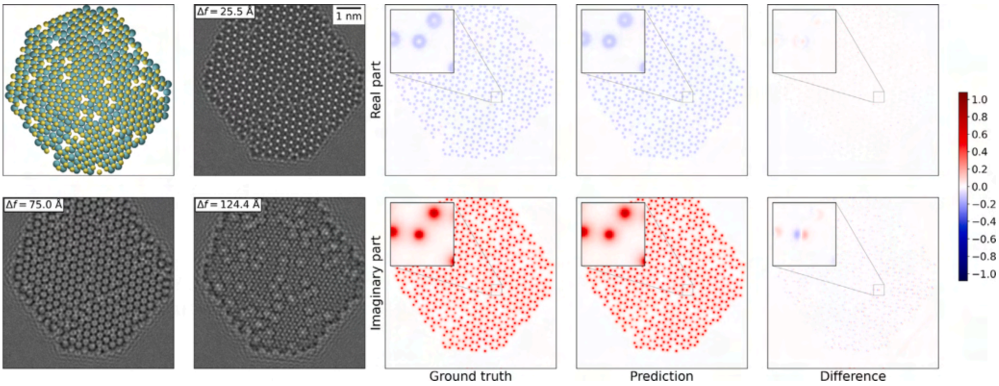


Fig. 1. Reconstruction of the exit wave of a MoS_2 nanoparticle by a network trained on unsupported MoS_2 nanoparticles. The first two columns show the structure and the three simulated HRTEM images. The third column shows the real and imaginary part of the actual wave function (the ground truth). The fourth column shows the exit wave reconstructed by the neural network, and the last column shows the difference. The inset highlights the point with the largest deviation, an atom misplaced by 7 pm (0.6 pixels). The Root Mean Square Error (RMSE) is 0.0062. The colourmap indicates the scale used for plotting the exit wave. The same scale is used in all the following figures.

Table 1
Microscope parameters. For each image series, a set of microscope parameters are drawn within the limits given here, except the acceleration voltage which is kept constant. All distributions are uniform, except for the dose which is exponential. The defocus of the first image is picked so the defocus of all images are within the bound specified.

Parameter	Lower bound	Upper bound
Acceleration voltage	50 keV	
Defocus (Δf)	−150 Å	150 Å
Spherical aberration (C_s)	−15 μm	15 μm
2-fold order astigmatism		
Amplitude (C_{12})	0	25 Å
Angle	0	2π
Coma		
Amplitude (C_{21})	0	600 Å
Angle	0	2π
Focal spread	5 Å	20 Å
Blur	0.5 Å	1.5 Å
Electron dose	$10^{2.5}$ Å^{-2}	$10^{5.0}$ Å^{-2}
Resolution	0.10 Å	0.11 Å
MTF c_1	−0.6	0.2
MTF c_2	0.1	0.2
MTF c_3	0.6	1.8

Adam algorithm [34], and saw similar but slightly less stable results, whereas Adam with the AMSgrad modification gave almost identical results to RMSprop. Increasing the learning rate above 1×10^{-3} would make the training unstable, and decreasing it below 5×10^{-4} was detrimental to the learning. Training curves showing the loss function of the training and validation set are shown in the SOI (Fig. S2). In spite of the reuse of pre-generated images, the training curves do not show signs of overfitting. We therefore did not use regularization in the neural network. Training using mean absolute error (MAE) as loss function lead to doubling the error in placement of the atomic columns compared to MSE, and was therefore not used.

The sharp potential of the nucleus causes some amount of annular structures to appear in the exit wave, in spite of the application of Debye–Waller smearing. This fine structure contain little or no information of value when analysing the exit waves. However, the neural network will attempt to recover this structure, leading to an overall small degradation of its ability to recover more important information about the main peaks associated with the atomic columns. For simplicity, we have filtered the exit waves prior to training by folding them with a Gaussian with a spread of 15 pm, see SOI Fig. S4. This leads to a significant improvement in the network performance, in particular when it comes to extracting quantitative information from the peak values.

3. Results and discussion

Fig. 1 shows the simplest situation, where the network is trained and tested with unsupported MoS_2 nanoparticles. The figure shows the real and imaginary parts of the exit wave used to simulate the images (the “ground truth”), and the exit wave reconstructed by the neural network (the “prediction”). For thin samples, the interaction between the electron wave and the sample mainly results in a phase shift of the wave [20]. This is also the case for the data in the figure, where the main part of the signal is in the imaginary part.

The difference plot in Fig. 1 shows that the network clearly reconstructs the imaginary part of the exit wave both qualitatively and quantitatively. We see that all peaks are reconstructed correctly, and that the neural network both reconstructs the periodic lattice and the deviations from periodicity such as vacancies, including single sulphur vacancies where a single sulphur atom leaves a weaker peak than the usual two atoms. The system shown in Fig. 1 was chosen as the median of the validation set, half the systems in the validation set perform worse, and half perform better. In the SOI Section S5 we show some of the worst systems in the validation set, even the five percentile sample is reconstructed quite well.

Fig. 2 shows the more complex situation, where the network is trained on graphene-supported MoS_2 nanoparticles. The way the training set is constructed does not guarantee that the full MoS_2 nanoparticle is overlapping with the support, so in this case the network needs to learn to recognize both supported and unsupported MoS_2.

The network is able to reconstruct both the part of the wave function coming from the support and from the nanoparticle, in spite of the signal from the support being much weaker than from the nanoparticle. The network is even able to correctly find the carbon vacancies that have been introduced in the support. It should be noted that if the network is trained for a shorter time (50 epochs instead of 200), it loses its ability to find the carbon atoms below the nanoparticle. The largest deviation in the reconstructed exit wave comes from a slight misplacement of the atoms in the MoS_2 layer, the maximal error in the placement of an atom is 9.7 pm, corresponding to a single pixel. This system is again chosen as the median of the validation set.

Finally, the method was tested on the C2DB database of 3393 proposed two-dimensional materials [29]. Again we show the median system, a nanoparticle of CoCl, (Fig. 3). We see how all atoms are placed correctly, but the detailed shape of the peaks in the imaginary part of the wave function is not well reproduced, the network predicts somewhat smoother peaks. In addition, the network does not always identify positions where single atoms are missing, leaving only one

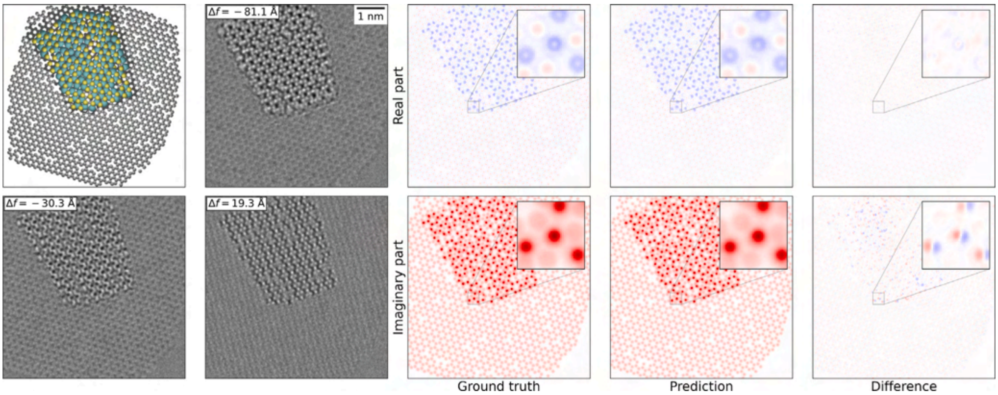


Fig. 2. Reconstruction of the exit wave of a MoS_2 nanoparticle supported on graphene, by a network trained on graphene-supported MoS_2 nanoparticles. The panels are the same as in Fig. 1. It is seen that the network locates the atoms both in the MoS_2 nanoparticle, and in the substrate. The worst spot in the prediction where an atom is misplaced by a single pixel. It is worth noticing that the graphene support is also reconstructed correctly, including the vacancies in the graphene. The RMSE is 0.0122 and the colorbar is the same as in Fig. 1.

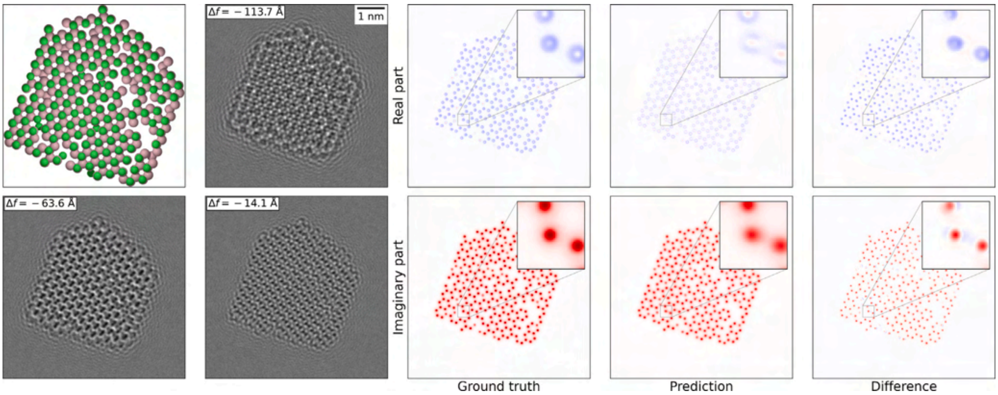


Fig. 3. Reconstruction of the exit wave of a CoCl nanoparticle by a network trained on the C2DB (see text). The panels are the same as in Fig. 1. The network correctly determines the positions of all the atoms (the maximal deviation is 8 pm or 0.77 pixel), but does not correctly reproduce the sharpness of the peaks in the wave function. The RMSE is 0.0263.

atom in the atomic column. Each position in the apparent hexagonal lattice contain both a Co and a Cl atom, alternately oriented with the Co or Cl on top, and staggered in the z direction.

In order to obtain a more quantitative measure of the performance of the networks, we have created histograms of the root-mean-square error (RMSE) of all the images in the validation sets, see Fig. 4. In general, the networks are better at reproducing the strong signal in the imaginary part of the exit wave than the weaker real part. It is seen that the performance of the network decreases somewhat as the complexity of the data set is increased, going from unsupported MoS_2 to supported MoS_2 to the C2DB dataset. It is not surprising that the network can be trained for better performance on the simpler datasets. As a "baseline", we also show the histogram produced from one of the datasets where the predictions are compared with randomly chosen other exit waves in the dataset (the Y-scramble method) rather than with the correct exit wave. This shows the performance of a hypothetical network learning the overall properties of exit waves but learning nothing about the specific systems, i.e. it acts as a "null hypothesis".

It is also seen that the relative error is significantly larger for the real part of the exit wave. This is because its magnitude is 3–4 times smaller than the imaginary part (this can e.g. be seen by the position of the peaks in the Argand plots in Fig. 5). It is only in the simplest case (unsupported MoS_2) that the network performs well on the real part.

We also test how networks trained on the C2DB dataset performs on the supported MoS_2 and vice versa. Unsurprisingly, the network trained on supported MoS_2 performs poorly on the C2DB dataset, as the latter contains a far richer variety of structures. On the other hand, the network trained on the C2DB generates a very broad distribution of results when applied to the database of supported MoS_2 structures (the red curve in Fig. 4). Our interpretation is that this is because the network correctly analyses the parts of the system where the MoS_2 and graphene only overlap a little, but performs badly where they overlap. While these systems are not inherently more complicated than in the C2DB, they differ in a fundamental way, as there are two different lattices in the system (the lattice of graphene and the one of MoS_2), whereas all systems in the C2DB training set only contain a single (but often more complicated) crystal lattice. This illustrates the importance of training the network on systems that are similar to the final application.

The purpose of an exit wave reconstruction is usually to extract quantitative information about the atomic columns. This is often done in form of an Argand plot, where the peak values of the wave function at the locations of the atoms are plotted in the complex plane [19]. It is therefore not enough that an exit wave reconstructed by neural networks visually and statistically resemble the actual exit wave, it

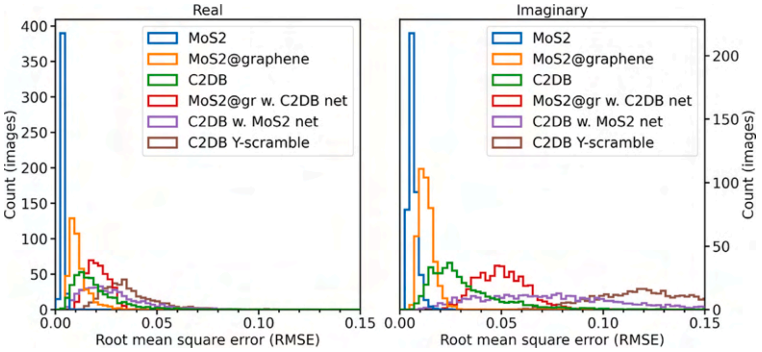


Fig. 4. Histograms of the root-mean-square error of the images in the validation sets for the various networks, showing their relative performance. Blue is the network trained and tested on unsupported MoS_2, orange is graphene supported MoS_2, and green is the C2DB. It is seen that the performance of the network decreases somewhat as the complexity of the data set is increased. The brown line is a baseline, this is the performance obtained if the network does not at all recognize the structure, obtained using the Y-scramble method (see text). The red curve shows the supported MoS_2 validation set with the network trained on the more diverse C2DB. As the samples contain two separate lattices, it is outside the training set of the C2DB. Validating the C2DB test set with the network trained on MoS_2 also gives bad results (purple curve), as the C2DB contains structures too far from what is observed in MoS_2.

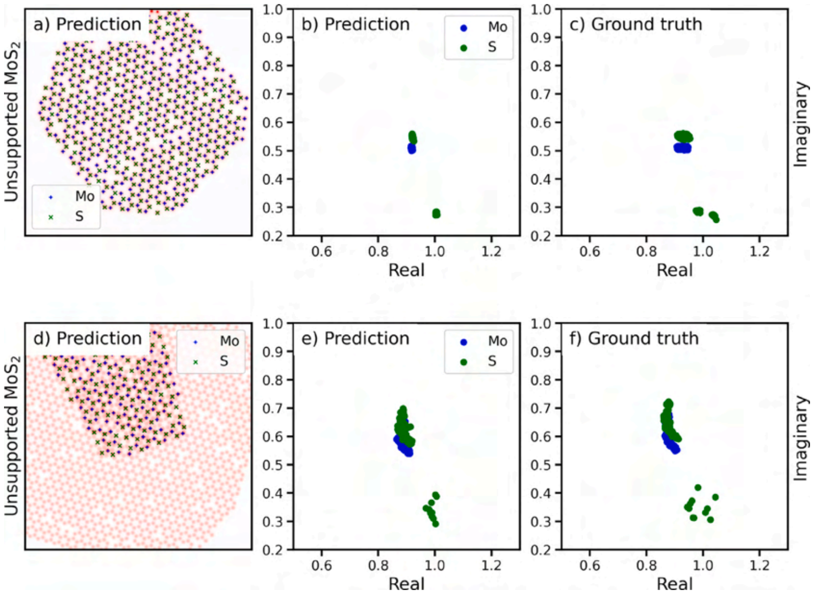


Fig. 5. Argand plots of the complex value of the exit wave function at the local maxima of the change in wave function ($|\Psi_{\text{exit}} - 1|$). Top row: unsupported MoS_2 (the same system as Fig. 1). Bottom row: supported MoS_2 (same as Fig. 2). a+d: The imaginary part of the reconstructed wave function. The peaks are marked with blue plusses or green crosses, depending on whether they correspond to a Mo or S atomic column position. b+e: The Argand plots of the reconstructed wave function. The separate points in the lower part of the plot corresponds to columns with a single S atom instead of two. c+f: The similar Argand plots made from the ground truth exit wave function.

should also permit analysis in an Argand plot. This is shown in Fig. 5, where we show Argand plots of both the unsupported and supported nanoparticles from Figs. 1 and 2. For the unsupported nanoparticle, the Argand plot is just able to distinguish between a single Mo atom (atomic number $Z = 42$) and a sulphur dimer (sum of atomic numbers $\sum Z = 32$). The sulphur vacancies, where there is only a single sulphur atom in the atomic column ($Z = 16$) are clearly separated from the other types. It is, however, not possible to determine if the missing atom was above or below the plane of the Mo atoms, although that information was present in the original wave function (shown as "ground truth", where we see that the spots corresponding to single S atoms is split into two nearby spots, as would be expected from atoms with different z coordinate, see Chen et al. [19]).

For the case of supported MoS_2 [Fig. 5(d–f)], the picture is less clear. The Argand plot still clearly separates the sulphur vacancies from the other atomic columns, but there is a larger spread on the column values, and no longer a clear separation between columns containing two S atoms or a single Mo atom. However, if the same analysis is done on the ground truth exit waves (Fig. 5(f)), the situation is the same. This is most likely due to interference from the substrate.

In the Argand plot, the position along the imaginary axis is largely indicative of the total atomic number of the atomic column in the weak phase limit [20]. The positions of the Argand points are also affected by the z-height of the column relative to the plane of the exit wave. This is mainly due to the propagation in free space that

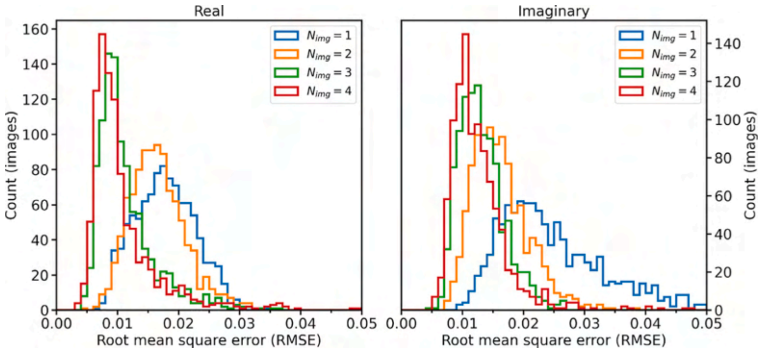


Fig. 6. Test of how the number of input images affects the performance. A single input image (blue curve) clearly does not give a good reconstruction of the exit wave. Already with two images (orange curve), good performance is obtained, at least for the imaginary part. Three images (green) as used in the rest of this work gives an improvement, whereas four images (red) gives only a marginal further improvement.

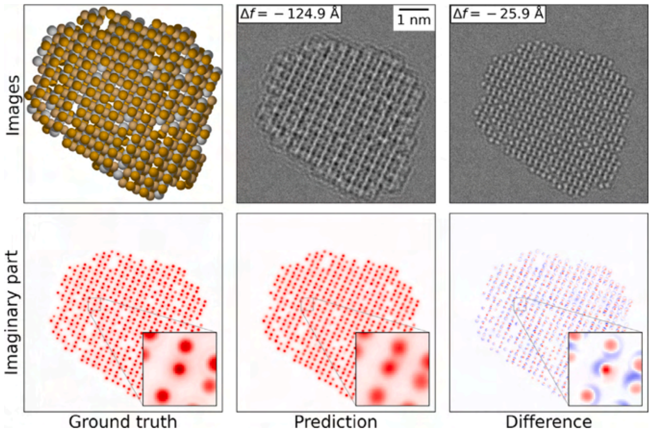


Fig. 7. Reconstruction of the exit wave of a $AgCuTe_2$ nanoparticle by a network trained on the C2DB. To save space we do not show the real part of the exit wave, and only the first and the last of the three images. The network does not recognize that the copper atoms are systematically slightly offset from the high-symmetry positions.

further change phase linearly with the atomic column height z [35]. This effect is clearly not reproduced by the neural network, as it cannot distinguish between single sulphur vacancies on the two sides of the nanoparticle [Fig. 5(b+c)]. It is possible that a network could be trained to distinguish these features by including training data where they are more prominent, i.e. a larger concentration of single sulphur vacancies and perhaps samples with higher tilt angles, producing height differences.

As a significant amount of information about the exit wave is encoded in how the image changes with defocus, it is our working hypothesis that a number of images are necessary for a neural network to be able to reconstruct the exit wave. This is verified in Fig. 6, showing the performance of networks trained on the same C2DB training sets but with a different number of input images. It is seen that some information about the exit wave can be gained from even a single image, but a dramatic improvement is seen going to two input images. A small further improvement is seen when increasing the number of images to three or four, and we decided to use three images in the rest of this work. In the simulations with two, three or four images, the total range of defocus from the first to the last image were in each case 10 nm.

4. When the network fails

No neural network is perfect, and it is important to be aware of the kind of failures that can occur when analysing an image. We illustrate this with two kinds of errors observed in the C2DB database.

The first case is silver copper telluride ($AgCuTe_2$), shown in Fig. 7. On one hand, the method reliably finds all the vacancies in the structure, a task that would be very difficult by visual inspection of the three images. On the other hand, the network fails to discover a small spontaneous breaking of the symmetry in the structure: the Cu atoms are slightly displaced compared to the rectangular lattice formed by the Ag and Te atoms. This is a highly unusual configuration, and the neural network interprets it as the far more common symmetric configuration.

In the second case, the network is locally inserting extra atoms into the structure, creating unphysical defects, see Fig. 8. This kind of errors should be relatively easy to spot for the scientist.

The cases in Figs. 7 and 8 were chosen manually. In the SOI, we give examples of some of the worst and best results of the networks, selected solely from the RMSE of the prediction.

As the examples here show, it is difficult to train a single network to 3400 different materials, even if they are two-dimensional. The networks trained to a single material (MoS_2), with or without support, do not exhibit these failure modes. It is therefore recommended to

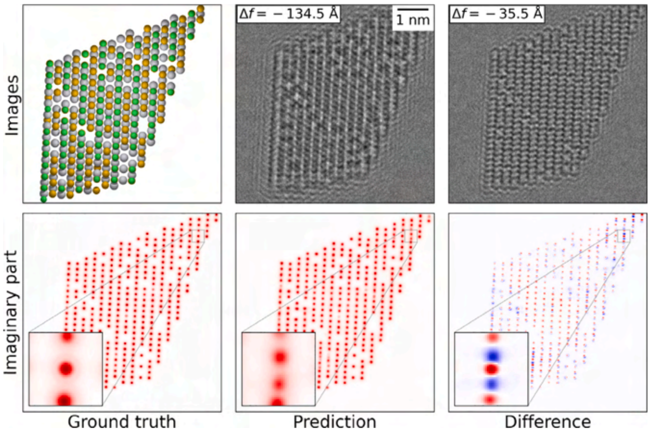


Fig. 8. An example of the neural network inserting extra atoms in several places in the system. The system is PtSeCl.

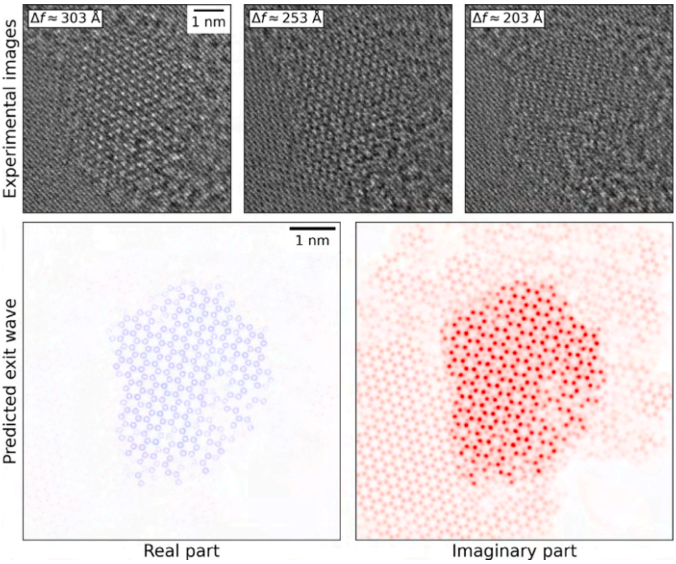


Fig. 9. Applying the neural network to three experimental images (top row) results in the exit wave function depicted in the bottom row. The network is clearly able to identify atomic positions in the MoS_2 nanoflake, but is not able to distinguish between Mo and S_2. The network also provides a best guess on the positions of carbon atoms in the support, but as the support is graphite and the network was trained on single graphene layers as support, that cannot be considered reliable. The defocus values are as reported by MacTempas (overfocus), which uses the opposite sign convention from abTEM and this work. The colours are the same as in Fig. 1.

train networks to smaller classes of materials matching the kinds of systems being studied experimentally. Furthermore, the kinds of errors shown here can be detected by training two or more different networks to similar data sets, and detecting when the networks differ in their prediction.

5. Application to experimental data

We apply the method to experimental data, a focal series of a MoS_2 model catalyst recorded on the TEAM 0.5 transmission electron microscope at 50 keV beam energy. The data analysed here is similar to what was published recently by Chen et al. [20], and we refer to that publications for details regarding the experimental setup.

In their publication, Chen et al. used focal series of 20–44 images to reconstruct the wave functions. Here, we have selected three images from their focal series for analysis by the neural network.

As the resolution of this image series is significantly lower than what we have otherwise been using in this work (0.227 Å/pixel instead of 0.105 Å/pixel) we retrained a network for this resolution, based on the same data set of supported MoS_2, but resampled to resolutions in the interval from 0.215 to 0.235 Å/pixel. The lower resolution had only a small detrimental effect on the network performance when tested on the validation set. We then selected three experimental images with a difference in defocus of 50 Å, to match the defocus difference between the three images used to train the network. The images and the resulting exit wave are shown in Fig. 9. As can be seen, a clear exit wave is reconstructed, showing the honeycomb lattice of the supported MoS_2 nanoflake, and of the supporting graphite lattice (few layers of graphene). However, an Argand plot is not able to distinguish the lattice points of the Mo and S sublattice (not shown), consistent with what we saw in simulated images (see Fig. 5, panel e and f). In both cases, the reason is the same. Some peaks in the wave function of the MoS_2 coincide with peaks from the graphite, some do not, and that leads to greater variation between peaks than the difference between a Mo atom and two S atoms.

In their publication, Chen et al. [20] were able to distinguish between peaks from Mo and S atomic columns, but their analysis of the exit wave is also more elaborate. First, they Fourier filtered their images, removing spatial frequencies coming from the graphene support from the exit wave. Second, even if a clear distinction of the peak imaginary values of the Mo and two S atomic columns were made, it is worth noticing that the chemical interpretation of the relative intensities calls for caution. As reported by Chen et al., the peak values can be severely reduced and the imaginary parts be broadened across a nanocrystal due to heterogeneous vibrations response of the sample under illumination. Chen et al. offers a framework for an interpretation of the exit wave function. This interpretation is independent of the way in which the exit wave function is reconstructed, which is the prime objective for the present analysis.

With even just a few images, the network can already capture the main arrangement of the atomic columns based on an experimental focal series of low-dose HRTEM images. Further inclusion of images from the focal series might help in better account for the column intensities and role of high order aberrations on the contrast blurring in the experimental image. For a full qualitative analysis of the experimental data, networks would have to be trained to specifically take into account a more realistic model for the vibrations of the atoms, as well as the more complicated multilayer support in the experimental data. In addition, the network should be trained to handle carbon contamination of the sample.

6. Comparison to traditional exit wave reconstruction

To be able to compare this method with more traditional methods for exit wave reconstructions, we have applied the algorithm of Gerchberg and Saxton [36], as implemented in MacTempas version 2.4.50, to three simulated image series of graphene supported MoS_2. The systems were selected according to how well they had been reconstructed by the neural network, we chose the 25, the 50 and the 75 percentile images (Figures S10, 2 and S11, respectively).

The generated data sets contain eleven images with a 1 nm change in defocus between each of them, leading to a total defocus range of 10 nm, the same that was used for the neural networks. All eleven images are used for the Gerchberg–Saxton(GS) exit wave reconstruction, whereas only three (the first, middle and last) were used for reconstructions with the neural network.

The GS exit wave reconstruction algorithm was given the actual values of the defocus, the spherical aberration (C_s) and the focal spread, instead of determining them through an optimization process as is usually done. No coma or 2-fold astigmatism was assumed in the reconstruction process, although both coma and astigmatism were present in the images.

In contrast, the neural network does not require any of this information, it is trained to reconstruct the wave function a few Ångström below the lowest atom in the sample without further knowledge of neither the exact values of the defocus, nor of the aberrations of the microscope, except that they are within the intervals used to train the neural network (Table 1).

A comparison between the neural network and the more traditional exit wave reconstruction is shown in Fig. 10. At first sight, it looks like the neural network strongly outperforms the traditional reconstruction, the difference between the reconstructed image and the original ground truth wave function is much smaller for the neural network reconstruction. However, this is mainly because the longest wavelengths in the exit wave have not been reconstructed by the Gerchberg–Saxton algorithm, leading to the phase of the wave function locally being averaged to zero. It is thus more fair to compare the reconstructed wave in Fig. 10(e) with a ground truth wave function where the longest wavelengths have been filtered out (panel f), using a Gaussian filter with a width of eight pixels (0.9 Å). In this case, visual inspection indicate that the error of the two models are of similar magnitude, although the neural network appears to be performing best. This is confirmed by calculating the Root Mean Square Error for the CNN reconstruction (i.e. for the difference between panel c and d) and for the Gerchberg–Saxton(panels e and f). The RMSE is 0.013 and 0.061, respectively.

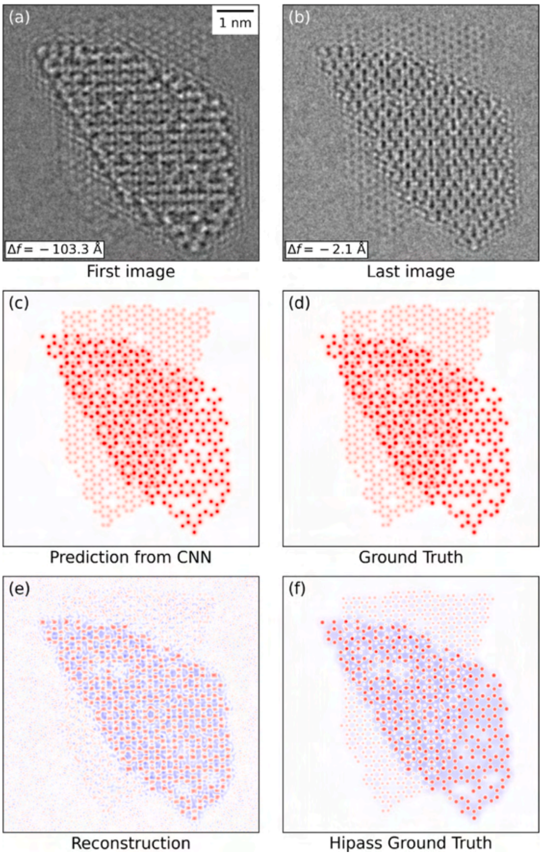


Fig. 10. Comparing the neural network with a traditional algorithm for exit wave reconstruction. (a) and (b) The first and last images in the image series. (c) The exit wave reconstructed by the network. (d) The ground truth (correct) exit wave. (e) The exit wave reconstructed by the Gerchberg–Saxton algorithm. The large deviations are due to the long wavelength part of the exit wave not being reconstructed. (f) The ground truth wave function with the longest wavelengths removed. The colourmap for the exit wave is identical to the one in Fig. 1.

The system shown in Fig. 10 is the 25-percentile system. Similar plots for the 50 and the 75-percentile systems are shown in the SOI (Figures S13 and S14). It should be noted that the Gerchberg–Saxton reconstruction of the 50-percentile image is of significantly lower quality than the two others, although the neural network did not have problems with this image series. This could be due to those images having both 2-fold astigmatism and coma in the upper end of the range shown in Table 1.

Inclusion of more aberrations than the ones in Table 1 might change these conclusions, and might require using more images for the neural network reconstruction to be reliable. It does, however, appear that a neural network is able to quickly give a reconstructed exit wave of a quality at least comparable to a traditional exit wave reconstruction from only a few images.

7. Conclusions

Convolutional Neural Networks are a promising alternative to traditional exit wave reconstruction, with the obvious advantage that they only require a few images instead of a long image sequence, that the data processing is fast enough to be done in real time at the microscope, and that detailed knowledge of the aberration parameters of the microscope is not needed. It does, however, require that the networks are optimized for the systems at hand.

As expected, the method works best for simpler systems, illustrated here with unsupported and graphene-supported MoS_2 nanoparticles, where the exit waves are reproduced with a fidelity that allows for both qualitative and quantitative analysis. For significantly more complicated structures, illustrated here with the relatively diverse C2DB dataset, the network overall performs well, but fails to reconstruct some details in some of the more complex materials. Nevertheless, even in the more complicated materials, the majority of the structure including the positions of point defects is recovered by the neural network.

One could hope that the neural network had learned to generally invert the Contrast Transfer Function of the microscope. That is, however, not the case. The network utilizes knowledge about “likely” structures based on the kind of structures it has seen in the training set, and must be trained on structures similar to the ones it will be used to analyse. On the other hand, this use of prior knowledge of the systems is probably what enables the network to reconstruct the exit wave based on only three input images, and without knowledge of the actual parameters of the CTF. It should be pointed out that including further aberrations than the ones used in this work (Table 1) may require using more than three images as input to the neural network.

In summary, we have demonstrated that neural networks can be trained to reconstruct the exit wave function of a varied class of two-dimensional materials, with only three HRTEM images with different defocus as input to the network. We can train and validate the network on simulated data, and then apply it to analyse experimentally obtained data, demonstrated here with the case of MoS_2 supported on graphene.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The code is available on github [37]. The trained networks, the scripts and data used to train the network and generate all figures except Fig. 9 are available from the DTU Data repository at doi:10.11583/DTU.15263655. The experimental data used in Fig. 9 belong to the authors of Ref. [20].

Acknowledgements

We would like to thank Sophie K. Kaptain and Dr. Daniel Kelly for technical assistance in connection with the MacTempas exit wave reconstructions.

Funding

The authors acknowledge financial support from the Independent Research Fund Denmark (DFF-FTP) through grant no. 9041-00161B. L.P.H. was financially supported by The Danish Council for Technology and Innovation (08-044837). The Center for Visualizing Catalytic Processes is sponsored by the Danish National Research Foundation (DNRF146). Electron microscopy is supported by the Molecular Foundry, which is supported by the Office of Science, the Office of Basic Energy Sciences, the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.ultramic.2022.113641.

References

[1] S.V. Kalinin, et al., Lab on a beam—Big data and artificial intelligence in scanning transmission electron microscopy, MRS Bull. 44 (2019) 565–575, http://dx.doi.org/10.1557/mrs.2019.159.

[2] S.R. Spurgeon, et al., Towards data-driven next-generation transmission electron microscopy, Nature Mater. 20 (2020) 274–279, http://dx.doi.org/10.1038/s41563-020-00833-z.

[3] T.M. Quan, D.G.C. Hildebrand, W.-K. Jeong, FusionNet: A deep fully residual convolutional neural network for image segmentation in connectomics, 2016, arXiv.org (2016) 1612.05360.

[4] E.A. Holm, et al., Overview: Computer vision and machine learning for microstructural characterization and analysis, Metall. Mater. Trans. A 37 (2020) 5985–5999, http://dx.doi.org/10.1007/s11661-020-06008-4.

[5] S.M. Azimi, D. Britz, M. Engstler, M. Fritz, F. Mücklich, Advanced steel microstructural classification by deep learning methods, Sci. Rep. 8 (2018) 2128, http://dx.doi.org/10.1038/s41598-018-20037-5.

[6] B.L. DeCost, B. Lei, T. Francis, E.A. Holm, High throughput quantitative metallography for complex microstructures using deep learning: A case study in ultrahigh carbon steel, Microsc. Microanal. 25 (2019-02) 21–29, http://dx.doi.org/10.1017/s1431927618015635.

[7] R. Lin, R. Zhang, C. Wang, X.-Q. Yang, H.L. Xin, TEMImageNet training library and AtomSegNet deep-learning models for high-precision atom segmentation, localization, denoising, and deblurring of atomic-resolution images, Sci. Rep. 11 (2021) 5386, http://dx.doi.org/10.1038/s41598-021-84499-w.

[8] J.L. Vincent, et al., Developing and evaluating deep neural network-based denoising for nanoparticle TEM images with ultra-low signal-to-noise, Microsc. Microanal. 27 (2021) 1431–1447, http://dx.doi.org/10.1017/s1431927621012678.

[9] J. Madsen, et al., A deep learning approach to identify local structures in atomic-resolution transmission electron microscopy images, Adv. Theory Simul. 1 (2018) 1800037, http://dx.doi.org/10.1002/adts.201800037.

[10] M. Ziatdinov, et al., Deep learning of atomically resolved scanning transmission electron microscopy images: Chemical identification and tracking local transformations, ACS Nano 11 (2017) 12742–12752, http://dx.doi.org/10.1021/acsnano.7b07504.

[11] C.-H. Lee, et al., Deep learning enabled strain mapping of single-atom defects in two-dimensional transition metal dichalcogenides with sub-picometer precision, Nano Lett. 20 (2020-05) 3369–3377, http://dx.doi.org/10.1021/acs.nanolett.0c00269.

[12] J.M. Ede, J.J.P. Peters, J. Sloan, R. Beanland, Exit wavefunction reconstruction from single transmission electron micrographs with deep learning, 2020, arXiv.org (2020) 2001.10938.

[13] Heindl Meyer, Reconstruction of off-axis electron holograms using a neural net, J. Microsc. 191 (2008) 52–59, http://dx.doi.org/10.1046/j.1365-2818.1998.00343.x.

[14] M. Op de Beeck, D. Van Dyck, W. Coene, Wave function reconstruction in HRTEM: The parabola method, Ultramicroscopy 64 (1996) 167–183, http://dx.doi.org/10.1016/0304-3991(96)00058-7.

[15] A. Thust, W. Coene, M. Op de Beeck, D. Van Dyck, Focal-series reconstruction in HRTEM: simulation studies on non-periodic objects, Ultramicroscopy 64 (1996) 211–230, http://dx.doi.org/10.1016/0304-3991(96)00011-3.

[16] W.-K. Hsieh, F.-R. Chen, J.-J. Kai, A.I. Kirkland, Resolution extension and exit wave reconstruction in complex HREM, Ultramicroscopy 98 (2004-01) 99–114, http://dx.doi.org/10.1016/j.ultramic.2003.08.004.

[17] L.J. Allen, W. McBride, N.L. O'Leary, M.P. Oxley, Exit wave reconstruction at atomic resolution, Ultramicroscopy 100 (2004-07) 91–104, http://dx.doi.org/10.1016/j.ultramic.2004.01.012.

[18] P.C. Tiemeijer, M. Bischoff, B. Freitag, C. Kisielowski, Using a monochromator to improve the resolution in TEM to below 0.5 Å . Part II: application to focal series reconstruction, Ultramicroscopy 118 (2012-07) 35–43, http://dx.doi.org/10.1016/j.ultramic.2012.03.019.

[19] F.R. Chen, C. Kisielowski, D.V. Dyck, 3D reconstruction of nanocrystalline particles from a single projection, Micron 68 (2015-01) 59–65, http://dx.doi.org/10.1016/j.micron.2014.08.009.

[20] F.-R. Chen, et al., Probing atom dynamics of excited Co-Mo-S nanocrystals in 3D, Nature Comm. 12 (2021) 5007, http://dx.doi.org/10.1038/s41467-021-24857-4.

[21] D. Van Dyck, I. Lobato, F.-R. Chen, C. Kisielowski, Do you believe that atoms stay in place when you observe them in HREM? Micron 68 (2015) 158–163, http://dx.doi.org/10.1016/j.micron.2014.09.003.

[22] A.C. Ferrari, et al., Science and technology roadmap for graphene, related two-dimensional crystals, and hybrid systems, Nanoscale 7 (2015-03) 4598–4810, http://dx.doi.org/10.1039/c4nr01600a.

[23] G.R. Bhimanapati, et al., Recent advances in two-dimensional materials beyond graphene, ACS Nano 9 (2015) 11509–11539, http://dx.doi.org/10.1021/acsnano.5b05556.

[24] I. Chorkendorff, J.W. Niemantsverdriet, Concepts of Modern Catalysis, second ed., Wiley-VCH, Weinheim, 2007.

[25] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, in: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, vol. 9351, Springer International Publishing, 2015, pp. 234–241, http://dx.doi.org/10.1007/978-3-319-24574-4_28.

[26] F. Chollet, Deep Learning with Python, Manning, 2018.

[27] https://www.tensorflow.org/.

[28] A.H. Larsen, et al., The atomic simulation environment - a Python library for working with atoms, J. Phys. Condens. Matter. 29 (2017) 273002, http://dx.doi.org/10.1088/1361-648x/aa680e.

[29] S. Haastrup, et al., The computational 2D materials database: high-throughput modeling and discovery of atomically thin crystals, 2D Mater. 5 (2018-10) 042002, http://dx.doi.org/10.1088/2053-1583/aacfc1.

[30] P. Goodman, A.F. Moodie, Numerical evaluations of N-beam wave functions in electron scattering by the multi-slice method, Acta Crystallogr. Sect. A 30 (1974-03) 280–290, http://dx.doi.org/10.1107/s056773947400057x.

[31] E.J. Kirkland, Advanced computing in electron microscopy, 2010, http://dx.doi.org/10.1007/978-1-4419-6533-2.

[32] J. Madsen, T. Susi, The abTEM code: transmission electron microscopy from first principles, Open Res. Europe 1 (2021) 24, http://dx.doi.org/10.12688/openreseurope.13015.2.

[33] E.M. Mannebach, et al., Dynamic structural response and deformations of monolayer MoS2 visualized by femtosecond electron diffraction, Nano Lett. 15 (2015) 6889–6895, http://dx.doi.org/10.1021/acs.nanolett.5b02805.

[34] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017, arXiv.org (2017) 1412.6980.

[35] D. Van Dyck, J.R. Jinschek, F.-R. Chen, ‘Big Bang’ tomography as a new route to atomic-resolution electron tomography, Nature 486 (2012) 243–246, http://dx.doi.org/10.1038/nature11074.

[36] R.W. Gerchberg, W.O. Saxton, A practical algorithm for the determination of phase from image and diffraction plane pictures, Optik 35 (1972) 237–256.

[37] https://gitlab.com/schiotz/NeuralNetwork_HRTEM/-/tree/ExitWave.

8.2 Publications II: Quantifying Noise Limitations of Neural Network Segmentations in High-Resolution Transmission Electron Microscopy.

Matthew Helmi Leth Larsen
William Bang Lomholdt
Cuauhtemoc Nuñez Valencia
Thomas W. Hansen
Jakob Schiøtz

Reprinted from arXiv - submitted to Elsevier *Ultramicroscopy*. arXiv:2302.12629.

Quantifying Noise Limitations of Neural Network Segmentations in High-Resolution Transmission Electron Microscopy.

Matthew Helmi Leth Larsen[a], William Bang Lomholdt[b], Cuauhtemoc Nuñez Valencia[a], Thomas W. Hansen[b], Jakob Schiøtz[a]

[a]*Computational Atomic-scale Materials Design (CAMD), Department of Physics, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

[b]*National Center for Nano Fabrication and Characterization, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

Abstract

Motivated by the need for low electron dose transmission electron microscopy imaging, we report the optimal frame dose (*i.e.* $e^-/\text{Å}^2$) range for object detection and segmentation tasks with neural networks. The MSD-net architecture shows promising abilities over the industry standard U-net architecture in generalising to frame doses below the range included in the training set, for both simulated and experimental images. It also presents a heightened ability to learn from lower dose images. The MSD-net displays mild visibility of a Au nanoparticle at 20-30 $e^-/\text{Å}^2$, and converges at 200 $e^-/\text{Å}^2$ where a full segmentation of the nanoparticle is achieved. Between 30 and 200 $e^-/\text{Å}^2$ object detection applications are still possible. This work also highlights the importance of modelling the modulation transfer function when training with simulated images for applications on images acquired with scintillator based detectors such as the Gatan Oneview camera. A parametric form of the modulation transfer function is applied with varying ranges of parameters, and the effects on low electron dose segmentation is presented.

Keywords: HR-TEM, Machine Learning, Modulation Transfer Function, Signal-to-noise, Beam damage

---

1. Introduction

High-resolution transmission electron microscopy (HR-TEM) is a primary method to characterise materials at the atomic scale, and is a method where an abundance of data can be obtained. HR-TEM can provide a greater temporal resolution, as opposed to scanning transmission electron microscopy, by illuminating the entire sample simultaneously. It does this, however, at the cost of the signal-to-noise ratio (SNR). Increasing the frame dose of the image can assist in increasing the interpretability of each image, but it is usually undesired to do so to avoid electron beam induced effects that are yet to be completely understood [1, 2, 3]. The averaging of many images is another method to increase the SNR. This method, however, lowers the temporal resolution and can be a tedious task that requires complex image alignment to ensure sensible results. Image alignment is also frame dose dependent as the alignment between low SNR images becomes difficult. Addressing SNR related issues has led to a quest in developing denoising methods [4, 5], often using neural networks [6, 7].

The field has seen a steady increase in applying machine learning solutions to solve various tasks of analysing and interpreting data. The importance of including machine learning in a standard workflow for HR-TEM characterisation is highlighted in multiple works [8, 9]. Pipelines for training neural networks for segmentation with hand labelled experimental data have been developed, such as the pipeline by Groschner *et al.* [10], where segmented nanoparticles are classified to acquire sufficient statistics on various classes of nanoparticles. The segmentation of nanoparticles is also useful for tracking dynamic behaviour across frames [11]. Simulated images provide possibilities for large amounts of accurately labelled images for training and have been used to identify and analyse atomic columns in experimental HR-TEM sequences [12, 13]. Vincent *et al.* also used simulated images to train neural networks for denoising [7]. Due to the successes of deep learning powered analysis of HR-TEM images many are seeking to understand the important aspects of optimising the generalisability and applications of deep learning models, and attempt to determine the best models available in comparison to eachother and other thresholding or clustering methods [14, 15].

In this work we look to continue the search for optimal simulated training data and neural networks, by considering the specific task of low SNR segmentation. We will report the optimal frame dose range for reliable segmentations by neural networks and how to control that range by tuning simulations. For this task we focus on the segmentation of metallic nanoparticles, more specifically CeO_2 supported Au nanoparticles. Au nanoparticles serve as a useful system for studying catalytic properties of metallic nanoparticles [16].

Preprint submitted to Elsevier *February 27, 2023*

2. Methods

This work utilises two different network architectures to perform the segmentations. The first is the well recognised U-net [17] architecture used by Madsen *et al.* [12], with the only modification that the skip connections are now concatenations rather than element-wise additions [18]. The other is the MSD-net introduced by Pelt *et al.* [19], which was highlighted to be a robust network against low SNR images. Both networks are convolutional neural networks that differ in their approach to capturing information from varying spatial distances in the image. The U-net's approach gives its characteristic U-shaped architecture due to the numerous down-sampling and up-sampling layers to spread the convolutional kernel over patterns with large spatial coverage. The MSD-net in contrast maintains the same resolution throughout all layers, but dilates the convolutional kernel to spread the weights over a larger spatial region. Specific hyper-parameters regarding the network architectures can be seen in the supplementary information section Appendix A.1.

To train the neural networks, supported nanoparticles are constructed with the Atomic Simulation Environment (ASE) [20] and HR-TEM images are simulated using abTEM [21]. For this work, Au face centred cubic structures of varying sizes are generated so that the [110] crystal direction is aligned with the optical axis, with a slight random tilt off zone axis of up to 3°. At a random layer from the centre of the nanoparticle the [111] direction is truncated, effectively slicing a (111) plane. The exposed face of the FCC structure is attached to a (111) plane of a CeO_2 slab. These systems are randomly rotated about the optical axis, and contain varying sizes of nanoparticles. This replicates realistic interfaces between Au and CeO_2 identified by Liu *et al.* [16]. The image simulation then applies the parameters shown in Table 1. This is a relatively cheap operation, since the expensive part of the image simulation is generating the exitwave by computing the multislice algorithm [22]. Applying imaging imperfections such as the contrast transfer function (CTF) and the modulation transfer function (MTF) can be done multiple times on the same exitwave, which generates multiple images of the same nanoparticle with varying imaging conditions. These images are pre-generated and stored to allow for a controlled comparison between networks on the exact same dataset. Training epochs cycle through the different sets of images of the same atomic system. Each set of images is referred to as an image epoch. With 300 training epochs and 10 image epochs, each image is reused 30 times in training.

Mask labels are binary images separating the Au nanoparticle from the surrounding vacuum and the CeO_2 support. This is generated by computing the convex hull of the atomic coordinates. The neural networks are trained to map a single HR-TEM image to a binary mask label image, which will provide the pixels containing the nanoparticle separate from the pixels containing substrate and vacuum. An example of this is shown in Fig. A.10. Once the network is trained it can be applied to an image, either simulated or experimental, and will return a probability map. Each pixel will be classified to belong to either the nanoparticle or the background (vacuum or substrate) class at some probability. This is referred to as the network inference or prediction and a threshold is applied at 0.9 to generate a binary predicted mask of the pixels that the network classify with at least 90% confidence.

Parameter	L	U	Unit
Acceleration voltage	300		keV
Defocus (Δf)	-200	200	Å
Spherical aberration (C_s)	0	12.45	μm
Focal spread	5	20	Å
Blur	0.1	0.8	Å
Frame dose	10^2	10^6	$e^-/\text{Å}^2$
Resolution	0.07	0.08	Å/pixel

Table 1: Microscope parameters. For each image series, a set of microscope parameters are drawn within the limits given here, except the acceleration voltage which is kept constant. L and U denote the lower and upper limits, respectively. All distributions are uniform, except for the electron dose which is exponential.

Experimental images of a CeO_2 supported Au nanoparticle were acquired on an image corrected FEI Titan 80-300 ECELL at 300 keV, using a Gatan OneView detector. The Au nanoparticle is imaged initially at a frame dose of 10 $e^-/\text{Å}^2$, where noise dominates the images. The doserate is continuously increased to above 1000 $e^-/\text{Å}^2$, where atomic columns are relatively visible. The Au nanoparticle is situated in vacuum at $200°C$, so it is expected that the nanoparticle is relatively inert. We refer to work by Lomholdt *et al.* [23] for experimental details and details of the SNR at varying frame doses.

Our approach here is to utilise the segmentation of the final frame of this continuously increasing doserate series as a pseudo-ground truth to gauge the performance of the network at lower frame doses. Due to drift in the images of the nanoparticle, pixel-wise scores such as the F1-Score will not be used, instead we will measure the area of segmentation, where the ground truth area will be the target.

For scintillating material based detectors, such as the Gatan OneView, the approximation of pure Poissonian noise in HR-TEM images breaks down, since the spectral profile of the noise is altered by the modulation transfer function (MTF) [24, 25]. This function is the Fourier transform of the point spread function, which is an intrinsic property of the scintillating material [26]. Here we study the parametric form from Lee *et al.* [27], shown in Eq. 1.

$$MTF(\tilde{q}) = (1 - C) \cdot \frac{1}{1 + (\frac{\tilde{q}}{c_0})^{c_3}} + C \tag{1}$$

where the spatial frequencies are normalised by the Nyquist frequency, which is related to the sampling, s of the detector by $\tilde{q} = q/q_N = 2 \cdot q \cdot s$. The limits of the function are 1 for $\tilde{q} \to 0$ due to a normalisation and C for $q \to \infty$. The function is fitted following the noise method described in Ref. [26].

3. Results

The following section will present results presenting how to achieve reliable and robust low dose segmentations of HR-TEM images. This will be divided into firstly a comparison of two neural network architectures and their ability to differentiate signal from noise. The neural networks will be trained on simulated data with different electron dose ranges to gauge their abilities to learn from training datasets of varying difficulty. After this a detailed analysis on how to tune performance by optimising the MTF will be presented.

3.1. Dataset Dose Limits and Neural Network Comparison

An obvious first step in approaching segmentation of low SNR data is to investigate the performance of various architectures. Here we use the experimental image series to benchmark the U-net against the MSD-net and identify the best neural network for low SNR HR-TEM segmentation. We test the neural networks abilities to train on two different ranges of frame dose presented in Table 2. The low frame dose range covers the range of the experimental data series in this work, where-as the high frame dose range feeds the network much clearer simulated images, making it an easier dataset to learn.

Parameter	L	U	Unit
High frame dose range	10^2	10^6	$e^-/\text{Å}^2$
Low frame dose range	10^1	10^4	$e^-/\text{Å}^2$

Table 2: Frame dose ranges for simulated images.

Comparisons will be made by plotting the segmented area versus frame dose to identify when each network begins to detect the nanoparticle and when the segmented area converges *i.e.* the network identifies the entire nanoparticle. The final frame segmentation should be understood as a a pseudo-ground truth. The aim is to segment a similar area as in the ground truth at as low frame dose as possible. The area may not be exact due to possible morphological changes in the nanoparticle. The morphological changes and slight drift disallows the use of, for example, an F1-Score against the given frame and the ground truth for the experimental data series, since the overlap of the segmentations will not be sensible at a pixel-wise level. The ideal case of these plots will be a step function, meaning the entire nanoparticle is immediately identified, at some minimal frame dose.

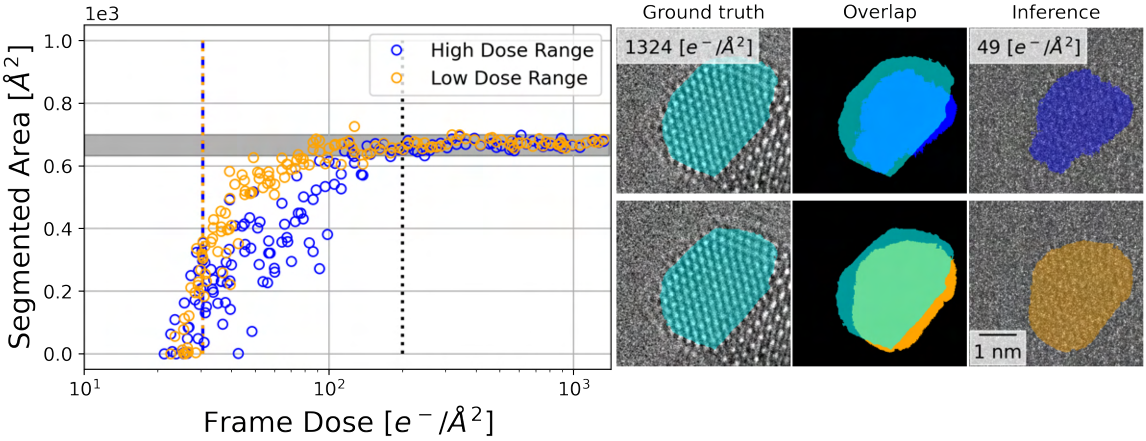
The MSD-net presents an ability to generalise outside of the training data range, whereas the U-net only performs within the training data range. Training the MSD-net on the low frame dose range increases the visibility of the nanoparticle between 40-100 $e^-/\text{Å}^2$ by ~50%, as seen in Fig. 1(a), however it is also a noteworthy feature that the MSD-net is able to segment significant regions below the lower limit of the high frame dose range when trained on the high frame dose range. This proves a superior ability to separate noise to signal and generalise beyond the limits of the training set, which means the MSD-net is a better candidate for when data is limited. Fig. 1(b) visualises the improvement in low dose segmentation by overlapping the ground truth segmentation (cyan coloured mask), with the segmentation from the two networks trained on each dose range (colour coded mask). The MSD-net trained on both dose ranges show a lower limit at 20-30 $e^-/\text{Å}^2$.

Predictions from the U-net are not as robust as the MSD-net. Fig. 2(b) shows that the U-net trained on the low frame dose range is not entirely reliable due to the difficulties in defining the edges of the nanoparticle. Comparing the low dose trained MSD-net segmentation at 49 $e^-/\text{Å}^2$ in Fig. 1(b) to the low dose trained U-net at 49 $e^-/\text{Å}^2$ in Fig. 2(b), the segmented area is similar however the edges are more well defined in the segmentation from the MSD-net.

The low dose trained U-net achieves some visibility a few $e^-/\text{Å}^2$ below the MSD-net, as seen in Fig. 2, however the over segmentation at higher dose highlights that this network is possibly being triggered by noise as well. In Fig. A.11 the F1-Score as a function of training epochs with simulated data highlight that the low dose range dataset becomes a harder problem for both neural networks to learn and generalise. Both networks show greater signs of overfitting. The MSD-net seems to handle more difficult problems better than the U-net. Finally Fig. 3 compares the two networks trained on the high frame dose range data, showcasing the performance gain in the low frame dose regime with the MSD-net segmenting 50% of the nanoparticle at a ~70% lower frame dose, and both show a reliable convergence at 200 $e^-/\text{Å}^2$.

3.2. Frame dose limit

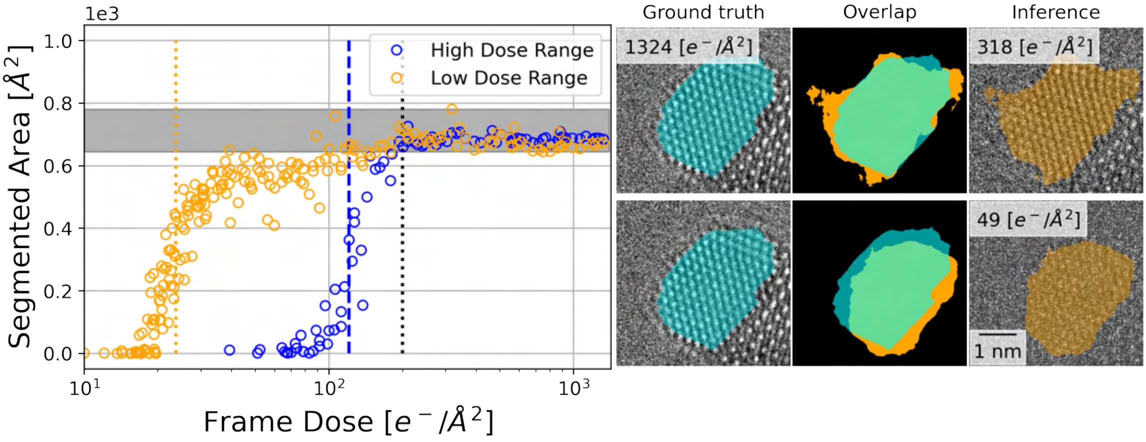
Low dose segmentation performance relies on a proper modelling of the noise in the simulated data. The following will described how noise characteristics are extracted from experimental data and modelled in simulated data. Eq. 1 was fitted to the azimuthally averaged centred Fourier transform of the vacuum region in the images, as done in [26, 27]. See Fig. A.12 and A.13 for the fitted MTF of the first frame and last frame of the experimental image series, respectively. Fig. 4 presents a distribution of fitted parameters dependent on the frame dose of the experimental images. The parameter points with white dots are the fits with an $R^2 \geq 0.98$.



(a)

(b)

Figure 1: Comparison of the MSD-net trained with simulated images within the high frame dose range to simulated images within the low frame dose range (see Table 2). (a) Segmented area of each model as a function of frame dose. Black dashed line represents the convergence of the area of segmentation, symbolising where the network segmented the entire nanoparticle. The minimum and maximum area beyond this point form the shaded grey bar as a visual aid for the target area of segmentation. Colour coded dashed lines for each model is shown representing the frame dose at which the model achieves 50% segmentation of the nanoparticle. (b) Colour coded examples of the segmentation at 49 $e^-/\text{Å}^2$ overlapped with the segmentation of the final frame (ground truth). This highlights the ability to achieve low dose segmentation.



(a)

(b)

Figure 2: Comparison of the U-net trained with simulated images within the high frame dose range to simulated images within the low frame dose range (see Table 2). (a) Segmented area of each model as a function of frame dose. Black dashed line represents the convergence of the area of segmentation, symbolising where the network segmented the entire nanoparticle. The minimum and maximum area beyond this point from the shaded grey bar as a visual aid for the target area of segmentation. Colour coded dashed lines for each model is shown representing the frame dose at which the model achieves 50% segmentation of the nanoparticle. (b) Top: maximum segmentation overlapped with the segmentation of the final frame (ground truth). This highlights any over-segmented areas, due to difficulties in defining the boundaries of the nanoparticle. Bottom: Segmentation at 49 $e^-/\text{Å}^2$ overlapped with the segmentation of the final frame (ground truth). This highlights the ability to achieve low dose segmentation.

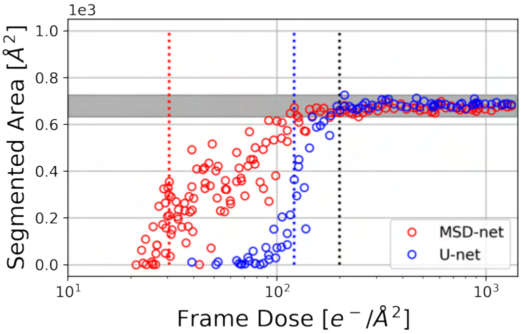


Figure 3: Comparison of the MSD-net's and U-net's ability to learn to distinguish between signal and noise. The plot shows the segmented area of each model as a function of frame dose. Black dashed line represents the convergence of the area of segmentation, symbolising where the network segmented the entire nanoparticle. The minimum and maximum area beyond this point from the shaded grey bar as a visual aid for the target area of segmentation. Colour coded dashed lines for each model is shown representing the frame dose at which the model achieves 50% segmentation of the nanoparticle.

It is vital to understand the role of each parameter in Eq. 1, to interpret the distributions in Fig. 4. The c_0 parameter represents the $\tilde{q}$ value of the half-maximum, which increases with frame dose and converges around 200 $e^-/\text{Å}^2$. This means that at frame doses above 200 $e^-/\text{Å}^2$ spatial frequencies up to $\sim 0.27 \cdot q_N$ are maintained at at least half maximum, but for lower doses the function is narrower. As a result spatial frequencies above $c_0 \cdot q_N$ are effectively filtered out at higher frame doses. The curvature of the function is consistently close to a Lorentzian form, as revealed by the c_3 parameter (higher values approach a low pass step function). The C parameter reveals the change in the tail of the function *i.e.* the $q \to \infty$ limit. C is non-zero at lower frame dose and approaches 0 at higher doses. We interpret this as a fraction of the noise that is not subject to the point spread function of the scintillator, but is generated later in the detection process. We label this part of the noise "readout noise" although it may come from more than one source.

All parameters show most variation below 200 $e^-/\text{Å}^2$. The variation below this limit is likely due to the transition between readout noise and shot noise as the dominating noise source [28].

The readout noise appears after the scintillating material and is therefore not affected by the MTF. The contributions of the readout noise and shot noise are modelled separately in our simulations and in order to extract the fractional contribution of each noise source we look at the C dependency of the frame dose in electrons per pixel. This value represents the noise floor in the image. At higher frame dose this noise floor is washed out by shot noise, but at lower dose the readout noise dominates. The shot noise is modelled as a Poisson distribution, $P(\lambda = N_D)$, where N_D is the frame dose in electrons per pixel. The read out noise is also modelled as a Poisson distribution, $P(\lambda = N_0)$, where N_0 is a constant noise floor. The total noise is a sum of the two Poisson distributions

$$P(\lambda = N_D) + P(\lambda = N_0) = P(\lambda = (N_D + N_0)) \quad (2)$$

meaning that each pixel intensity, $I_{x,y}$ in the final image is within the distribution

$$I_{x,y} = N_D + N_0 \pm \sqrt{N_D + N_0}. \quad (3)$$

In Fig. 5, we extract N_0 from the fractional contribution of the readout noise deviation to the total noise deviation.

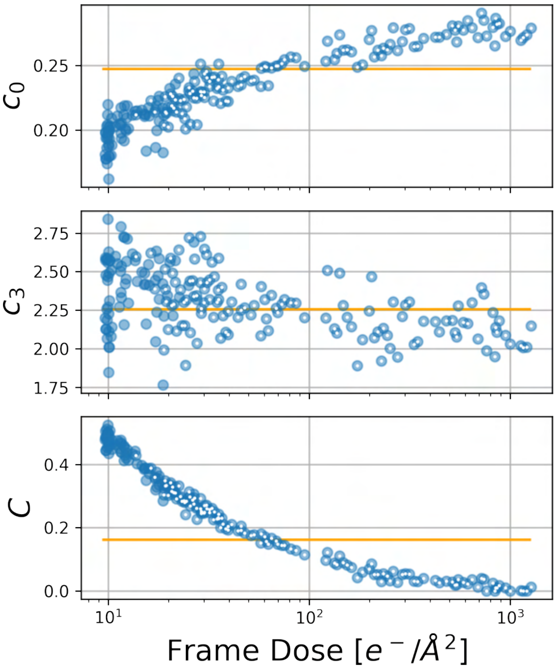


Figure 4: The fitted parameters of Eq. 1 for each frame in the data series. Fits with $R^2 \geq 0.98$ are marked by a white dot. The orange line represents the mean value.

The approach we have taken to model the variations of c_0 and c_3 is to randomise the parameters within a given range as done by Madsen et al. [12], which alters the spectral profile of the shot noise. The extracted N_0 is varied at ±50%, *i.e.* $N_0 \in [0.005, 0.015]$ when applied to simulated images. In Fig. A.14, we show that with the fitted N_0 applied as a separate Poissonian noise source, we are able to replicate the C parameter dependency on the frame dose from MTFs fitted to a simulated image series of vacuum at increasing frame dose. The parameters c_0 and c_3 show random values within the given range as expected.

Table 3 summarises 5 different MTF parameter ranges applied to 5 simulated datasets of identical atomic systems. All apply the microscope parameters from Table 1. An identical MSD-net is trained on each dataset. The

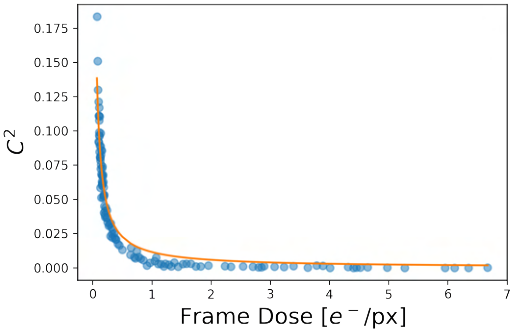


Figure 5: The correlation of the C parameter of Eq. 1 to the frame dose as electrons per pixel. The fraction of readout noise, N_0 is extracted from its fractional contribution to the total noise. The orange lines shows the fitted curve with $R^2 = 0.94$ and $N_0 = 0.01$.

first model, MTF 1, will take the parameters from Ref. [12]. Motivations behind the other MTF models will be explained as the results are discussed.

MTF	c_0		c_3		N_0	
	L	U	L	U	L	U
1	0.4	0.6	2.0	3.0	0.005	0.015
2	0.4	0.6	1.0	5.0	0.005	0.015
3	0.2	0.3	1.0	5.0	0.005	0.015
4	0.25	0.6	2.0	5.0	0.005	0.015
5	0.6	0.8	2.0	5.0	0.005	0.015

Table 3: The various MTF models studied in the work. 1-5 represents different ranges of parameters in Eq. 1.

Higher values of c_0 in MTF 1 and MTF 2 assist the network in low frame dose segmentation. Fig. 6(a) presents the visibility of the entire nanoparticle as a function of frame dose from 3 MSD-net models trained on MTF models 1-3 individually. MTF 1 and MTF 2 show very similar performance, which highlights that it is not important that the form of the MTF is exactly Lorentzian. This was determined by ranging c_3 over a larger range in MTF 2. Following this result, MTF 3 samples a range of c_0 more suited to the fitted ranges in Fig. 4. The result of this range were detrimental on the low dose performance, delaying the visibility of the nanoparticle by ∼25% in electron dose.

Applying the range of c_0 within the experimental fit by MTF 3 provide more refined segmentations at higher frame dose. The variations in the segmented area at higher doses for the MTF 3 trained model are smaller than that of the MTF 1 and 2. The variations of MTF 1 and MTF 2 are shown by the shaded grey region. These variations can be due to difficulties in defining the boundaries between the nanoparticle and substrate/vacuum in the images. Fig. 6(b) shows the maximum segmented area of MTF 2 (top) and MTF 3 (bottom). Here it is seen that the MTF 2 trained model has minor difficulties in defining the border of the nanoparticles.

We speculate that preserving spatial frequencies up to around half-Nyquist is preferential for the networks to learn to differentiate signal from noise. In practice this means having a c_0 that ranges up to or slightly above $0.5 \cdot q_N$. Values of c_0 within the fitted range from experimental data is however also necessary for refining boundaries of the segmented areas.

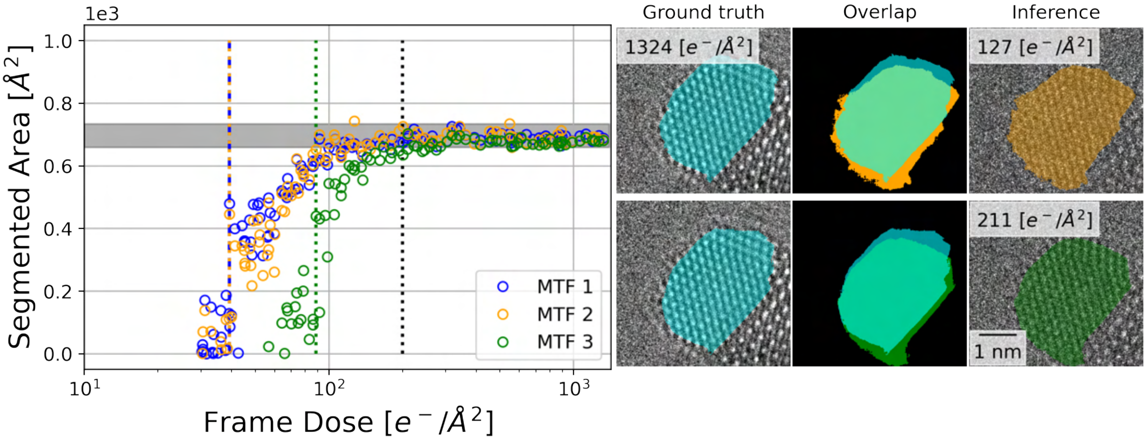
MTF 4 ranges between the lower limit of the fitted c_0 from MTF 3 and the upper limits of c_0 from MTF 1 and 2, and achieves the low dose performance of MTF 1 and 2 and the refined boundaries of MTF 3 at higher dose. MTF 5 samples larger c_0, beyond half-Nyquist, with the same range as MTF 1 and 2. Fig. 7(a) presents the performance of MTF models 2,4, and 5. MTF 5 shows much more visibility at low dose frames, however much more sporadic variations in the higher frame dose regime. Fig. 7(b) shows the maximum segmented area of MTF 5, which highlights its weakness in identifying the borders of the nanoparticle; The segmentation bleeds into the surrounding vacuum. This renders the low dose segmentations of MTF 5 unreliable, as it seems it is being triggered by noise in the vacuum. MTF 5 has such a high c_0 range that it approaches a noise profile appropriate for direct electron counting detectors such as the Gatan K2/3 camera [25]. The maximum segmentation of MTF 4 in contrast shows very sharp separations between the nanoparticle and its surroundings. This proves that ranging c_0 such that it covers the range fitted from experimental data, but also such that it retains spatial frequencies up to half-Nyquist is ideal for optimal segmentation performance across the entire frame dose range.

We note that in all cases the area of segmentation converges at approximately 200 $e^-/\text{Å}^2$, depicted by the dashed black line. This is the first reported lower limit of frame dose for a reliable full segmentation.

The overlap in the middle of Figs. 6(b) and 7(b) shows the morphological changes in the nanoparticle but also highlight that both are segmentations of the same nanoparticle. Here we also highlight that the segmentation is sensible and we show that this is at a level where a human interpreter would have difficulties being certain of the presence of the entire nanoparticle.

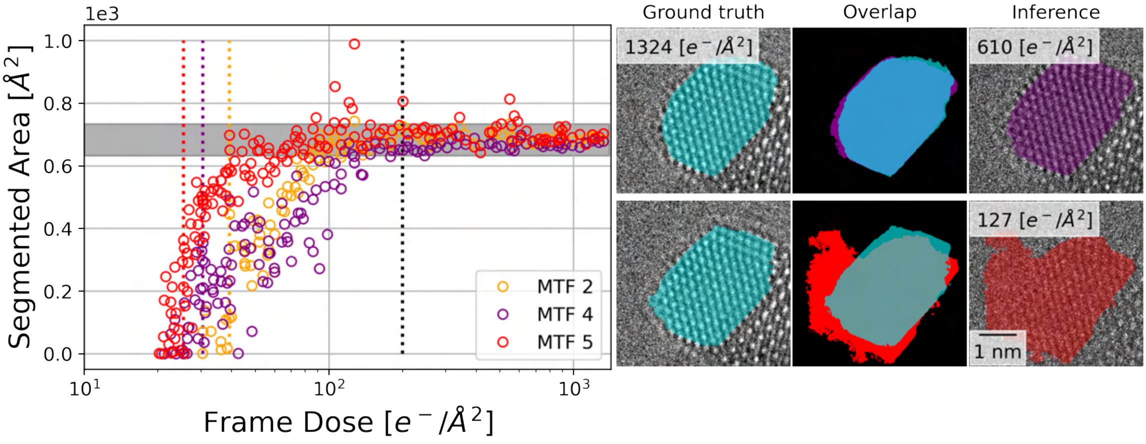
The segmentations below 200 $e^-/\text{Å}^2$ are not full segmentations and cannot be used to for example measure the area, but can still be used for object detection purposes and regional Fourier transform extraction. For these purposes 50% of segmentation would be a sensible minimum, represented by the colour coded dashed lines in Figs. 1(a), 2(a), 6(a), and 7(a).

To further prove the performance of the MTF 4 model, we show that an MSD-net trained with the MTF 4 model performs better on a simulated dataset with the MTF 3 model applied, compared to an MSD-net trained with the MTF 3 model. An image series was simulated to repli-



(a) (b)

Figure 6: Comparison of the first 3 MTF models presented in Table 3. (a) Segmented area of each model as a function of frame dose. Black dashed line represents the convergence of the area of segmentation, symbolising where the network segmented the entire nanoparticle. The minimum and maximum area beyond this point from the shaded grey bar as a visual aid for the target area of segmentation. Colour coded dashed lines for each model is shown representing the frame dose at which the model achieves 50% segmentation of the nanoparticle. MTF 2 shows the best performance (most nanoparticle visibility at the lowest frame dose). (b) Colour coded examples of the maximum segmentation overlapped with the segmentation of the final frame (ground truth). This highlights any over-segmented areas, due to difficulties in defining the boundaries of the nanoparticle.



(a) (b)

Figure 7: Comparison of the last 3 MTF models presented in Table 3. (a) Segmented area of each model as a function of frame dose. Black dashed line represents the convergence of the area of segmentation, symbolising where the network segmented the entire nanoparticle. The minimum and maximum area beyond this point from the shaded grey bar as a visual aid for the target area of segmentation. Colour coded dashed lines for each model is shown representing the frame dose at which the model achieves 50% segmentation of the nanoparticle. MTF 4 shows the best performance (most nanoparticle visibility at the lowest frame dose and a tight convergence of the segmented area). (b) Colour coded examples of the maximum segmentation overlapped with the segmentation of the final frame (ground truth). This highlights any over-segmented areas, due to difficulties in defining the boundaries of the nanoparticle.

cate the experimental series. This simulated series contains 1000 images of a CeO_2 supported Au nanoparticle in the [110] zone axis positioned similarly to the experimental sample, with a blur range of [0.1, 0.2] Å, focal spread range of [5, 6] Å, defocus range of [48, 50] Å, C_s range of [7, 8] μm, a frame dose from [10^1, 10^4] $e^-/\text{Å}^2$, and the MTF 3 model. In this case we have exact ground truths and the F1-Score will be used as a metric. Fig. A.15a presents a histogram of the F1-Score for each image in the simulated series for both networks in mention. It is immediately apparent that the MTF 4 dataset trains a network that outperforms the MTF 3 trained network on images with the MTF 3 ranges applied. Fig. A.15b displays the F1-Score against frame dose, which highlights the improved performance of the MTF 4 dataset trained MSD-net is primarily on low frame dose simulated images, with a higher mean F1-Score in the range of 10-100 $e^-/\text{Å}^2$.

4. Conclusions

In this work we have investigated the quantitative limit of low SNR HR-TEM image segmentation. A continuously increasing frame dose HR-TEM image series of a CeO_2 supported Au nanoparticle was acquired to rate the performance of neural network segmentations at low frame dose and compare the segmentation to the final (highest frame dose) frame.

The results show that the neural networks are achieving human level performance, which means it can safely be equipped for large-scale automated analysis to relieve the human operator of repetitive tasks.

The MSD-net showed promising low SNR performance compared to the industry standard U-net. The MSD-net showed an ability to generalise outside the frame dose range of the training set and provide reliable segmentations, with well defined boundaries down to 20 $e^-/\text{Å}^2$. The U-net was only able to operate within the given training range, and when trained with lower dose images, presented weaknesses in defining boundaries between the nanoparticle and its surroundings. This highlights the MSD-net's superior ability to learn the differences between signal and noise.

A parametric form of the MTF was fitted to all frames in the HR-TEM series resulting in a frame dose dependent range of parameters. The noise contributions were separated into two sources, shot noise and readout noise, and the fractional contribution of each was extracted from the noise floor given by the C parameter of Eq. 1. It was shown that modelling the MTF to retain spatial frequencies up to $\sim 0.5 \cdot q_N$ with at least half-maximum assisted the MSD-net in detecting the nanoparticle at lower frame dose. It seems that modelling the MTF with a wide range of parameters that both consist of parameters within the fitted range and parameters that maintain spatial frequencies up to half-Nyquist allows the MSD-net to operate in both the shot noise and readout noise dominated regimes.

All results converged at 200 $e^-/\text{Å}^2$. This is the first report of a frame dose limit for reliable neural network segmentations. Neural network predictions with HR-TEM images between 20-100 $e^-/\text{Å}^2$ are still useful for object detection purposes.

Knowledge of these frame dose limits will provide the community with realistic expectations from deep learning models and an ability to design experiments that are optimised for their needs from deep learning powered analysis tools, whether it be object detection in live low dose imaging, or statistical data accumulating of morphological properties, amongst others.

Funding

The authors acknowledge financial support from the Independent Research Fund Denmark (DFF-FTP) through grant no. 9041-00161B.

Availability of data and materials

Competing interests

The authors declare that they have no competing interests.

References

[1] P. E. Batson, Motion of Gold Atoms on Carbon in the Aberration-Corrected STEM, Microscopy and Microanalysis 14 (1) (2008) 89–97. `doi:10.1017/s1431927608080197`.

[2] D. V. Dyck, I. Lobato, F.-R. Chen, C. Kisielowski, Do you believe that atoms stay in place when you observe them in HREM?, Micron 68 (2015) 158–163. `doi:10.1016/j.micron.2014.09.003`.

[3] R. F. Egerton, Radiation damage to organic and inorganic specimens in the TEM, Micron 119 (November 2018) (2019) 72–87. `doi:10.1016/j.micron.2019.01.005`.
URL `https://doi.org/10.1016/j.micron.2019.01.005`

[4] H. Du, A nonlinear filtering algorithm for denoising HR(S)TEM micrographs, Ultramicroscopy 151 (2015) 62–67. `doi:10.1016/j.ultramic.2014.11.012`.
URL `http://dx.doi.org/10.1016/j.ultramic.2014.11.012`

[5] T. Furnival, R. K. Leary, P. A. Midgley, Denoising time-resolved microscopy image sequences with singular value thresholding, Ultramicroscopy 178 (2017) 112–124. `doi:10.1016/j.ultramic.2016.05.005`.
URL `http://dx.doi.org/10.1016/j.ultramic.2016.05.005`

[6] R. Lin, R. Zhang, C. Wang, X.-Q. Yang, H. L. Xin, TEMImageNet training library and AtomSegNet deep-learning models for high-precision atom segmentation, localization, denoising, and deblurring of atomic-resolution images, Scientific Reports 11 (1) (2021) 5386. `doi:10.1038/s41598-021-84499-w`.

[7] J. L. Vincent, R. Manzorro, S. Mohan, B. Tang, D. Y. Sheth, E. P. Simoncelli, D. S. Matteson, C. Fernandez-Granda, P. A. Crozier, Developing and Evaluating Deep Neural Network-Based Denoising for Nanoparticle TEM Images with Ultra-Low Signal-to-Noise, Microscopy and Microanalysis 27 (6) (2021) 1431–1447. `doi:10.1017/S1431927621012678`.

[8] S. R. Spurgeon, C. Ophus, L. Jones, A. Petford-Long, S. V. Kalinin, M. J. Olszta, R. E. Dunin-Borkowski, N. Salmon, K. Hattar, W. C. D. Yang, R. Sharma, Y. Du, A. Chiaramonti, H. Zheng, E. C. Buck, L. Kovarik, R. L. Penn, D. Li, X. Zhang,

M. Murayama, M. L. Taheri, Towards data-driven next-generation transmission electron microscopy, Nature Materials 20 (3) (2021) 274–279. doi:10.1038/s41563-020-00833-z.
URL http://dx.doi.org/10.1038/s41563-020-00833-z

[9] K. P. Treder, C. Huang, J. S. Kim, A. I. Kirkland, Applications of deep learning in electron microscopy, Microscopy 71 (August 2021) (2022) I100–I115. doi:10.1093/jmicro/dfab043.

[10] C. K. Groschner, C. Choi, M. C. Scott, Machine Learning Pipeline for Segmentation and Defect Identification from High-Resolution Transmission Electron Microscopy Data, Microscopy and Microanalysis 27 (3) (2021) 549–556. doi:10.1017/S1431927621000386.

[11] K. Faraz, T. Grenier, C. Ducottet, T. Epicier, Deep learning detection of nanoparticles and multiple object tracking of their dynamic evolution during in situ ETEM studies, Scientific Reports 12 (1) (2022) 1–15. doi:10.1038/s41598-022-06308-2.
URL https://doi.org/10.1038/s41598-022-06308-2

[12] J. Madsen, P. Liu, J. Kling, J. B. Wagner, T. W. Hansen, O. Winther, J. Schiøtz, A Deep Learning Approach to Identify Local Structures in Atomic-Resolution Transmission Electron Microscopy Images, Advanced Theory and Simulations 1 (8) (2018) 1–12. doi:10.1002/adts.201800037.

[13] M. Ragone, V. Yurkiv, B. Song, A. Ramsubramanian, R. Shahbazian-Yassar, F. Mashayek, Atomic column heights detection in metallic nanoparticles using deep convolutional learning, Computational Materials Science 180 (December 2019) (2020) 109722. doi:10.1016/j.commatsci.2020.109722.
URL https://doi.org/10.1016/j.commatsci.2020.109722

[14] J. P. Horwath, D. N. Zakharov, R. Mégret, E. A. Stach, Understanding important features of deep learning models for segmentation of high-resolution transmission electron microscopy images, npj Computational Materials 6 (1) (2020) 1–9. doi:10.1038/s41524-020-00363-x.
URL http://dx.doi.org/10.1038/s41524-020-00363-x

[15] K. M. Saaim, S. K. Afridi, M. Nisar, S. Islam, In search of best automated model: Explaining nanoparticle TEM image segmentation, Ultramicroscopy 233 (December 2021) (2022) 113437. doi:10.1016/j.ultramic.2021.113437.
URL https://doi.org/10.1016/j.ultramic.2021.113437

[16] P. Liu, T. Wu, J. Madsen, J. Schiøtz, J. B. Wagner, T. W. Hansen, Transformations of supported gold nanoparticles observed by: In situ electron microscopy, Nanoscale 11 (24) (2019) 11885–11891. doi:10.1039/c9nr02731a.

[17] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, Vol. 9351 of Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, 2015, pp. 234 – 241. doi:10.1007/978-3-319-24574-4_28.

[18] M. H. Leth Larsen, F. Dahl, L. P. Hansen, B. Barton, C. Kisielowski, S. Helveg, O. Winther, T. W. Hansen, J. Schiøtz, Reconstructing the exit wave of 2D materials in high-resolution transmission electron microscopy using machine learning, Ultramicroscopy 243 (November 2022) (2022) 113641. doi:10.1016/j.ultramic.2022.113641.
URL https://doi.org/10.1016/j.ultramic.2022.113641

[19] D. M. Pelt, J. A. Sethian, A mixed-scale dense convolutional neural network for image analysis, Proceedings of the National Academy of Sciences of the United States of America 115 (2) (2017) 254–259. doi:10.1073/pnas.1715832114.

[20] A. Hjorth Larsen, J. JØrgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Dułak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. SchiØtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, K. W. Jacobsen, The atomic simulation environment - A Python library for working with atoms, Journal of Physics Condensed Matter 29 (27) (2017). doi:10.1088/1361-648X/aa680e.

[21] J. Madsen, T. Susi, abTEM: ab Initio Transmission Electron Microscopy Image Simulation , Microscopy and Microanalysis 26 (S2) (2020) 448–450. doi:10.1017/s1431927620014701.

[22] E. J. Kirkland, Advanced Computing in Electron Microscopy, in: Advanced Computing in Electron Microscopy, 3rd Edition, Springer Cham, 2020, pp. 143–195. doi:10.1007/978-3-030-33260-0.
URL https://link.springer.com/book/10.1007/978-3-030-33260-0

[23] W. B. Lomholdt, [In Preparation] Signal-to-Noise Ratio Metrics for Quantifying High-Resolution Transmission Electron Microscopy Image Interpretability.

[24] G. McMullan, A. R. Faruqi, D. Clare, R. Henderson, Comparison of optimal performance at 300keV of three direct electron detectors for use in low dose electron microscopy, Ultramicroscopy 147 (2014) 156–163. doi:10.1016/j.ultramic.2014.08.002.
URL http://dx.doi.org/10.1016/j.ultramic.2014.08.002

[25] A. R. Faruqi, G. McMullan, Direct imaging detectors for electron microscopy, Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment 878 (May 2017) (2018) 180–190. doi:10.1016/j.nima.2017.07.037.
URL http://dx.doi.org/10.1016/j.nima.2017.07.037

[26] M. Vulovic, B. Rieger, L. J. Van Vliet, A. J. Koster, R. B. Ravelli, A toolkit for the characterization of CCD cameras for transmission electron microscopy, Acta Crystallographica Section D: Biological Crystallography 66 (1) (2010) 97–109. doi:10.1107/S0907444909031205.

[27] Z. Lee, H. Rose, O. Lehtinen, J. Biskupek, U. Kaiser, Electron dose dependence of signal-to-noise ratio, atom contrast and resolution in transmission electron microscope images, Ultramicroscopy 145 (2014) 3–12. doi:10.1016/j.ultramic.2014.01.010.
URL http://dx.doi.org/10.1016/j.ultramic.2014.01.010

[28] W. J. De Ruijter, J. K. Weiss, Methods to measure properties of slow-scan CCD cameras for electron detection, Review of Scientific Instruments 63 (10) (1992) 4314–4321. doi:10.1063/1.1143730.

Appendix A. Supplementary online information

Appendix A.1. Network architecture

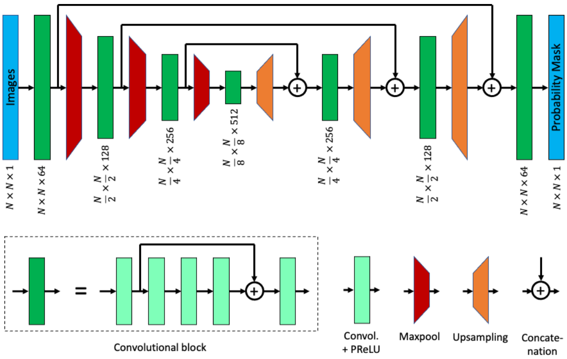


Figure A.8: The architecture of the U-net. Information flows from left to right. The first part of the network, the "encoding path", consists of convolutional processing blocks alternated by downsampling layers using the MaxPool method. The second part, the "decoding path", consists of convolutional blocks alternated by bilinear upsampling layers. Long skip connections ensures that the original spatial information can be maintained. Adapted from Madsen *et al.* [12]

The U-net architecture is identical to Ref. [18], with a difference in the initial number of channels. From hyperparameter tuning it was seen that increasing the channels minimised overfitting and provided more robust results for segmentations. The network consists of a downsampling (or "encoding")) path, where convolutional blocks alternate with downsampling layers, and an upsampling (or "decoding") path, where the convolutional blocks alternate with upsampling layers (see Fig. A.8). The convolutional blocks consist of 5 convolutional layers, with a short skip connection between the output of the first layer and the input of the fifth.

The downsampling is done with conventional MaxPool operations. Each time the resolution is cut in half in a MaxPool operation, the number of feature channels in the following convolutional block is doubled to maintain the information flow in the network. The upsampling is done using bilinear interpolation, and the following convolution block has the number of channels cut by a factor two. After each upsampling, information from the last layer with the same spatial resolution is added from the downsampling path, this is done by concatenating the channels. The first layer in the convolutional blocks in both paths will therefore have a different number of input channels from what is stated in the figure.

Each convolutional layer uses a 5×5 convolutional kernel, followed by a Parametric Leaky Rectifying Linear Unit. Hyperparameter tuning showed no significant improvement in increasing the kernel size above 5×5.

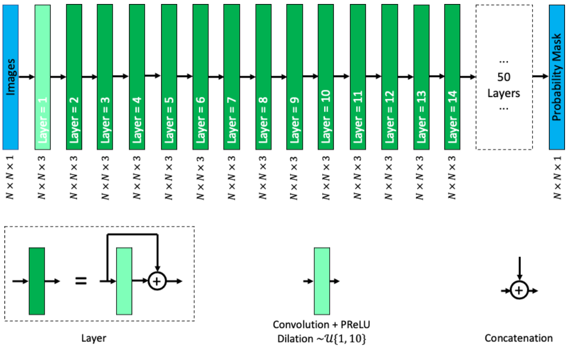


Figure A.9: The architecture of the MSD-net. The convolutional kernel dilation value is sampled from a uniform distribution in the range [1,10]. Information flows from left to right.

The MSD-net architecture shown in Fig. A.9, is an adaptation of the model introduced by Pelt *et al.* in Ref. [19]. The network consists of 50 layers, where each layer performs a convolution and immediately concatenates the output with the output of the previous layer. The first layer is an exception as it does not concatenate with the input image.

The reoccurring successive concatenations is what classifies this network architecture in the family of dense network architectures where all convolutional output filters are connected with eachother. This MSD-net implementation utilises a 9×9 convolutional kernel that is dilated to capture features at varying length scales. The dilation value is sampled from a uniform distribution in the range $[1, 10]$, and seeding is provided to reproduce multiple identical MSD-net models. This is what "Mixed-Scale" refers to in the name of the architecture.

Hyperparameter tuning showed signifcant improvements when increasing the number of output channels, kernel size, and number of layers, however Tensorflow's implementation of concatenation layers consumes large amount of memory, and so parameters of this architecture are limited by available memory.

All convolutional layers apply reflection padding to maintain dimensions.

Appendix A.2. Network training

The problem definition here, as presentsed in Figures A.8 and A.9, is to take a HR-TEM image and produce a probability mask of the pixels corresponding to the nanoparticle. An example of this is shown in Fig. A.10.

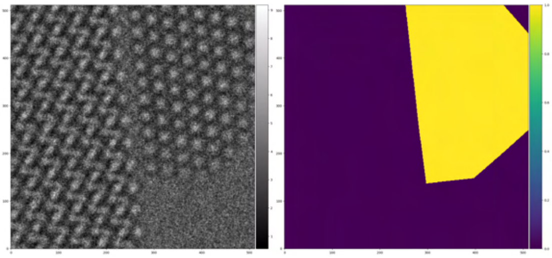
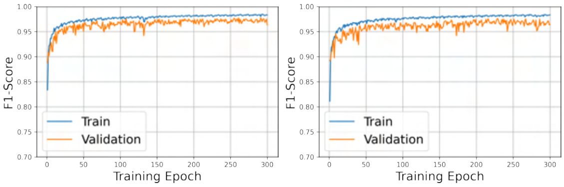


Figure A.10: Example of simulated image and its binary mark label.

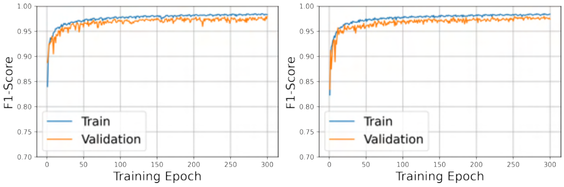
Binary masks are used so that the binary cross entropy loss (or its multi-class counterpart: the categorical cross entropy loss) can be applied for training. This is a loss function widely applied for segmentation problems and is natively implemented in Tensorflow.

Fig. A.11 shows the learning curves for all networks applied in this work. The F1-Score is shown instead of the binary cross entropy loss, as it has a more intuitive range of values, 1 being a pixel wise perfect segmentation and 0 the worst result possible. The F1-Score is a harmonic mean between the precision and recall, and properly gauges the ability to accurately classify both positives and negatives, *i.e.* identify both nanoparticle and background.

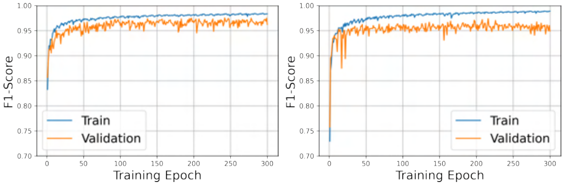
Training is executed using the ADAM gradient optimiser with the AMSGrad variant activated, and a learning rate of 0.001. Many training epochs in an attempt to obtain a well converged network, which is typically the most robust when generalising to experimental data.



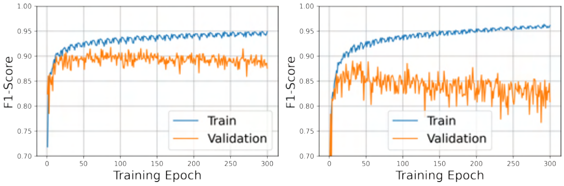
(a) MSD-net MTF 1 (b) MSD-net MTF 2



(c) MSD-net MTF 3 (d) MSD-net MTF 4



(e) MSD-net MTF 5 (f) U-net MTF 4



(g) MSD-net MTF 4 low dose range (h) U-net MTF 4 low dose range

Figure A.11: F1-Score learning curves for all networks applied in this work.

Appendix A.3. Noise Fitting

The power spectral density is defined as the magnitude squared of the Fourier transform. This is taken from a vacuum region of the HR-TEM image for every frame. This 2D-spectrum is averaged azimuthally to obtain a 1D-spectrum, as done in Ref. [26]. Frequencies are normalised by the Nyquist frequency, *i.e.* $\tilde{q} = q/q_N$. As shown in Figures A.12 and A.13, the spatial frequency filtering affect of the MTF is clear. The raw spectrum

$$MTF(\tilde{q}) = (c_1 - c_2) \cdot \frac{1}{1 + (\frac{\tilde{q}}{c_0})^{c_3}} + c_2 \qquad \text{(A.1)}$$

is fitted in the top of the two figures. The bottom of the two figures presents the fitted MTF normalised by the $q \to 0$ limit, c_1, *i.e.* Eq. 1.

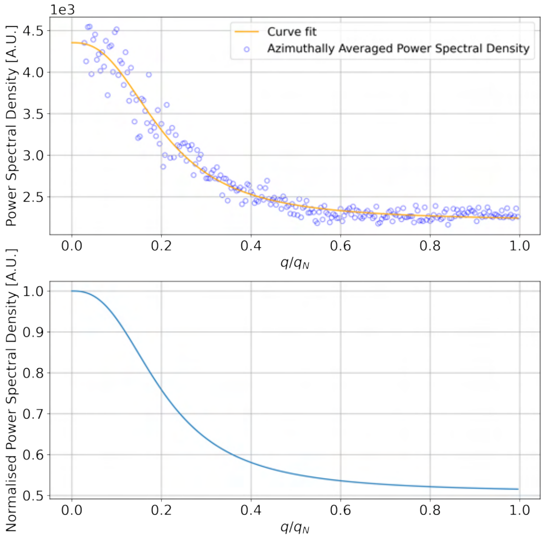


Figure A.12: MTF fitted from the power spectral density of the first frame at approximately 10 $e^-/\text{Å}^2$. Top: Raw fit (Eq. A.1) with $R^2 = 0.99$, $c_0 = 0.20$, $c_1 = 4353.29$, $c_2 = 2207.83$, $c_3 = 2.57$. Bottom: Normalised MTF (Eq. 1) with $c_0 = 0.20$, $c_3 = 2.57$, $C = 0.51$.

In Fig. 5, the readout noise contribution, N_0, is fitted from its fractional contribution the the total noise floor as a function of electron dose per pixel. A simulation of vacuum was made, where c_0 and c_3 were varied within a range of $[0.2, 0.3]$ and $[2, 3]$ respectively, and an addition Poisson noise is added with $N_0 \pm 50\%$. Fitting MTFs for the simulated image series presents the same behaviour of the C parameter as in Fig. 4, proving a sensible modelling of the readout and shot noise contributions.

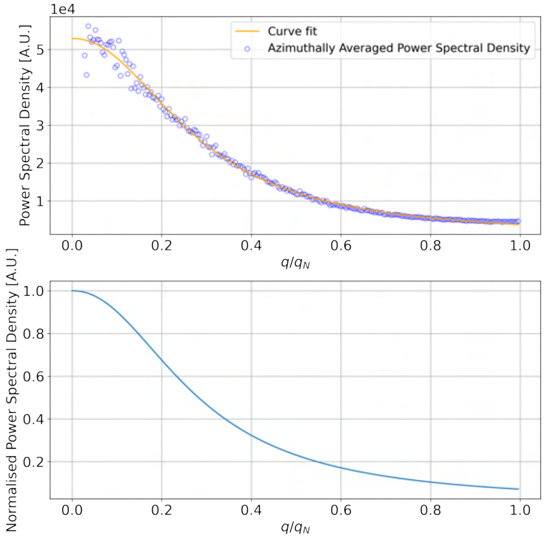


Figure A.13: MTF fitted from the power spectral density of the final frame at approximately 1300 $e^-/\text{Å}^2$. Top: Raw fit (Eq. A.1) with $R^2 = 0.99$, $c_0 = 0.28$, $c_1 = 52887.32$, $c_2 = 602.13$, $c_3 = 2.25$. Bottom: Normalised MTF (Eq. 1) with $c_0 = 0.28$, $c_3 = 2.15$, $C = 0.01$.

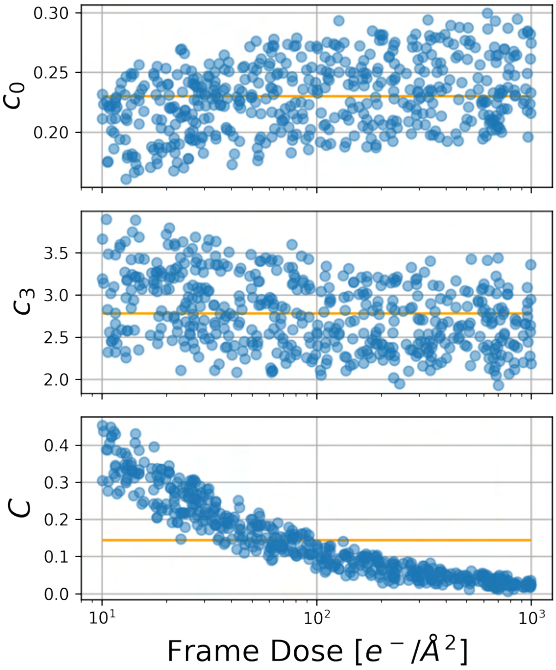
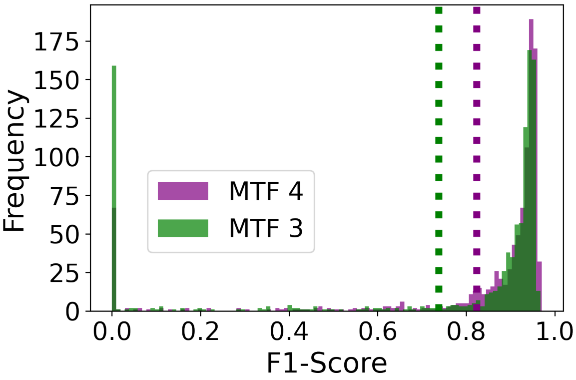
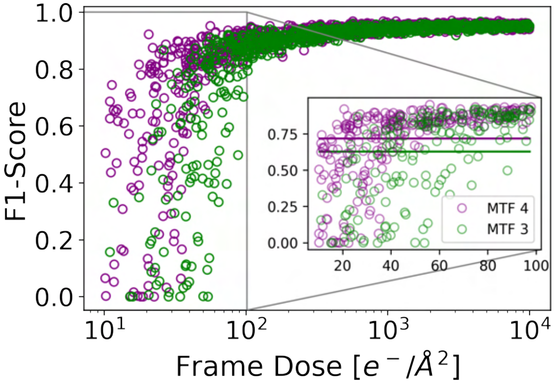


Figure A.14: Distribution of parameters for Eq. 1, fitted to a simulated vacuum image series of continuously increasing dose. This replicates the behaviour in Fig. 4.



(a)



(b)

Figure A.15: Comparing the networks trained separately on simulated data with the MTF 4 model and the MTF 3 model. Both networks are applied on a simulated continuously increasing dose image series of a CeO_2 supported Au nanoparticle with the MTF 3 model applied. The F1-Score ranges between 0 and 1, where 1 is a perfect segmentation. The MTF 4 model learns to distinguish noise and signal better than the MTF 3 model, seen by the higher mean F1-Score. Top: Histogram of F1-Score for each image. Dashed lines represent the colour coded mean. Bottom: F1-Score as a function of frame dose. Solid lines represent the mean within the inset of the very low dose regime.

8.3 Publication III: Identifying Atomic Positions in MoS2 with Neural Networks using Focal Series from High-Resolution Transmission Electron Microscopy.

Matthew Helmi Leth Larsen
Cuauhtemoc Nuñez Valencia
Jakob Schiøtz

Preprint. *In preparation.*

Identifying Atomic Positions in MoS_2 with Neural Networks using Focal Series from High-Resolution Transmission Electron Microscopy.

Matthew Helmi Leth Larsen[a], Cuauhtemoc Nuñez Valencia[a], Jakob Schiøtz[a]

[a] *Computational Atomic-scale Materials Design (CAMD), Department of Physics, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark*

Abstract

We demonstrate that a convolutional neural network is able to identify atomic species information from high-resolution transmission electron microscopy images. We train the networks on simulated images of the two-dimensional material molybdenum disulphide (MoS_2) to identify atomic columns containing a Mo atom, a single or two sulphur atoms. To do this reliably, the network needs a focal series of three images with a known difference in defocus. In this case the networks can achieve near perfect identification of all columns over a large range of microscope parameters. Three different network architectures were investiaged, they were all able to achieve this with 100 training images and with a number of parameters below 10^5.

Keywords: HR-TEM, 2D Materials, Machine Learning, Multi-class Segmentation

1. Introduction

High-resolution transmission electron microscopy (HR-TEM) can be used to directly visualise the arrangement of atoms in a sample, allowing researchers to determine the structure and properties of the material. One of the key advantages of HR-TEM is to study the dynamics of these atoms as they interact with each other and with their environment. The ability to identify and study atom dynamics is important for example to optimize functional materials, and to develop new ones, as the functional properties of these materials often depend on atomic-scale details. Properties like mechanical strength and electrical conductivity are often determined by atomic-scale defects, and the catalytic properties of nanoparticles often depend on the presence of specific atomic configurations [1]. In their operating environment, such materials may exhibit rapid changes in their atomic configuration [2, 3] By using HR-TEM, researchers can directly observe the movement of atoms in space and time, providing valuable insights into the mechanisms that govern their behaviour [4, 5]. Identifying atoms in HR-TEM can assist with understanding the formation of catalytically active sites [6, 7], morphological changes and changes to exposed facets under different gaseous environments [8], edge termination of 2D-materials [9, 10, 11], and identifying formation of vacancy sites [12, 13].

Time resolution can be of importance when tracking atom dynamics, which makes it beneficial to image samples with TEM rather than Scanning-TEM (STEM), where the entire sample is illuminated simultaneously. Intensities in images acquired in TEM mode are less interpretable than in STEM mode, making it increasingly difficult to extract information of atomic columns in TEM images. Focal series are often acquired in TEM mode to add an extra dimension of information. These focal series can be used to analytically reverse the effects of the contrast transfer function between the image and the exitwave function, in order to obtain the maximal information of atomic positions in all 3 spatial dimensions [14]. The number of images in the focal series is typically on the order of 20-50 images, heavily reducing the time resolution. The same authors of this work presented the possibility of using neural networks to solve this task with a focal series of only 3 images [15]. An approximate exitwave reconstruction can be performed in real-time with neural networks to get live information of atomic positions and defect formation.

Defect formation is an important parameter when attempting to understand electron beam induced effects on material samples under exposure. Studies have attempted to identify energy thresholds for defect formation in various material samples [13, 16, 17, 18, 19]. Avoiding beam damage in any material is essential to be able to discern the naturally occurring structures from the oned induced by the beam. It becomes particularly pertinent in industrially important but beam sensitive materials, such as molybdenum disulphide (MoS_2). MoS_2 is a common catalyst in the industry used to remove sulphur from crude oil, amongst other applications [20, 21]. Identification of various defects in monolayer 2D transition metal dichalcogenides has been presented for STEM images and utilising image stacking to reduce noise, meaning a lower time resolution [22, 23].

Ultimately, identifying individual atoms in material samples is a challenging task in HR-TEM. This is an area where machine learning can step in to assist the process. The most common defect that occurs in MoS_2 is single

S vacancies (hereafter referred to as 1S columns). Other possible defects are missing Mo atoms or holes where both S atoms are missing, otherwise samples will consist of Mo and 2S columns that are similar in intensity [13, 19]. This work will report advancements on the tailoring of deep learning neural networks for differentiating between individual atomic columns in MoS2 samples by multi-class segmentation. The ability to simulate HR-TEM images for any given material allows for an extensive and detailed training process, utilising amounts of data near impossible for a human operator to handle. This concept has previously been demonstrated for tracking Au atomic columns in HR-TEM image sequences with an industry standard U-net architecture [24, 25]. Similar to work with Au atomic columns [24] and exitwave reconstruction [15], training the network to distinguish between atomic species in multi-component material samples is possible by feeding the network a focal series of images. For example, providing three images of an MoS_2 sample at three different defocus settings allows a network to differentiate between atomic columns consisting of either 1 molybdenum, 2 sulphur, or 1 sulphur atom in HRTEM images. This work compares the performance of the U-net implementation from Ref. [26], U-net++ [27], and MSD-net [28] in distinguishing between the described atomic columns in simulated HR-TEM images of MoS_2.

2. Methods

The MSD-net and U-net/U-net++ take different approaches in identifying features across varying length scales. The U-net has its name from the characteristic U-shape due to the series of down-sampling layers to spread features across a large length scale, followed by an equal number of up-sampling layers to return to the original resolution. The MSD-net in contrast retains the same resolution throughout all layers, but dilates the kernel to spread its weights over a larger region. The dilation value iterates over some given range repeatedly. The MSD-net belongs to the family of dense-nets, as every layer in the network is connected to previous layers by concatenations following every convolution. The U-net++ takes the U-net architecture and includes more skip connections across outputs of layers at different resolutions, resulting in a more densely connected network. This provides a middle ground between the two alternatives, the U-net and MSD-net, in order to identify the best approach to this problem.

We use the F1-Score as a measure for the performance of the networks, as it gauges the ability to avoid both false positives and false negatives. In this work we differentiate between two different F1-Scores: The pixel F1-Score and the atomic F1-Score. The former is the typical F1-Score computed pixel wise, *i.e.* every pixel contributes to the true positives, true negatives, false positives, and false negatives. This is a very sensitive metric, which will highlight whether all atoms have been properly classified and placed with pixel wise precision. For most cases pixel wise precision is not necessary. The latter therefore identifies the atoms in the groundtruth, and queries the prediction for atoms found nearby. True positives (TP) are defined as atoms identified in the predictions that are placed within 10 pixels from a groundtruth atom. With a resolution of 0.1 Å/pixel, this corresponds to predicting the atom within 1 Å from the correct position. The false positives (FP) and false negatives (FN) are then defined as

$$\begin{aligned} \text{FP} &= \text{No. of predicted atoms - TP}, \\ \text{FN} &= \text{No. of groundtruth atoms - TP} . \end{aligned}$$

We have simulated HR-TEM images of bare monolayer MoS_2 in the (001) crystal direction. The MoS_2 systems are generated with a random percentage of S vacancies in the percentage range [0, 20], and Mo holes in the percentage range [0, 5]. Random degrees of tilt are applied between [0, 1] degrees, and randomly shaped flakes cut out of larger sheets. This is identical to the work in Ref. [15], where all atoms are perturbed based on a Gaussian distribution of 0.01 Å. The training set consists of 1000 MoS_2 systems, each imaged at 10 different randomly selected imaging conditions from Table 1, and a validation set of 500 MoS_2 systems only imaged once. For each sample a starting defocus value is randomly selected within the range, then two more defocus values are selected at +50 ± 1 Å, and +100 ± 1 Å, to simulate a focal series with a defocus step of 50 ± 1 Å. The training process follows Ref. [26] and involves iterating over batches of 1000 images, 10 times, referred to as image epochs, such that for 100 training epochs, each image is used 10 times. The images are therefore pre-generated to ensure an exact comparison between the three different networks in mention by executing an identical training process and testing on an identical validation set.

Parameter	L	U	Unit
Acceleration voltage	50		keV
Defocus (Δf)	-150	50	Å
Spherical aberration (C_s)	-15	15	μm
Focal spread	5	20	Å
Blur	0.05	0.25	Å
Resolution	0.1	0.11	Å/pixel
Electron dose	$10^{2.5}$	10^5	$e^-/\text{Å}^2$
Modulation Transfer Function			
c_0	0.25	0.6	
c_3	2.0	5.0	
Readout (N_0)	0.005	0.015	

Table 1: Microscope parameters. For each image series, a set of microscope parameters are drawn within the limits given here, except the acceleration voltage which is kept constant. All distributions are uniform, except for the dose which is exponential.

3. Results

In Figure 1 it is shown that neural networks are able to achieve perfect atomic column identification on a majority of the validation set when given a simulated HR-TEM focal series. The atomic F1-Score is in the range where very few atomic columns are being mis-classified.

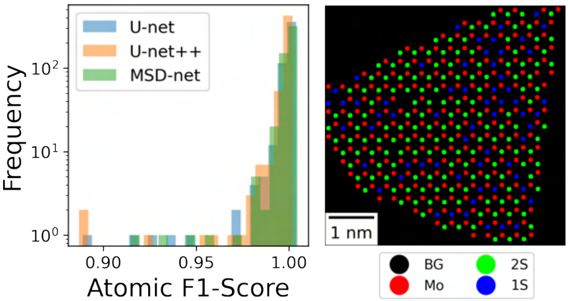


Figure 1: Left: Distribution of atomic F1-Scores for each network. All three present very strong detection of Mo, 2S, and 1S atomic columns. Right: Best MSD-net segmentation from the validation dataset, visualising the background (BG) and all the identified atomic columns of varying chemical composition.

The neural networks are able to classify the atomic columns beyond human capabilities. This is emphasised in Figure 2 that presents a line scan intensity of a simulated focal series from -150 to 150 Å with a defocus step of 50 Å. The line scan is placed along a line consisting of a two Mo columns, a single 2S column, and a single 1S column. The intensities show that at none of these defocus settings would a human interpreter be able to say with absolute certainty where the Mo, 2S, or 1S columns are. The images at 50, 100, and 150 Å defocus were fed to a neural network and the predictions were perfect, as presented below the image intensities of the line scan.

It is well known that neural networks train by minimising a highly muli-variate loss function space. This is done via the back propagation where gradients of the individual weights in the network are computed with respect to the loss function. Similar to fitting a polynomial one can argue that with the infinite increase of trainable weights one will eventually fit the dataset perfectly, however there is value in a model that fits the same data points with fewer parameters. There will be an in depth presentation of tuning these networks to solve the segmentation problem of classifying individual pixels into a 1Mo, 2S, 1S, or background class, where the network will have to use the limited focal series to learn the relative intensity differences between the classes.

3.1. Parameter Tuning

In Table 2 we highlight the significant difference in number of parameters between the U-net architecture and the MSD-net for parameters that maximise memory limitations of the supercomputing cluster used for training.

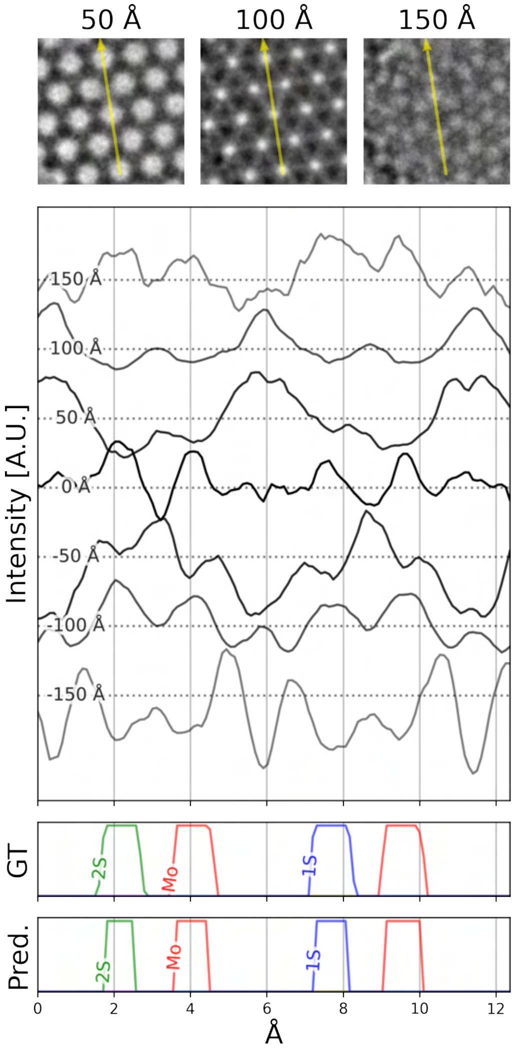


Figure 2: Simulated focal series where a line scan is placed over two Mo columns, a single 2S column, and a single 1S column. The intensities of the image line scans in the middle present the complexity of deducing atomic columns in HR-TEM images, however feeding a three image focal series (an example is shown at the top) to a neural network provides perfect atomic column identification, as shown at the bottom comparing the groundtruth (GT) and prediction (Pred.) line scans.

Due to the large amount of concatenation layers in the MSD-net, this configuration reaches the memory limits at far fewer parameters. The test loss evaluated after 300 training epochs shows greater performance from the U-net architectures.

With simulations we have a unique abundance of data and as a first test we search for the minimal training set size required for the performance of these models to converge. Limiting the training data size is one way to push

Network	Parameters	Time [hrs]	Test Loss
U-net	63,080,644	47.2	0.0473 ± 0.0005
U-net++	65,750,468	52.7	0.0473 ± 0.0007
MSD-net	277,054	52.7	0.0577 ± 0.0024

Table 2: Max attainable parameters with Float32 precision format in Tensorflow.

the networks to extremes and see which can withstand harder training conditions. The training set of 1000 images is limited to a training set size between 10 and 1000, but always evaluated on the validation set of 500 images. Figure 3 presents the mean and 10th percentile pixel F1-Score, and the mean atomic F1-Score. Each point is averaged over 3 replicas of each network, using the standard error for error bars. The result highlights a convergence in performance at 100 images in the training set, where the results show negligible error bars proving very reproducible results.

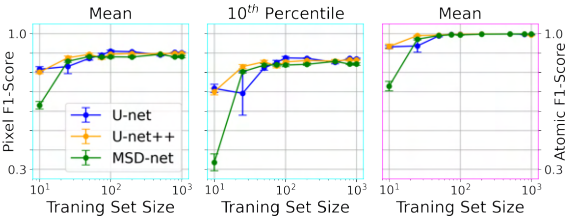


Figure 3: Left: Mean and Middle: 10th percentile pixel F1-Score averaged over 3 replicas of each network, and Right: the mean atomic F1-Score as a function of training set size, all validated on the same validation set of 500 images. All networks converge their performance at 100 images in the training set.

A mixed precision format was utilised within Tensorflow's framework, to do Float16 computations during training, which allowed for a boost in parameters for the MSD-net. This was an attempt to push the number of trainable parameters higher, but as a result also significantly affected the training and inferences times. This analysis is separated due to the hardware requirements restricting the use of mixed precision in Tensorflow. The MSD-net parameters and training times are presented in Table 3. The number of parameters were also tuned for the U-net and U-net++ by lowering the number of channels.

Each network presented in Table 3 is replicated 5 times and trained. The mean and 10th percentile atomic F1-Score is presented in Figure 4, where an optimal number of parameters is identified in the vicinity of 10^6 parameters. The networks here present very robust results, and the 10th percentile score reveals the networks ability of a near perfect identification of all atomic columns for 90% of the dataset between 10^5 to 10^7 parameters.

(a)

Kernel	Channels	Layers	Parameters	Time [hours]
5	3	25	68,329	20.9
5	3	50	277,054	52.7
5	3	80	713,149	106.9
5	5	50	768,004	78.1
9	3	50	894,958	99.7
5	7	50	1,503,954	108.9
5	9	50	2,484,904	141.8

(b)

Channels	U-net		U-net++	
	Parameters	Time [hours]	Parameters	Time [hours]
4	248,464	15.7	258,848	15.8
8	989,468	16.0	1,031,100	16.3
16	3,949,108	17.8	4,115,828	19.2
32	15,778,916	25.3	16,446,180	27.5
64	63,080,644	47.7	65,750,468	53.0

Table 3: Neural network models for parameter tuning. 5 replicas are trained for each. a) MSD-net. b) U-net and U-net++

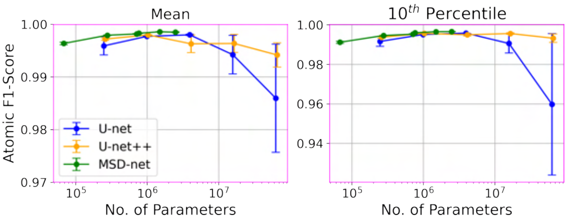


Figure 4: Left: mean and Right: 10th percentile atomic F1-Score averaged over 5 replicas, as a function of the number of trainable parameters in each neural network. See Table 3 for network details.

3.2. Generalisability and Limitations of the Method

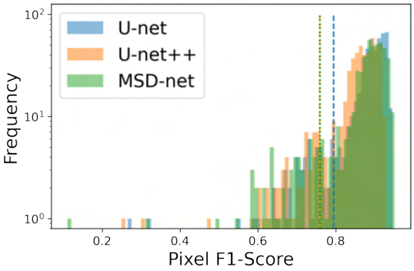
For this section an additional test dataset is simulated with the same conditions as in Table 1, however now the defocus range is [-150, 150] Å, to include defocus values outside the training range in Table 1 and gauge how well the networks generalise.

All networks perform perfectly on a majority of this test dataset. Figure 5b presents a histogram of the pixel F1-Score on this test dataset. All networks show similar performance. Table 5a presents the pixel F1-Score for median of the MSD-net result for each network, and the atomic F1-Score. Comparing the pixel F1-Score value to the predictions of each network in Figure 5c, it is apparent that the differences in scores are not obvious, in fact it is hard to understand why all scores are not perfect. The atomic F1-Score presents that all networks are achieving perfect identification of all atomic columns.

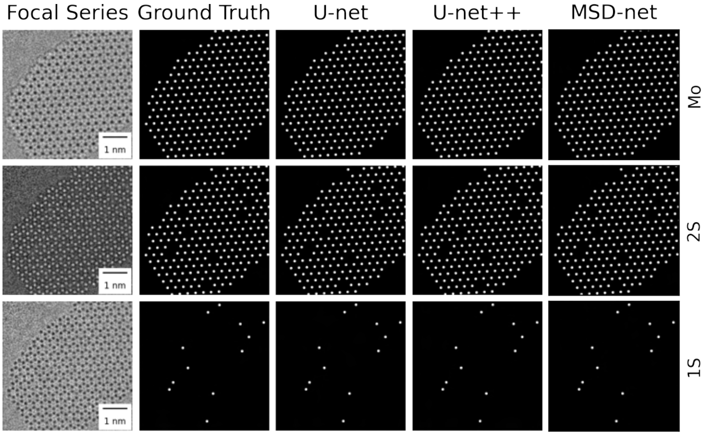
Looking at the differences between each network's prediction of the 1S class to the groundtruth in Figure 6, it is evident that the imperfect pixel F1-Score stems from the

Network	Pixel	Atomic
U-net	0.897	1.0
U-net++	0.885	1.0
MSD-net	0.891	1.0

(a)



(b)



(c)

Figure 5: Example of using the pixel wise F1-Score as a metric. (a) The median score for each network. (b) Histogram of the scores for all examples in the test dataset. Dashed lines represent the 10^{th} percentile. (c) Visualisation of the MSD-net's median example for each network. The pixel wise scores in (a) do not properly represent the result in (c).

fact that all atoms are not perfectly placed. The number of false positives and false negatives here correspond to the shift of the atoms, however the same number of false positives and false negatives could arise from a situation with more precisely predicted atoms and a number of completely missing or excess predicted atoms. This highlights the need for the atomic F1-Score as a reasonable metric. The mis-positioning is on the order of tens of picometres, presenting that this method is achieving picometre resolution determination of multi-component atomic columns.

The atomic F1-Score is used instead in Figure 7, where the histogram in Figure 7a presents a bimodal distribution, with one mode around the 10^{th} percentile scores. Figure 7b shows that this cluster of scores is occuring for defocus values outside the range of the training set, highlighting that these networks are not able to generalise outside the range of defocus values used for training. In Figure 7c it is clear that the low score is sensible as all networks show considerable weaknesses in identifying the defects of this focal series.

As a final note we train networks to solve the same problem, but instead with a single image, and a focal series of 2 images. Figure 8 shows the atomic F1-Score for a single image, focal series of 2 images, and a focal series of 3 images, from left to right, imaged with the conditions in Table 1. There is a large improvement from a single image to a focal series of two images, and minor improvement from a focal series of two to three images, which highlights the ability of each network to utilise the extra level of information in the focal series to differentiate between atomic column intensities, however a focal series of 20 images would not be necessary, reducing the time resolution of atomic position probing in TEM substantially.

4. Conclusions

It has been shown that neural networks are able to classify atomic columns in a multi-component material samples, in this case MoS_2, where the atomic columns of interest are 1Mo, 2S, and 1S columns. The networks exceed human performance. This is a challenging task, since HR-TEM intensities are not as interpretable as, for example, STEM image intensities. Typically, image averaging is used to interpret such intensities, however the neural networks require only a focal series of 3 images, imaged at a defocus step of 50 Å. This allows for probing atom dynamics with a fine temporal resolution.

The neural networks do not generalise to defocus value outside the training range, so the networks must be trained to the defocus range that the user plans to work with. The neural networks do not see much improvement when increasing the number of trainable parameters from 10^5 to 10^7 parameters, and do not require more than 100 images to train.

An ”atomic” F1-Score was introduced that truly gauges how well the neural networks are identifying atomic columns, without the need for pixel wise precision, which revealed that all atomic columns are being identified with the picometer resolution.

Neural network atomic column identification serves as a promising candidate for tracking changes in the atomic configuration, most importantly defect formation, when exposing samples to varying conditions such as changing electron doses.

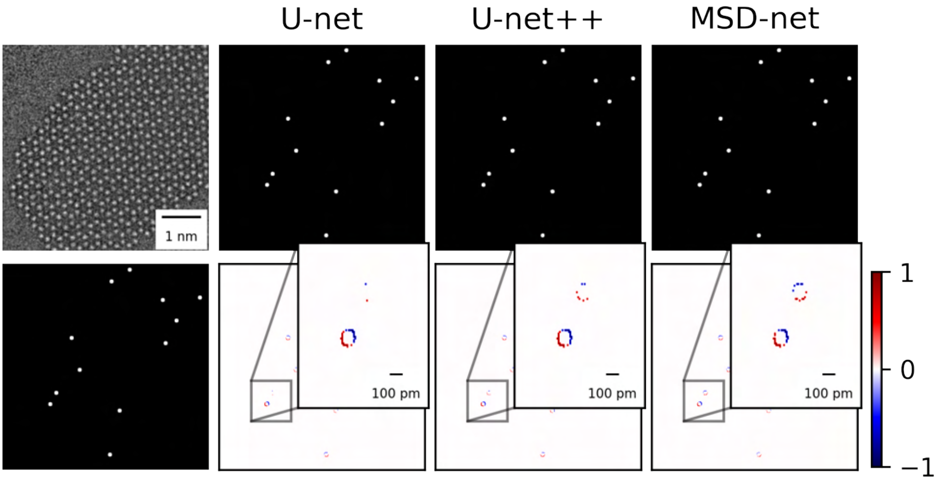


Figure 6: Difference between the 1S class of the median example presented in Figure 5

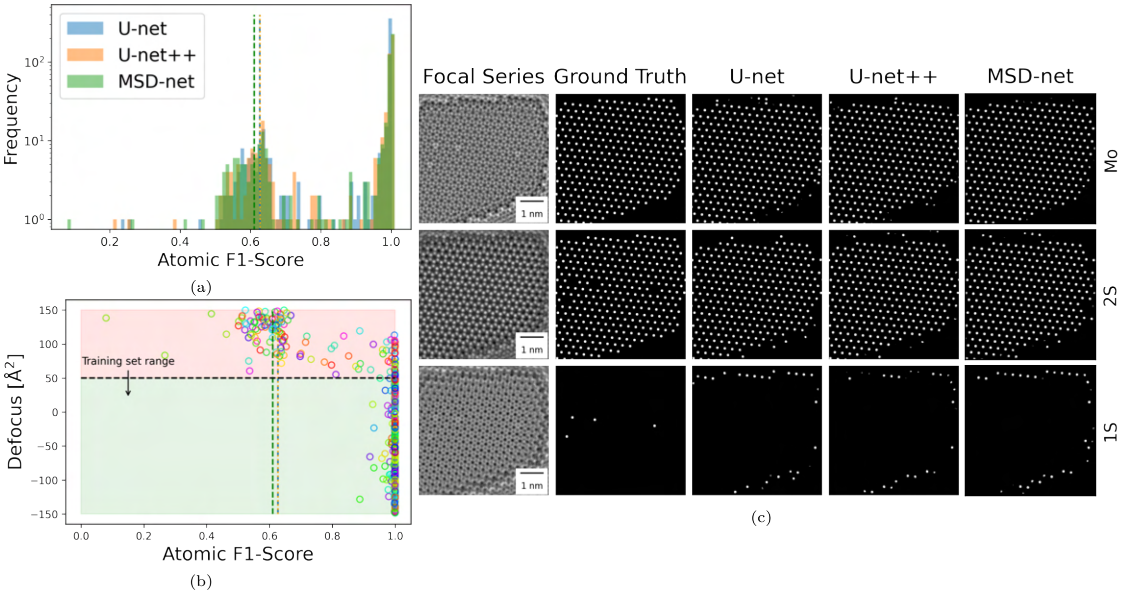
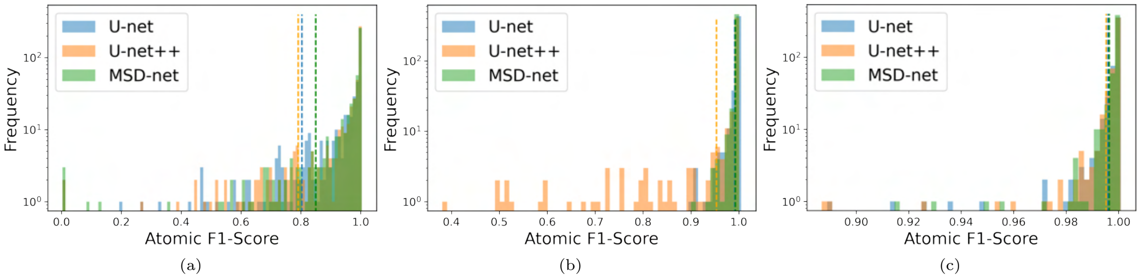


Figure 7: Performance of each network on the test dataset with a defocus range outside the training dataset range. (a) Histogram of atomic F1-Score for each network. (b) Atomic F1-Score versus the minimum defocus value in the focal series. Dashed lines in (a) and (b) represent the 10th percentile. (c) MSD-net's 10th percentile example for each network.



(a) (b) (c)

Figure 8: Ability of all three networks to classify Mo, 2S, or 1S columns given (a) a single image, (b) a focal series of 2 images, or (c) a focal series of 3 images. Dashed lines represent the 10^{th} percentile.

Funding

The authors acknowledge financial support from the Independent Research Fund Denmark (DFF-FTP) through grant no. 9041-00161B.

Competing interests

The authors declare that they have no competing interests.

References

[1] K. Honkala, A. Hellman, I. N. Remediakis, A. Logadottir, A. Carlsson, S. Dahl, C. H. Christensen, J. K. Nørskov, Ammonia synthesis from first-principles calculations, Science 307 (5709) (2005) 555 – 558. `doi:10.1126/science.1106435`.

[2] S. B. Vendelbo, C. F. Elkjaer, H. Falsig, I. Puspitasari, P. Dona, L. Mele, B. Morana, B. J. Nelissen, R. v. Rijn, J. F. Creemer, P. J. Kooyman, S. Helveg, Visualization of oscillatory behaviour of Pt nanoparticles catalysing CO oxidation, Nature materials 13 (9) (2014-09) 884 – 890. `doi:10.1038/nmat4033`.

[3] P. Liu, T. Wu, J. Madsen, J. Schiøtz, J. B. Wagner, T. W. Hansen, Transformations of supported gold nanoparticles observed by in situelectron microscopy, Nanoscale 11 (24) (2019) 11885 – 11891. `doi:10.1039/c9nr02731a`.

[4] P. E. Batson, Motion of gold atoms on carbon in the aberration-corrected STEM, in: Microscopy and Microanalysis, Vol. 14, 2008, pp. 89–97. `doi:10.1017/S1431927608080197`.

[5] A. De Wael, A. De Backer, L. Jones, A. Varambhia, P. D. Nellist, S. Van Aert, Measuring Dynamic Structural Changes of Nanoparticles at the Atomic Scale Using Scanning Transmission Electron Microscopy, Physical Review Letters 124 (10). `doi:10.1103/PhysRevLett.124.106105`.

[6] L. C. Gontard, L. Y. Chang, C. J. Hetherington, A. I. Kirkland, D. Ozkaya, R. E. Dunin-Borkowski, Aberration-corrected imaging of active sites on industrial catalyst nanoparticles, Angewandte Chemie - International Edition 46 (20) (2007) 3683–3685. `doi:10.1002/anie.200604811`.

[7] C. Kisielowski, L. W. Wang, P. Specht, H. A. Calderon, B. Barton, B. Jiang, J. H. Kang, R. Cieslinski, Real-time sub-Ångstrom imaging of reversible and irreversible conformations in rhodium catalysts and graphene, Physical Review B - Condensed Matter and Materials Physics 88 (2). `doi:10.1103/PhysRevB.88.024305`.

[8] T. Altantzis, I. Lobato, A. De Backer, A. Béché, Y. Zhang, S. Basak, M. Porcu, Q. Xu, A. Sánchez-Iglesias, L. M. Liz-Marzán, G. Van Tendeloo, S. Van Aert, S. Bals, Three-Dimensional Quantification of the Facet Evolution of Pt Nanoparticles in a Variable Gaseous Environment, Nano Letters 19 (1) (2019) 477–481. `doi:10.1021/acs.nanolett.8b04303`.

[9] L. P. Hansen, Q. M. Ramasse, C. Kisielowski, M. Brorson, E. Johnson, H. Topsøe, S. Helveg, Atomic-scale edge structures on industrial-style MoS 2 nanocatalysts, Angewandte Chemie - International Edition 50 (43) (2011) 10153–10156. `doi:10.1002/anie.201103745`.

[10] Q. Chen, H. Li, W. Xu, S. Wang, H. Sawada, C. S. Allen, A. I. Kirkland, J. C. Grossman, J. H. Warner, Atomically Flat Zigzag Edges in Monolayer MoS2 by Thermal Annealing, Nano Letters 17 (9) (2017) 5502–5507. `doi:10.1021/acs.nanolett.7b02192`.

[11] X. Zhao, D. Fu, Z. Ding, Y. Y. Zhang, D. Wan, S. J. Tan, Z. Chen, K. Leng, J. Dan, W. Fu, D. Geng, P. Song, Y. Du, T. Venkatesan, S. T. Pantelides, S. J. Pennycook, W. Zhou, K. P. Loh, Mo-Terminated Edge Reconstructions in Nanoporous Molybdenum Disulfide Film, Nano Letters 18 (1) (2018) 482–490. `doi:10.1021/acs.nanolett.7b04426`.

[12] E. L. Lawrence, B. D. Levin, T. Boland, S. L. Chang, P. A. Crozier, Atomic Scale Characterization of Fluxional Cation Behavior on Nanoparticle Surfaces: Probing Oxygen Vacancy Creation/Annihilation at Surface Sites, ACS Nano 15 (2) (2021) 2624–2634. `doi:10.1021/acsnano.0c07584`.

[13] S. Kretschmer, T. Lehnert, U. Kaiser, A. V. Krasheninnikov, Formation of Defects in Two-Dimensional MoS2 in the Transmission Electron Microscope at Electron Energies below the Knock-on Threshold: The Role of Electronic Excitations, Nano Letters 20 (4) (2020) 2865–2870. `doi:10.1021/acs.nanolett.0c00670`.

[14] F. R. Chen, D. Van Dyck, C. Kisielowski, L. P. Hansen, B. Barton, S. Helveg, Probing atom dynamics of excited Co-Mo-S nanocrystals in 3D, Nature Communications 12 (1). `doi:10.1038/s41467-021-24857-4`.
URL `http://dx.doi.org/10.1038/s41467-021-24857-4`

[15] M. H. Leth Larsen, F. Dahl, L. P. Hansen, B. Barton, C. Kisielowski, S. Helveg, O. Winther, T. W. Hansen, J. Schiøtz, Reconstructing the exit wave of 2D materials in high-resolution transmission electron microscopy using machine learning, Ultramicroscopy 243 (November 2022) (2022) 113641. `doi:10.1016/j.ultramic.2022.113641`.
URL `https://doi.org/10.1016/j.ultramic.2022.113641`

[16] A. W. Robertson, C. S. Allen, Y. A. Wu, K. He, J. Olivier, J. Neethling, A. I. Kirkland, J. H. Warner, Spatial control of defect creation in graphene at the nanoscale, Nature Communications 3. `doi:10.1038/ncomms2141`.

[17] N. Jiang, Electron beam damage in oxides: A review (12 2015). `doi:10.1088/0034-4885/79/1/016501`.

[18] C. Kisielowski, H. Frei, P. Specht, I. D. Sharp, J. A. Haber, S. Helveg, Detecting structural variances of Co3O4 catalysts by controlling beam-induced sample alterations in the vacuum of a transmission electron microscope, Advanced Structural and Chemical Imaging 2 (1). `doi:10.1186/s40679-016-0027-9`.

[19] H. P. Komsa, J. Kotakoski, S. Kurasch, O. Lehtinen, U. Kaiser, A. V. Krasheninnikov, Two-dimensional transition metal dichalcogenides under electron irradiation: Defect production and doping, Physical Review Letters 109 (3) (2012) 1–5. `doi:10.1103/PhysRevLett.109.035503`.

[20] Y. Zhu, Q. M. Ramasse, M. Brorson, P. G. Moses, L. P. Hansen, C. F. Kisielowski, S. Helveg, Visualizing the stoichiometry of industrial-style Co-Mo-S catalysts with single-atom sensitivity, Angewandte Chemie - International Edition 53 (40) (2014) 10723–10727. `doi:10.1002/anie.201405690`.

[21] R. F. Egerton, Radiation damage to organic and inorganic specimens in the TEM, Micron 119 (November 2018) (2019) 72–87. `doi:10.1016/j.micron.2019.01.005`.
URL `https://doi.org/10.1016/j.micron.2019.01.005`

[22] C. H. Lee, A. Khan, D. Luo, T. P. Santos, C. Shi, B. E. Janicek, S. Kang, W. Zhu, N. A. Sobh, A. Schleife, B. K. Clark, P. Y. Huang, Deep learning enabled strain mapping of single-atom defects in two-dimensional transition metal dichalcogenides with sub-picometer precision, Nano Letters 20 (5) (2020) 3369–3377. `doi:10.1021/acs.nanolett.0c00269`.

[23] Y. Lee, J. Lee, H. Chung, J. Kim, Z. Lee, In Situ Scanning Transmission Electron Microscopy Study of MoS2Formation on Graphene with a Deep-Learning Framework, ACS Omega 6 (33) (2021) 21623–21630. `doi:10.1021/acsomega.1c03002`.

[24] J. Madsen, P. Liu, J. Kling, J. B. Wagner, T. W. Hansen, O. Winther, J. Schiøtz, A Deep Learning Approach to Identify Local Structures in Atomic-Resolution Transmission Electron Microscopy Images, Advanced Theory and Simulations 1 (8) (2018) 1–12. `doi:10.1002/adts.201800037`.

[25] M. Ragone, V. Yurkiv, B. Song, A. Ramsubramanian, R. Shahbazian-Yassar, F. Mashayek, Atomic column heights detection in metallic nanoparticles using deep convolutional learning, Computational Materials Science 180 (December 2019) (2020) 109722. `doi:10.1016/j.commatsci.2020.109722`.
URL `https://doi.org/10.1016/j.commatsci.2020.109722`

[26] M. H. Leth Helmi, W. B. Lomholdt, C. N. Valencia, T. W. Hansen, J. Schiøtz, [In Progress] Quantifying Noise Limitations of Neural Network Segmentations in High-Resolution Transmission Electron Microscopy .

[27] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, Unet++: A nested u-net architecture for medical image segmentation, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11045 LNCS (2018) 3–11. `doi:10.1007/978-3-030-00889-5{\_}1`.

[28] D. M. Pelt, J. A. Sethian, A mixed-scale dense convolutional neural network for image analysis, Proceedings of the National Academy of Sciences of the United States of America 115 (2) (2017) 254–259. `doi:10.1073/pnas.1715832114`.

APPENDIX A

Appendix

A.1 An Example of a User Defined Input Parameters File for the Software Pipeline

```
    # Specify the type of task by the type of label
    # Mask, Blob, Disk, Gaussian, Exitwave
    ’label’: ”Mask”,
    #’label’: ”Disk”,
    #’label’: ”Exitwave”,

    # Number of images (at different microscope parameters)
    # per exit wave. (Only applies to training data)
    ’image_epochs’: 10,

    # Size of the images during training (x, y)
    ’image_size’: (512, 512),

    # Stochastic noise model to apply
    # poisson - shot noise approximation
    # None - no noise
    ’noise’: ”poisson”,

    # Number of classes in output of network, including the background
    # class.  Setting num_classes=1 means just a single class, no
    # background.  Otherwise, num_classes should be one higher than
    # the number of actual classes, to make room for the background
    # class.
    ’num_classes’: 1,
    #’num_classes’: 4,

    # Should there be a background class?
    ’null_class’: True,

    # Spot size of atomic columns in Angstrom
    ’spotsize’: 0.4,

    # Image resolution range in pixels/Angstrom
    #’sampling’: (0.17, 0.18),
    ’sampling’: (0.07, 0.08),
    #’sampling’: (0.09, 0.11),

    # Focal series if not None.
    # (number of images, change in focus, random part of change).
    ’multifocus’: None,
    #’multifocus’: (3, 50.0, 1.0),
```

```
## Noise parameters
# Range of the logarithm (base 10) of the dose in electrons/A^2
'log_dose': (2, 6),

# Range of blur
'blur': (0.1, 0.8),

# Range of focal spread
'focal_spread': (5, 20),

## CTF Parameters (values from microscope at Nanolab)
# Defocus range in Angstrom
'defocus': (-200, 200),
# Spherical abberation in Angstrom (1 micrometer = 1e4 A)
'ctf_c30': (0, 1.5*83000),
# 2-fold astigmatism in Angstrom
'ctf_c12': (0, 0),
# Axial coma in Angstrom
'ctf_c21': (0, 0),
# 3-fold astigmatism in Angstrom
'ctf_c23': (0, 0),
# Star abberation in Angstrom
'ctf_c32': (0, 0),
# 4-fold astigmatism in Angstrom
'ctf_c34': (0, 0),
# 5-fold astigmatism in Angstrom
'ctf_c45': (0, 0),

## MTF Parameters
# Deterministic modulation transfer function to apply
# None - no mtf (flat spatial distribution)
# parametric (c1,c2,c3)
"mtf": "parametric",
# Range of MTF parameters
'mtf_c1': (0.2, 0.6),
'mtf_c2': (0.0, 0.0),
'mtf_c3': (2.0, 5.0),

## Readout noise (constant lambda for additional poisson noise
## from the detector)
'readout': (0.005, 0.015),

# normalization distance in Angstrom
'normalizedistance': 12.0,

# How many images to save in debug folder (None=none, True=all).
'debug': 100,

# Seed for reproducibility
'seed': 100
```

Listing A.1: An example of an input parameters file from Table 3.2, which presents the various options the user has to set to prepare a neural network for a specific task and experimental data set in the software pipeline.

A.2 More Instructions on Unix Command-Line Usage

```
$ python train_imageepochs.py [Epochs]
  [Training Data] [Validation Data] [Output Folder]
  Optional: --restart [Restart from epoch number]
  Optional: --regularize [Amount of L2 regularization. Default:Disabled]
  Optional: --epochsave [Epoch interval for saving network]
  Optional: --adam [Adam optimiser passed as dict with parameters]
  Optional: --rmsprop [RMSProp learning rate]
  Optional: --limitdata [Truncate training data at a given number]
```

Listing A.2: An instruction on the *Unix* command-line usage of the *Python* script for training a U-net.

```
$ python train_uplusplus.py [Epochs]
  [Training Data] [Validation Data] [Output Folder]
  Optional: --restart [Restart from epoch number]
  Optional: --regularize [Amount of L2 regularization. Default:Disabled]
  Optional: --epochsave [Epoch interval for saving network]
  Optional: --adam [Adam optimiser passed as dict with parameters]
  Optional: --rmsprop [RMSProp learning rate]
  Optional: --limitdata [Truncate training data at a given number]
```

Listing A.3: An instruction on the *Unix* command-line usage of the *Python* script for training a U-net++.

```
# Data Generation (Atomic Models, Exit waves, Sites)
$ python make_fcc_cluster_supported.py example_data Au 3 1000
$ python make_fcc_cluster_supported.py example_data Au 3 500 --test

# Data Generation (Images and Labels)
$ python make_image_data.py example_data example_input_parameters.json
$ python make_image_data.py example_data example_input_parameters.json
  --test

# Neural Network Training (U-net)
$ python train_imageepochs.py 100 example_data example_data-test
  example_network
```

Listing A.4: Example of running the data generation and neural network segment of the workflow, where a training and validation/test data set of CeO_2 supported Au nanoparticles with a maximum off zone axis tilt of 3° are generated with images and labels defined by an input parameters file and a U-net is trained.

A.3 A Collection of the Method Attributes for the Python Objects for Handling Simulated and Experimental HR-TEM Data in the Software Pipeline

Table A.1: *Python* objects for handling and analysing both simulated and experimental data. *(...) will represent necessary user input - refer to the GitLab repository.

Class `Simulated_Data`

Initialises with a path to a *.npz* containing a simulated image and label, a *.json* input parameters file containing data parameters, and a *.json* parameters file with specific ctf parameters.

Attributes	Descriptions
`load()`	Loads the image, label, and sampling, as instance attributes.
`cp_local_standardise()`	Performs the local standardisation defined in Eq. (3.1) utilising CuPy [110] to accelerate Gaussian convolutions.
`infer( ... )`	Applies a trained neural network to the image. A Keras model is provided by the user.
`get_F1_score()`	Returns the F1-Score between the prediction and label.
`get_mse_score()`	Returns the MSE-Score between the prediction and label.
`plot()`	Plots the image and label, with scale bars.

Class `Experimental_Data`

Initialises class with just a path to an experimental file (*.dm3*/*.dm4*, or *TIFF*).

Attributes	Descriptions
`load()`	Loads the experimental data using Hyperspy along with metadata such as the sampling in nm/pixel.
`cp_local_standardise()`	Performs the local standardisation defined in Eq. (3.1) utilising CuPy [110] to acceleration Gaussian convolutions.

Continued on next page

Table A.1 – continued from previous page

`infer( ... )`	Applies a trained neural network to the image. A Keras model is provided by the user.
`crop()`	Crops the ROI using the mask segmentation. This for example crops a nanoparticle out of an image containing vacuum and substrate.
`fft()`	Returns the Fourier transform of the Image. If `crop()` was applied, the fft will be performed on the cropped image, allowing fft extraction of the nanoparticle absent of any surroundings.
`compute_dose( ... )`	Uses the region defined by corner and window size (should be a vacuum area) to compute the frame's electron dose based on the metadata. User provides the corner and window size.
`plot()`	Plots the image and label, with scale bars.

Bibliography

[1] I. Chorkendorff and J. Niemantsverdriet, “Oil Refining and Petrochemistry,” in *Concepts of modern catalysis and kinetics* (October 2003) Chap. 9, pages 349–376, `https://doi.org/10.1002/3527602658.ch9`.

[2] A. Beniya and S. Higashi, “Towards dense single-atom catalysts for future automotive applications,” Nature Catalysis **2**, 590 (2019), `http://dx.doi.org/10.1038/s41929-019-0282-y`.

[3] X. Huang, C. Liu, and P. Zhou, “2D semiconductors for specific electronic applications: from device to system,” npj 2D Materials and Applications **6**, `10.1038/s41699-022-00327-3` (2022).

[4] Y. Lecun, Y. Bengio, and G. Hinton, “Deep learning,” Nature **521**, 436 (2015).

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” Communications of the ACM **60**, 84 (2017).

[6] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, “Image Segmentation Using Deep Learning: A Survey,” (2020), `http://arxiv.org/abs/2001.05566`.

[7] G. Hinton, L. Deng, D. Yu, G. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, and B. Kingsbury, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Processing Magazine **29**, 82 (2012).

[8] L. Ding, Y. Wang, R. Laganière, D. Huang, and S. Fu, “Convolutional neural networks for multispectral pedestrian detection,” Signal Processing: Image Communication **82**, `10.1016/j.image.2019.115764` (2020).

[9] E. Gómez-de-Mariscal, M. Maka, A. Kotrbová, V. Pospíchalová, P. Matula, and A. Muñoz-Barrutia, “Deep-Learning-Based Segmentation of Small Extracellular Vesicles in Transmission Electron Microscopy Images,” Scientific Reports **9**, `10.1038/s41598-019-49431-3` (2019).

[10] X. Liu, L. Song, S. Liu, and Y. Zhang, “A review of deep-learning-based medical image segmentation methods,” Sustainability (Switzerland) **13**, 1 (2021).

[11] J. Madsen, P. Liu, J. Kling, J. B. Wagner, T. W. Hansen, O. Winther, and J. Schiøtz, “A Deep Learning Approach to Identify Local Structures in Atomic-Resolution Transmission Electron Microscopy Images,” Advanced Theory and Simulations **1**, 1 (2018).

[12] M. Ragone, V. Yurkiv, B. Song, A. Ramsubramanian, R. Shahbazian-Yassar, and F. Mashayek, "Atomic column heights detection in metallic nanoparticles using deep convolutional learning," Computational Materials Science **180**, `10.1016/j.commatsci.2020.109722` (2020).

[13] K. Faraz, T. Grenier, C. Ducottet, and T. Epicier, "Deep learning detection of nanoparticles and multiple object tracking of their dynamic evolution during in situ ETEM studies," Scientific Reports **12**, `10.1038/s41598-022-06308-2` (2022).

[14] C. K. Groschner, C. Choi, and M. C. Scott, "Machine Learning Pipeline for Segmentation and Defect Identification from High-Resolution Transmission Electron Microscopy Data," Microscopy and Microanalysis **27**, 549 (2021).

[15] J. M. Ede, J. J. P. Peters, J. Sloan, and R. Beanland, "Exit Wavefunction Reconstruction from Single Transmission Electron Micrographs with Deep Learning," (2020), `http://arxiv.org/abs/2001.10938`.

[16] S. V. Kalinin, A. R. Lupini, O. Dyck, S. Jesse, M. Ziatdinov, and R. K. Vasudevan, "Lab on a beam-Big data and artificial intelligence in scanning transmission electron microscopy," in Mrs bulletin, Vol. 44, 7 (July 2019), pages 565–575.

[17] M. Botifoll, I. Pinto-Huguet, and J. Arbiol, "Machine learning in electron microscopy for advanced nanocharacterization: current developments, available tools and future outlook," Nanoscale Horizons **7**, 1427 (2022).

[18] S. R. Spurgeon, C. Ophus, L. Jones, A. Petford-Long, S. V. Kalinin, M. J. Olszta, R. E. Dunin-Borkowski, N. Salmon, K. Hattar, W. C. D. Yang, R. Sharma, Y. Du, A. Chiaramonti, H. Zheng, E. C. Buck, L. Kovarik, R. L. Penn, D. Li, X. Zhang, M. Murayama, and M. L. Taheri, "Towards data-driven next-generation transmission electron microscopy," Nature Materials **20**, 274 (2021).

[19] K. Sytwu, C. Groschner, and M. C. Scott, "Understanding the Influence of Receptive Field and Network Complexity in Neural Network-Guided TEM Image Analysis," Microscopy and Microanalysis **28**, 1896 (2022).

[20] J. P. Horwath, D. N. Zakharov, R. Mégret, and E. A. Stach, "Understanding important features of deep learning models for segmentation of high-resolution transmission electron microscopy images," npj Computational Materials **6**, `10.1038/s41524-020-00363-x` (2020).

[21] Z. J. Ding, C. Li, B. Da, and J. Liu, "Charging effect induced by electron beam irradiation: a review," Science and Technology of Advanced Materials **22**, 932 (2021).

[22] J. Kotakoski, C. H. Jin, O. Lehtinen, K. Suenaga, and A. V. Krasheninnikov, "Electron knock-on damage in hexagonal boron nitride monolayers," Physical Review B - Condensed Matter and Materials Physics **82**, 1 (2010).

[23] F. Banhart, "Irradiation effects in carbon nanostructures," Reports on Progress in Physics **62**, 1181 (1999).

[24] N. Jiang and J. C. Spence, "On the dose-rate threshold of beam damage in TEM," Ultramicroscopy **113**, 77 (2012), `http://dx.doi.org/10.1016/j.ultramic.2011.11.016`.

[25] N. Jiang, "Electron beam damage in oxides: A review," Reports on Progress in Physics **79**, `10.1088/0034-4885/79/1/016501` (2015).

[26] A. Kryshtal, M. Mielczarek, and J. Pawlak, "Effect of electron beam irradiation on the temperature of single AuGe nanoparticles in a TEM," Ultramicroscopy **233**, 113459 (2022), `https://doi.org/10.1016/j.ultramic.2021.113459`.

[27] T. Susi, J. C. Meyer, and J. Kotakoski, "Quantifying transmission electron microscopy irradiation effects using two-dimensional materials [arXiv]," Arxiv (2019).

[28] C. Kisielowski, L. W. Wang, P. Specht, H. A. Calderon, B. Barton, B. Jiang, J. H. Kang, and R. Cieslinski, "Real-time sub-Ångstrom imaging of reversible and irreversible conformations in rhodium catalysts and graphene," Physical Review B - Condensed Matter and Materials Physics **88**, `10.1103/PhysRevB.88.024305` (2013).

[29] F. R. Chen, D. Van Dyck, C. Kisielowski, L. P. Hansen, B. Barton, and S. Helveg, "Probing atom dynamics of excited Co-Mo-S nanocrystals in 3D," Nature Communications **12**, `10.1038/s41467-021-24857-4` (2021), `http://dx.doi.org/10.1038/s41467-021-24857-4`.

[30] E. J. Kirkland, "The Transmission Electron Microscope," in *Advanced computing in electron microscopy*, 3rd edition (Springer Cham, 2020) Chap. 2, pages 9–36, `https://link.springer.com/book/10.1007/978-3-030-33260-0`.

[31] F. R. Chen, C. Kisielowski, and D. Van Dyck, "3D reconstruction of nanocrystalline particles from a single projection," Micron **68**, 59 (2015).

[32] E. J. Kirkland, "Preface," in *Advanced computing in electron microscopy*, 3rd edition (Springer Cham, 2020), pages i–xiii, `https://link.springer.com/book/10.1007/978-3-030-33260-0`.

[33] E. J. Kirkland, "Theory of Calculation of Images of Thick Specimens," in *Advanced computing in electron microscopy*, 3rd edition (Springer Cham, 2020), pages 143–195, `https://link.springer.com/book/10.1007/978-3-030-33260-0`.

[34] D. F. Lynch and A. F. Moodie, "Numerical evaluation of low energy electron diffraction intensities: I. The perfect crystal with no upper layer lines and no absorption," Surface Science **32**, 422 (1972), `https://www.sciencedirect.com/science/article/pii/0039602872901707`.

[35] E. J. Kirkland, "Calculation of Images of Thin Specimens," in *Advanced computing in electron microscopy*, 3rd edition (Springer Cham, 2020) Chap. 5, pages 99–142, `https://link.springer.com/book/10.1007/978-3-030-33260-0`.

[36] E. J. Kirkland, "Some Image Approximations," in *Advanced computing in electron microscopy*, 3rd edition (Springer Cham, 2020) Chap. 3, pages 37–80, `https://link.springer.com/book/10.1007/978-3-030-33260-0`.

[37] K. Ishizuka, "Contrast transfer of crystal images in TEM," Ultramicroscopy **5**, 55 (1980), `https://www.sciencedirect.com/science/article/pii/030439918090011X`.

[38] L. J. Allen, W. McBride, N. L. O'Leary, and M. P. Oxley, "Exit wave reconstruction at atomic resolution," Ultramicroscopy **100**, 91 (2004).

[39] O. L. Krivanek, N. Dellby, and A. R. Lupini, "Towards sub-Å electron beams," Ultramicroscopy **78**, 1 (1999).

[40] A. Bleloch and Q. Ramasse, "Lens Aberrations: Diagnosis and Correction," in *Aberrationcorrected analytical transmission electron microscopy* (September 2011), pages 55–87, https://doi.org/10.1002/9781119978848.ch4.

[41] "Appendix A: Aberration Notation," in *Aberrationcorrected analytical transmission electron microscopy* (September 2011), pages 263–266, https://doi.org/10.1002/9781119978848.app1.

[42] D. Van Dyck, "Atomic Resolution Electron Microscopy," in *Handbook of nanoscopy* (April 2012), pages 45–79, https://doi.org/10.1002/9783527641864.ch2.

[43] O. Scherzer, "The theoretical resolution limit of the electron microscope," Journal of Applied Physics **20**, 20 (1949).

[44] A.-C. Milazzo, P. Leblanc, F. Duttweiler, L. Jin, J. C. Bouwer, S. Peltier, M. Ellisman, F. Bieser, H. S. Matis, H. Wieman, P. Denes, S. Kleinfelder, and N.-H. Xuong, "Active pixel sensor array as a detector for electron microscopy," Ultramicroscopy **104**, 152 (2005), https://www.sciencedirect.com/science/article/pii/S0304399105000513.

[45] Z. Lee, H. Rose, O. Lehtinen, J. Biskupek, and U. Kaiser, "Electron dose dependence of signal-to-noise ratio, atom contrast and resolution in transmission electron microscope images," Ultramicroscopy **145**, 3 (2014), http://dx.doi.org/10.1016/j.ultramic.2014.01.010.

[46] W. J. De Ruijter and J. K. Weiss, "Methods to measure properties of slow-scan CCD cameras for electron detection," Review of Scientific Instruments **63**, 4314 (1992).

[47] E. L. Lawrence, B. D. Levin, B. K. Miller, and P. A. Crozier, "Approaches to Exploring Spatio-Temporal Surface Dynamics in Nanoparticles with in Situ Transmission Electron Microscopy," Microscopy and Microanalysis **26**, 86 (2020).

[48] R. Manzorro, Y. Xu, J. L. Vincent, R. Rivera, D. S. Matteson, and P. A. Crozier, "Exploring Blob Detection to Determine Atomic Column Positions and Intensities in Time-Resolved TEM Images with Ultra-Low Signal-to-Noise," Microscopy and Microanalysis **28**, 1917 (2022).

[49] J. L. Vincent, R. Manzorro, S. Mohan, B. Tang, D. Y. Sheth, E. P. Simoncelli, D. S. Matteson, C. Fernandez-Granda, and P. A. Crozier, "Developing and Evaluating Deep Neural Network-Based Denoising for Nanoparticle TEM Images with Ultra-Low Signal-to-Noise," Microscopy and Microanalysis **27**, 1431 (2021).

[50] Gatan, *Gatan OneView*, https://www.gatan.com/products/tem-imaging-spectroscopy/oneview-camera.

[51] G. McMullan, A. R. Faruqi, D. Clare, and R. Henderson, "Comparison of optimal performance at 300keV of three direct electron detectors for use in low dose electron microscopy," Ultramicroscopy **147**, 156 (2014), `http://dx.doi.org/10.1016/j.ultramic.2014.08.002`.

[52] A. R. Faruqi and G. McMullan, "Direct imaging detectors for electron microscopy," Nuclear Instruments and Methods in Physics Research, Section A: Accelerators, Spectrometers, Detectors and Associated Equipment **878**, 180 (2018), `http://dx.doi.org/10.1016/j.nima.2017.07.037`.

[53] M. Vulovic, B. Rieger, L. J. Van Vliet, A. J. Koster, and R. B. Ravelli, "A toolkit for the characterization of CCD cameras for transmission electron microscopy," Acta Crystallographica Section D: Biological Crystallography **66**, 97 (2010).

[54] R. W. Gerchberg and W. O. Saxton, "A Practical Algorithm for the Determination of Phase from Image and Diffraction Plane Pictures," Optik (Stuttgart) **35**, 237 (1972), `https://cir.nii.ac.jp/crid/1572261550522209664.bib?lang=en`.

[55] W. Coene, G. Janssen, M. Op De Beeck, and D. Van Dyck, "Phase retrieval through focus variation for ultra-resolution in field-emission transmission electron microscopy," Physical Review Letters **69**, 3743 (1992).

[56] D. Van Dyck and M. Op De Beeck, "A simple intuitive theory for electron diffraction," Ultramicroscopy **64**, 99 (1996).

[57] S. Van Aert, P. Geuens, D. Van Dyck, C. Kisielowski, and J. R. Jinschek, "Electron channelling based crystallography," Ultramicroscopy **107**, 551 (2007).

[58] W. Sinkler and L. D. Marks, "Dynamical direct methods for everyone," Ultramicroscopy **75**, 251 (1999).

[59] A. Wang, F. R. Chen, S. Van Aert, and D. Van Dyck, "Direct structure inversion from exit waves. Part I: Theory and simulations," Ultramicroscopy **110**, 527 (2010), `http://dx.doi.org/10.1016/j.ultramic.2009.11.024`.

[60] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Feedforward Networks," in *Deep learning* (MIT Press, 2016) Chap. 6, pages 164–223, `http://www.deeplearningbook.org`.

[61] "Neural Networks," in *The principles of deep learning theory: an effective theory approach to understanding neural networks*, edited by D. A. Roberts and S. Yaida (Cambridge University Press, Cambridge, 2022) Chap. 2, pages 37–52, `https://www.cambridge.org/core/books/principles-of-deep-learning-theory/pretraining/9E54A59B9D1D04773CF9EF5B778C2527`.

[62] I. Goodfellow, Y. Bengio, and A. Courville, "Machine Learning Basics," in *Deep learning* (MIT Press, 2016) Chap. 5, pages 96–161.

[63] T. Hastie, R. Tibshirani, and J. H. Friedman, "Neural Networks," in *The elements of statistical learning : data mining, inference, and prediction* (Springer, 2001), pages 389–416.

[64] S. Jadon, "A survey of loss functions for semantic segmentation," 2020 Ieee Conference on Computational Intelligence in Bioinformatics and Computational Biology, Cibcb 2020, `10.1109/CIBCB48159.2020.9277638` (2020).

[65] I. Goodfellow, Y. Bengio, and A. Courville, "Optimization for Training Deep-Models," in *Deep learning* (MIT Press, 2016) Chap. 8, pages 271–325.

[66] J. M. Phillips, "Gradient Descent," in *Mathematical foundations for data analysis* (Springer International Publishing, 2021) Chap. 6, pages 125–142.

[67] "Gradient-Based Learning," in *The principles of deep learning theory: an effective theory approach to understanding neural networks*, edited by D. A. Roberts and S. Yaida (Cambridge University Press, Cambridge, 2022) Chap. 7, pages 191–198, `https://www.cambridge.org/core/books/principles-of-deep-learning-theory/pretraining/9E54A59B9D1D04773CF9EF5B778C2527`.

[68] S. J. Reddi, S. Kale, and S. Kumar, "On the convergence of adam and beyond [arXiv]," Arxiv (2019).

[69] J. Brownlee, *Deep Learning With Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras* (Machine Learning Mastery, 2016), `https://books.google.dk/books?id=K-ipDwAAQBAJ`.

[70] S. Basodi, C. Ji, H. Zhang, and Y. Pan, *Gradient amplification: an efficient way to train deep neural networks*, 2020.

[71] A. F. Agarap, "Deep learning using rectified linear units (relu) [arxiv]," Arxiv (2018).

[72] L. Trottier, P. Giguère, and B. Chaib-draa, "Parametric exponential linear unit for deep convolutional neural networks [arxiv]," Arxiv (2018).

[73] F. Alrasheedi, X. Zhong, and P. C. Huang, "<italic>padding module</italic>: learning the padding in deep neural networks," Ieee Access, 1 (2023).

[74] I. Goodfellow, Y. Bengio, and A. Courville, "Convolutional Networks," in *Deep learning* (MIT Press, 2016) Chap. 9, pages 326–366, `http://www.deeplearningbook.org`.

[75] A. Hjorth Larsen, J. JØrgen Mortensen, J. Blomqvist, I. E. Castelli, R. Christensen, M. Duak, J. Friis, M. N. Groves, B. Hammer, C. Hargus, E. D. Hermes, P. C. Jennings, P. Bjerre Jensen, J. Kermode, J. R. Kitchin, E. Leonhard Kolsbjerg, J. Kubal, K. Kaasbjerg, S. Lysgaard, J. Bergmann Maronsson, T. Maxson, T. Olsen, L. Pastewka, A. Peterson, C. Rostgaard, J. SchiØtz, O. Schütt, M. Strange, K. S. Thygesen, T. Vegge, L. Vilhelmsen, M. Walter, Z. Zeng, and K. W. Jacobsen, "The atomic simulation environment - A Python library for working with atoms," Journal of Physics Condensed Matter **29**, `10.1088/1361-648X/aa680e` (2017).

[76] J. Madsen and T. Susi, " abTEM: ab Initio Transmission Electron Microscopy Image Simulation," Microscopy and Microanalysis **26**, 448 (2020).

[77] R. Lin, R. Zhang, C. Wang, X. Q. Yang, and H. L. Xin, "Temimagenet training library and atomsegnet deep-learning models for high-precision atom segmentation, localization, denoising, and deblurring of atomic-resolution images," Scientific Reports **11**, 5386 (2021).

[78] T. Developers, "TensorFlow," `10.5281/ZENODO.6574269` (2022), `https://zenodo.org/record/6574269`.

[79] E. J. Kirkland, "Atomic Potentials and Scattering Factors," in *Advanced computing in electron microscopy* (Springer US, 2010), pages 243–260.

[80] A. K. Datye, D. S. Kalakkad, M. H. Yao, and D. J. Smith, "Comparison of metal-support interactions in pt/TiO2 and Pt/CeO2," Journal of Catalysis **155**, 148 (1995).

[81] E. Aneggi, M. Boaro, S. Colussi, C. de Leitenburg, and A. Trovarelli, "Ceria-Based Materials in Catalysis: Historical Perspective and Future Trends," in *Handbook on the physics and chemistry of rare earths*, Vol. 50, edited by J.-C. G. Bünzli, V. K. B. T. .-. H. o. t. P. Pecharsky, and C. of Rare Earths (Elsevier, 2016) Chap. 289, pages 209–242, `https://www.sciencedirect.com/science/article/pii/S0168127316300046`.

[82] J. L. Vincent and P. A. Crozier, "Atomic level fluxional behavior and activity of CeO2-supported Pt catalysts for CO oxidation," Nature Communications **12**, 1 (2021).

[83] M. Goyal and B. R. Gupta, "Shape and size dependent thermophysical properties of nanocrystals," Chinese Journal of Physics **56**, 282 (2018), `https://doi.org/10.1016/j.cjph.2017.12.014`.

[84] M. Frei and F. E. Kruis, "Fully automated primary particle size analysis of agglomerates on transmission electron microscopy images via artificial neural networks," Powder Technology **332**, 120 (2018), `https://doi.org/10.1016/j.powtec.2018.03.032`.

[85] M. J. Yacamán, J. A. Ascencio, H. B. Liu, and J. Gardea-Torresdey, "Structure shape and stability of nanometric sized particles," Journal of Vacuum Science & Technology B: Microelectronics and Nanometer Structures **19**, 1091 (2001).

[86] P. Liu, T. Wu, J. Madsen, J. Schiøtz, J. B. Wagner, and T. W. Hansen, "Transformations of supported gold nanoparticles observed by: In situ electron microscopy," Nanoscale **11**, 11885 (2019).

[87] T. M. Project, "Materials Data on CeO2 by Materials Project," `https://doi.org/10.17188/1195334` (2020), `https://www.osti.gov/dataexplorer/servlets/purl/1195334`.

[88] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, "Commentary: The Materials Project: A materials genome approach to accelerating materials innovation," APL Materials **1**, 11002 (2013), `https://doi.org/10.1063/1.4812323`.

[89] R. Kumar, W. Zheng, X. Liu, J. Zhang, and M. Kumar, "MoS2-Based Nanomaterials for Room-Temperature Gas Sensors," Advanced Materials Technologies **5**, 1901062 (2020), `https://doi.org/10.1002/admt.201901062`.

[90] H. P. Komsa, J. Kotakoski, S. Kurasch, O. Lehtinen, U. Kaiser, and A. V. Krasheninnikov, "Two-dimensional transition metal dichalcogenides under electron irradiation: Defect production and doping," Physical Review Letters **109**, 1 (2012).

[91] F. Bertoldo, S. Canulescu, R. R. Unocic, Y. C. Lin, X. Sang, A. A. Puretzky, Y. Yu, D. Miakota, C. M. Rouleau, J. Schou, K. S. Thygesen, and D. B. Geohegan, "Intrinsic defects in MoS2 grown by pulsed laser deposition: From monolayers to bilayers," ACS Nano **15**, 2858 (2021).

[92] S. Kretschmer, T. Lehnert, U. Kaiser, and A. V. Krasheninnikov, "Formation of Defects in Two-Dimensional MoS2 in the Transmission Electron Microscope at Electron Energies below the Knock-on Threshold: The Role of Electronic Excitations," Nano Letters **20**, 2865 (2020).

[93] L. S. Byskov, J. K. Nørskov, B. S. Clausen, and H. Topsøe, "Edge termination of MoS2 and CoMos catalyst particles," Catalysis Letters **64**, 95 (2000).

[94] K. Lee, J. Park, S. Choi, Y. Lee, S. Lee, J. Jung, J.-Y. Lee, F. Ullah, Z. Tahir, Y. S. Kim, G.-H. Lee, and K. Kim, "STEM Image Analysis Based on Deep Learning: Identification of Vacancy Defects and Polymorphs of MoS2," Nano Letters **22**, 4677 (2022), `https://doi.org/10.1021/acs.nanolett.2c00550`.

[95] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," (2015).

[96] T. M. Quan, D. G. C. Hildebrand, and W. K. Jeong, "FusionNet: A Deep Fully Residual Convolutional Neural Network for Image Segmentation in Connectomics," Frontiers in Computer Science **3**, `10.3389/fcomp.2021.613981` (2021).

[97] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) **11045 LNCS**, 3 (2018).

[98] D. M. Pelt and J. A. Sethian, "A mixed-scale dense convolutional neural network for image analysis," Proceedings of the National Academy of Sciences of the United States of America **115**, 254 (2017).

[99] F. d. l. Peña, E. Prestat, V. T. Fauske, P. Burdet, J. Lähnemann, P. Jokubauskas, T. Furnival, M. Nord, T. Ostasevicius, K. E. MacArthur, D. N. Johnstone, M. Sarahan, J. Taillon, T. Aarholt, pquinn-dls, V. Migunov, A. Eljarrat, J. Caron, C. Francis, T. Nemoto, T. Poon, S. Mazzucco, actions-user, N. Tappy, N. Cautaerts, S. Somnath, T. Slater, M. Walls, F. Winkler, and H. W. Ånes, "hyperspy/hyperspy: Release v1.7.3," `10.5281/ZENODO.7263263` (2022), `https://zenodo.org/record/7263263`.

[100] L. Najman and M. Schmitt, "Watershed of a Continuous Function," Signal Processing **38**, 99 (1994).

[101] T. Friedrich, C.-P. Yu, J. Verbeeck, and S. Van Aert, "Phase Object Reconstruction for 4D-STEM using Deep Learning," Microscopy and Microanalysis **29**, 395 (2023), `https://doi.org/10.1093/micmic/ozac002`.

[102] S. Haastrup, M. Strange, M. Pandey, T. Deilmann, P. S. Schmidt, N. F. Hinsche, M. N. Gjerding, D. Torelli, P. M. Larsen, A. C. Riis-Jensen, J. Gath, K. W. Jacobsen, J. Jørgen Mortensen, T. Olsen, and K. S. Thygesen, "The Computational 2D Materials Database: high-throughput modeling and discovery of atomically thin crystals," 2D Materials **5**, 42002 (2018), `https://dx.doi.org/10.1088/2053-1583/aacfc1`.

[103] M. N. Gjerding, A. Taghizadeh, A. Rasmussen, S. Ali, F. Bertoldo, T. Deilmann, N. R. Knøsgaard, M. Kruse, A. H. Larsen, S. Manti, T. G. Pedersen, U. Petralanda, T. Skovhus, M. K. Svendsen, J. J. Mortensen, T. Olsen, and K. S. Thygesen, "Recent progress of the Computational 2D Materials Database (C2DB)," 2D Materials **8**, 44002 (2021), `https://dx.doi.org/10.1088/2053-1583/ac1059`.

[104] B. H. Savitzky, I. El Baggari, C. B. Clement, E. Waite, B. H. Goodge, D. J. Baek, J. P. Sheckelton, C. Pasco, H. Nair, N. J. Schreiber, J. Hoffman, A. S. Admasu, J. Kim, S.-W. Cheong, A. Bhattacharya, D. G. Schlom, T. M. McQueen, R. Hovden, and L. F. Kourkoutis, "Image registration of low signal-to-noise cryo-STEM data," Ultramicroscopy **191**, 56 (2018), `https://www.sciencedirect.com/science/article/pii/S0304399117304369`.

[105] C.-H. Lee, A. Khan, D. Luo, T. P. Santos, C. Shi, B. E. Janicek, S. Kang, W. Zhu, N. A. Sobh, A. Schleife, B. K. Clark, and P. Y. Huang, "Deep Learning Enabled Strain Mapping of Single-Atom Defects in Two-Dimensional Transition Metal Dichalcogenides with Sub-Picometer Precision," Nano Letters **20**, 3369 (2020), `https://doi.org/10.1021/acs.nanolett.0c00269`.

[106] H. J. Lipkin, "Physics of Debye-Waller Factors," 1 (2004), `http://arxiv.org/abs/cond-mat/0405023`.

[107] E. M. Mannebach, R. Li, K. A. Duerloo, C. Nyby, P. Zalden, T. Vecchione, F. Ernst, A. H. Reid, T. Chase, X. Shen, S. Weathersby, C. Hast, R. Hettel, R. Coffee, N. Hartmann, A. R. Fry, Y. Yu, L. Cao, T. F. Heinz, E. J. Reed, H. A. Dürr, X. Wang, and A. M. Lindenberg, "Dynamic Structural Response and Deformations of Monolayer MoS2 Visualized by Femtosecond Electron Diffraction," Nano Letters **15**, 6889 (2015).

[108] J. J. Mortensen, L. B. Hansen, and K. W. Jacobsen, "Real-space grid implementation of the projector augmented wave method," Physical Review B **71**, 35109 (2005), `https://link.aps.org/doi/10.1103/PhysRevB.71.035109`.

[109] H. J. C. Berendsen, J. P. M. Postma, W. F. van Gunsteren, A. DiNola, and J. R. Haak, "Molecular dynamics with coupling to an external bath," The Journal of Chemical Physics **81**, 3684 (1984), `https://doi.org/10.1063/1.448118`.

[110] R. Okuta, Y. Unno, D. Nishino, S. Hido, and C. Loomis, "CuPy: A NumPy-compatible library for NVIDIA GPU calculations," Workshop on Machine Learning Systems (LearningSys) at Neural Information Processing Systems (NIPS), 1 (2017), `http://learningsys.org/nips17/assets/papers/paper_16.pdf`.

Amid the rise of *artificial intelligence* (AI), the world is beginning to witness the power of *big-data driven machine learning* models in many contexts. From your smart phone to your bank, machine learning models are paving the way we interact with our surroundings and make decisions that should maximise our safety, happiness, and fortune in the years to come. Fret not, as there aren't any indications of an AI takeover of society, but merely tools that can automate much of the work we're already doing.

Companies and research groups across the globe are investigating the ways machine learning can bring value to their work - so what about the field of *electron microscopy*?

Most are familiar with the optical microscope from their high school labs. The electron microscope made its debut in the early 1930s and earned its creators the Nobel prize due to its ability to visualise objects at the atomic scale – that's around 0.000000001 of a metre – where individual atoms are visible. The electron microscope is the go-to machine for companies and researchers to characterise and study materials. For some it may be to determine the quality of the synthesis of a given material and for others it may be to investigate what interactions happen between a material's atoms with atoms from a surrounding gas.

Despite the benefits of *electron microscopy*, it remains a daunting task to operate these microscopes and often requires years of expertise and countless hours at the lab. This thesis explores the applications of *deep learning* to assist microscopists on their quest to gather data from the atomic scale. *Deep learning* models – known as *neural networks* - are models inspired by the functions of the human brain and have proven themselves powerful at capturing complex features in images of any kind.

This Ph.D. explores tools for microscopists to utilise the power of *deep learning neural networks* to automate the analysis of *high-resolution transmission electron microscopy* images and extract *large-scale data*. Tools have been investigated to provide microscopists with live approximate chemical information to help them highlight areas of particular interest before manual analysis. An investigation has also been made to see whether *neural networks* can exceed human-level performance in seeing past noise and imperfections in the image and still extract the same quality of information.

Technical University of Denmark
DTU Physics
Department of Physics

Fysikvej 311
2800 Kongens Lyngby, Denmark
Phone: +45 4525 3344
info@fysik.dtu.dk

www.fysik.dtu.dk