Machine Learning-Based High-Throughput Processing of Chemical Imaging Data

Hongyang Dong

Submitted in fulfilment of the requirements

for the award of the degree of

Doctor of Philosophy

Department of Chemistry

University College London

11th September 2023

Declaration

I, Hongyang Dong, confirm that the work presented in my thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Abstract

Experimental powder diffraction patterns have improved dramatically in quality and volume due to advancements in X-ray sources, optics, and detectors over the past decades. During this PhD, we have established a machine learning-based chemical image analysis pipeline to accelerate and enhance the acquisition of chemical imaging datasets. Phase Quantification Neural Network (PQ-Net) is a regression, deep convolutional neural network that provides quantitative analysis of powder X-ray diffraction patterns and has been applied to recently acquired data to accelerate data analysis. The network is trained with simulated datasets and tested with experimental X-ray diffraction computed tomography (XRD-CT) datasets and shows great potential as a tool for real-time analysis of diffraction data during *in situ* experiments due to its ability to yield results at least ten times faster (i.e. minutes rather than hours).

We also used a self-supervised learning approach, the SingleDigit2Image (SD2I) network, to overcome the problem related to insufficient projections. The network is tested with simulated data and experimental synchrotron X-ray micro-tomography and XRD-CT data. Statistical analysis revealed that the results are more accurate than the images reconstructed by filtered back projection (FBP) and iterative algorithms. Based on the SD2I, we developed the ParallaxNet to solve the parallax artefact in XRD-CT images. The parallax artefact arises from the incorrectly recorded diffraction angle caused by the different starting positions of scattered beams, which is the main barrier to applying the XRD-CT to samples on the scale of centimetres. The ParallaxNet is tested with one simulated and two real experimental datasets and correctly refined the diffraction reflections in all pixels.

Furthermore, a new 3D image decomposition method Self2Comp is developed to decompose the XRD-CT image with the standard diffraction patterns of chemical components. This method requires no prior knowledge about the sample and can extract more components than conventional image-decomposing methods like non-negative matrix factorization (NMF).

Statement of Impact

The advancements in this PhD research represent a significant leap forward in the field of X-ray powder diffraction analysis and computed tomographic (CT) artefact removal. By leveraging machine learning and neural network technologies, the work promises to revolutionise the way chemical imaging datasets are analysed and applied. Specifically, the Phase Quantification Neural Network (PQ-Net) can accelerate the powder diffraction data analysis, providing unprecedented speed that opens the door to real-time, *in situ* experiments.

Beyond speed, our research addresses critical limitations in the field. For instance, the SingleDigit2Image (SD2I) network improves the accuracy of chemical imaging data and overcomes the challenge of insufficient projections. Our work even extends to resolving issues that have previously inhibited the application of X-ray diffraction computed tomography (XRD-CT) on a larger scale, such as the parallax artefact, through the development of ParallaxNet.

The impact of these innovations is not confined to academic circles; it has substantial real-world implications. Speeding up and improving the accuracy of diffraction data analysis will inevitably benefit a range of sectors, including material science, healthcare, and manufacturing. Since the publication of our paper, PQ-Net has not only gained academic recognition but has also been officially licensed for broader applications. This advanced technology is now being employed in new experiments and has received overwhelmingly positive responses. By resolving long-standing technical issues on synchrotron CT techniques and dramatically increasing the speed of analysis, this research sets the stage for applications that could change the way we understand and interact with material studies.

UCL Research Paper Declaration Form
referencing the doctoral candidate's own published work(s)

Please use this form to declare if parts of your thesis are already available in another format, e.g. if data, text, or figures:

- *have been uploaded to a preprint server*
- *are in submission to a peer-reviewed publication*
- *have been published in a peer-reviewed publication, e.g. journal, textbook.*

This form should be completed as many times as necessary. For instance, if you have seven thesis chapters, two of which containing material that has already been published, you would complete this form twice.

1. For a research manuscript that has already been published (if not yet published, please skip to section 2)

a) **What is the title of the manuscript?**

A deep convolutional neural network for real-time full profile analysis of big powder diffraction data

b) **Please include a link to or doi for the work**

https://doi.org/10.1038/s41524-021-00542-4

c) **Where was the work published?**

Npj Computational Materials

d) **Who published the work?** (e.g. OUP)

Nature Publishing Group

e) **When was the work published?**

21 May 2021

f) **List the manuscript's authors in the order they appear on the publication**

Hongyang Dong, Keith T. Butler, Dorota Matras, Stephen W. T. Price, Yaroslav Odarchenko, Rahul Khatry, Andrew Thompson, Vesna Middelkoop, Simon D. M. Jacques, Andrew M. Beale and Antonis Vamvakeros

g) **Was the work peer reviewed?**

Yes

h) **Have you retained the copyright?**

Yes

i) **Was an earlier form of the manuscript uploaded to a preprint server?** (e.g. medRxiv). If 'Yes', please give a link or doi)

No

If 'No', please seek permission from the relevant publisher and check the box next to the below statement:

☒

*I acknowledge permission of the publisher named under **1d** to include in this thesis portions of the publication named as included in **1c**.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)

   a) **What is the current title of the manuscript?**
   b) **Has the manuscript been uploaded to a preprint server?** (e.g. medRxiv; if 'Yes', please give a link or doi)
   c) **Where is the work intended to be published?** (e.g. journal names)
   d) **List the manuscript's authors in the intended authorship order**
   e) **Stage of publication** (e.g. in submission)

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)

   H.D. and A.V. designed and implemented the PQ-Net. H.D., A.V. and S.D.M.J. created the MATLAB, Python and TOPAS scripts for the creation of the diffraction libraries. S.W.T.P. and Y.O. were responsible for aspects of testing. H.D., A.V. and D.M. performed the Rietveld analysis of the XRD-CT data. K.B. advised on the deep ensembles while R.K. and A.T. advised on the CNN architectures. A.V., S.D.M.J., D.M., V.M. and A.M.B. designed the XRD-CT experiments and acquired the tomographic data. H.D. developed and tested the deep ensembles. H.D. and A.V. are responsible for writing the manuscript with feedback given by all authors. S.D.M.J., A.M.B. and A.V. directed the research.

4. **In which chapter(s) of your thesis can this material be found?**

   Chapter 3

5. **e-Signatures confirming that the information above is accurate** (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work)

   Candidate

   Hongyang Dong

   Date:

   20/09/2023

   Supervisor/ Senior Author (where appropriate)

   Andrew M. Beale

   Date

   20/09/2023

UCL Research Paper Declaration Form

referencing the doctoral candidate's own published work(s)

Please use this form to declare if parts of your thesis are already available in another format, e.g. if data, text, or figures:

- *have been uploaded to a preprint server*
- *are in submission to a peer-reviewed publication*
- *have been published in a peer-reviewed publication, e.g. journal, textbook.*

This form should be completed as many times as necessary. For instance, if you have seven thesis chapters, two of which containing material that has already been published, you would complete this form twice.

1. For a research manuscript that has already been published (if not yet published, please skip to section 2)

a) **What is the title of the manuscript?**

A scalable neural network architecture for self-supervised tomographic image reconstruction

b) **Please include a link to or doi for the work**

https://doi.org/10.1039/D2DD00105E

c) **Where was the work published?**

Digital Discovery

d) **Who published the work?** (e.g. OUP)

The Royal Society of Chemistry

e) **When was the work published?**

02 Jun 2023

f) **List the manuscript's authors in the order they appear on the publication**

Hongyang Dong, Simon D. M. Jacques, Winfried Kockelmann, Stephen W. T. Price, Robert Emberson, Dorota Matras, Yaroslav Odarchenko, Vesna Middelkoop, Athanasios Giokaris, Olof Gutowski, Ann-Christin Dippel, Martin von Zimmermann, Andrew M. Beale, Keith T. Butler and Antonis Vamvakeros

g) **Was the work peer reviewed?**

Yes

h) **Have you retained the copyright?**

Yes

i) **Was an earlier form of the manuscript uploaded to a preprint server?** (e.g. medRxiv). If 'Yes', please give a link or doi)

doi:10.26434/chemrxiv-2022-xnlqz-v2

If 'No', please seek permission from the relevant publisher and check the box next to the below statement:

☐

*I acknowledge permission of the publisher named under **1d** to include in this thesis portions of the publication named as included in **1c**.*

2. **For a research manuscript prepared for publication but that has not yet been published** (if already published, please skip to section 3)

   a) **What is the current title of the manuscript?**
   b) **Has the manuscript been uploaded to a preprint server?** (e.g. medRxiv; if 'Yes', please give a link or doi)
   c) **Where is the work intended to be published?** (e.g. journal names)
   d) **List the manuscript's authors in the intended authorship order**
   e) **Stage of publication** (e.g. in submission)

3. **For multi-authored work, please give a statement of contribution covering all authors** (if single-author, please skip to section 4)

   H. D. designed and implemented the self-supervised reconstruction CNNs with contributions from A. V., K. T. B. and discussions with W. K. H. D. and A. V. prepared the SD2I python scripts and notebooks for the nDTomo software suite. H. D., A. V. and R. E. performed the various tests involving the reconstruction CNNs. S. W. T. P., R. E., Y. O. and A. G. were responsible for aspects of testing. A. V., D. M., Y. O., V. M., S. W. T. P., S. D. M. J. and A. M. B. designed the micro-CT and XRD-CT experiments and acquired the tomographic data. O. G., A. C. D. and M. Z. were responsible for P07 instrumentation and setup at the PETRA III, DESY. H. D., A. V. and K. T. B. are responsible for writing the manuscript with feedback given by all co-authors.

4. **In which chapter(s) of your thesis can this material be found?**

   Chapter 4

5. **e-Signatures confirming that the information above is accurate** (this form should be co-signed by the supervisor/ senior author unless this is not appropriate, e.g. if the paper was a single-author work)

   Candidate

   Hongyang Dong

   Date:

   20/09/2023

   Supervisor/ Senior Author (where appropriate)

   Andew M. Beale

   Date

   20/09/2023

Acknowledgements

I would like to express my deepest thanks to my supervisor, Prof. Andrew Beale, for his invaluable guidance and for providing me the opportunity to participate in this exciting project. I am also profoundly grateful to Dr. Antony Vamvakeros for his consistent support and prompt replies throughout this project. From our in-depth discussions on machine learning methods to his advice on chemical imaging techniques, working with you has been a truly delightful experience. Special thanks to my co-supervisor, Prof. David Scanlon. Although we were only able to meet online due to the pandemic, his patience and extensive knowledge have been invaluable assets on my PhD journey.

I also wish to offer my sincere thanks to all the staff at Finden Ltd. In particular, I am indebted to Dr. Simon Jacques for his invaluable mental and academic support throughout my PhD journey. The opportunity he provided for industry collaboration has been an invaluable experience. Many thanks go to Dr. Keith Butler for his generous assistance with AI algorithms, which have significantly benefited my PhD projects. I would like to express gratitude to all my colleagues and friends at the Research Complex at Harwell, especially to Chengwu, Sining, and Yiyi, for their incredible support while I was living abroad.

Additionally, I would like to thank my closest friends in China—Tianhao, Junzhe, Ziqiang, Boyuan, Xudong, and his wife Xilin—for your enduring and patient support, which has helped me overcome countless challenges. Special thanks to Chenyu Wang in Toronto; you have always been there when I needed you, helping me remain positive and focused in my life.

Last but not least, my deepest thanks go to all my family members in China, especially my mother, Mei Jia, and my father, Yujie Dong. I could not have completed nine years of study in the UK without your love, understanding, and support. I am also thankful for the companionship of my cat, Dino, during the lockdown. Finally, I wish to extend special thanks to my girlfriend, Xinyi Zhong. You have stood by my side for the past seven years. We have both invested

tremendous effort into our individual goals and our shared future, and I eagerly look forward to celebrating this joy with you.

This PhD thesis is dedicated to my grandparents.

Publications

- Jacquot, C. *et al.* A multi-scale study of 3D printed Co-Al2O3 catalyst monoliths versus spheres. 100538 (2023) doi:10.1016/j.ceja.2023.100538.
- Dong, H. *et al.* A scalable neural network architecture for self-supervised tomographic image reconstruction. *Digit. Discov.* **2**, 967–980 (2023).
- Matras, D. *et al.* Emerging chemical heterogeneities in a commercial 18650 NCA Li-ion battery during early cycling revealed by synchrotron X-ray diffraction tomography. *J. Power Sources* **539**, 231589 (2022).
- Dong, H. *et al.* A deep convolutional neural network for real-time full profile analysis of big powder diffraction data. *Npj Comput. Mater.* **7**, 1–9 (2021).
- Cycling Rate‐Induced Spatially‐Resolved Heterogeneities in Commercial Cylindrical Li‐Ion Batteries - Vamvakeros - 2021 - Small Methods - Wiley Online Library. https://onlinelibrary.wiley.com/doi/full/10.1002/smtd.202100512.
- Vamvakeros, A. *et al.* DLSR: a solution to the parallax artefact in X-ray diffraction computed tomography data. *J. Appl. Crystallogr.* **53**, (2020).

Table of Contents

Glossary of Acronyms

Term	Defination
AdaGrad	Adaptive Gradient
AdaIN	Adaptive Instance Normalisation
Adam	Adaptive Moment Estimation
ART	Algebraic Reconstruction technique
ARU-Net	Artefact Removal U-Net
AI	Artificial Intelligent
ANN	Artificial Neural Networks
AUTOMAP	Automated Transform by Manifold Approximation
CT	Computed Tomography
CGLS	Conjugate Gradient Least Squares
CBAM	Convolutional Block Attention Module
CONV	Convolutional layers
CNN	Convolutional Neural Networks
DCNN	Deep Convolutional Neural Networks
DNN	Deep Neural Networks
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DFR	Direct Fourier Reconstruction
ESRF	European Synchrotron Radiation Facilities
ELU	Exponential Linear Unit
EBS	Extremely Brilliant Source
FFT	Fast Fourier Transform
FBP	Filtered Back Projection
FWHM	Full Width at Half Maximum

FC	Fully Connected layers (Dense layers)
GAN	Generative Adversarial Network
GB	Gigabyte
GPUs	Graphics Processing Units
GT	Ground Truth
HAC	Hierarchical Agglomerative Clustering
IR	Infrared
ICSD	Inorganic Crystal Structure Database
KL-NMF	Kullback-Leibler NMF
LHS	Left Hand Side
ML	Machine Learning
MR	Magnetic Resonance
POOL	Max-Pooling layers
MAE	mean absolute error
MSE	mean squared error
MCR-ALS	Multivariate Curve Resolution-Alternating Least-Squares
NIST	National Institute of Standards and Technology
NN	Neural networks
NMF	Non-negative Matrix Factorization
NMR	Nuclear Magnetic Resonance
OVO	One-versus-One
Optimizer	optimisation function
OGCSC	Oriented Grouping-Constrained Spectral Clustering
PDF-CT	Pair Distribution Function Computed Tomography
PQ	Parameter Quantification
POX	Partial Oxidation of Methane

PSNR	Peak Signal-to-Noise Ratio
PQ-Net	Phase Quantification Neural Network
PURE	Poisson Unbiased Risk Estimator
PCA	Principal Component Analysis
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RNN	Recurrent Neural Network
RHS	Reft Hand Side
RED-Net	Residual Encoder-Decoder Network
SD2If	SD2I uses only full image resolution
SD2Iu	SD2I with upsampling layers
Self2Comp	Self to Components Archtiecture
SART	Simultaneous Algebraic Reconstruction Technique
SIRT	Simultaneous Iterative Reconstruction Technique
SCXRD	Single-Crystal X-ray Diffraction
SD2I	Single-Digit-to-Image
SAXS	Small-Angle X-ray Scattering
SPCA	Smoothed PCA
SOFCs	Solid Oxide Fuel Cells
SGD	Stochastic Gradient Descent
SSIM	Structural Similarity Index Measure
SVMs	Support Vector Machines
t-SNE	t-distributed Stochastic Neighbor Embedding
TB	Terabyte
TEM	Transmission Electron Microscopy
VGG	Visual Geometry Group

WGS	Water Gas Shift
X-ray CT	X-ray Absorption Computed Tonography
micro-CT	X-ray Absorption CT on micrometer scale
nano-CT	X-ray Absorption CT on nanometer scale
XAS	X-ray Absorption Spectroscopy
XRD	X-ray Diffraction
XRD-CT	X-ray diffraction patterns
XRF	X-ray Fluorescence
XRF-CT	X-ray Fluorescence Computed Tomography

List of Symbols

Symbol	**Defination**
λ	Wavelength
Å	Angstrom
ppm	Parts per million
n	An integer that represents the order of reflection.
d	Spacing between the crystal planes
θ	Incident angle between the beam and the normal to the lattice planes
Q	Momentum change
I_o	Intensity of the incident beam
I_t	Intensity recorded on the detector

μ	Linear attenuation coefficient
x_{path}	The distance of the beam travels through the sample
npr	Number of projections
ntr	Number of translation steps
$\mathbf{A}$	Weight matrix
x_j, y_j, z_j	Positions of the *j-th* atom in the unit cell
O_j	Occupancy of the *j-th* atom
h, *k*, *l*	Miller indices
f_j	Scattering factor of the *j-th* atom
F^2_{hkl}	Structure factor
L	Lorentz correction factor
P	Polarisation correction factor
A	Absorption correction factor
j_{hkl}	Factor that depends on the symmetry of the material
s	Combination of the remaining factors in the structure factor equation
I(2θ)	Intensentiy function according to the 2θ diffraction angle
$\mathbf{L}$	Normalised Lorentzian function
$\mathbf{G}$	Normalised Gaussian function
H	Peak width at FWHM
η	Factor in Pesudo-Voiget peak shape to control the ratio between the Lorenzen part and the Gaussian part.

w	Half of the peak width at FWHM
D	Crystallite size
B	Constant in the peak shape functions normally between 0.89 and 0.94.
I_{tot}	Final simulated intensities at each angle
y_i^{obs}	Observed intensity at point *i* on the experimental diffraction pattern
y_i^{sim}	Simulated intensity at point *i* on the experimental diffraction pattern
w_i	Weight of each observed value
R_p	Profile R-factor
R_{wp}	Weighted Profile R-factor
R_{exp}	Expected R-factor
N	Number of observation points
P	Number of parameters that are refined
C	Number of constraints
a	Positive coefficient controls the shape of the Leaky ReLu activation function
b	Positive coefficient controls the shape of ELU activation function
x	Element in the output vector of a layer
β_1	Constant in Adam optimizor, controls the first moment bias correction
β_2	Constant in Adam optimizor, controls the second moment bias correction
α	Learning rate in Adam optimizor

ε	Small constant in Adam optimizor to prevent the squaroot to be zero
L_D	dynamic range of the input images
K_1, K_2	Constants in the SSIM loss function
$\mu_{\mathbf{x}}, \mu_{\mathbf{y}}$	Mean value among the input vector **x** and **y**
σ	Standard deviation
x, y	Input vectors (images) to the loss functions
p_{ic}	confidence that the sample *i* belongs to the cluster c.
L_{MSE}	Mean squared error loss function
L_{MAE}	Mean absolute error loss function
L_{SSIM}	Structural Similarity Index Measure loss function
$L_{\text{Cross-Entropy}}$	Cross-entropy loss function
r	Fraction controls the ratio of MSE and SSIM parts in the joint loss function
m, n	Image sizes
k	Factor controls the size of SD2I architectures
c	Number of diffraction channels
G(a)	Generated reconstruction image
a_c	control constant sent into the generator
R_S	Ratio between the sample to detector distance and the cos(2θ)
θ_{middle}	The diffraction angle on the detector from the centre of the sample
θ_{new}	Simulated diffraction angle after encountered the parallax artefact

$\mathbf{c}_i$	Position of the centroid on the cluster i.
Si	Cluster i.
O	Big O notation
A	Adjacency matrix
L	Graph Laplacian matrix
D	Diagonal degree matrix
L_{rw}	Random walk normalised Laplacian
$P_{\min}$	Precision of the cluster with the smallest size
P_{avg}	Average precision of all clusters
V	Original dataset matrix
W	Matrix including a set of basis images
H	Matrix of coefficients for each basis image
f	Vector of importance of each cluster predicted by Self2Comp.
S	Simulated matrix

List of Figures

List of Tables

Chapter 1: Introduction

1.1 Motivation

During the past decades, there has been a significant rise in the quantity and quality of chemical imaging data due to progress in beam sources, optics, and detector technologies. Nevertheless, these developments have a downside which involves handling the large amount of data gathered in these experiments [1]–[3]. Taking modern synchrotron experiments as an example, with the emergence of fourth-generation synchrotron facilities globally, the potential to obtain numerous Terabytes (TBs) of X-ray diffraction (XRD) data per dynamic experiment is now a reality [4]. Traditional techniques for data analysis, such as least-squares minimization methods, are unable to keep pace with the rapid rate of XRD data collection. As a result, there is a demand for alternative methods that can deliver results that are both swift and precise [5].

While experimental equipment continues to develop, the role of machine learning (ML) in addressing big data challenges is constantly being explored by experts [6]. ML generally involves using algorithms to teach the computer how to learn based on a set of experimental or simulated data and give certain assumptions, and it allows the computer to perform tasks that are not specifically programmed [7]. In many fields of research, ML has demonstrated its ability to derive more accurate and reliable results than humans, especially in problems that involve repetition, specialization, and big data [8]. Big data often contains many variables and factors, and the relationships between them are complex and difficult to discern by traditional methods. However, ML can find their connections and potential causal relationships, thus discovering hidden information and patterns in the data [9].

It is well recognised that ML approaches, especially deep neural networks, have great potential to be applied to chemical imaging data in terms of fast analysis of experimental data [10], enhancing image resolution [11], and solving common artifacts that affect data quality [12]–[14]. Previous work only used ML

techniques in one or two stages of data acquisition. Although they have achieved impressive results, the lack of comprehensive analysis of data from beginning to end often leads to models that do not consider the needs of people under laboratory conditions in terms of model size, accuracy, or generalization [15]–[17]. This has resulted in the situation that although new models are continuously developed, people still find it difficult to use them in real experiments.

Therefore, we saw great potential to build up a new chemical imaging analysing pipeline that used ML methods on all procedures that are currently labour-intensive. We first focused on X-ray diffraction computed tomography (XRD-CT), because it is a relatively new and advanced computed tomography (CT) method that contains many similar components to other CT methods and therefore suffers from a big data problem that is ripe for tackling using ML methods [18]. Subsequently, the learnings from the application of ML methods for XRD-CT can potentially be applied to more CT methods like neutron-CT and X-ray absorption CT (X-ray CT).

1.2 Chemical Imaging

Chemical imaging is an approach that enables the visualization of the spatial organization and concentration of chemical elements within a sample, often down to the atomic level. By merging principles from analytical chemistry, microscopy, and imaging, this technique delivers comprehensive data on the location of various chemical compounds within the sample [19]. Chemical imaging is versatile, as it can be applied to diverse materials such as solids, liquids, and gasses, and is utilized across fields like materials science [19], pharmaceuticals [20], environmental science [21], and biology [22]. Different chemical imaging methods can be classified by their sources used, the interaction between the sources and the sample, and the data acquisition/processing methods [23]–[25].

Wave sources can interact with the matter in a range of different ways which include absorption, emission, scattering, and diffraction. Chemical imaging methods will often quantify these interactions, and generally, different detection setups are needed to detect different types of interactions [26]–[28].

For example, infrared (IR) spectroscopy [29] measures the absorption of IR rays (λ ~ μm) by the sample. Data on the molecular vibrations can be obtained, which is linked to the chemical signature of the molecules, and chemical bonds present [30]. IR spectroscopy has a wide range of applications due to its ability to identify and study chemical substances [31]–[33]. In IR image processing, AI is being extensively used in image classification [34], image enhancement [35], and both quantitative and qualitative data analysis [36].

The X-ray absorption spectroscopy (XAS) used the transmission/absorption of different energies (λ ~ Å) of infrared and X-ray beams on the sample to characterize the material being measured. During XAS, an adjustable X-ray source, like a synchrotron radiation source, produces monochromatic X-rays with a range of energies [26]. As the energy of the X-rays is adjusted, the sample takes in the X-rays, ultimately leading to the excitation of core electrons to unoccupied electronic states. The X-ray absorption spectrum is derived by evaluating the absorption coefficient in relation to the energy of the incoming X-rays [37]. Moreover, since light absorption is directly proportional to the path length and the concentration of the solution (i.e the Beer Lambert Law), X-ray absorption with mass attenuation is an essential physical mechanism employed in medical imaging to produce images that depict the internal structures of diverse organs [38].

Emission is another interaction between the source light and the sample that is applied in chemical imaging. Fluorescence microscopy is a typical emission-based technique [39]. This method works by illuminating a sample with light of a particular wavelength, which excites certain molecules in the sample. After some non-radiative transition, these molecules can emit

fluorescence at a longer wavelength (lower energy). By detecting the emitted light, it is possible to visualize and analyze the distribution of fluorescent molecules. Fluorescence microscopy methods normally use UV-visible radiation (λ ~ nm) as the source and measure the UV-Visible or near-infrared light response at the detectors. This technique normally provides information on molecular properties, such as structure, environment, or concentration [39]. The X-ray fluorescence (XRF) imaging technique is based on the same phenomenon of the visible light fluorescence techniques, however, with an excitation source in the Å and which is, therefore, higher in energy, can penetrate deeper into the sample [40]. XRF allows for analysing the elemental composition information of a sample in a non-destructive way and with higher sensitivity (~ ppm).

1.3 Chemical Computed Tomographic (CT) Methods

1.3.1 X-Ray Absorption CT

X-ray absorption CT, when the pixel size of the recreated images is on the micrometre scale (typically 1-10 μm), is also known as micro-CT or μ-CT [41]. This technology results in a three-dimensional matrix. The original reconstructed data volume can be utilized for localized characterization, such as identifying morphological features in specific sample areas, or even as a model (after segmentation and mesh creation) for assessing the material's transport properties [42]. Recent advancements in full-field X-ray microscopy or nano-CT have facilitated data collection with voxel sizes on the nanometer scale (tens to hundreds of nm) [43]. This high-resolution X-ray CT data enables the examination of a sample's microstructure and direct measurement of its physical properties, such as porosity, tortuosity, and pore size distribution. X-ray CT is widely used in material science and functional materials studies. This is especially apt for examining the microstructural changes caused by fatigue. X-ray CTs allow us to investigate the materials in a non-destructive way and allows for repeated examination of the same sample over time without causing damage [44]. Chowdhury et al. analyzed the micro-scale damage in

carbon-carbon composites, intending to apply their findings in the aerospace sector [45]. Another recent study employed in situ micro-CT to investigate the mechanism of fracture initiation and propagation in lightweight cement foams, a new building material with high strength and low density. The findings revealed a distinct correlation between the occurrence of cracks and the presence of two standard hollow glass microspheres [46]. *In situ* micro-CT can also be utilized for biomimetic material analysis. In 2018, Wang et al. performed *in situ* compression stress tests on pomelo peel. By employing micro-CT to analyze the hierarchical structure of the peel, they investigated its physical properties in terms of stress resistance, damping ability, and energy absorption [47].

Micro-CT imaging enables a thorough examination of the intricate internal structure of energy storage devices, ranging from tiny batteries to complete fuel cells, providing a high level of detail. This technique proves useful in quality control processes and aids in identifying defective components [48]–[50]. In 2016, Finegan et al. used both *operando* and multi-scale micro-CT to analyse the catastrophic breakdown of lithium-ion batteries that can happen at various length scales and within a short time frame [51]. This comprehensive approach allows for the investigation of the interdependent dynamic structural, thermal, and electrical responses associated with the degradation process. Eastwood et al. applied both micro-CT and nano-CT to studying low atomic number materials that are commonly found in Li-ion battery electrodes [52]. In this paper, they quantitively extracted crucial characteristics within a statistically representative region of interest, encompassing over a thousand particles. Meanwhile, X-ray CT is also widely applied in studying solid oxide fuel cells (SOFCs) [53]. Nano-CT, has become a prevalent tool for exploring the microstructure around the Nickel electro-catalyst located at the anode [54]. This full-field X-ray microscopy technique can deliver reconstructed data with a spatial resolution of less than 50 nanometers. Nano-CT not only makes it feasible to measure physical properties but also enables an investigation into the interconnectivity, triple phase boundary, and the relative volume of the various phases present in the sample.

1.3.2 X-Ray Diffraction Computed Tomography (XRD-CT)

XRD-CT couples XRD with a scanning CT approach using a monochromatic pencil beam. Then it can re-create a 3D model of a physical object layer by layer without destroying the original sample [55]. Harding et al. initially showed the principle of XRD-CT in 1987 with a laboratory diffractometer [56]. Over the past ten years, however, it has become clear that its full capacity for materials characterization can be leveraged using a synchrotron beam [57]. In conventional X-ray tomography, the images that are obtained give maps of density within the object and the composing pixels contain single grayscale values. In scatter-based tomography, each pixel instead contains a scattering pattern or equivalent chemical signal (a 1D array (or higher) of numbers). An XRD-CT image can show the X-ray diffraction patterns on each pixel of the reconstructed image. X-ray scatter-based tomography allows unprecedented insight into the chemical and physical state of functional materials and devices. Such tomographic approaches can be used as research tools but also offer the potential for routine scanning for inspection systems and medical scanning [18]. An X-ray scatter-based tomography slice is a data cube with two spatial dimensions and one scattering dimension.

This form of chemical tomography, similar to techniques like X-ray Fluorescence Computed Tomography (XRF-CT) and Pair Distribution Function Computed Tomography (PDF-CT), is capable of detecting both the chemical state and physical form of the substances present within the system [58]. This makes it an extremely valuable resource for characterizing catalyst reactions and heterogeneous materials. Jacques et al. first conducted an *in situ* XRD-CT study, examining the evolution of Ni species during the activation of a Ni/Al_2O_3 hydrogenation catalyst [59]. They claimed that this method has the potential to facilitate the chemical imaging of substances inside bulk objects that are ongoing physicochemical changes. Beale et al. utilized XRD-CT technology to study the impact of sulfur (S) on $Cu/ZnO/Al_2O_3$ catalysts that are commonly used in Water Gas Shift (WGS) reactions [60]. The spatially-resolved diffraction

patterns derived from the XRD-CT examinations disclosed that Cu and Zn sulfides were formed during the H_2S treatment. The study also found that the thickness of the CuS and ZnS shells increased as the concentration of H_2S increased. In 2022, Matras et al. applied the XRD-CT on Li-ion cylindrical batteries under charge/discharge conditions [61]. By analysing the spatially-resolved diffraction patterns, numerous chemical inconsistencies associated with the distribution of lithium in both the cathode and anode were determined, and they correctly detected the lattice parameter on the cathode around the aluminium tab. Moreover, Vamvakeros et al. performed XRD-CT on an operational catalytic membrane reactor used for the oxidative coupling of methane [57]. Their findings showed the significance of evolving solid-state chemistry during the catalytic reaction. Middelkoop et al. used XRD-CT to study the reactivity of 3D-printed Ni/Al_2O_3 monolithic structures in relation to the methanation reaction of CO_2 [55]. They claimed that the XRD-CT disclosed the spatial changes in physico-chemical characteristics under operating conditions.

1.4 Machine Learning in Chemical Image Processing

In recent years, due to the continuous development of experimental equipment and their precision, the size of experimental data has increased at an astonishing speed. Analyzing the increasingly large volume of experimental data sometimes cannot be done using traditional data analysis methods, because they often fall short in terms of speed and feasibility [62]. Machine learning (ML), as an emerging discipline, is increasingly favoured in the processing of large chemical data and high-throughput chemical image analysis [7], [63].

Neural networks (NN) are a crucial machine learning technique with significant implications in the fields of computer vision and image processing. The earliest idea of NN was proposed in 1943 by McCulloch and Pitts [64], and it has grown in popularity after the publication of the Hopfield networks in 1982 [65], along with the introduction of the backpropagation learning algorithm for multi-layer

networks by Rumelhart et al. in 1986 [66]. In the following two decades, NN have become widely applied in fields such as chemical engineering [67], principle component analysis [68], base function approximation [69], and catalyst design [70]. However, due to the limitation of computer processing power and memory, storage, programming languages, and the design of these early-stage NNs, these early-stage developments in NNs did not have much impact on chemical research [71].

Recently, Convolutional Neural Networks (CNN), a deep learning method, has shown important potential in molecular and material sciences. For instance, Rao and Liu implemented 3D-CNN for heterogeneous material homogenization[72]. They used a computational homogenization approach to generate a high-fidelity training dataset and claimed that their CNN architecture is also suitable for facilitating material design. Ziatdinov and his colleagues meanwhile, successfully applied 2D-CNN to analyse scanning transmission electron microscopy images [73].

By the 2000s, with the emergence of deep neural networks (DCNN) [74] and CNNs [75], the use of AI in image processing and chemical data analysis began to receive increasing attention. These advanced NNs normally have more hidden layers, so that they can handle more complex problems [71]. For example, Zhu et al. proposed the automated transform by manifold approximation (AUTOMAP) CNN architecture that can directly reconstruct CT images from its sinograms [76]. Since this method is a supervised learning technique, training a model requires a large training dataset that mixes simulated and real experimental data, and is tested on another experimental dataset. They utilized the widely recognized ImageNet database for simulating CT images [77]. Since this database contains real-life images and covers all features in the real dataset, the network became more robust and could subsequently be applied to all experimental data that are not seen by the model. The CNN has also been applied to solve more specific problems in

chemical CT images like angular undersampling artefacts [78]–[81], metal artefacts [82], image noise [83]–[85] and blurring [86].

CNN is also often used to analyze chemical data. Nowadays, XRD phase identification has become a common bottleneck in materials characterisation and screening [87]. Assuming we have some previous knowledge about the crystalline phases present in the sample under investigation, phase identification can still take hours for an expert crystallographer [88]. For an XRD-CT image, there are normally thousands or even millions of XRD patterns to be analysed, therefore, their recent research has focused on using machine learning methods for phase identification.

Lee et al. proposed a CNN structure used for XRD pattern identification, which also has the potential to predict fractions of phases approximately [10]. The same group also demonstrated a classification CNN model that can extract crystal-system, extinction-group, and space-group information from XRD patterns [14]. Wang et al. demonstrated a phase identification CNN which was trained with simulated patterns with noise extracted from real experimental data [89]. However, most of the recent works focus on producing classification models for phase identification. In other words, they can only predict whether a phase is present or not in a diffraction pattern. Lee's team expanded its model to roughly classify the fraction range that phases lie on [10]. The authors claimed that their model possesses the fastest training time whilst retaining good accuracy (about 99 %). This structure can predict the weight percentages of 38 phases of crystals in one pattern, within a matter of a few minutes. Furthermore, this model can predict the fraction of one phase lying in the 66-99 % range, 33-66 % range or 0-33 % range as three different outputs.

However, like many other supervised learning methods, CNN phase identification has some important shortcomings [88]. First of all, a trained model cannot predict phases which are not included in the original training dataset, and those new phases may also have a big influence on the performance of

predicting other phases [90]. This disadvantage becomes very dangerous when we attempt to use a CNN model to decipher the phases present in a diffraction pattern, as it can produce completely incorrect results without any warning [87]. Secondly, the input scale and output scale are fixed, so it is hard to get a generally applied CNN model that can adapt to the different Q ranges and the number of bins.

Moreover, chemical image segmentation involves breaking down images of chemical samples, microstructures, or chemical reactions into distinct segments or areas, each representing a unique object or point of interest. The primary goal is to modify or simplify the image's representation to facilitate easier analysis [91]–[93]. The basic concept involves carrying out segmentation through the application of filters on a 2D input image. In the research conducted by Zhang et al., various types of information were presented as 2D images and fed into the CNN's input layer via different image channels (like the Red, Green, and Blue channels) [94]. Leist et al. proposed a 2D CNN architecture that allows for automatic defect identification and strain mapping in transmission electron microscopy (TEM) images [95]. This architecture used multiple residual blocks that enable the gradient to be back-propagated directly to the preceding layers, thus addressing the issue of vanishing gradients. This subsequently facilitates the training of significantly deeper neural networks [96].

The CNN-based approach is not only restricted to the 2D domain however. Aggarwal et al. proposed a 3D CNN architecture to predict the possible binding site on a protein that a ligand can bind to, and then they used another 3D CNN model to visualize the protein [97]. In 2022, Muin used CNN to capture reliable geometrical information on foam structures in 3D micro-CT images. The model explored in this research utilizes supervised learning based on labelled ground truth data, aiming to precisely discern pixel distributions at air-liquid interfaces [98].

1.5 Research Hypothesis and Aims

We have already seen outstanding contributions from ML methods in solving various individual analytical problems and it also becomes more possible to design an image processing method that can solve multiple problems at the same time. In this thesis, I will demonstrate my work on the application of diverse machine learning algorithms for CT image reconstruction. In the following chapters, a study of the feasibility of having a regression CNN model to predict the scale factor, crystallite size and lattice parameters of multiple phases in one XRD pattern simultaneously is demonstrated. The model is trained with at least 50,000 real experimental or simulated patterns. Using modern graphics processing units (GPUs) to accelerate the pattern-simulation process is also an attractive secondary goal, although as back-up, diffraction analysis software such as TOPAS can be used [99]. The simulated patterns should have a wider range of parameters than real experimental data to include all possible experimental patterns. We are assuming that the experimental data are perfect powder diffraction patterns, however, many factors can affect the shape of the real data. Therefore, measures to justify the robustness of a model trained with simulated patterns acting on real experimental data need also to be developed.

The second objective of this thesis is to employ a combination of supervised and unsupervised learning techniques to reconstruct CT images that contain multiple types of artefact, including angular undersampling and parallax artefacts. I will introduce the Single-Digit-to-Image (SD2I) [100] and the ParallaxNet architectures, which are designed to accurately reconstruct CT images using fewer projection data and to handle large sample sizes in XRD-CT, respectively. The reconstructed images were compared with filtered back projection (FBP) results and conventional iterative methods. Meanwhile, I use statistical methods, like the t-distributed Stochastic Neighbor Embedding (t-SNE), to study and visualize the impact of the diversity of training datasets

acting on the trained CNN model, which can also give an indication regarding the extent we can trust the results from a trained CNN.

The last part of this thesis will focus on AI-based image clustering methods. We will introduce the Self-to-Components (Self2Comp) algorithm that can serve as a more effective replacement for conventional data clustering methods such as Non-negative Matrix Factorization (NMF) and Principal Component Analysis (PCA), thereby significantly reducing the number of patterns to be analyzed and accelerating the X-ray Diffraction Computed Tomography (XRD-CT) image processing pipeline. Successfully minimizing the volume of data for analysis could increase the efficiency of the image processing pipeline. The validation of this hypothesis would necessitate the design, application, and comparative evaluation of this new AI-based image clustering method against conventional methods, with emphasis on its capacity to reduce pattern complexity and expedite processing times.

1.6 References

[1] S. J. Lusher, R. McGuire, R. C. van Schaik, C. D. Nicholson, and J. de Vlieg, ‘Data-driven medicinal chemistry in the era of big data’, *Drug Discov. Today*, vol. 19, no. 7, pp. 859–868, 2014.

[2] I. V. Tetko, O. Engkvist, U. Koch, J.-L. Reymond, and H. Chen, ‘BIGCHEM: challenges and opportunities for big data analysis in chemistry’, *Mol. Inform.*, vol. 35, no. 11–12, pp. 615–621, 2016.

[3] H. Zheng, J. Hou, M. D. Zimmerman, A. Wlodawer, and W. Minor, ‘The future of crystallography in drug discovery’, *Expert Opin. Drug Discov.*, vol. 9, no. 2, pp. 125–137, 2014.

[4] G. Pacchioni, ‘An upgrade to a bright future’, *Nat. Rev. Phys.*, vol. 1, no. 2, Art. no. 2, Feb. 2019, doi: 10.1038/s42254-019-0019-5.

[5] C. Wang, U. Steiner, and A. Sepe, ‘Synchrotron big data science’, *Small*, vol. 14, no. 46, p. 1802291, 2018.

[6] V. V. Kolisetty and D. S. Rajput, ‘A review on the significance of machine learning for data analysis in big data’, *Jordanian J. Comput. Inf. Technol. JJCIT*, vol. 6, no. 01, pp. 155–171, 2020.

[7] J. F. Rodrigues, L. Florea, M. C. de Oliveira, D. Diamond, and O. N. Oliveira, ‘Big data and machine learning for materials science’, *Discov. Mater.*, vol. 1, pp. 1–27, 2021.

[8] J. Qiu, Q. Wu, G. Ding, Y. Xu, and S. Feng, ‘A survey of machine learning for big data processing’, *EURASIP J. Adv. Signal Process.*, vol. 2016, no.

1, p. 67, May 2016, doi: 10.1186/s13634-016-0355-x.
[9] D. Bhamare and P. Suryawanshi, ‘Review on Reliable Pattern Recognition with Machine Learning Techniques’, *Fuzzy Inf. Eng.*, vol. 10, no. 3, pp. 362–377, Jul. 2018, doi: 10.1080/16168658.2019.1611030.
[10] J.-W. Lee, W. B. Park, J. H. Lee, S. P. Singh, and K.-S. Sohn, ‘A deep-learning technique for phase identification in multiphase inorganic compounds using synthetic XRD powder patterns’, *Nat. Commun.*, vol. 11, no. 1, pp. 1–11, 2020.
[11] M. Ebner *et al.*, ‘An automated framework for localization, segmentation and super-resolution reconstruction of fetal brain MRI’, *Neuroimage*, vol. 206, 2019.
[12] M. Bal and L. Spies, ‘Metal artifact reduction in CT using tissue-class modeling and adaptive prefiltering’, *Med. Phys.*, vol. 33, no. 8, pp. 2852–2859, 2006.
[13] S. Ghadrdan, J. Alirezaie, J.-L. Dillenseger, and P. Babyn, ‘Low-dose computed tomography image denoising based on joint wavelet and sparse representation’, in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2014, pp. 3325–3328. doi: 10.1109/EMBC.2014.6944334.
[14] H. Lee, J. Lee, H. Kim, B. Cho, and S. Cho, ‘Deep-Neural-Network-Based Sinogram Synthesis for Sparse-View CT Image Reconstruction’, *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 3, no. 2, pp. 109–119, Mar. 2019, doi: 10.1109/TRPMS.2018.2867611.
[15] N. Hampe, J. M. Wolterink, S. G. Van Velzen, T. Leiner, and I. Išgum, ‘Machine learning for assessment of coronary artery disease in cardiac CT: a survey’, *Front. Cardiovasc. Med.*, vol. 6, p. 172, 2019.
[16] S. Wang and R. M. Summers, ‘Machine learning and radiology’, *Med. Image Anal.*, vol. 16, no. 5, pp. 933–951, 2012.
[17] H. Khalid *et al.*, ‘A comparative systematic literature review on knee bone reports from mri, x-rays and ct scans using deep learning and machine learning methodologies’, *Diagnostics*, vol. 10, no. 8, p. 518, 2020.
[18] A. M. Beale, S. D. Jacques, E. K. Gibson, and M. Di Michiel, ‘Progress towards five dimensional diffraction imaging of functional materials under process conditions’, *Coord. Chem. Rev.*, vol. 277, pp. 208–223, 2014.
[19] A. Dazzi and C. B. Prater, ‘AFM-IR: Technology and applications in nanoscale infrared spectroscopy and chemical imaging’, *Chem. Rev.*, vol. 117, no. 7, pp. 5146–5173, 2017.
[20] A. A. Gowen, C. P. O’donnell, P. J. Cullen, and S. E. J. Bell, ‘Recent applications of chemical imaging to pharmaceutical process monitoring and quality control’, *Eur. J. Pharm. Biopharm.*, vol. 69, no. 1, pp. 10–22, 2008.
[21] S. Primpke, M. Godejohann, and G. Gerdts, ‘Rapid identification and quantification of microplastics in the environment by quantum cascade laser-based hyperspectral infrared chemical imaging’, *Environ. Sci. Technol.*, vol. 54, no. 24, pp. 15893–15903, 2020.
[22] D. Kurouski, A. Dazzi, R. Zenobi, and A. Centrone, ‘Infrared and Raman chemical imaging and spectroscopy at the nanoscale’, *Chem. Soc. Rev.*, vol. 49, no. 11, pp. 3315–3347, 2020.
[23] C. Pasquini, ‘Near infrared spectroscopy: A mature analytical technique

with new perspectives–A review', *Anal. Chim. Acta*, vol. 1026, pp. 8–36, 2018.
[24] P. Rostron, S. Gaber, and D. Gaber, 'Raman spectroscopy, review', *laser*, vol. 21, p. 24, 2016.
[25] J. T. Irvine, D. C. Sinclair, and A. R. West, 'Electroceramics: characterization by impedance spectroscopy', *Adv. Mater.*, vol. 2, no. 3, pp. 132–138, 1990.
[26] P. Zimmermann *et al.*, 'Modern X-ray spectroscopy: XAS and XES in the laboratory', *Coord. Chem. Rev.*, vol. 423, p. 213466, 2020.
[27] B. Wu and J. S. Becker, 'Imaging techniques for elements and element species in plant science', *Metallomics*, vol. 4, no. 5, pp. 403–416, 2012.
[28] R. H. Wilson, K. P. Nadeau, F. B. Jaworski, B. J. Tromberg, and A. J. Durkin, 'Review of short-wave infrared spectroscopy and imaging methods for biological tissue characterization', *J. Biomed. Opt.*, vol. 20, no. 3, pp. 030901–030901, 2015.
[29] J. Ryczkowski, 'IR spectroscopy in catalysis', *Catal. Today*, vol. 68, no. 4, pp. 263–381, 2001.
[30] A. Barth, 'Infrared spectroscopy of proteins', *Biochim. Biophys. Acta BBA-Bioenerg.*, vol. 1767, no. 9, pp. 1073–1101, 2007.
[31] L. Fernández-Carrasco, D. Torrens-Martín, L. M. Morales, and S. Martínez-Ramírez, 'Infrared spectroscopy in the analysis of building and construction materials', *Infrared Spectrosc. Sci. Eng. Technol.*, vol. 510, 2012.
[32] J. Luypaert, D. L. Massart, and Y. Vander Heyden, 'Near-infrared spectroscopy applications in pharmaceutical analysis', *Talanta*, vol. 72, no. 3, pp. 865–883, 2007.
[33] J. Haas and B. Mizaikoff, 'Advances in mid-infrared spectroscopy for chemical analysis', *Annu. Rev. Anal. Chem.*, vol. 9, pp. 45–68, 2016.
[34] F. B. de Santana, W. B. Neto, and R. J. Poppi, 'Random forest as one-class classifier and infrared spectroscopy for food adulteration detection', *Food Chem.*, vol. 293, pp. 323–332, 2019.
[35] X. Kuang, X. Sui, Y. Liu, Q. Chen, and G. Gu, 'Single infrared image enhancement using a deep convolutional neural network', *Neurocomputing*, vol. 332, pp. 119–128, 2019.
[36] R. Gautam, S. Vanga, F. Ariese, and S. Umapathy, 'Review of multidimensional data processing approaches for Raman and infrared spectroscopy', *EPJ Tech. Instrum.*, vol. 2, pp. 1–38, 2015.
[37] F. De Groot, 'High-resolution X-ray emission and X-ray absorption spectroscopy', *Chem. Rev.*, vol. 101, no. 6, pp. 1779–1808, 2001.
[38] J. H. Lovett and H. H. Harris, 'Application of X-ray absorption and X-ray fluorescence techniques to the study of metallodrug action', *Curr. Opin. Chem. Biol.*, vol. 61, pp. 135–142, 2021.
[39] J. W. Lichtman and J.-A. Conchello, 'Fluorescence microscopy', *Nat. Methods*, vol. 2, no. 12, pp. 910–919, 2005.
[40] T. D. T. Oyedotun, 'X-ray fluorescence (XRF) in the investigation of the composition of earth materials: a review and an overview', *Geol. Ecol. Landsc.*, vol. 2, no. 2, pp. 148–154, 2018.
[41] S. R. Stock, *Microcomputed tomography: methodology and applications*.

CRC press, 2019.
[42] N. Dal Ferro, P. Charrier, and F. Morari, ‘Dual-scale micro-CT assessment of soil structure in a long-term fertilization experiment’, *Geoderma*, vol. 204, pp. 84–93, 2013.
[43] A. Singhal, J. C. Grande, and Y. Zhou, ‘Micro/nano-CT for visualization of internal structures’, *Microsc. Today*, vol. 21, no. 2, pp. 16–22, 2013.
[44] L. Vásárhelyi, Z. Kónya, Á. Kukovecz, and R. Vajtai, ‘Microcomputed tomography–based characterization of advanced materials: a review’, *Mater. Today Adv.*, vol. 8, p. 100084, 2020.
[45] P. Chowdhury, H. Sehitoglu, and R. Rateick, ‘Damage tolerance of carbon-carbon composites in aerospace application’, *Carbon*, vol. 126, pp. 382–393, 2018.
[46] H. K. Bas, W. Jin, N. Gupta, and R. K. Behera, ‘In-situ micro-CT characterization of mechanical properties and failure mechanism of cementitious syntactic foams’, *Cem. Concr. Compos.*, vol. 90, pp. 50–60, 2018.
[47] B. Wang, B. Pan, and G. Lubineau, ‘Morphological evolution and internal strain mapping of pomelo peel using X-ray computed tomography and digital volume correlation’, *Mater. Des.*, vol. 137, pp. 305–315, 2018.
[48] J. Scharf *et al.*, ‘Bridging nano-and microscale X-ray tomography for battery research by leveraging artificial intelligence’, *Nat. Nanotechnol.*, vol. 17, no. 5, pp. 446–459, 2022.
[49] F. Li and Z. Zhou, ‘Micro/nanostructured materials for sodium ion batteries and capacitors’, *Small*, vol. 14, no. 6, p. 1702961, 2018.
[50] Z. Ju, X. Xu, X. Zhang, K. U. Raigama, and G. Yu, ‘Towards fast-charging high-energy lithium-ion batteries: From nano-to micro-structuring perspectives’, *Chem. Eng. J.*, vol. 454, p. 140003, 2023.
[51] D. P. Finegan *et al.*, ‘Investigating lithium-ion battery materials during overcharge-induced thermal runaway: an operando and multi-scale X-ray CT study’, *Phys. Chem. Chem. Phys.*, vol. 18, no. 45, pp. 30912–30919, 2016, doi: 10.1039/C6CP04251A.
[52] D. S. Eastwood *et al.*, ‘The application of phase contrast X-ray techniques for imaging Li-ion battery electrodes’, *Nucl. Instrum. Methods Phys. Res. Sect. B Beam Interact. Mater. At.*, vol. 324, pp. 118–123, Apr. 2014, doi: 10.1016/j.nimb.2013.08.066.
[53] J. Malzbender, R. W. Steinbrech, and L. Singheiser, ‘A review of advanced techniques for characterising SOFC behaviour’, *Fuel Cells*, vol. 9, no. 6, pp. 785–793, 2009.
[54] K. N. Grew *et al.*, ‘Nondestructive nanoscale 3D elemental mapping and analysis of a solid oxide fuel cell anode’, *J. Electrochem. Soc.*, vol. 157, no. 6, p. B783, 2010.
[55] V. Middelkoop *et al.*, ‘3D printed Ni/Al2O3 based catalysts for CO2 methanation-a comparative and operando XRD-CT study’, *J. CO2 Util.*, vol. 33, pp. 478–487, 2019.
[56] G. Harding, J. Kosanetzky, and U. Neitzel, ‘X-ray diffraction computed tomography’, *Med. Phys.*, vol. 14, no. 4, pp. 515–525, 1987.
[57] A. Vamvakeros *et al.*, ‘Real time chemical imaging of a working catalytic membrane reactor during oxidative coupling of methane’, *Chem.*

Commun., vol. 51, no. 64, pp. 12752–12755, 2015.
[58] C. Vanhoof, J. R. Bacon, U. E. Fittschen, and L. Vincze, '2020 atomic spectrometry update–a review of advances in X-ray fluorescence spectrometry and its special applications', *J. Anal. At. Spectrom.*, vol. 35, no. 9, pp. 1704–1719, 2020.
[59] S. D. Jacques *et al.*, 'Dynamic X-Ray Diffraction Computed Tomography Reveals Real-Time Insight into Catalyst Active Phase Evolution', *Angew. Chem. Int. Ed.*, vol. 50, no. 43, pp. 10148–10152, 2011.
[60] A. M. Beale *et al.*, 'Chemical imaging of the sulfur-induced deactivation of Cu/ZnO catalyst bodies', *J. Catal.*, vol. 314, pp. 94–100, 2014.
[61] D. Matras *et al.*, 'Emerging chemical heterogeneities in a commercial 18650 NCA Li-ion battery during early cycling revealed by synchrotron X-ray diffraction tomography', *J. Power Sources*, vol. 539, p. 231589, Aug. 2022, doi: 10.1016/j.jpowsour.2022.231589.
[62] Y. Chen, H. Chen, A. Gorkhali, Y. Lu, Y. Ma, and L. Li, 'Big data analytics and big data science: a survey', *J. Manag. Anal.*, vol. 3, no. 1, pp. 1–42, Jan. 2016, doi: 10.1080/23270012.2016.1141332.
[63] G. B. Goh, N. O. Hodas, and A. Vishnu, 'Deep learning for computational chemistry', *J. Comput. Chem.*, vol. 38, no. 16, pp. 1291–1307, 2017.
[64] W. S. McCulloch and W. Pitts, 'A logical calculus of the ideas immanent in nervous activity', *Bull. Math. Biophys.*, vol. 5, pp. 115–133, 1943.
[65] J. J. Hopfield, 'Neural networks and physical systems with emergent collective computational abilities.', *Proc. Natl. Acad. Sci.*, vol. 79, no. 8, pp. 2554–2558, 1982.
[66] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, 'Learning representations by back-propagating errors', *nature*, vol. 323, no. 6088, pp. 533–536, 1986.
[67] A. B. Bulsari, *Neural networks for chemical engineers*. Elsevier Science Inc., 1995.
[68] M. A. Kramer, 'Nonlinear principal component analysis using autoassociative neural networks', *AIChE J.*, vol. 37, no. 2, pp. 233–243, 1991, doi: 10.1002/aic.690370209.
[69] B. R. Bakshi and G. Stephanopoulos, 'Wave-net: a multiresolution, hierarchical neural network with localized learning', *AIChE J.*, vol. 39, no. 1, pp. 57–81, 1993.
[70] S. Katare, J. M. Caruthers, W. N. Delgass, and V. Venkatasubramanian, 'An intelligent system for reaction kinetic modeling and catalyst design', *Ind. Eng. Chem. Res.*, vol. 43, no. 14, pp. 3484–3512, 2004.
[71] V. Venkatasubramanian, 'The promise of artificial intelligence in chemical engineering: Is it here, finally?', *AIChE J.*, vol. 65, no. 2, pp. 466–478, 2019.
[72] C. Rao and Y. Liu, 'Three-dimensional convolutional neural network (3D-CNN) for heterogeneous material homogenization', *Comput. Mater. Sci.*, vol. 184, p. 109850, 2020.
[73] M. Ziatdinov *et al.*, 'Deep learning of atomically resolved scanning transmission electron microscopy images: chemical identification and tracking local transformations', *ACS Nano*, vol. 11, no. 12, pp. 12742–12752, 2017.

[74] G. Montavon, W. Samek, and K.-R. Müller, ‘Methods for interpreting and understanding deep neural networks’, *Digit. Signal Process.*, vol. 73, pp. 1–15, 2018.
[75] J. Gu *et al.*, ‘Recent advances in convolutional neural networks’, *Pattern Recognit.*, vol. 77, pp. 354–377, 2018.
[76] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, ‘Image reconstruction by domain-transform manifold learning’, *Nature*, vol. 555, pp. 487–492, 2018, doi: 10.1038/nature25988.
[77] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, ‘Imagenet: A large-scale hierarchical image database’, in *2009 IEEE conference on computer vision and pattern recognition*, Ieee, 2009, pp. 248–255.
[78] D. H. Ye, G. T. Buzzard, M. Ruby, and C. A. Bouman, ‘Deep back projection for sparse-view CT reconstruction’, in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, IEEE, 2018, pp. 1–5.
[79] A. R. Podgorsak, M. M. Shiraz Bhurwani, and C. N. Ionita, ‘CT artifact correction for sparse and truncated projection data using generative adversarial networks’, *Med. Phys.*, vol. 48, no. 2, pp. 615–626, 2021.
[80] Z. Zhao, Y. Sun, and P. Cong, ‘Sparse-view CT reconstruction via generative adversarial networks’, in *2018 IEEE Nuclear Science Symposium and Medical Imaging Conference Proceedings (NSS/MIC)*, IEEE, 2018, pp. 1–5.
[81] S. Xie *et al.*, ‘Artifact removal using improved GoogLeNet for sparse-view CT reconstruction’, *Sci. Rep.*, vol. 8, no. 1, p. 6700, 2018.
[82] A. Š. Trbalić, A. Trbalić, D. Demirović, E. Skejić, and D. Gleich, ‘CT Metal Artefacts Reduction Using Convolutional Neural Networks’, in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, IEEE, 2019, pp. 251–255.
[83] Q. Yang, P. Yan, M. K. Kalra, and G. Wang, ‘CT Image Denoising with Perceptive Deep Neural Networks’. arXiv, Feb. 22, 2017. doi: 10.48550/arXiv.1702.07019.
[84] B. Kim, M. Han, H. Shim, and J. Baek, ‘A performance comparison of convolutional neural network-based image denoising methods: The effect of loss functions on low-dose CT images’, *Med. Phys.*, vol. 46, no. 9, pp. 3906–3923, 2019.
[85] A. A. Hendriksen, D. M. Pelt, and K. J. Batenburg, ‘Noise2inverse: Self-supervised deep convolutional denoising for tomography’, *IEEE Trans. Comput. Imaging*, vol. 6, pp. 1320–1335, 2020.
[86] D. Yim, B. Kim, and S. Lee, ‘A deep convolutional neural network for simultaneous denoising and deblurring in computed tomography’, *J. Instrum.*, vol. 15, no. 12, p. P12001, 2020.
[87] F. Oviedo *et al.*, ‘Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks’, *Nat. Commun.*, vol. 5, no. 1, pp. 2057–3960, May 2019, doi: 10.1038/s41524-019-0196-x.
[88] V. Stanev, V. V. Vesselinov, A. G. Kusne, G. Antoszewski, I. Takeuchi, and B. S. Alexandrov, ‘Unsupervised phase mapping of X-ray diffraction data by nonnegative matrix factorization integrated with custom clustering’, *Npj*

Comput Mater, vol. 4, p. 43, 2018.
[89] H. Wang *et al.*, ‘Rapid Identification of X-ray Diffraction Patterns Based on Very Limited Data by Interpretable Convolutional Neural Networks’, *J. Chem. Inf. Model.*, vol. 60, no. 4, pp. 2004–2011, 2020.
[90] J. A. Aguiar, M. L. Gong, and T. Tasdizen, ‘Crystallographic prediction from diffraction and chemistry data for higher throughput classification using machine learning’, *Comput Mater Sci*, vol. 173, p. 109409, 2020.
[91] M. I. Razzak, S. Naz, and A. Zaib, ‘Deep learning for medical image processing: Overview, challenges and the future’, *Classif. BioApps Autom. Decis. Mak.*, pp. 323–350, 2018.
[92] J. Wei *et al.*, ‘Machine learning in materials science’, *InfoMat*, vol. 1, no. 3, pp. 338–358, 2019.
[93] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, ‘The rise of deep learning in drug discovery’, *Drug Discov. Today*, vol. 23, no. 6, pp. 1241–1250, 2018.
[94] W. Zhang *et al.*, ‘Deep convolutional neural networks for multi-modality isointense infant brain image segmentation’, *NeuroImage*, vol. 108, pp. 214–224, 2015.
[95] C. Leist, M. He, X. Liu, U. Kaiser, and H. Qi, ‘Deep-Learning Pipeline for Statistical Quantification of Amorphous Two-Dimensional Materials’, *ACS Nano*, vol. 16, no. 12, pp. 20488–20496, Dec. 2022, doi: 10.1021/acsnano.2c06807.
[96] K. He, X. Zhang, S. Ren, and J. Sun, ‘Deep Residual Learning for Image Recognition’, *2016 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*, pp. 770–778, 2016.
[97] R. Aggarwal, A. Gupta, V. Chelur, C. V. Jawahar, and U. D. Priyakumar, ‘DeepPocket: Ligand Binding Site Detection and Segmentation using 3D Convolutional Neural Networks’, *J. Chem. Inf. Model.*, vol. 62, no. 21, pp. 5069–5079, Nov. 2022, doi: 10.1021/acs.jcim.1c00799.
[98] S. Rubaiya Muin *et al.*, ‘Dynamic X-ray micotomography of microfibrous cellulose liquid foams using deep learning’, *Chem. Eng. Sci.*, vol. 248, p. 117173, Feb. 2022, doi: 10.1016/j.ces.2021.117173.
[99] A. A. Coelho, ‘TOPAS and TOPAS-Academic: an optimization program integrating computer algebra and crystallographic objects written in C++’, *J. Appl. Crystallogr.*, vol. 51, no. 1, pp. 210–218, 2018.
[100] H. Dong *et al.*, ‘A scalable neural network architecture for self-supervised tomographic image reconstruction’, *Digit. Discov.*, p., 2023, doi: 10.1039/D2DD00105E.

Chapter 2: Methods

2.1 X-Ray Diffraction (XRD)

The phenomenon of X-ray diffraction (XRD) was discovered by Laue with his co-workers Friedrich and Knipping in 1912, for which he was rightly awarded the Nobel Prize [1]. The discovery was made with a single crystal of Copper Sulphate Hydrate. Although the quality of the obtained diffraction pattern is not anything close to what can be achieved today, it was a monumental discovery in the 20th century, which provided a powerful tool for many subsequent pieces of research from superconductors to the structure of DNA [2].

X-ray diffraction is the result of constructive and destructive interference of X-ray electromagnetic waves [3]. The elastic scattering of X-ray photons by atoms can give constructive interference and form sharp spots on a photographic film. This collection of spots is called a diffraction pattern. When the signal is radially integrated, the resultant spots are represented as peaks on a 1-dimensional plot and from the position of the peak, we can easily derive the space between crystal planes by the famous Bragg's law [4]:

$$n\lambda = 2d\sin\theta \tag{2.1}$$

where n is an integer that represents the order of reflection, λ is the incident x-rays' wavelength, d is the spacing between the crystal planes and θ is the incident angle between the beam and the normal to the lattice planes as shown in Figure 2.1.

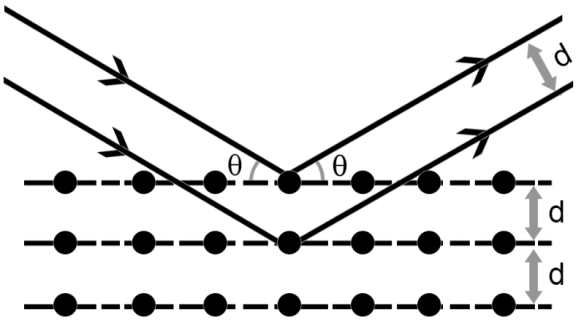

Figure 2.1: Schematic representation of Bragg's Law.

Moreover, instead of using θ, more commonly 2θ is used as the x-axis for the diffraction patterns. To make each peak more separable and patterns in the same form as the experimental data, I will use the momentum change Q as the x-axis for diffraction patterns in this report [6]. Here Q is defined as:

$$Q = \frac{4\pi \sin(\theta)}{\lambda} \tag{2.2}$$

Here the θ and λ have the same definition as in Bragg's law. Since the wavelength, λ is known for the chosen X-ray source. Then we can easily convert the recorded 2θ into Q values.

Due to the phenomenon above, the different lattices will result in different characteristic peaks in the XRD patterns [1]. By comparing the experimental patterns with the characteristic peaks, we can determine the properties of crystalline phases [2]. However, it is not a very easy approach because there are many factors that can affect the position, height and broadness of characteristic peaks [7]. For example, if the crystallite size of a lattice is below 100 nm, X-rays will start to be out of phase, which leads to incomplete destructive interference in scattering directions, and then causes line broadening [8]. Therefore, a more powerful and intelligent way is needed to extract the information from XRD patterns.

After these initial discoveries of XRD, the methodology progressively evolved, culminating in what we know today as X-ray powder diffraction. X-ray powder diffraction analysis involves the use of a powdered form of the sample which is composed of numerous tiny randomly oriented crystals [9]. This random orientation allows X-rays to be scattered in all directions, resulting in a comprehensive diffraction pattern. However, the task becomes more difficult as the symmetry of the unit cell decreases. This is because the complexity of indexing the peaks and measuring the various reflections escalates with the increasing intricacy of the crystal structure. As the structure grows more complex, the number of lines in the diffraction patterns also increases, leading to an unavoidable overlap of reflections, especially when the unit cell symmetry decreases.

Conversely, Single-Crystal X-ray Diffraction (SCXRD) demands a single, high-quality crystal of the substance under investigation [10]. It is widely applied in determining the arrangement of atoms within a crystal [11]–[13]. Considering the different types of information they furnish, X-ray powder diffraction and SCXRD have diverse applications. X-ray powder diffraction is frequently employed in the field of materials science for studying the properties of bulk materials, whereas SCXRD is typically utilised in chemistry and biology to explore the structures of individual and often very large molecules.

Moreover, Small-Angle X-ray Scattering (SAXS), is another analytical method utilised to examine the structure and morphology of materials at a nanoscopic level, irrespective of their crystalline or non-crystalline nature [14]. It has the capability to define the dimensions, form, and arrangement of these structures, spanning from roughly 1 nm to about 100 nm, it is an important method in investigating polymers and biological macromolecules, particularly in their native environment. In 2015, Kikhney et al. applied the SAXS to quantitatively analyse intrinsically disordered proteins in solution [15]. They also claimed that SAXS can be effectively integrated with supplementary methods like X-ray crystallography, and nuclear magnetic resonance (NMR).

2.2 Computed Tomography (CT)

2.2.1 X-ray absorption computed tomography

X-ray radiography is based on the principle of illuminating an object with an X-ray beam and measuring the intensity of the transmitted X-rays. The relationship between the length of the path light travels through the sample and the amount of light absorbed is well descriptive by the Beer-Lambert law [16]:

$$I_{\mathrm{t}} = I_{\mathrm{o}} \mathrm{e}^{-\mu x_{\mathrm{path}}} \tag{2.3}$$

Here I_{o} is the intensity of the incident beam and I_{t} is the intensity recorded on the detector, x_{path} is the distance of the beam travels through the sample, and μ is the linear attenuation coefficient. This formula provides a mathematical

relationship stating that the absorbance of a sample is directly proportional to the concentration of the absorbing species and the distance the light travels through the sample.

X-ray absorption computed tomography (X-ray CT) is an imaging technique that utilises X-rays to study the structure of objects. Since the first medical X-ray CT was developed by Hounsfield, X-ray CT scanners have been widely used in hospitals and industry for materials characterization. The key advantage of this technique is its non-destructive nature, allowing repeated testing and further analysis of the same sample after the tomographic measurement. The initial version of X-ray CT employed a narrow X-ray beam, also known as a "pencil" beam, which is discussed in more detail in a later section on X-ray diffraction computed tomography (XRD-CT). But currently, in synchrotrons, X-ray CT scans are commonly carried out using a parallel beam geometry [17].

In synchrotron settings, a sample is exposed to a monochromatic X-ray beam, and numerous individual radiographs (projections) are taken as the sample rotates, usually covering angles of 0 to 180 °. The recorded intensities are represented in a sinogram; each row corresponds to one individual radiograph. The schematic of the X-ray CT scan and an example sinogram is shown in Figure 2.2.

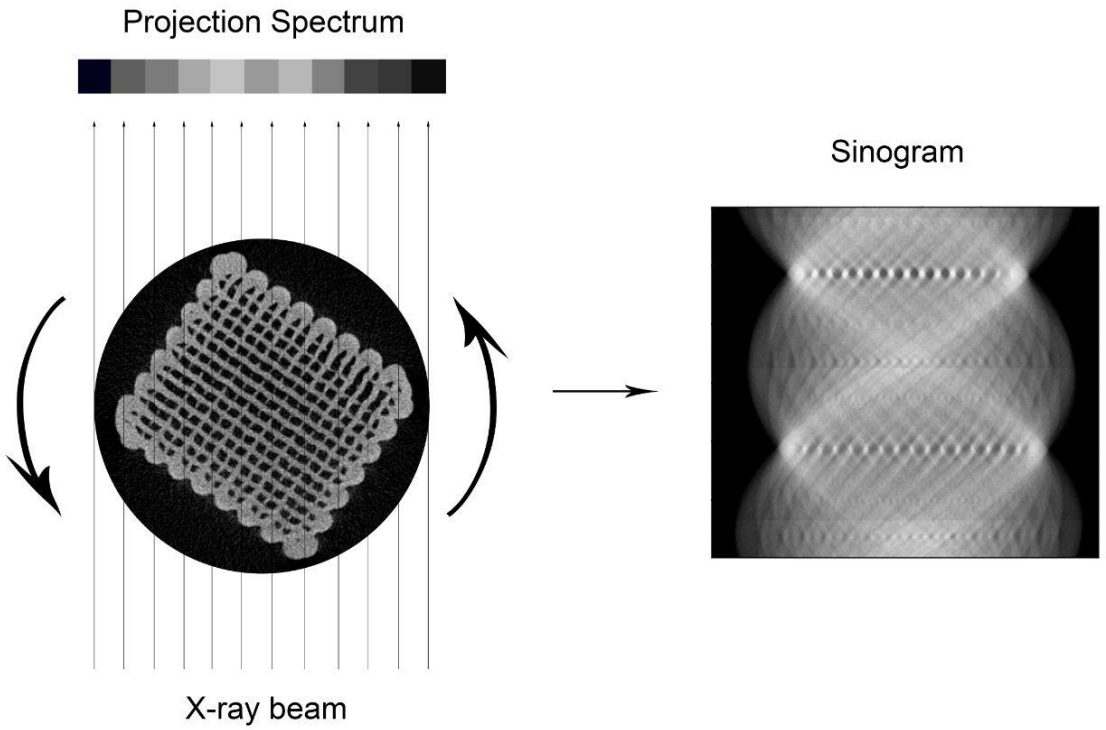

Figure 2.2: X-ray absorption CT sampling in the real-space domain (left). An example sinogram that contains the spectrum of all angles and translations (right).

Each pixel in the sinogram corresponds to the detected beam strength at a specific angle and translation step. Several algorithms exist for image reconstruction from the recorded sinogram. Both iterative algorithms and algorithms using Fourier analysis of the projection data are widely recognized which will be elaborately illustrated in the following sections [18].

2.2.2 X-ray diffraction computed tomography

Unlike X-ray CT which is built on the X-ray attenuation of different materials, XRD-CT is based on distinct diffraction signals from the crystal phases in the sample. This allows the identification and separation of different crystals, even those with similar densities. In the resulting XRD-CT images, each pixel is linked to a unique diffraction pattern, and by analysing these patterns we can extract multiple chemical information and then get a better understanding of the material than the image from X-ray CT.

XRD-CT is primarily utilised at synchrotron facilities where the exceptional X-rays produced, in combination with cutting-edge detectors, allow for dynamic XRD-CT scans with quick acquisition times (less than 100 ms for each diffraction pattern) [19]. The spatial resolution of XRD-CT images depends on the size of the incident X-ray beam. While using slits can reduce the beam size to a few microns or even less, it also diminishes the brightness of the X-ray beam and increases the time required to acquire quality data. Figure 2.3 illustrates the data acquisition procedure of XRD-CT.

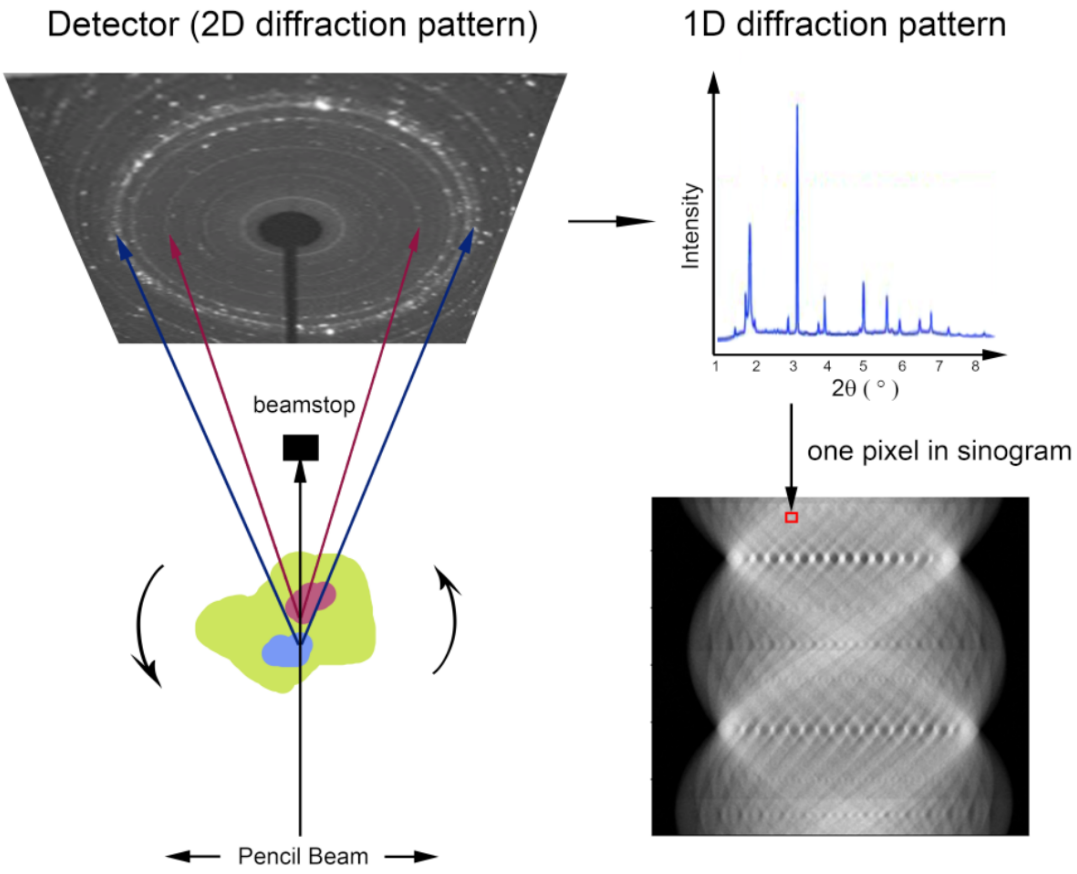

Figure 2.3: X-ray diffraction CT sampling in the real-space domain. Two diffracted beams from two crystals (red and blue) reach different positions on the detector and form the 2D powder diffraction pattern. Then the 2D diffraction pattern is converted to a 1D pattern after integration of the average value from the diffraction rings. By moving the pencil beam and rotating the sample we can then get the XRD-CT sinogram where after back projection each pixel in the image contains a full diffraction pattern. The Diffraction pattern shown in the figure is adapted from previous work by Antonios et al. on Mn–Na–W/SiO2 fixed-bed reactor [20].

In the process of performing XRD-CT with a pencil beam, a concentrated, slim beam of X-rays is targeted at a sample, which is then rotated with a small angle step. With each rotation, the diffracted X-rays from the sample's crystal structures are captured, forming a 2D diffraction pattern. By taking the average of all points which have the same distance from the pattern centre, the 2D diffraction pattern is transferred to a 1D diffraction pattern and forms a pixel in the sinogram. After the sample has finished all angular steps (by 180 or 360 °), the sample will be moved by a translation step *t* which is perpendicular to the incident X-ray beam, and preferably, each *t* is close to the horizontal size of the X-ray beam. The total number of translation steps *n* can be calculated by $n = s / t$ where *s* is the scope size. In practice, the *s* is normally slightly larger than the sample size to make sure all information from the sample has been taken. Meanwhile, there is another way to take the XRD-CT data which swap the order of angular and translation steps (i.e. finish all translation steps before rotating to the next angle) [20]. Whichever the data acquisition method is applied, the best number of angular steps obeys the Nyquist sampling theorem to ensure a faithful reconstruction (i.e. without artefacts) [21]:

$$npr = ntr \times \frac{\pi}{2} \tag{2.4}$$

Where npr and ntr represents the number of projections and the number of translation steps in a sinogram. The sinogram of XRD-CT is then a 3D volume that contains information on the sample's crystalline components.

2.3 Computed Tomographic Image Reconstruction

The back-projection algorithm is the most straightforward of all tomographic reconstruction methodologies. This method is a simple summation of the projections on all projection angles. It implies that each projection spectrum is spread across the image in the direction they were gathered during the tomographic scan. Consequently, the reconstructed image is the cumulative

effect of the back-projected projection spectra. Despite its simplicity, the back-projection algorithm normally results in blurred images because of the uneven sampling in the frequency domain during a Computed Tomography (CT) scan. Since the sampling rates around the centre of a CT image are greater than those around the edge of the scope, the density of radial points decreases as one moves further from the sample's centre which results in the blurring of the back projection images.

The Direct Fourier Reconstruction (DFR) algorithm is another technique used for the reconstruction of images in CT scans. According to the Fourier slice theorem, the Fourier transformed projection sinogram can be utilised to construct the Fourier transform of the (reconstructed) image, and subsequently, the image can be reconstructed by employing the inverse Fourier transform on the 2D frequency spectrum of the image [22]. The Fourier transform is a mathematical technique that allows the conversion of signals from the time (or spatial) domain to the frequency domain. Because of the implementation of the Fast Fourier Transform (FFT), the DFR can perform CT image reconstruction very efficiently in modern computers and can create high-resolution images that preserve details and minimise artefacts [23]. However, this method requires the data to be collected from a significant number of evenly spaced angles around the object, although this may not always be feasible or practical in real-world CT imaging scenarios. Moreover, the quality of the final image can be influenced by various factors, such as noise, the quality of the initial projection data, and the specific execution of the algorithm [24].

Filtered Back Projection (FBP) is the most commonly used CT image reconstruction technique. The FBP can take advantage of both the back projection method and the DFR methods [25]. In this process, a sharpening filter, such as a ramp filter, is used on the projection data before back projection takes place. In the frequency domain of the sinogram after applying the Fourier transformation, the ramp filter, a linear function that directly corresponds to the absolute frequency, is applied directly to each projection’s frequency with

multiplication operations. Then the inverse Fourier transform is applied to convert the sinogram back to the spatial domain, and the back projection is applied to the filtered sinogram. This arrangement allows for the dampening of low-frequency elements while simultaneously escalating high-frequency ones, this makes the image look sharper, particularly along edges and minute details. However, in the spatial domain, the ramp filter is a sinc function, the infinite range of which poses computational challenges. In order to make these calculations practicable, the function is generally approximated and cut short, resulting in a windowed sinc function. This is achieved by using various window functions such as Hamming and Hanning to curtail the sinc function. This measure assists in eliminating ring artefacts and managing noise amplification in the final images. Moreover, the FBP can sometimes result in artefacts and can be affected by noise [26]. In this thesis, I will show that the FBP fails when the number of projections is insufficient (leads to angular-undersampling artefacts), and the overall performance is worse than when using iterative methods. Despite these potential drawbacks, the FBP algorithm continues to be a popular choice because of its straightforward application and quick execution time.

Apart from direct reconstruction methods like the FBP and the back projection, iterative tomographic reconstruction algorithms are also commonly used in modern CT techniques. For example, the Algebraic Reconstruction Technique (ART) is an iterative method that rebuilds images from their projections [27]. The ART algorithm begins with an initial guess of the CT image and then progresses in a series of steps. On each step, the algorithm only takes one angle’s projection data in the sinogram and optimises the image to decrease the difference between the selected projection data and the current projection data of the estimated image along the same path. This cycle is repeated until the solution stabilises. The most straightforward ART variant, known as Kaczmarz's method, deals with each projection data separately [28]. However, more sophisticated versions can simultaneously consider multiple line integrals [29]. The Radon transform is the most commonly used technique in converting any

image into its sinogram domain [30], however, due to the computational complexity of the Radon transform, the weight matrix is also sometimes used in performing ART or other iterative methods [31]. The weight matrix is defined by how much each pixel of the reconstruction image adds to each individual projection. By using the weight matrix, one just needs to perform the matrix multiplication between the weight matrix and the reconstructed image to get the sinogram.

The Simultaneous Iterative Reconstruction Technique (SIRT) [32], is another iterative reconstruction method. Different from ART, SIRT updates the reconstructed image with all projections together in a loop. This approach usually leads to more consistent convergence behaviour than the ART method, however, it also requires larger computational resources, especially for the reconstruction of a 3D volume.

The Simultaneous Algebraic Reconstruction Technique (SART) also utilises all projections to update the reconstructed image on each step [33], but it uses a weighted contribution from each projection line in the image refinement process. Each pixel's update is weighted according to the length of the path the projection line takes through that pixel. This accounts for the fact that a pixel has a greater influence on a projection if the projection line covers more of that pixel. Moreover, to prevent certain artefacts and to improve the algorithm's convergence, the updates can be normalised by a factor associated with the total length of all projection lines crossing each pixel. In general, SART typically reaches the desired solution more quickly and is able to produce images with less blurring, and more shaper details compared to SIRT. However, one drawback is that the SART is less robust to the noise than the SIRT. Therefore, the choice of SIRT and SART highly depends on the specific sinograms and the goal of reconstruction.

Conjugate Gradient Least Squares (CGLS) is another iterative technique employed for addressing large, complex image reconstruction problems [34].

Unlike SIRT, the objective of this algorithm is to update the image by reducing the least squares function between the generated sinogram and the original sinogram. One key advantage of CGLS lies in its typically rapid convergence when compared to many other iterative techniques. Nevertheless, it is important to note that CGLS is sensitive to noise within the sinogram. As with any reconstruction method, the selection of CGLS depends on a variety of task-specific factors, such as the size and quality of the dataset, computational resources available, and the balancing act between desired image clarity and noise reduction. In this thesis, I will use these iterative methods as the basis and explore the possibility of applying Machine Learning (ML) techniques in designing new image reconstruction algorithms.

2.4 Simulation of diffraction patterns

Preparing various and large databases for learning is quite important in enhancing the performance of ANN models [35]. Because of the limited number of experimental data, simulated XRD patterns will be used to enrich the scale of our training datasets.

There are many existing packages and software that can be used to simulate XRD patterns like PowderCell [36], VESTA [37], and TOPAS [38]. These tools have already been verified to simulate XRD patterns precisely. But most of them are time-consuming for creating a large number of patterns at the same time. For example, it takes 4 hours to create 100,000 patterns with TOPAS, but at least 1 million patterns are required to obtain a model that can give a precise result. This is not acceptable because we need to recreate the training data many times for different combinations of materials in our research. Therefore, we decided to generate a more powerful Python package using a graphics processing unit (GPU) in the computer.

The first step is to calculate the intensity of diffraction events using the structure factor equation [39]:

$$F_{hkl} = \sum_j O_j \, f_j \, \mathrm{e}^{-2\pi \mathrm{i}(hx_j + ky_j + lz_j)} \, \mathrm{e}^{\frac{-\mathrm{B}(\sin\theta)^2}{\lambda}} \tag{2.5}$$

Here O_j is the occupancy of atom *j*. x_j , y_j and z_j are the positional coordinates. These values come from the lattice parameters a, b, c and the angles α, β, γ. The *h*, *k* and *l* in the equation define a reciprocal lattice point corresponding to the real-space plane in the Miller indices (*hkl*). B is the Debye-Waller factor that describes the reduction of X-ray scattering due to the effects of thermal motion. Finally, λ is the wavelength of the incident X-ray beam and f_j is the scattering factor of the *j-th* atom, which is unique for each element and can be found in many databases such as the International Crystallographic Tables [40] and the NIST chemistry webbook [41].

However, the structure factor is not exactly the intensity in the XRD patterns. To convert the structure factor into actual intensity, some additional factors need to be applied:

$$\mathrm{I} = \mathrm{s} j_{hkl} \mathrm{LPA} F^2_{hkl} \tag{2.6}$$

Here F^2_{hkl} is the structure factor in the above. L, P, and A are the Lorentz, polarisation, and absorption corrections, respectively [42], whilst j_{hkl} is a factor that depends on the symmetry of the material. And s is the combination of the remaining factors that need to be taken into account like the data acquisition time and the displacement caused by the choice of profile peak shape functions. [43]. A diffraction pattern is then the collection of peaks for every combination of Miller indices.

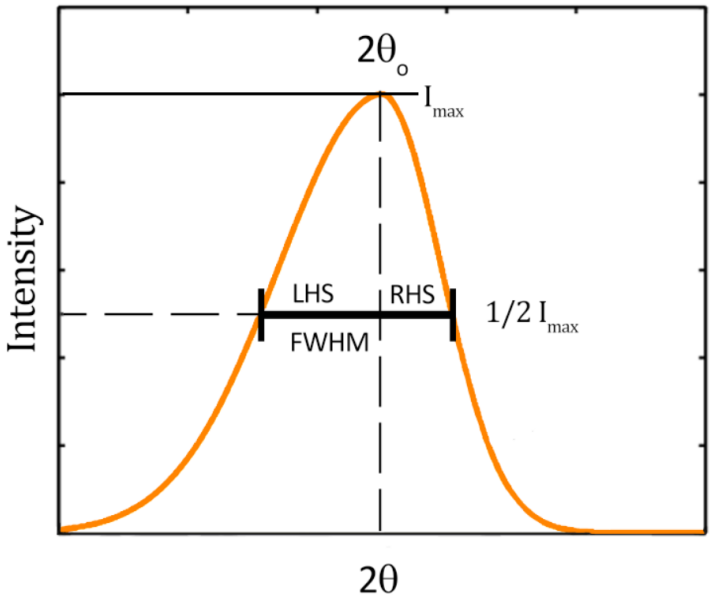

Figure 2.4: Constituent parts of an example peak shape in XRD patterns.

For simulating the peak shape, it is represented with some characteristic factors as shown in Figure 2.4. The peak width Full Width at Half Maximum (FWHM) concerns the broadness of a peak. Changing the crystallite size of a material mainly affects the FWHM factor [44]. A peak can be asymmetric (LHS and RHS FWHM are not the same), this is a particular problem for powder XRD patterns from neutron spallation source instruments [45]. The tails on both sides of the peak should approach 0, which represents how well the mathematical model works in representing the peak shape. Then the shape of a peak can be calculated by statistical models like the Gaussian function [46]:

$$\mathrm{I}(2\theta) = \mathrm{I}_{max} e^{\frac{-\pi(2\theta - 2\theta_0)^2}{\beta^2}} \tag{2.7}$$

with

$$\beta = \frac{1}{2} H \left(\frac{\pi}{\log_e 2} \right)^{1/2} \tag{2.8}$$

the Lorentzian (or Cauchy) peak function [47]:

$$I(2\theta) = \frac{w^2}{w^2 + (2\theta - 2\theta_0)^2} \tag{2.9}$$

Or the pseudo-Voigt function [48]:

$$\mathrm{I}(2\theta) = \mathrm{I}_{hkl}[\eta \mathbf{L}(2\theta - 2\theta_0) + (1-\eta)\,\mathbf{G}(2\theta - 2\theta_0)] \tag{2.10}$$

Here $\mathbf{L}$ and $\mathbf{G}$ are normalised Lorentzian and Gaussian functions respectively, which are defined as:

$$\mathbf{L} = \frac{2/H\pi}{1 + 4(2\theta - 2\theta_0)^2/H^2} \tag{2.11}$$

And

$$\mathbf{G} = \frac{\sqrt{4\log_e 2/\pi}}{H} \exp\left(\frac{-4\log_e 2(2\theta - 2\theta_0)^2}{H^2}\right) \tag{2.12}$$

Here

In these functions, H means the peak width at FWHM, and w = 0.5H. The pseudo-Voigt function can be simply considered as the combination of the other two functions by applying an η factor lying between 0 and 1 to adjust the fractions of each function used. Then, the effect of crystallite size on the FWHM can be calculated by the famous Scherrer equation [49]:

$$\mathrm{FWHM}(2\theta) = H = \frac{\mathrm{b}\lambda}{D\cos\theta} \tag{2.13}$$

Here D is the crystallite size which will be predicted by our neural networks, λ is the X-ray wavelength, 2θ is the scattering angle in radius, and b is a constant, normally between 0.89 and 0.94, depending on the choice of peak function. This equation shows an inverse relationship between the broadness of the peak and the crystallite size [49].

Lastly, we are going to combine the patterns of different phases into a whole simulated XRD pattern. Instead of simply adding intensities all together at each angle (or q value), we are going to multiply each phase by a unique scale factor before adding them together. Those scale factors represent the density of a type of crystal in the mixture material. Therefore, these scale factors can also be used in calculating the weight percentages of each crystal. To represent it as functions, the final simulated intensities at each angle $\mathrm{I}_{\mathrm{tot}}$ are:

$$\mathrm{I}_{\mathrm{tot}}(2\theta) = \sum_j c_j \mathrm{I}_j(2\theta) \tag{2.14}$$

Here c_j is the scale factor for each phase [50].

In conclusion, we are going to simulate training patterns by varying the lattice parameters a, b, c, the crystallite size and the scale factor for each phase. Those will also be the predictions that our neural network models make.

2.5 Rietveld Refinement

Rietveld refinement is a method employed in the field of materials science for refining a structure model and extracting physicochemical information from powder diffraction data without directly solving a crystal structure [51]. Unlike the Le Bail (or Pawley) analysis [52], [53], which performs whole pattern fitting of intensities and peaks that are not bound by the chemistry of crystals, Rietveld refinement focuses on minimising the difference between the simulated diffraction patterns and the experimental data with the prior knowledge about the complete crystal structure models. Rietveld refinement offers significant insights into various aspects of crystalline materials, including atomic positions, thermal vibrations, and other structural parameters.

The Rietveld refinement technique utilises a least-squares minimization algorithm and considers the positions, intensities, and shapes of the powder diffraction peaks. The primary objective of Rietveld refinement is to minimise the residual function, which quantifies the remaining mismatch between the model and experimental data:

$$\text{residual} = \sum_{i}^{n} w_i (y_i^{\text{obs}} - y_i^{\text{sim}})^2 \tag{2.15}$$

where y_i^{obs} is the observed intensity at point *i* on the experimental diffraction pattern and y_i^{sim} is the intensity of the simulated pattern at the same position, and $w_i = 1/y_i^{\text{obs}}$. The simulated diffraction pattern is calculated based on the functions in the previous section, and to assess the accuracy of a crystal structure model in Rietveld refinement, several statistical factors, known as R factors, are utilised. The key factor is determined by the equation:

$$R_{\mathrm{p}} = \frac{\sum_i \left| y_i^{\mathrm{obs}} - y_i^{\mathrm{sim}} \right|}{\sum_i \left| y_i^{\mathrm{obs}} \right|} \tag{2.16}$$

$$R_{\mathrm{wp}} = \left(\frac{\sum_i w_i \left| y_i^{\mathrm{obs}} - y_i^{\mathrm{sim}} \right|^2}{\sum_i w_i \left| y_i^{\mathrm{obs}} \right|^2} \right)^{1/2} \tag{2.17}$$

$$R_{\mathrm{exp}} = \left(\frac{\mathrm{N} - \mathrm{P} + \mathrm{C}}{\sum_i \left| y_i^{\mathrm{obs}} \right|} \right)^{1/2} \tag{2.18}$$

In these equations, N represents the number of observation points, P denotes the number of parameters that are refined, and C is the number of constraints. However, while these statistical factors provide valuable information, the preferred method for evaluating the Rietveld refinement is a visual inspection of the fitted and experimental data. Ideally, the difference plot should exhibit a flat line, indicating a successful fit between the observed and calculated data. Visual assessment, combined with the statistical factors and conformity to well-established chemical principles such as bond lengths and coordination numbers consistent with the Shannon Tables of crystallographic radii [54], offers a comprehensive evaluation of the refinement quality and the validity of the crystal structure model.

2.6 Artificial Neural Network (ANN)

The Artificial Neural Network (ANN) is a very important tool for implementing Artificial Intelligence (AI) and ML, which are completely different from traditional task-specific programming; these can learn from examples and give predictions based on the trained model [55]. An ANN normally has an input layer consisting of a number of neurons equal to the number of input features, an output layer which has a size that depends on the problem being solved, and several specialised hidden layers between the input and output layers. Each hidden layer in the network receives a signal from the previous layer or node, performs one or multiple computations, and then sends the resultant signal to the next

layer or node. Signals are all real numbers, and then the output layer is the sum of a series of nonlinear functions from the input value [56].

2.6.1 Type of Layers in ANN

The fully connected layer is the most commonly used hidden layer, which is inspired by biological neural networks [56]. Fully connected layers are made of several nodes. Each node is connected to all outputs from the previous layer, performs multiplications based on the weights that are trained, and passes the output to the next layer. It can create n^2 trainable weights when two fully connected layers with n nodes are connected together, therefore, the use of multiple large-scale fully connected layers in combination may impose a computational burden on both training and applying the ANN model.

The dropout layer is a kind of layer that is normally used after fully connected layers to reduce overfitting in artificial neural networks [57]. By applying a dropout layer, a proportion of weights in the fully connected layer are omitted while training. The omitted weights are randomly selected and will differ in each training epoch. The dropout layers can help to prevent overfitting. Overfitting occurs when a model obtains excellent accuracy during its training but performs inadequately on testing. This typically happens when the training dataset is small but the number of trainable weights is large. Applying the dropout layers can effectively force the model to generalise and learn the overall characteristics of the dataset rather than focusing on individual data pairs, and then improve the performance of the model.

The batch normalisation layer is frequently utilised in neural networks for normalising the activations of a specific layer [58]. It achieves this by dividing the batch standard deviation and subtracting the batch mean. Its purpose is to enhance the efficiency and stability of the model during training by mitigating the internal covariate shift which is caused by the changing distribution of the inputs to each layer. Batch normalisation equalises the distribution of activations

across batches, thereby mitigating the internal covariate shift. This facilitates faster convergence and higher accuracy in the model's predictions.

A pooling layer is used to shrink the scale of matrices. The idea of a pooling layer is to take one only value in a section of the matrix [59], [60]. There are many types of pooling layers, like a max-pooling layer which only takes the maximum value in the pooling kernel, an average pooling layer that takes the average value in the kernel, and so on. There is normally much redundancy information in the experimental images. Disposing of unnecessary information with pooling layers during training will not affect the performance of an ANN model, but it can reduce the size of the model significantly.

In contrast, an upsampling layer is frequently employed in neural networks for image or signal processing tasks, aiming to enhance the spatial resolution of input data by expanding it in horizontal, vertical, or both dimensions. This technique is also referred to as "deconvolution" or "transposed convolution." Upsampling layers are commonly used in tandem with down-sampling or pooling layers to restore the original spatial resolution of the data, and they are pivotal in applications such as image generation, image segmentation, and super-resolution.

There are many other operation layers that I will use in this thesis. For example, the flattening layer can convert any output of layers into a 1D array. A reshaped layer can convert a signal into any shape, including increasing or reducing the dimension of a matrix. Lastly, the mathematics layers perform basic addition, subtraction, multiplication, and division operations between two layers that have the same shape.

2.6.2 Activation Functions

For each trainable node, an activation function is needed to define the relationship between the output of one node and the input of the next and determine whether a node is activated or not. The choice of activation function

is related to the problems that the model is designed for and the positions of the layers. For example, the linear activation function does not perform any modification of the output number, and the Rectified Linear Unit (ReLU) function keeps only positive values from the output signal [61]. By applying ReLU, all negative values of one layer's output are set as 0 before being sent to the next layer, which helps the ANN to reduce the number of activated neurons needed to be learned, and has fewer vanishing gradient problems [62], [63]. However, many reports have proved that ReLU may cause dead neurons on the last/output layer [64], [65]. Therefore, the alternatives, like the Leaky ReLU and exponential linear unit (ELU) activation function, are becoming more popular [65]–[68]. The leaky ReLU and ELU are defined as:

$$\text{Leaky ReLU: } f(x) = \begin{cases} x & x > 0 \\ \mathrm{a}x & x \leq 0 \end{cases} \tag{2.19}$$

$$\text{ELU: } f(x) = \begin{cases} x & x > 0 \\ \mathrm{b}(\mathrm{e}^{x} - 1) & x \leq 0 \end{cases} \tag{2.20}$$

Here a and b are positive coefficients that control the shape of the function which needs to be tuned according to the problem. Moreover, a Softplus function, a continuous activation function that only outputs positive values, is also a popular replacement for the ReLU function:

$$\text{Softplus: } f(x) = \log(1 + \mathrm{e}^{x}) \tag{2.21}$$

This function is suitable for problems that are strictly non-negative. The continuous derivative makes the model easier to optimise with gradient-based methods such as stochastic gradient descent.

The Sigmoid function only outputs numbers between 0 and 1, so it is normally chosen as the activation function at the final layer of a classification ANN [69]. Meanwhile, the Softmax function is commonly employed as the activation function in the output layer of a neural network that is designed for multi-class classification tasks [70]. It works by taking in a vector containing real numbers and converting it into a probability distribution that reflects the potential classes. This allows the network to estimate the probability of an input being assigned to

each of the potential classes. The Sigmoid and Softmax function is defined as follows:

$$\text{Sigmoid: } f(x) = \frac{1}{1+e^{-x}} \tag{2.22}$$

$$\text{Softmax: } f(x_i) = \frac{e^{x_i}}{\sum e^{x_j}} \quad for\ j = 1, \ldots, n_{\text{neurons}} \tag{2.23}$$

Here x represents the element in the output vector of a layer, and n_{neurons} represents the total number of neurons. Both activation functions are suitable for classification methods with positive values as both inputs and outputs. For problems with both positive and negative values as input, the tanh activation function can limit the output within a range of -1 to 1 [71]:

$$\tanh: f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{2.24}$$

If the input data has a mean of zero, the tanh function is beneficial since it produces outputs that are centred around zero. This property can be helpful in minimising the impact of bias in the network. The activation functions are summarised in Figure 2.5. In this thesis, the ReLU function is used as the activation function for all hidden layers in the presented architectures.

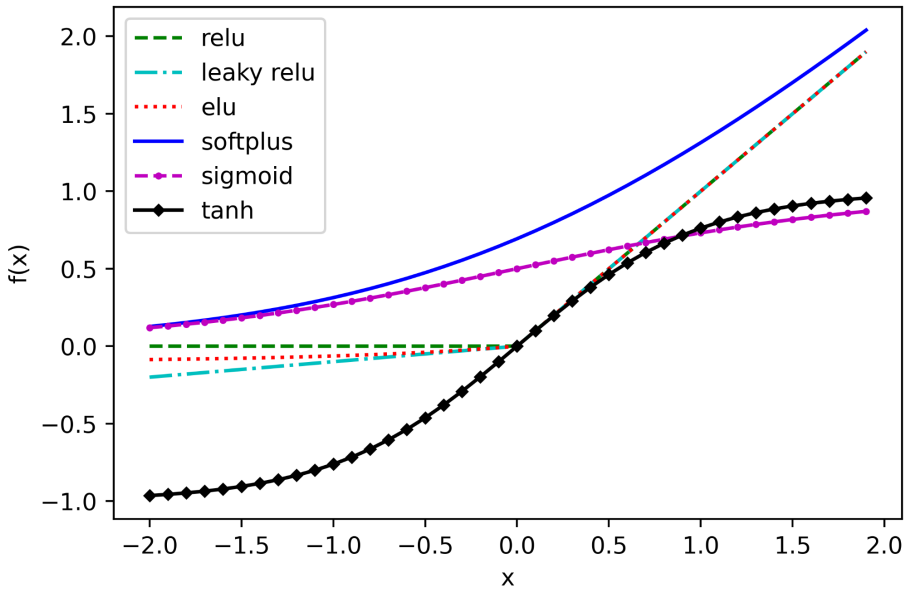

Figure 2.5: The result of different activation functions on the x values between -2 and 2. Both a and b coefficients of leaky ReLU and ELU functions are set as 0.1.

2.6.3 Training Algorithms

Supervised learning is one of the most commonly used ML methods to train ANNs that can give predictions on datasets that have never been seen before [72]. Based on the intended use of the model, supervised learning models can be categorised into classification models and regression models. Classification models normally only give 0s and 1s as output to identify to which cluster the input data belongs [55]. Meanwhile, regression models can give more specific predictions on quantities of the predictions [73]. No matter what models are used, supervised learning approaches always need a large number of pairs of well-labelled training data to obtain well-trained ANN models. The optimization methods are applied to fit the model with the labelled datasets, and after the model is well trained, the model can be applied to new unlabeled datasets. Normally in supervised learning approaches, the dataset can be split into three sub-datasets which are training, validation, and test datasets. The training dataset is the largest dataset used for training the model, and the validation dataset is a small percentage (normally 10 %) of the training dataset. The validation dataset never takes part in the training processes, but they are calculated at each loop to observe the performance of the model while training. When the accuracy of the validation dataset has stopped decreasing, the training process should be stopped immediately to avoid overfitting problems. Finally, the testing dataset is a group of data that is used to test the performance of the model after it is well-trained. Different from the validation dataset, the test dataset is normally chosen from another data frame (different experiment, different camera, etc.) that has not been seen in the training dataset. If the model is trained with simulated data, the test dataset is normally the real experimental data for which there is a ground truth.

In recent years, unsupervised learning has become increasingly popular, because of the characteristic of not requiring a large amount of training data [74]. In unsupervised learning, an ML model is trained on a dataset without any labelled output or guidance. The objective is to identify patterns and connections in the data without being given a specific prediction target.

Examples of unsupervised learning algorithms include clustering [75], dimensionality reduction [76], and image processing [77]. Unsupervised learning is also based on modern optimization methods. In contrast to supervised learning, the unsupervised model does not have any pre-knowledge about the dataset, which is generally applied in more adaptable applications [78].

To implement optimisations on both supervised and unsupervised learning models in an efficient manner, we need to choose an appropriate loss function. In most learning networks, the loss function is used to compute the error between the predicted output and the ground truth value given in the training dataset. Different loss functions will have a significant impact on the performance of the model because they will show different errors for the same prediction [79]. The performance of the loss function depends strongly on the model type. The mean squared error (MSE) and mean absolute error (MAE) functions are very useful in regression problems, but for classification problems, binary cross-entropy function, negative log-likelihood and soft margin classifiers are more popular [80].

Then, to achieve an accurate prediction, one needs to minimise the error calculated by the loss function on each node. The choice of optimisation algorithm (optimiser) can significantly affect the performance of training an ANN [81]. In ANNs, the loss on the predictions is propagated backwards to a previous layer, where it is used to modify the weights [82]. Then the weights are adjusted by the chosen optimizer. Optimisation functions usually need to take the partial derivative of the loss functions, and then the weights are adjusted in the opposite direction.

2.6.4 Optimizers

To achieve an accurate prediction, one needs to minimise the error calculated by the loss function on each node. In neural networks, the error is propagated backwards to a previous layer, where it is used to modify the weights [82]. Then

the weights are adjusted by the chosen optimisation function (Optimizer). In this thesis, the Adaptive Moment Estimation (Adam) method will be mainly used as the optimisation function [83]. The Adam algorithm was an extension of the stochastic gradient descent method, which was invented by Diederik Kingma and Jimmy Ba in 2015. The Adam algorithm is expressed in Algorithm 1. As shown in the table, the Adam algorithm takes four hyper-parameters which are normally set as $\beta_1 = \beta_2 = 0.9$, $\varepsilon = 10^{-8}$ and a small α normally around 0.001. Then, we can see from the last update step; that the updated weight is approximately bound by the learning rate, thus making it relatively easier to choose the scale of the learning rate, especially in some situations where it is possible to estimate the scale of optimised result in advance. In many recent publications, it has been determined that the convergence speed of Adam is much faster than traditional optimisers like the Adaptive Gradient (AdaGrad) and the Stochastic Gradient Descent (SGD), especially when treating large models and datasets [84].

Algorithm 1: Pseudocode - Adaptive Moment Estimation (Adam) method

$M_o \leftarrow 0$
$R_o \leftarrow 0$
for *i in range (npr)* **do**
  $M_t = \beta_1 M_{t-1} + (1-\beta_1)\nabla L_t(W_{t-1})$ (1^{st} moment estimate)
  $R_t = \beta_2 R_{t-1} + (1-\beta_2)\nabla^2 L_t(W_{t-1})$ (2^{nd} moment estimate)
  $\hat{M}_t = M_t/(1-(\beta_1)^t)$ (1^{st} moment bias correction)
  $\hat{R}_t = R_t/(1-(\beta_2)^t)$ (2^{nd} moment bias correction)
  $W_t = W_{t-1} - \alpha\frac{\hat{M}_t}{\sqrt{\hat{R}_t}+\epsilon}$
Return *3D_grid*

However, there are also many reports arguing that although Adam reaches convergence quicker, SGD has better generalisation capabilities compared to Adam, leading to enhanced performance in the end, especially for image detection networks [84], [85]. In standard Gradient Descent, the model's parameters are adjusted according to the gradient of the error with respect to the entire dataset. This means for just a single update, gradients for the whole dataset have to be calculated. In contrast, SGD updates parameters for each individual training data pair, which proves more computationally efficient for

large datasets. In essence, it picks one data point (or a small group of data points) at random from the complete dataset during each iteration, significantly reducing the computational load. However, due to its random nature, it can also cause convergence to the minimum to be less accurate. As a result, variations of SGD, such as mini-batch SGD and SGD with momentum, are more frequently employed to balance the speed of convergence and stability of the final parameters [86].

Moreover, there are more candidate optimizers that have shown their advantages in specific situations. For example, RMSprop, AdaGrad, and AdaDelta offer distinct benefits when it comes to ANN optimization [86], [87]: RMSprop adapts the learning rate for each weight, which is particularly useful for non-stationary objectives or scenarios with significant noise; AdaGrad implements varying updates for features based on their frequency of occurrence, advantageous for handling sparse data; and AdaDelta, as an improvement on AdaGrad, seeks to mitigate the continuous reduction of its learning rate, thereby enhancing its overall performance.

2.6.5 Loss Functions

In machine learning, a loss function serves as a metric to evaluate the difference between the predicted outcomes of a model and the real values. The main goal during the training of the model is to minimise the loss function, Depending on the task at hand, various types of loss functions are employed. Mean Absolute Error (MAE) and Mean Squared Error (MSE) are commonly used loss functions for regression problems. MAE calculates the average of the absolute differences between the predicted and actual values, ignoring their positive or negative direction, making it more resistant to outliers. Conversely, MSE squares the differences before taking the average, leading to larger errors having a disproportionately larger impact and being sensitive to outliers. The MAE and MSE are defined as [88]:

$$L_{\mathrm{MAE}}(\mathbf{x},\mathbf{y}) = \frac{1}{\mathrm{N}}\sum_{i=0}^{\mathrm{N}} \left|\left|\mathbf{x}_i - \mathbf{y}_i\right|\right| \tag{2.25}$$

$$L_{\mathrm{MSE}}(\mathbf{x},\mathbf{y}) = \frac{1}{\mathrm{N}}\sum_{i=0}^{\mathrm{N}}\left(\mathbf{x}_i - \mathbf{y}_i\right)^2 \tag{2.26}$$

The Structural Similarity Index Measure (SSIM) is another commonly used loss function when comparing the differences between images. Compared to pixel-based loss measures such as MAE and MSE, it more accurately aligns with human visual perception. By taking into account changes in texture, brightness, and contrast, SSIM can help generate models that yield more visually pleasing results. The SSIM loss can be expressed as [89]:

$$L_{\mathrm{SSIM}}(\mathbf{x},\mathbf{y}) = 1 - \frac{(2\mu_{\mathrm{x}}\mu_{\mathrm{y}} + \mathrm{C}_1)(2\sigma_{\mathrm{xy}} + \mathrm{C}_2)}{(\mu_{\mathrm{x}}^2 + \mu_{\mathrm{y}}^2 + \mathrm{C}_1)(\sigma_{\mathrm{x}}^2 + \sigma_{\mathrm{y}}^2 + \mathrm{C}_2)} \tag{2.27}$$

Here,

$$\mathrm{C}_1 = (\mathrm{K}_1\mathrm{L}_{\mathrm{D}})^2 \tag{2.28}$$

$$\mathrm{C}_2 = (\mathrm{K}_2\mathrm{L}_{\mathrm{D}})^2 \tag{2.29}$$

$$\sigma_{xy} = \frac{1}{\mathrm{N}-1}\sum_{i=1}^{\mathrm{N}}(\mathbf{x}_i - \mu_x)(\mathbf{y}_i - \mu_y) \tag{2.30}$$

The $\mu_{\mathbf{x}}$, $\mu_{\mathbf{y}}$ are the average of all pixels in the input sinograms and the σ_{x}, σ_{y} represent their standard deviations. L_{D} is the dynamic range of the input images. K_1 and K_2 are two constants that are set as 0.01 and 0.03.

For classification tasks, the role of a loss function is to quantify the dissimilarity between the class predicted by the model and the true classes. Cross-entropy loss is frequently employed in binary and multi-class classification tasks. The cross-entropy loss is defined as [90]:

$$L_{\mathrm{Cross-Entropy}} = -\frac{1}{\mathrm{N}}\sum_{i=0}\sum_{c=1}^{\mathrm{M}} y_{ic}\log(p_{ic}) \tag{2.31}$$

Here M is the number of clusters, y_{ic} is the label of the sample (0 or 1), and p_{ic} is the confidence that the sample belongs to the cluster c. In binary classification

problems, the sum of p_{ic} values for the two predicted classes is typically equal to 1, so p_{ic} can be considered as the predicted probability distribution of the classes. However, in multi-class classification problems, depending on the choice of activation functions, the sum may not necessarily be 1, as the confidence for each class is independent. The most commonly used activation function to ensure that the sum of confidences for all classes equals 1 is softmax. Moreover, there are many other loss functions that are used for classification models. For example, the Hinge loss, originally used with Support Vector Machines (SVMs), aims to discover a decision boundary that enlarges the separation between classes and is also applicable in the classification ANNs [91], [92]. Negative log likelihood is an expansion of binary cross-entropy for scenarios with more than two classes [93]. And the Zero-one Loss is the loss function to test incorrect classifications [94]. It is typically employed for assessing the model's performance rather than for the training process.

2.7 Machine Learning Architectures

2.7.1 Convolutional Neural Network (CNN)

The Convolution Neural Network (CNN), a type of ANN, is widely applied to image identification, natural language processing, financial time series, etc. [95]. A CNN normally contains one or a series of convolutional layers and other common ANN layers like pooling layers and fully connected layers. Figure 2.6 shows a regression CNN developed for quantitative analysis of the XRD pattern [96]. For processing a 2D image, the input tensor should have three dimensions for each picture, height, width, and colour channels (depths). In a convolutional layer, this image will be abstracted by a kernel which will only look at a small part of the image each time. The kernel will take and record the features of this part of the image and then jump to the next part and repeat the steps. Then the collection of features can form a feature map which will be passed to the next

layer. The nodes in the kernel are the parameters refined during the training process of the CNN, and they are the same for taking features of different parts of an image [97]. For the 1D XRD pattern identification problem, the process is similar but uses only two dimensions for each pattern.

Each convolutional layer will have some definition factors, the n x n kernel size defines the scope of kernels, the number of filters defines the number of kernels used in this layer (how many features this layer will take), stride defines the pixel shifts over the input image (the gap a filter going to the next part of the image), and padding is the number of pixels added on the edges of original pictures to make sure the feature map is not too small.

A complete CNN will have many other types of layers. As shown in Figure 2.6, A pooling layer is used to shrink the scale of pictures. For a 9x9 matrix, after applying a pooling layer with kernel 3x3 size, it will become a 3x3 matrix with all features remaining. Secondly, a flatten layer is used to convert multidimensional layers into a 1D layer with 1D output [98]. This layer is always used in those multi-dimension identification problems like word embedding and image identification. A flatten layer does not have any trainable weight. Lastly, a traditional fully connected layer takes all values in the previous layer to all values in the next layer. There are many other types of layers that were used in our CNN architecture, such as a dropout layer to avoid model over-fitting, a batch normalisation layer to decrease the learning time, and an up-sampling layer to increase the data size [99].

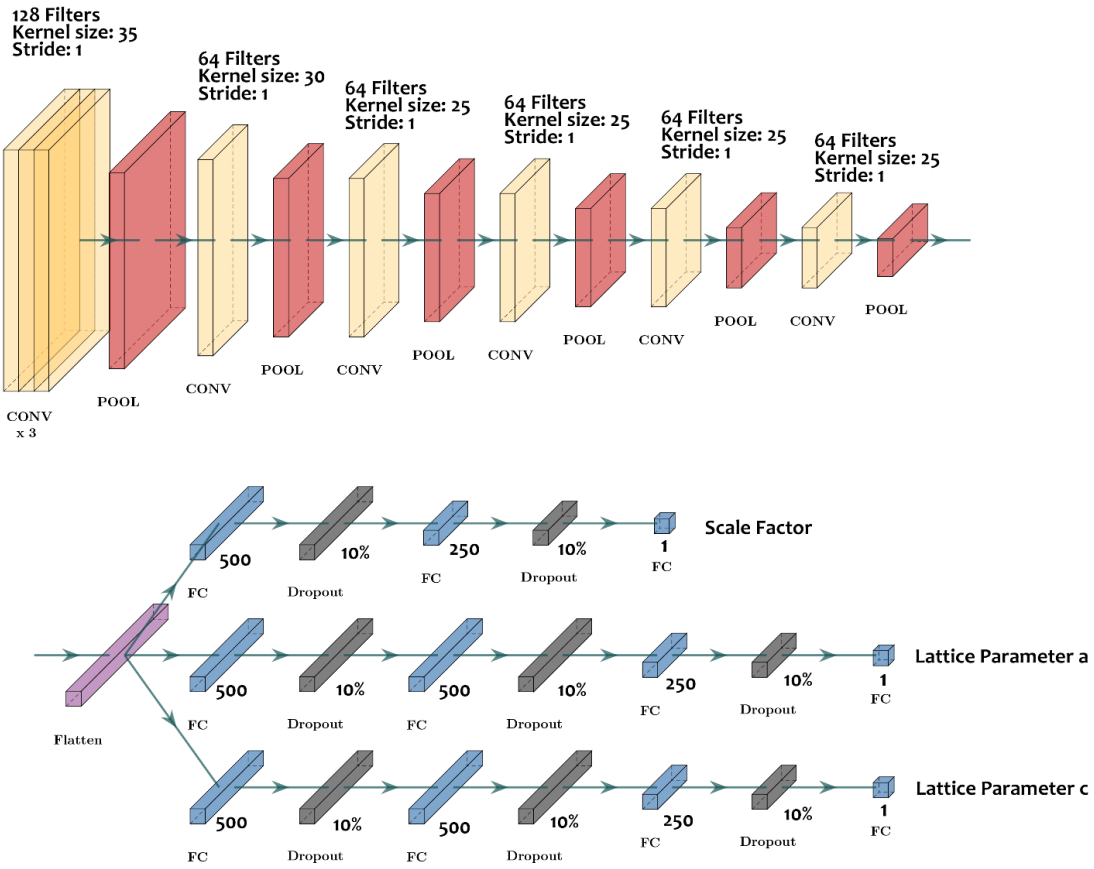

Figure 2.6: The single-phase phase quantification neural network (PQ-Net) CNN [96], all parameters are shown in the figure. This CNN contains 1D convolutional layers (CONV), max-pooling layers (POOL), fully connected layers (FC), a flatten layer and dropout layers. This network receives diffraction patterns as input and outputs the scale factor, lattice parameter a and lattice parameter c of the crystal.

2.7.2 Dilated CNN

A CNN with dilated convolutional layers is called dilated CNN. Compared to the traditional convolutional layers, a kernel in the dilated convolutional layer is not a square (or rectangular) which covers all pixels in the region, instead, it leaves a blank between each node of the kernel [100]. The number of empty pixels between two nodes in the kernel is the so-called dilation rate. Therefore, assuming we have a convolutional kernel with the size of 3x3, it can cover 5X5 regions in a picture if the dilation rate is equal to 1. It covers larger regions with fewer trainable nodes than a traditional convolutional layer, so it is very useful when the computer resources are limited. The dense dilated CNN was

developed by Sethian and co-workers in 2018 [100]. They made all layers densely connected to each other, so the model could learn the best route and find the most suitable dilation rate to use. However, because of the large number of connections in the architecture, the training time will be longer than normal CNN architectures like U-Net and AUTOMAP. In our experience, when the trainable parameters are the same, this architecture can take twice as long.

2.7.3 Encoder-Decoder Network

An encoder-decoder network is a kind of neural network design frequently employed for tasks like generating images, translating languages, domain transferring, and more [101]–[103]. This type of architecture normally has two parts: an encoder takes the input, processes it, and creates a condensed representation known as the context vector, and a decoder uses the context vector created by the encoder to generate the output, which could be a series of words, an image, or another type of data. For tasks that involve transforming one domain into another (like sinogram-image reconstruction), where input and output lengths can differ and vary, the context vector acts as a means of packing information from a variable-length sequence into a static-sized representation.

For autoencoders, a specific kind of encoder-decoder network is used in image processing, the aim during training is to reduce the disparity between the input and output images [104]. This encourages the network to learn how to effectively recreate the original image from the context vector, making the context vector capture the image's most significant features. This means that the context vector itself can be used for clustering images and represents the inputs into a vector of features [105].

2.7.4 U-Net

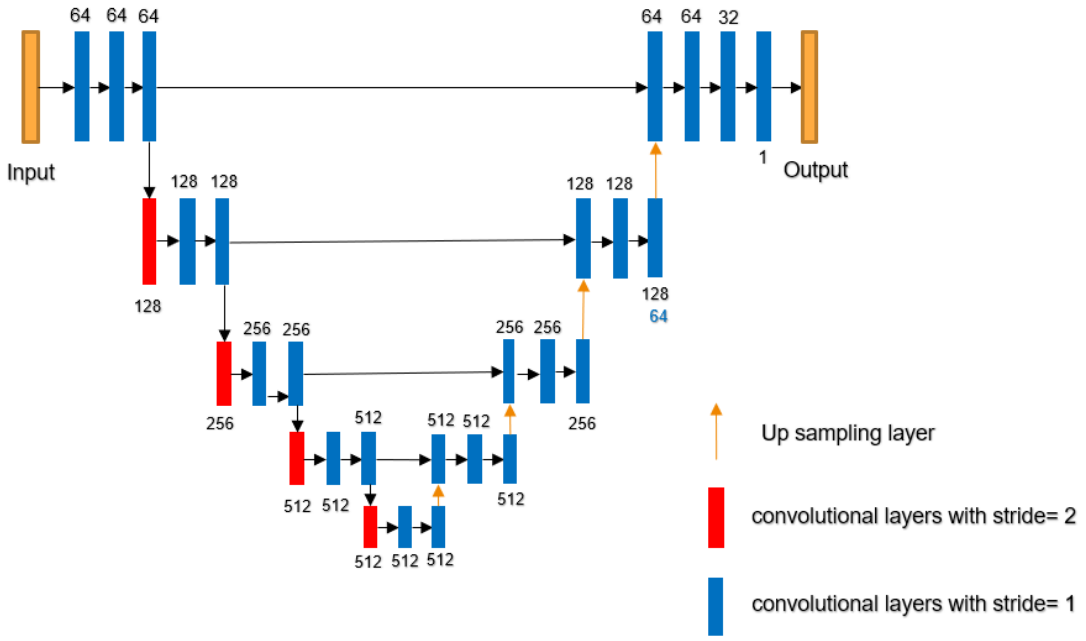

Figure 2.7: An example of 5 levels U-Net architecture with 512x512 image size as input [106]. All convolutional layers in the figure are 2D convolutional layers, and their kernel numbers and sizes as shown in the legend.

The U-Net architecture was first developed for biomedical image segmentation problems [106]. A U-Net architecture contains one down-scaling part and one up-scaling part as shown in Figure 2.7. Each part has multiple levels connected by max-pooling and up-sampling layers, and inside the levels, there are normal convolutional 2D layers. In the up-scaling levels, they do not only take the output from lower levels as input, but they also acquire output images from the same level in the down-scaling part. As a result, each level in the up-scaling part has two inputs containing the features captured in the previous level and in the down-scaling convolutional layers, which makes the model more robust. Compared to traditional CNN architectures, the U-Net can yield more precise segmentation with fewer training datasets [107], [108]. According to the original paper, the segmentation of a 512 x 512 image takes less than a second on a modern GPU [106]. Depending on the image size and how deep the architecture people want, users can easily add or remove levels to fit their

problem. Therefore, its applications are expanding into many areas like image de-noising and clearing [109].

The U-Net has already been well used in refining images. In 2019, a paper appeared using the U-Net to reconstruct an incomplete CT image [110]. They demonstrated that the U-Net could also be useful to enhance the resolution of CT images with a low number of projections. In the same year, Ebner's team used U-Net to do localisation, segmentation and super-resolution reconstruction of magnetic resonance imaging results for analysing foetal brains [111]. A very important paper published recently shows U-Net to be a very powerful method for tackling sinogram-image reconstruction problems [29]. They fed the U-Net with Gaussian noise and trained it with experimental images. The loss function is calculated according to the sinograms while training, so therefore can be considered a type of unsupervised learning method without requiring pre-training. They claimed that the method improved the state-of-the-art results in the low data regime. In their paper, they also compare the results with many other structures like the AUTOMAP and Autoencoder [112].

2.7.5 Generative Adversarial Network (GAN)

A Generative Adversarial Network (GAN) is a powerful unsupervised learning method designed by Goodfellow and his colleagues in 2014 [113]. As shown in Figure 2.8, A GAN structure contains at least two parts: a generator and a discriminator. The generator and the discriminator in GAN are trained together in a competitive setup. In this arrangement, the generator aims to produce new images that look similar to the input images, while the discriminator tries to accurately distinguish between real and artificially generated data. This technique prompts the generator to create data that is more and more convincing and realistic as the training evolves. In the original theory, the two parts do not have to be neural networks, but in modern GAN architectures, both of them are normally deep neural networks [114]. Applications of a GAN cover many image processing fields. For example, Javid and co-workers use a GAN

to remove the rain inside normal pictures taken by camera [115]. And Putra's team used GAN to generate real faces from portrayals [114].

Of course, GAN is also becoming a very important tool for CT image reconstruction. A paper in 2020 used both a GAN and U-Net to remove noise and artefacts in low-dose CT to avoid damage that high amounts of radiation can have on patients [116]. Another team led by Liu created the TomoGAN, another image-denoising model based on a GAN developed for low-dose X-ray imaging [109]. Yang's team proposed GANrec architecture in 2019 [117]. They added the Radon transformation into the architecture to enable GANrec to reconstruct the image. The back-projection image from the sinogram is used as the input to the generator, and then the generator learns how to reconstruct an image from the back-projection.

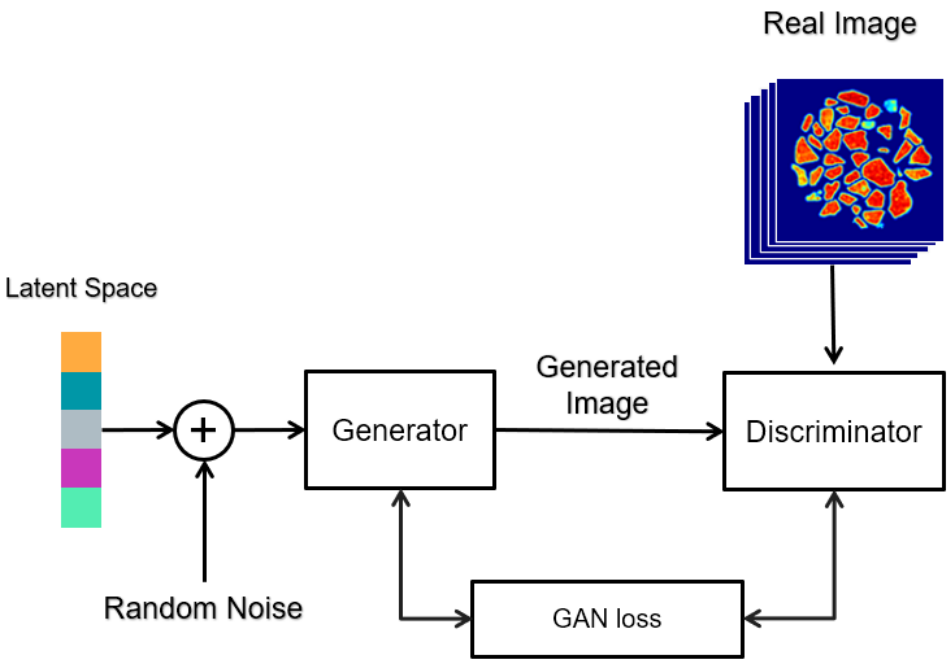

Figure 2.8: A typical GAN architecture for generating similar images. Reproduced from its original paper written by I. Goodfellow et al. [113]. Random noise is fed into the generator to generate a fake image. Then the discriminator predicts whether the generated image is real or not. Those two parts of GAN contest with each other while training to make the generator produces similar images.

2.8 References

[1] A. K. Chatterjee, ‘8 - X-Ray Diffraction’, in *Handbook of Analytical Techniques in Concrete Science and Technology*, V. S. Ramachandran and J. J. Beaudoin, Eds., Norwich, NY: William Andrew Publishing, 2001, pp. 275–332. doi: 10.1016/B978-081551437-4.50011-4.

[2] J. Epp, ‘4 - X-ray diffraction (XRD) techniques for materials characterization’, in *Materials Characterization Using Nondestructive Evaluation (NDE) Methods*, G. Hübschen, I. Altpeter, R. Tschuncky, and H.-G. Herrmann, Eds., Woodhead Publishing, 2016, pp. 81–124. doi: 10.1016/B978-0-08-100040-3.00004-3.

[3] H. Konno, ‘Chapter 8 - X-ray Photoelectron Spectroscopy’, in *Materials Science and Engineering of Carbon*, M. Inagaki and F. Kang, Eds., Butterworth-Heinemann, 2016, pp. 153–171. doi: 10.1016/B978-0-12-805256-3.00008-8.

[4] W. H. Bragg and W. L. Bragg, ‘The reflection of X-rays by crystals’, *Proc. R. Soc. Lond. Ser. Contain. Pap. Math. Phys. Character*, vol. 88, no. 605, pp. 428–438, 1913, doi: 10.1098/rspa.1913.0040.

[5] Z. Qiu, ‘Transition Metal-Based Electrocatalysts for Alkaline Water Splitting and CO2 Reduction’, PhD Thesis, 2019.

[6] E. Bandiello, C. Popescu, J. Á. Sans, D. Errandonea, and M. Bettinelli, *$PrVO_4$ under High Pressure: Effects on Structural, Optical and Electrical Properties*. 2020.

[7] K. T. Butler, D. W. Davies, H. Cartwright, O. Isayev, and A. Walsh, ‘Machine learning for molecular and materials science’, *Nature*, vol. 559, p. 547, 2018.

[8] R. Taleei and M. Shahriari, ‘Monte Carlo simulation of X-ray spectra and evaluation of filter effect using MCNP4C and FLUKA code’, *Appl. Radiat. Isot.*, vol. 67, no. 2, pp. 266–271, 2009, doi: https://doi.org/10.1016/j.apradiso.2008.10.007.

[9] A. Chauhan and P. Chauhan, ‘Powder XRD technique and its applications in science and technology’, *J Anal Bioanal Tech*, vol. 5, no. 5, pp. 1–5, 2014.

[10] G. M. Sheldrick, ‘A short history of SHELX’, *Acta Crystallogr. A*, vol. 64, no. 1, pp. 112–122, 2008.

[11] P. Chauhan, S. Javed, D. C. Levendis, and M. Fernandes, ‘Hydrophobicity directed guest-inclusion for structure-elucidation of enclatherated guests within a crystalline sponge by SC-XRD’, *J. Mol. Struct.*, vol. 1251, p. 132054, Mar. 2022, doi: 10.1016/j.molstruc.2021.132054.

[12] S. Datta and D. J. W. Grant, ‘Crystal structures of drugs: advances in determination, prediction and engineering’, *Nat. Rev. Drug Discov.*, vol. 3, no. 1, Art. no. 1, Jan. 2004, doi: 10.1038/nrd1280.

[13] S. B. Marganakop *et al.*, ‘SCXRD, DFT and molecular docking based structural analyses towards novel 3-piperazin-1-yl-benzo[d]isothiazole and 3-piperidin-4-yl-benzo[d]isoxazoles appended to quinoline as pharmacological agents’, *J. Mol. Struct.*, vol. 1248, p. 131442, Jan. 2022, doi: 10.1016/j.molstruc.2021.131442.

[14] B. R. Pauw, ‘Everything SAXS: small-angle scattering pattern collection

and correction', *J. Phys. Condens. Matter*, vol. 25, no. 38, p. 383201, 2013.
[15] A. G. Kikhney and D. I. Svergun, 'A practical guide to small angle X-ray scattering (SAXS) of flexible and intrinsically disordered proteins', *FEBS Lett.*, vol. 589, no. 19, Part A, pp. 2570–2577, Sep. 2015, doi: 10.1016/j.febslet.2015.08.027.
[16] D. F. Swinehart, 'The beer-lambert law', *J. Chem. Educ.*, vol. 39, no. 7, p. 333, 1962.
[17] 'X-ray microtomography of materials: International Materials Reviews: Vol 44, No 4'. https://www.tandfonline.com/doi/abs/10.1179/095066099101528261 (accessed Sep. 12, 2023).
[18] W. Stiller, 'Basics of iterative reconstruction methods in computed tomography: a vendor-independent overview', *Eur. J. Radiol.*, vol. 109, pp. 147–154, 2018.
[19] U. Kleuker, P. Suortti, W. Weyrich, and P. Spanne, 'Feasibility study of x-ray diffraction computed tomography for medical imaging', *Phys. Med. Biol.*, vol. 43, no. 10, p. 2911, 1998.
[20] A. Vamvakeros *et al.*, 'Interlaced X-ray diffraction computed tomography', *J. Appl. Crystallogr.*, vol. 49, no. 2, pp. 485–496, Apr. 2016, doi: 10.1107/S160057671600131X.
[21] H. J. Landau, 'Sampling, data transmission, and the Nyquist rate', *Proc. IEEE*, vol. 55, no. 10, pp. 1701–1706, Oct. 1967, doi: 10.1109/PROC.1967.5962.
[22] J. Radon, 'On the determination of functions from their integral values along certain manifolds', *IEEE Trans. Med. Imaging*, vol. 5, no. 4, pp. 170–176, 1986.
[23] J. Als-Nielsen and D. McMorrow, *Elements of modern X-ray physics*. John Wiley & Sons, 2011.
[24] R. M. Mersereau, 'Direct Fourier transform techniques in 3-D image reconstruction', *Comput. Biol. Med.*, vol. 6, no. 4, pp. 247-IN4, 1976.
[25] X. Pan, E. Y. Sidky, and M. Vannier, 'Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?', *Inverse Probl.*, vol. 25, no. 12, p. 123009, 2009.
[26] M. J. Willemink and P. B. Noël, 'The evolution of image reconstruction for CT—from filtered back projection to artificial intelligence', *Eur. Radiol.*, vol. 29, pp. 2185–2195, 2019.
[27] R. Gordon, 'A tutorial on ART (algebraic reconstruction techniques)', *IEEE Trans. Nucl. Sci.*, vol. 21, no. 3, pp. 78–93, 1974.
[28] T. G. Feeman and T. G. Feeman, 'Algebraic reconstruction techniques', *Math. Med. Imaging Beginner's Guide*, pp. 137–161, 2015.
[29] H. Guan and R. Gordon, 'Computed tomography using algebraic reconstruction techniques (ARTs) with different projection access schemes: a comparison study under practical situations', *Phys. Med. Biol.*, vol. 41, no. 9, p. 1727, 1996.
[30] S. Helgason and S. Helgason, *The radon transform*, vol. 2. Springer, 1980.
[31] V.-G. Nguyen and S.-J. Lee, 'Parallelizing a matched pair of ray-tracing projector and backprojector for iterative cone-beam CT reconstruction', *IEEE Trans. Nucl. Sci.*, vol. 62, no. 1, pp. 171–181, 2015.

[32] J. Trampert and J.-J. Leveque, ‘Simultaneous iterative reconstruction technique: Physical interpretation based on the generalized least squares solution’, *J. Geophys. Res. Solid Earth*, vol. 95, no. B8, pp. 12553–12559, 1990.
[33] A. H. Andersen and A. C. Kak, ‘Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm’, *Ultrason. Imaging*, vol. 6, no. 1, pp. 81–94, 1984.
[34] Y. Saad, *Iterative methods for sparse linear systems*. SIAM, 2003.
[35] F. Oviedo *et al.*, ‘Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks’, *Npj Comput Mater*, vol. 5, p. 60, 2019.
[36] W. Kraus and G. Nolze, ‘POWDER CELL–a program for the representation and manipulation of crystal structures and calculation of the resulting X-ray powder patterns’, *J. Appl. Crystallogr.*, vol. 29, no. 3, pp. 301–303, 1996.
[37] K. Momma and F. Izumi, ‘VESTA 3 for three-dimensional visualization of crystal, volumetric and morphology data’, *J. Appl. Crystallogr.*, vol. 44, no. 6, pp. 1272–1276, 2011.
[38] A. A. Coelho, ‘TOPAS and TOPAS-Academic: an optimization program integrating computer algebra and crystallographic objects written in C++’, *J. Appl. Crystallogr.*, vol. 51, no. 1, pp. 210–218, 2018.
[39] B. Borie, ‘X-Ray Diffraction in Crystals, Imperfect Crystals, and Amorphous Bodies.’, *J. Am. Chem. Soc.*, vol. 87, no. 1, pp. 140–141, 1965.
[40] T. Hahn, U. Shmueli, and J. W. Arthur, *International tables for crystallography*, vol. 1. Reidel Dordrecht, 1983.
[41] P. J. Linstrom and W. G. Mallard, ‘The NIST Chemistry WebBook: A chemical data resource on the internet’, *J. Chem. Eng. Data*, vol. 46, no. 5, pp. 1059–1063, 2001.
[42] R. Reynolds, ‘The Lorentz-polarization factor and preferred orientation in oriented clay aggregates’, *Clays Clay Miner.*, vol. 34, no. 4, pp. 359–367, 1986.
[43] C. Rajendran, F. S. Dworkowski, M. Wang, and C. Schulze-Briese, ‘Radiation damage in room-temperature data acquisition with the PILATUS 6M pixel detector’, *J. Synchrotron Radiat.*, vol. 18, no. 3, pp. 318–328, 2011.
[44] S. Susman, D. Price, K. Volin, R. Dejus, and D. Montague, ‘Intermediate-range order in binary chalcogenide glasses: The first sharp diffraction peak’, *J. Non-Cryst. Solids*, vol. 106, no. 1–3, pp. 26–29, 1988.
[45] H. Schober *et al.*, ‘Tailored instrumentation for long-pulse neutron spallation sources’, *Nucl. Instrum. Methods Phys. Res. Sect. Accel. Spectrometers Detect. Assoc. Equip.*, vol. 589, no. 1, pp. 34–46, 2008.
[46] K. Lan and J. W. Jorgenson, ‘A hybrid of exponential and gaussian functions as a simple model of asymmetric chromatographic peaks’, *J. Chromatogr. A*, vol. 915, no. 1–2, pp. 1–13, 2001.
[47] M. Haeri, S. Goldberg, and J. L. Gilbert, ‘The voltage-dependent electrochemical impedance spectroscopy of CoCrMo medical alloy using time-domain techniques: Generalized Cauchy–Lorentz, and KWW–Randles functions describing non-ideal interfacial behaviour’, *Corros. Sci.*, vol. 53, no. 2, pp. 582–588, 2011.

[48] T. Ida, M. Ando, and H. Toraya, ‘Extended pseudo-Voigt function for approximating the Voigt profile’, *J. Appl. Crystallogr.*, vol. 33, no. 6, pp. 1311–1316, 2000.
[49] U. Holzwarth and N. Gibson, ‘The Scherrer equation versus the’Debye-Scherrer equation’’, *Nat. Nanotechnol.*, vol. 6, no. 9, pp. 534–534, 2011.
[50] R. Hill and C. Howard, ‘Quantitative phase analysis from neutron powder diffraction data using the Rietveld method’, *J. Appl. Crystallogr.*, vol. 20, no. 6, pp. 467–474, 1987.
[51] H. M. Rietveld, ‘A profile refinement method for nuclear and magnetic structures’, *J. Appl. Crystallogr.*, vol. 2, no. 2, pp. 65–71, 1969.
[52] G. S. Pawley, ‘Unit-cell refinement from powder diffraction scans’, *J. Appl. Crystallogr.*, vol. 14, no. 6, pp. 357–361, 1981.
[53] A. Le Bail, H. Duroy, and J. L. Fourquet, ‘Ab-initio structure determination of LiSbWO6 by X-ray powder diffraction’, *Mater. Res. Bull.*, vol. 23, no. 3, pp. 447–452, 1988.
[54] R. D. Shannon, ‘Revised effective ionic radii and systematic studies of interatomic distances in halides and chalcogenides’, *Acta Crystallogr. A*, vol. 32, no. 5, Art. no. 5, Sep. 1976, doi: 10.1107/S0567739476001551.
[55] S. Dreiseitl and L. Ohno-Machado, ‘Logistic regression and artificial neural network classification models: a methodology review’, *J. Biomed. Inform.*, vol. 35, no. 5–6, pp. 352–359, 2002.
[56] F. Günther and S. Fritsch, ‘neuralnet: Training of neural networks’, *R J.*, vol. 2, no. 1, pp. 30–38, 2010.
[57] P. Baldi and P. J. Sadowski, ‘Understanding dropout’, *Adv. Neural Inf. Process. Syst.*, vol. 26, 2013.
[58] S. Ioffe and C. Szegedy, ‘Batch normalization: Accelerating deep network training by reducing internal covariate shift’, in *International conference on machine learning*, pmlr, 2015, pp. 448–456.
[59] R. Socher, E. H. Huang, J. Pennin, C. D. Manning, and A. Y. Ng, ‘Dynamic pooling and unfolding recursive autoencoders for paraphrase detection’, in *Advances in neural information processing systems*, 2011, pp. 801–809.
[60] M. Sun, Z. Song, X. Jiang, J. Pan, and Y. Pang, ‘Learning pooling for convolutional neural network’, *Neurocomputing*, vol. 224, pp. 96–104, 2017.
[61] Y. Li and Y. Yuan, ‘Convergence analysis of two-layer neural networks with relu activation’, in *Advances in neural information processing systems*, 2017, pp. 597–607.
[62] M. Hein, M. Andriushchenko, and J. Bitterwolf, ‘Why relu networks yield high-confidence predictions far away from the training data and how to mitigate the problem’, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 41–50.
[63] H. Ide and T. Kurita, ‘Improvement of learning for CNN with ReLU activation by sparse regularization’, in *2017 international joint conference on neural networks (IJCNN)*, IEEE, 2017, pp. 2684–2691.
[64] Y. Shin and G. E. Karniadakis, ‘Trainability of ReLU networks and data-dependent initialization’, *J. Mach. Learn. Model. Comput.*, vol. 1, no. 1, 2020.

[65] J. Xu, Z. Li, B. Du, M. Zhang, and J. Liu, ‘Reluplex made more practical: Leaky ReLU’, in *2020 IEEE Symposium on Computers and communications (ISCC)*, IEEE, 2020, pp. 1–7.
[66] A. K. Dubey and V. Jain, ‘Comparative study of convolution neural network’s relu and leaky-relu activation functions’, in *Applications of Computing, Automation and Wireless Systems in Electrical Engineering: Proceedings of MARC 2018*, Springer, 2019, pp. 873–880.
[67] B. Ding, H. Qian, and J. Zhou, ‘Activation functions and their characteristics in deep neural networks’, in *2018 Chinese control and decision conference (CCDC)*, IEEE, 2018, pp. 1836–1841.
[68] A. D. Rasamoelina, F. Adjailia, and P. Sinčák, ‘A review of activation function for artificial neural network’, in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, IEEE, 2020, pp. 281–286.
[69] J. Han and C. Moraga, ‘The influence of the sigmoid function parameters on the speed of backpropagation learning’, in *From Natural to Artificial Neural Computation: International Workshop on Artificial Neural Networks Malaga-Torremolinos, Spain, June 7–9, 1995 Proceedings 3*, Springer, 1995, pp. 195–201.
[70] R. A. Dunne and N. A. Campbell, ‘On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function’, in *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, Citeseer, 1997, p. 185.
[71] M. M. Lau and K. H. Lim, ‘Review of adaptive activation function in deep neural network’, in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*, IEEE, 2018, pp. 686–690.
[72] I. Muhammad and Z. Yan, ‘SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY.’, *ICTACT J. Soft Comput.*, vol. 5, no. 3, 2015.
[73] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, ‘A regression approach to speech enhancement based on deep neural networks’, *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 1, pp. 7–19, 2014.
[74] H. U. Dike, Y. Zhou, K. K. Deveerasetty, and Q. Wu, ‘Unsupervised learning based on artificial neural network: A review’, in *2018 IEEE International Conference on Cyborg and Bionic Systems (CBS)*, IEEE, 2018, pp. 322–327.
[75] N. Grira, M. Crucianu, and N. Boujemaa, ‘Unsupervised and semi-supervised clustering: a brief survey’, *Rev. Mach. Learn. Tech. Process. Multimed. Content*, vol. 1, no. 2004, pp. 9–16, 2004.
[76] R. Zebari, A. Abdulazeez, D. Zeebaree, D. Zebari, and J. Saeed, ‘A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction’, *J Appl Sci Technol Trends*, vol. 1, no. 2, pp. 56–70, 2020.
[77] Y. Zhang, S. Liu, C. Dong, X. Zhang, and Y. Yuan, ‘Multiple cycle-in-cycle generative adversarial networks for unsupervised image super-resolution’, *IEEE Trans. Image Process.*, vol. 29, pp. 1101–1112, 2019.
[78] R. Sathya and A. Abraham, ‘Comparison of supervised and unsupervised learning algorithms for pattern classification’, *Int. J. Adv. Res. Artif. Intell.*, vol. 2, no. 2, pp. 34–38, 2013.

[79] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, ‘Loss functions for image restoration with neural networks’, *IEEE Trans. Comput. Imaging*, vol. 3, no. 1, pp. 47–57, 2016.
[80] Q. Wang, Y. Ma, K. Zhao, and Y. Tian, ‘A comprehensive survey of loss functions in machine learning’, *Ann. Data Sci.*, pp. 1–26, 2020.
[81] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, ‘State-of-the-art in artificial neural network applications: A survey’, *Heliyon*, vol. 4, no. 11, p. e00938, 2018.
[82] X.-S. Yang, ‘Firefly algorithm, stochastic test functions and design optimisation’, *Int. J. Bio-Inspired Comput.*, vol. 2, no. 2, pp. 78–84, 2010.
[83] J. Zhou *et al.*, ‘Adaptive moment estimation for polynomial nonlinear equalizer in PAM8-based optical interconnects’, *Opt. Express*, vol. 27, no. 22, pp. 32210–32216, 2019.
[84] N. S. Keskar and R. Socher, ‘Improving generalization performance by switching from adam to sgd’, *ArXiv Prepr. ArXiv171207628*, 2017.
[85] P. Zhou, J. Feng, C. Ma, C. Xiong, and S. C. H. Hoi, ‘Towards theoretically understanding why sgd generalizes better than adam in deep learning’, *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 21285–21296, 2020.
[86] S. Ruder, ‘An overview of gradient descent optimization algorithms’, *ArXiv Prepr. ArXiv160904747*, 2016.
[87] S. H. Haji and A. M. Abdulazeez, ‘Comparison of optimization techniques based on gradient descent algorithm: A review’, *PalArchs J. Archaeol. EgyptEgyptology*, vol. 18, no. 4, pp. 2715–2743, 2021.
[88] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, ‘On mean absolute error for deep neural network based vector-to-vector regression’, *IEEE Signal Process. Lett.*, vol. 27, pp. 1485–1489, 2020.
[89] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, ‘Image quality assessment: from error visibility to structural similarity’, *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.
[90] Z. Zhang and M. Sabuncu, ‘Generalized cross entropy loss for training deep neural networks with noisy labels’, *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.
[91] J. Jin, K. Fu, and C. Zhang, ‘Traffic sign recognition with hinge loss trained convolutional neural networks’, *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1991–2000, 2014.
[92] G. Xu, Z. Cao, B.-G. Hu, and J. C. Principe, ‘Robust support vector machines based on the rescaled hinge loss function’, *Pattern Recognit.*, vol. 63, pp. 139–148, 2017.
[93] H. Yao, D. Zhu, B. Jiang, and P. Yu, ‘Negative log likelihood ratio loss for deep neural network classification’, in *Proceedings of the Future Technologies Conference (FTC) 2019: Volume 1*, Springer, 2020, pp. 276–282.
[94] R. Kohavi and D. H. Wolpert, ‘Bias plus variance decomposition for zero-one loss functions’, in *ICML*, Citeseer, 1996, pp. 275–83.
[95] S.-C. B. Lo, H.-P. Chan, J.-S. Lin, H. Li, M. T. Freedman, and S. K. Mun, ‘Artificial convolution neural network for medical image pattern recognition’, *Neural Netw.*, vol. 8, no. 7–8, pp. 1201–1214, 1995.

[96] H. Dong *et al.*, ‘A deep convolutional neural network for real-time full profile analysis of big powder diffraction data’, *Npj Comput. Mater.*, vol. 7, no. 1, Art. no. 1, May 2021, doi: 10.1038/s41524-021-00542-4.
[97] H. Sedghi, V. Gupta, and P. M. Long, ‘The singular values of convolutional layers’, *ArXiv Prepr. ArXiv180510408*, 2018.
[98] B. A. Ashqar and S. S. Abu-Naser, ‘Image-Based Tomato Leaves Diseases Detection Using Deep Learning’, 2018.
[99] Z. Tang, W. Jiang, Z. Zhang, M. Zhao, L. Zhang, and M. Wang, ‘DenseNet with Up-Sampling block for recognizing texts in images’, *Neural Comput. Appl.*, pp. 1–9, 2019.
[100] D. M. Pelt and J. A. Sethian, ‘A mixed-scale dense convolutional neural network for image analysis’, *Proc. Natl. Acad. Sci.*, vol. 115, no. 2, pp. 254–259, 2018.
[101] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, ‘On the properties of neural machine translation: Encoder-decoder approaches’, *ArXiv Prepr. ArXiv14091259*, 2014.
[102] V. Badrinarayanan, A. Kendall, and R. Cipolla, ‘Segnet: A deep convolutional encoder-decoder architecture for image segmentation’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, 2017.
[103] K. Cho *et al.*, ‘Learning phrase representations using RNN encoder-decoder for statistical machine translation’, *ArXiv Prepr. ArXiv14061078*, 2014.
[104] C. Doersch, ‘Tutorial on variational autoencoders’, *ArXiv Prepr. ArXiv160605908*, 2016.
[105] S. Yang, W. Zhu, and Y. Zhu, ‘Residual encoder-decoder network for deep subspace clustering’, in *2020 IEEE International Conference on Image Processing (ICIP)*, IEEE, 2020, pp. 2895–2899.
[106] O. Ronneberger, P. Fischer, and T. Brox, ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241.
[107] E. Shelhamer, J. Long, and T. Darrell, ‘Fully Convolutional Networks for Semantic Segmentation’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, pp. 640–651, 2017.
[108] M. Allan *et al.*, ‘2018 Robotic Scene Segmentation Challenge’, *ArXiv*, vol. abs/2001.11190, 2020.
[109] Z. Liu, T. Bicer, R. Kettimuthu, D. Gursoy, F. De Carlo, and I. Foster, ‘TomoGAN: low-dose synchrotron x-ray tomography with generative adversarial networks: discussion’, *J. Opt. Soc. Am. A*, vol. 37, no. 3, p. 422, Feb. 2020, doi: 10.1364/josaa.375595.
[110] J. Dong, J. Fu, and Z. He, ‘A deep learning reconstruction framework for X-ray computed tomography with incomplete data’, *PloS One*, vol. 14, no. 11, p. e0224426, 2019.
[111]M. Ebner *et al.*, ‘An automated framework for localization, segmentation and super-resolution reconstruction of fetal brain MRI’, *Neuroimage*, vol. 206, 2019.
[112] M. A. Kramer, ‘Nonlinear principal component analysis using

autoassociative neural networks', *AIChE J.*, vol. 37, no. 2, pp. 233–243, 1991, doi: 10.1002/aic.690370209.
[113] I. J. Goodfellow *et al.*, 'Generative Adversarial Networks', *ArXiv*, vol. abs/1406.2661, 2014.
[114] D. H. Putra and T. Basaruddin, 'Face Reconstruction with Generative Adversarial Network', in *ICMLSC 2019*, 2019.
[115] K. Javed, G. Hussain, F. Shaukat, and S. O. Hwang, 'A neural network approach to remove rain using reconstruction and feature losses', *Neural Comput. Appl.*, pp. 1–10, 2019.
[116] J. Brusokas and L. Petkevi\vcius, 'Analysis of Deep Neural Network Architectures and Similarity Metrics for Low-Dose CT Reconstruction', *2020 IEEE Open Conf. Electr. Electron. Inf. Sci. EStream*, pp. 1–6, 2020.
[117] X. Yang *et al.*, 'Tomographic reconstruction with a generative adversarial network', *J. Synchrotron Radiat.*, vol. 27, no. 2, pp. 486–493, Mar. 2020, doi: 10.1107/S1600577520000831.

Chapter 3: Fast Analysis of Big Powder Diffraction Data

3.1 Abstract

We present the Parameter Quantification Network (PQ-Net), a regression deep convolutional neural network providing quantitative analysis of powder X-ray diffraction patterns from multi-phase systems. The network is tested against simulated and experimental datasets of increasing complexity with the last one being an X-ray diffraction computed tomography dataset of a multi-phase Ni-Pd/CeO_2-ZrO_2/Al_2O_3 catalytic material system consisting of ca. 20,000 diffraction patterns. It is shown that the network predicts accurate scale factors, lattice parameters and crystallite size maps for all phases, which are comparable to those obtained through full profile analysis using the Rietveld method, whilst also providing a reliable uncertainty measurement of the results. The main advantage of PQ-Net is its ability to yield these results, orders of magnitude faster than via conventional data processing (i.e. Rietveld refinement of individual diffraction datasets) demonstrating its potential as a new tool for real-time diffraction data analysis during *in situ/operando* experiments.

Note this work has been published in the following journal:

Dong, H., Butler, K.T., Matras, D. et al. A deep convolutional neural network for real-time full profile analysis of big powder diffraction data. NPJ Comput. Mater. 7, 74 (2021) as well as being applied in a recent paper as the data analysis approach in place of conventional full profile Rietveld analysis [1].

3.2 Introduction

Over the past decade, advancements in X-ray sources, optics and detector technologies have led to a dramatic increase in the volume and data quality of experimental powder diffraction patterns [2]–[8]. These technical advances are beginning to make high-throughput powder diffraction measurements a reality not just at synchrotron facilities but also in the laboratory [9], [10]. It is now common to acquire diffraction patterns with a sufficient signal-to-noise ratio in matters of minutes with laboratory diffractometers and in milliseconds at X-ray diffraction (XRD) dedicated beamlines at synchrotron facilities. *In situ* and *operando* ultra-fast and/or spatially-resolved multi-dimensional XRD experiments of functional materials and devices previously considered technically infeasible have already been demonstrated [11], [12]. The first 5D *operando* tomographic diffraction imaging experiment (three spatial, one scattering and one dimension to denote time/imposed state) to study a multi-component catalytic reactor for the partial oxidation of methane was presented for example in 2018 [13]. However, these advances come at a cost and this is related to the challenge of handling the large volume of data collected during these experiments.

It is now possible to acquire many Terabytes (TBs) of XRD data per dynamic experiment and this is expected to increase significantly with the advent of the fourth generation synchrotron facilities all around the world, such as the Extremely Brilliant Source (EBS) of the European Synchrotron Radiation Facilities (ESRF) and MAX IV (Sweden) and the scheduled upgrades for Diamond-II (United Kingdom), Petra IV (Germany) and Advanced Photon Source-Upgrade (USA) [14]. It is currently well-accepted that it is the data analysis that is emerging as the bottleneck for measurement science and not the data acquisition and/or the experiment itself. Conventional data analysis methods, such as least-squares minimisation approaches, are not able to keep up with the data collection rates and there is a need for alternative methods which can provide both fast and accurate results [15]. As an example, the Rietveld method, often employed in XRD data when performing full profile

analysis, can yield invaluable physicochemical information regarding the material system under investigation (e.g. lattice parameters, crystallite sizes/strain, atom/site occupancies and weight percentages for mixed-phase systems) but does not scale well with big data. Equally important is the data crunching/visualisation capabilities which are not developed in tandem with the data acquisition capabilities and as a result, dynamic beamtime experiments are largely driven 'blind'.

Nowadays, XRD phase identification has become a common bottleneck in material characterisation and screening [16]. Assuming we have some previous knowledge about the crystalline phases present in the sample under investigation, phase identification can still take hours even for an expert crystallographer [17]. Therefore, there is currently great interest in exploiting machine learning methods for phase identification.

In recent years, deep learning methods involving convolutional neural networks (CNNs) have gained popularity and attracted a lot of attention due to the aforementioned properties and also due to their unique scalability (i.e. ability to handle big data) [18]–[20]. Although primarily explored for applications in medical imaging and tomography [21], [22], their potential for materials science and especially for spectroscopic/scattering techniques is gaining momentum [23], [24]. Regarding XRD, previous studies have mainly focused on developing CNNs for crystal structure prediction, space group classification and phase identification [22], [25]–[30]. For example, Park et al. demonstrated a classification CNN model that can extract crystal-system and space-group information from an XRD pattern [25]. Tatlier used an Artificial Neural Network (ANN) to predict the structures of crystalline materials and zeolites from XRD patterns and it was claimed that the ANN performed better than manual regression [31]. There are also some algorithms that can achieve phase identification without Neural Networks, but they are generally less efficient and/or less accurate than Neural Network models [32]–[37]. It should be pointed out though that most studies focus on developing classification models for

phase identification; that is predicting whether a phase is present or not in an XRD pattern.

Lee et al. proposed a CNN structure used for XRD pattern identification, which also has the potential to roughly predict fractions of phases [22]. The same group also demonstrated a classification CNN model that can extract crystal-system, extinction-group, and space-group information from XRD patterns [38]. However, most of the recent works focus on producing classification models for phase identification. In other words, they can only predict whether a phase is present or not in a diffraction pattern. For example, Lee et al. expanded its model to roughly classify the fraction range that phases lie on [22]. The authors claimed that this particular approach has the fastest training time whilst retaining a good accuracy (about 99 %). For example this approach can predict 38 phases in one pattern simultaneously whilst only taking a few minutes. Furthermore, by adding more nodes in the later layers, the expanded model (shown as blue arrows) has the potential to predict the fractions of phases. For example, for 114 output layers, the model can predict the fraction of one phase lying in the 66-99, 33-66 or 0-33 % range. It is a very smart way to convert a classification problem to a regression problem and serves as a good starting point for the work presented herein. The goal here though is to build up a powerful model that can retrieve not only accurate fractions but also reliable lattice parameters. Therefore, a true regression model is desired.

Another recent work used a similar CNN structure to perform phase identification [30]. This CNN model possesses 1012 nodes inside the last fully connected layer, which means their structure has the potential to predict 1012 different materials. For data preparation, they added the noise taken from real experimental data into the simulated patterns to enhance the robustness of the training models. To use the model, they applied a Fast Fourier Transformation (FFT) on the real data with a high error to get better predictions. They claimed

they got a 96.7 % accuracy rate for the top 5 rankings on many test datasets with 30 hold-out samples.

However, like many other supervised learning methods, CNN phase identification has some important shortcomings [17]. First of all, a trained model cannot predict phases which are not included in the original training dataset, and those new phases may also have a big influence on the prediction of other phases [39]. This disadvantage is risky when trying to use CNN to figure out which phases are in a diffraction pattern because it can give a wrong output without any warning [16]. Secondly, the input scale and output scale are fixed, so it is hard to get a generally applied CNN model that can adapt to the different q ranges and the number of bins. A recurrent neural network (RNN) can solve this problem [40], but it is necessary to balance the adapted input range and required accuracy since the training dataset cannot be infinitely large.

In this chapter, a neural network for parameter quantification (PQ) is demonstrated that we term PQ-Net, the first regression CNN model to be applied in 1D powder diffraction data for extracting physico-chemical information. Specifically, PQ-Net is trained with simulated 1D XRD patterns and it is shown that it can yield accurate predictions for scale factors, crystallite sizes and lattice parameters for both simulated and experimental XRD data. For the latter, a very challenging X-ray diffraction computed tomography (XRD-CT) dataset was chosen in order to test the performance of the network and its feasibility for deployment for real-world applications. We focus on the scale factors, lattice parameters and crystallite sizes because they are sufficient to describe and understand the materials and process performance under operational conditions [13], [41], [42]. This exemplar dataset was acquired using a multi-component Ni-Pd/CeO_2-ZrO_2/Al_2O_3 catalyst and consists of ca. 20,000 diffraction patterns. The results from the analysis of this five-phase system are also compared with results obtained using the Rietveld method with the TOPAS software [43].

3.3 Method

3.3.1 Simulated diffraction libraries

To use the PQ-Net, one needs to simulate the training library according to their experimental data frame. In this chapter, All diffraction libraries were simulated using MATLAB scripts and the TOPAS software v7 with CPU. The neural networks were trained using GPU and a CPU/GPU comparison Table is shown in Table 3.1. A 10 % validation split was applied to the training data for each diffraction library used in this work to train the PQ-Net. The scale factors, lattice parameters and crystallite size values for each pattern were created using random sampling from a fixed range. For each parameter, the minimum value was subtracted from the corresponding array and subsequently normalised. This step proved very important for the stability of the network as it allowed all parameters to have values in the same range (0-100). The minimum and maximum values are recorded in a text file, and then the models' predictions on the experimental dataset can be easily converted to real values before normalisation. A 3XS Data Science Workstation C264X2 with 2x Intel Xeon Silver 4216, 350 GB Random Access Memory (RAM) and 2x Quadro RTX 8000 was used for the development and training of the neural networks used in this work and the Rietveld analysis of the diffraction patterns.

Table 3.1: Differences in Models' Performance with CPU and GPU on the single-phase PQ-Net.

Training library size		Training time per epoch (s)	Predicting time on 1k patterns (s)
1,000	CPU	27	5.54
	GPU	2	0.61
10,000	CPU	279	5.54
	GPU	14	0.61
50,000	CPU	1396	5.54
	GPU	70	0.61

3.3.2 XRD-CT dataset

In this chapter I'll use the previously published XRD-CT measurements of a Ni-Pd/CeO_2-ZrO_2/Al_2O_3 catalyst that were performed at beamline station ID31 of the ESRF using a 70 keV monochromatic X-ray beam focused to have a spot size of 20 × 20 μm as our test dataset [13]. The 2D powder diffraction patterns were collected using the state-of-the-art Pilatus3 X CdTe 2 M hybrid photon counting area detector. The total acquisition time per point was 20 ms. Tomographic measurements were made with 225 translation steps (translation step size of 20 μm) covering 0–180 ° angular range, in steps of 1.125 ° (i.e. 160 line scans). The detector calibration was performed using a CeO_2 NIST standard. Every 2D diffraction image was converted to a 1D powder diffraction pattern after applying an appropriate filter (i.e. 1 % trimmed mean filter) to remove outliers using in-house developed MATLAB scripts [44]. The final XRD-CT images were reconstructed using the filtered back projection algorithm and were cropped to a 151 × 151 × 2048 reconstructed data volume (22801 diffraction patterns in total).

3.3.3 Rietveld analysis

Full profile analysis of the spatially-resolved diffraction patterns present in the XRD-CT data was performed using the Rietveld method. MATLAB scripts were used in combination with the TOPAS software v7 to perform the analysis in a sequential manner, processing one diffraction pattern at a time. A mask was applied so that only the pixels in the images corresponding to sample regions would be processed which can significantly decrease the number of patterns to be analysed. Rietveld analysis was performed using the summed diffraction pattern of each XRD-CT dataset prior to the Rietveld analysis of the XRD-CT data in order to have a good starting model before performing the batch Rietveld analysis. A 2nd-degree Chebyshev polynomial was used to model the background as it was observed to be fairly linear and the scale factor, lattice parameters and crystallite size were refined for each phase in the model. The results from the refinements were imported into MATLAB in order to create the various figures presented in this work (e.g. phase distribution maps based on

the scale factors or weight percentages, lattice parameters, and so on). The Rietveld analysis of the XRD-CT data presented in this work was based on the intensity of the scale factors.

3.4 Results & Discussion

3.4.1 Single-phase PQ-Net architecture and simulated XRD-CT dataset

The architecture of the PQ-Net for a single-phase of cubic symmetry is presented in Figure 3.1. The PQ-Net architecture can be divided into three main parts. The first part (pattern-block), which decreases the size of the XRD pattern and extracts local features, involves three convolutional layers (128 filters, kernel size of 35 and stride equal to one) followed by a max-pooling layer (stride equal to two). The second part, which is replicated for each crystalline phase used in the model, includes a sequence of five convolutional and max-pooling layers ending with a flatten layer; this extracts features from the XRD pattern related to the specific phase (phase-block). The third and final part (parameter-block) contains two fully connected (dense) layers ending with one single output per phase parameter. A glossary explaining the various layers in more depth is available in the SI.

The parameter-block part of the architecture contained the majority of the trained weights so our initial work focused on minimising both the depth and width of the dense layers. It was found that the width of the second dense layer has to be doubled for the crystallite size and lattice parameters as these are more challenging parameters to quantify compared to scale factors. A dropout layer has been used after each fully connected layer (apart from the final layers) to prevent overfitting. Regarding the training process, it was seen that using mean absolute error (MAE) as the loss function instead of mean squared error (MSE) yielded a more accurate model; this is not surprising as MAE is known to better handle outliers present in the training data. This feature is crucial as there are diffraction patterns in the training data that contain low intensity reflections

and the network has to adapt and learn to ignore them. It should be noted that the various parameters were normalised prior to the training of the network since their magnitude varied significantly (i.e. between scale factors, lattice parameters and crystallite sizes). Adam was used as the optimisation algorithm and the learning rate was set to 0.0005 for all case studies presented in this chapter[45].

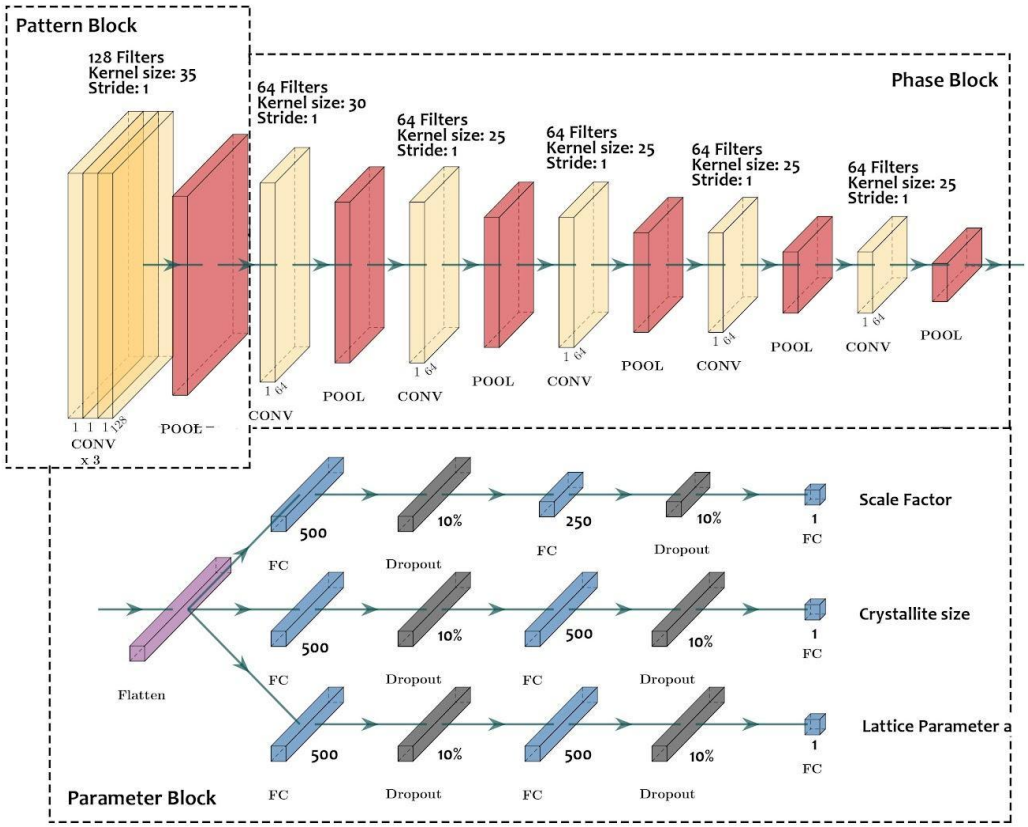

Figure 3.1: Regression CNN architecture for Ni phantom experimental system featuring a single phase. CONV represents 1-D convolutional layers, Pool represents max-pooling layers, FC represents fully connected layers, and Dropout represents dropout layers with 10 % dropout rate. The network consists of eight convolutional layers, six max-pooling layers, one flatten layer, nine fully connected layers and six dropout layers in total. There are three routes connected to the flatten layer which give predictions for scale factor, crystallite size and lattice parameter a, respectively. Each route has three fully connected layers whose scales are shown in the figure. The number of filters, kernel sizes and the stride of convolutional layers are also given in the figure. All max-pooling layers have stride equal to 2.

As a proof-of-concept study, the PQ-Net was first trained with noiseless and zero-background XRD patterns of a Ni fcc structure (ICSD: 64989) simulated

with the TOPAS software v7 [43]. Each simulated diffraction pattern has 2000 bins which cover 0-10 $Å^{-1}$ Q-range, and the parameters are randomly selected from the range:

- Ni scale factor range: random sampling between 0 - 4 × 10^{-9}
- Ni lattice parameter range: 3.55 ± 0.025 Å
- Ni crystallite size range: 2-25 nm

These diffraction patterns were used as the training data with a 10 % validation split. The impact of the diffraction library size (i.e. size of the training data) was investigated and the results are summarised in Table 3.2 there was no apparent change in performance with libraries containing more than 100 K patterns. There is also a sudden drop in MAE above 10 K patterns, implying that accurate results could be obtained with libraries as small as 20 K. It should also be pointed out that the MAE values provide an indication of accuracy regarding the predictions of the network but a PQ-Net model trained with relatively high MAE can potentially still yield accurate predictions. All models are trained with a batch size of 32.

Table 3.2: Impact of Library Size.

Library size	Epochs	MAE	Val MAE
1K	21	17.3640	28.2726
2K	24	7.8272	7.7344
5K	17	6.0237	7.1791
10K	13	5.8395	8.1244
20K	12	4.0839	3.8190
50K	11	2.8442	3.0839
100K	25	1.2830	1.4234
250K	13	1.2046	1.6934

To evaluate the performance of the PQ-Net we simulated a phantom XRD-CT dataset using a scale factor map derived from the Rietveld analysis of an experimental XRD-CT dataset from a catalyst sample [46]. One advantage of

this dataset is the realistic nature of this intensity map compared to idealised ones, such as the Shepp-Logan phantom. More important though is that one can easily segment the various particles, assign different properties to them and generate multiple XRD-CT datasets with varying chemistry in an efficient manner. More details regarding the phantom simulation can be found in our previous work [47]. These 120 x 120 (14400) diffraction patterns were used as test data for the single-phase PQ-Net. The results from a single-phase PQ-Net trained with 100 K patterns applied to this test set are presented in panel in Figure 3.2. The ground truth (GT) maps for the various parameters are presented in the first row, the results from the PQ-Net in the second row and the difference between the two maps in the third row. The scale factor is the easiest parameter to train the PQ-Net for, so it is not surprising that it yields very accurate maps with a relative error per pixel of less than 5 % (Figure 3.2).

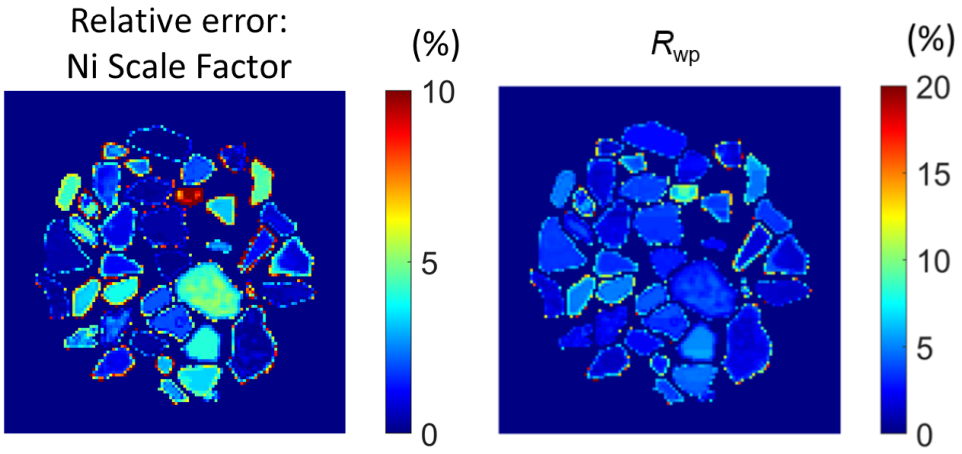

Figure 3.2: Left: Relative error map for the Ni scale factor by comparing the results obtained with the PQ-Net and the ground truth. Right: R_{wp} map as calculated with TOPAS software v7.

Impressively though it can be seen that the error for the crystallite size is below 1 nm for all catalyst particles and similarly the error for the lattice parameter is below 10^{-3} Å. The weighted profile R-factor (R_{wp}) also is below 5 % for the majority of pixels(Figure 3.2). To illustrate this further, the mean diffraction pattern from three catalyst particles exhibiting different properties were extracted and analysed with PQ-Net. The results are presented in panel b of Figure 3.3 and it can be clearly seen that the network is able to accurately predict the three patterns (R_{wp} of 3.144, 1.191 and 1.835 % respectively). The corresponding Ni parameters are summarised in Table 3.3.

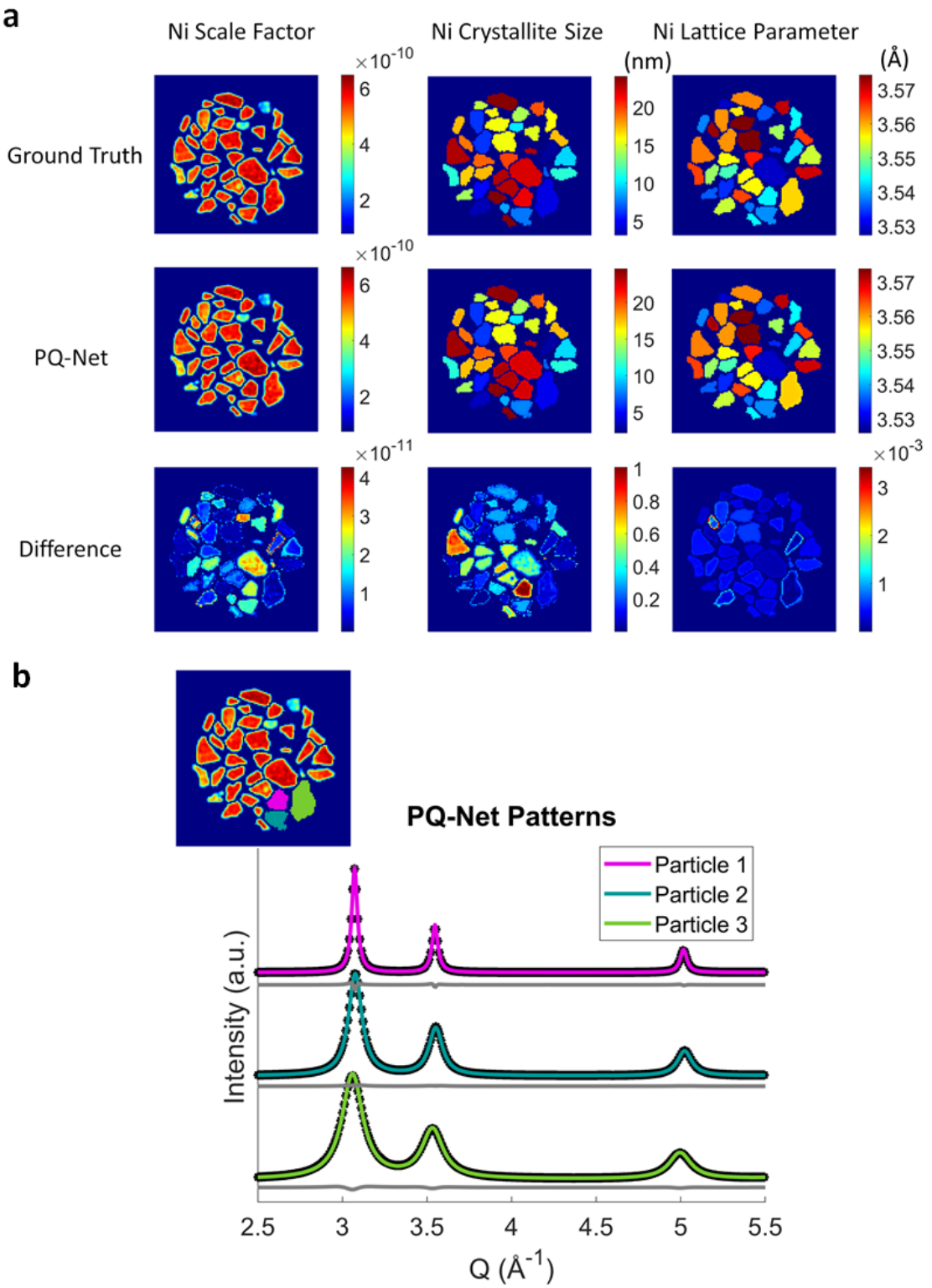

Figure 3.3: Panel a: Ground truth maps (top row), results obtained with the PQ-Net (middle row) and their absolute difference (bottom row). First column corresponds to scale factors, second column to crystallite sizes (nm) and third column to lattice parameter (Å). The colour scale bars represent the values obtained through Rietveld analysis for each pixel in the maps. Panel b: Comparison between the average diffraction patterns extracted from three particles of interest and the TOPAS generated patterns using the parameters predicted by the PQ-Net.

Table 3.3:The predicted and ground truth lattice parameters and crystallite sizes of the three selected particles that are shown in Figure 3.3.

Ni		Lattice Parameter (Å)	Crystallite Size (nm)
Particle 1	Ground Truth	3.544	21.6
	PQ-Net	3.543	23.4
Particle 2	Ground Truth	3.539	7.6
	PQ-Net	3.539	7.6
Particle 3	Ground Truth	3.558	4.5
	PQ-Net	3.558	4.6

3.4.2 Influence of layer sizes and optimizers.

During our network development, we explored the impact of various parameters related to network architecture and the training process itself. Based on the simulated single-phase Ni dataset I showed above, we tested the influence of the numbers of filters and the layer sizes of the 1D convolutional layers and the fully connected layers respectively. Table 3.4 shows the MAEs of the validation dataset when the parameters are halved or doubled. We then draw the conclusion that increasing the filter numbers does not necessarily mean better performance, but it can easily lead to local minima especially when the training dataset is small. By considering the balance of the model's performance and the training time, we chose to use the parameter settings shown in Figure 3.1.

Table 3.4: The MAEs of Validation Dataset with Different Layer Sizes in Single-Phase PQ-Net

	With 0.5 x filter numbers	With original convolutional layer size	With 2 x filter numbers
With 0.5 x nodes in fully connected layers	5.3	5.49	6.08
With original fully connected layer size	3.47	3.28	4.07
With 2 x nodes in fully connected layers	3.21	3.23	3.01

The choice of optimisation algorithms (optimizers) is another factor that influences the models' performance. We tested five commonly used optimizers which are Stochastic Gradient Descent (SGD) [48], Adam [45], SGD with momentum [49], Adagrad [50], and RMSProp [51]. All training is performed on single-phase PQ-Net architectures on 10,000 simulated patterns and assessed using the MAE loss function. Their performance is shown in Figure 3.4. We then decided to use the Adam optimizer on all our works shown in this chapter, because it works the best among all methods tested in terms of both convergence speed and accuracy.

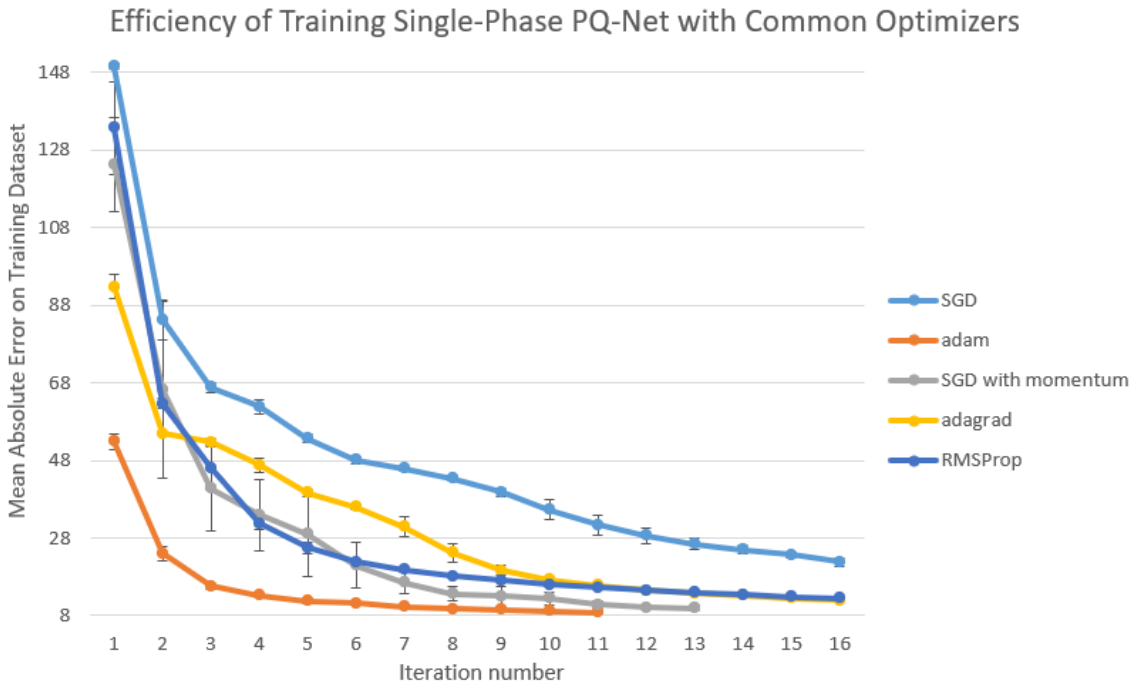

Figure 3.4: Comparison among the performance of common optimizers. MAE is chosen as the loss function. The training was performed on single-phase PQ-Net architectures with 10,000 training patterns with learning rates as 0.0005. Not all optimizers reach 16 epochs because they have got enough accuracy, and the validation loss did not improve anymore.

3.4.3 Multi-phase deep ensemble PQ-Net and simulated XRD-CT dataset

The next logical step was to test the PQ-Net against simulated multi-phase XRD patterns which is a closer approximation to most experimental data and closer to real-world applications. Here, we chose a five-phase system that corresponds to the chemistry encountered in the experimental XRD-CT dataset

presented in the next section. The choice of this material system was based on the complexity and its challenging nature; it is a multi-component system with various phases exhibiting different symmetries and their physico-chemical properties vary spatially as it will be shown in the next section.

The five crystalline phases modelled here are NiO (ICSD: 9866), PdO (ICSD: 24692), CeO_2 (ICSD: 72155), ZrO_2 (ICSD: 66781) and theta-Al_2O_3 [52]. The parameters refined here are the scale factors, lattice parameters and crystallite size for each phase. Poisson noise and a varying linear background (2nd degree Chebyshev polynomial) were also introduced in the diffraction libraries used to train the multi-phase PQ-Net. The parameter ranges for simulating the training diffraction patterns are shown in Table 3.5.

Table 3.5:The ranges of parameters used in simulating the five-phase POX dataset.

	Phase 1 NiO	**Phase 2 CeO_2**	**Phase 3 ZrO_2**	**Phase 4 PdO**	**Phase 5 Al_2O_3**
Scale Factor	0-1 × 10-9	0-1 × 10-10	5.350 ± 0.05	3.010 ± 0.02	0-1 × 10-11
Crystallite Size (nm)	2-50	2-50	2-50	2-50	2-50
Lattice Parameter a (Å)	4.137 ± 0.02	5.350 ± 0.05	3.589 ± 0.02	3.010 ± 0.02	11.965 ± 0.02
Lattice Parameter c (Å)			5.230 ± 0.02	5.350 ± 0.02	11.133 ± 0.02
Lattice Parameter b (Å)					11.071 ± 0.02
Lattice Parameter beta (°)					103.246 ± 1

To predict all 20 parameters with one architecture, we firstly tried the one-go CNN architecture shown in Appendix 1, Figure S1.1. and it can be seen that it is a simple extension of the one presented in Figure 3.1. However, the network easily converges to focus on only a few parameters due to the different difficulties of refining each parameter. We split the CNNs according to the number of phases used in the dataset to force the CNN not to ignore any minor phase or parameter. To balance the scale of different parameters during training, and make all parameters have equal contribution to the loss function, all the parameters are normalised from 0-100 during training, and the normalisation factors are recorded to refine the correct values afterwards. The multi-phase PQ-Net architecture we finally used is presented in Figure 3.5. [52]

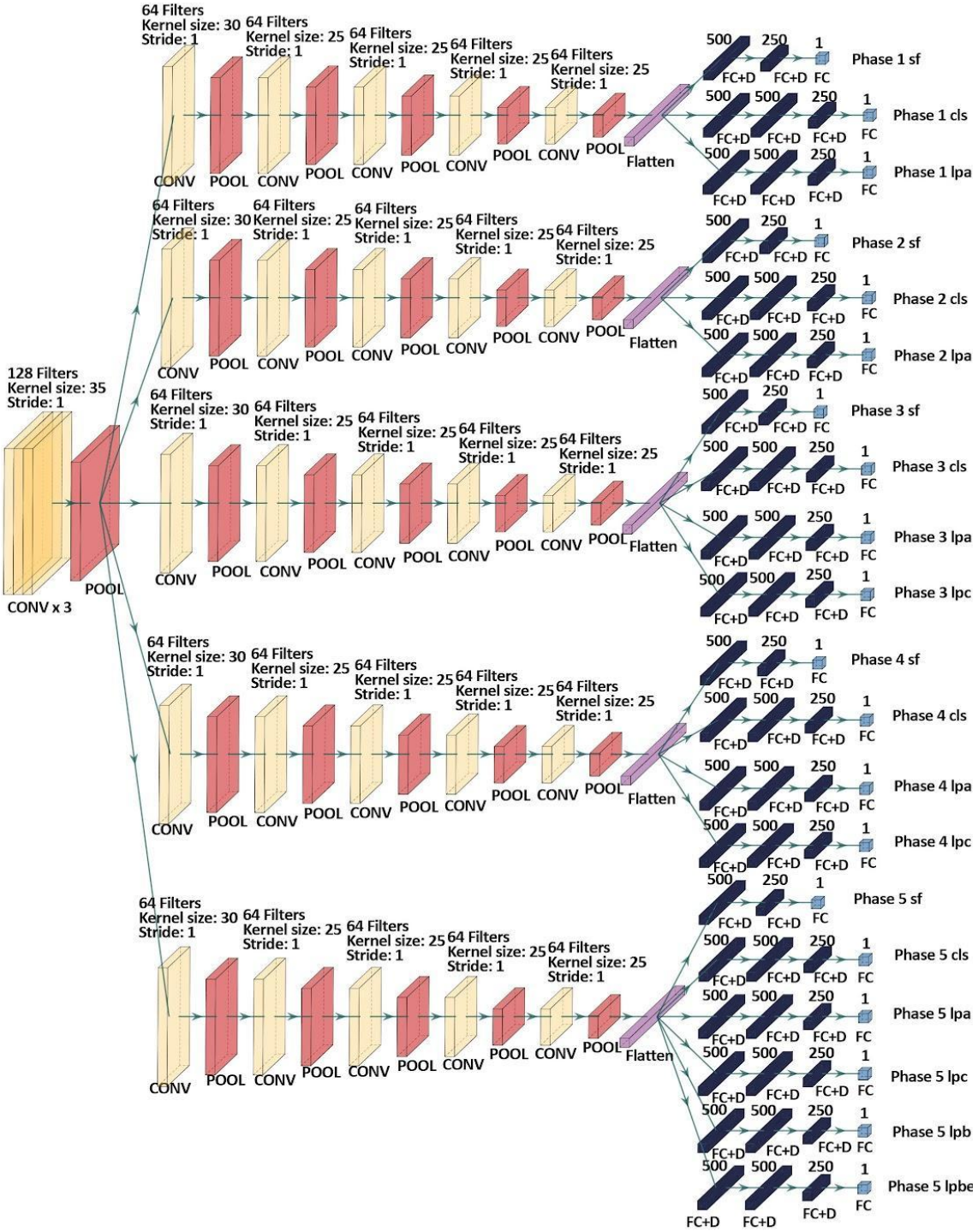

Figure 3.5: Regression CNN architecture for POX catalyst experimental system with five phases. CONV represents 1-D convolutional layers, POOL represents max-pooling layers, FC represents fully connected layers, and FC+D represents a fully connected layer and a dropout layer. The CNN splits after the first max-pooling layer and forms five sub-CNNs corresponding to the five phases in the experimental system. As shown in the schematic, each parameter has its own route after flattening layers. Scales of fully connected layers are shown above them. There is also a 10 % dropout layer after each hidden fully connected layer. The number of filters, kernel sizes and the stride of convolutional layers are also given. All max-pooling layers have strides equal to 2. The sf, cls, lpa, lpb, lpc, and lpbe represent the scale factor, crystallite size, lattice parameters a, b, c, and beta respectively.

The Influence of training library size on the models' accuracy is shown in Table 3.6. Each training pattern contains 740 bins which cover the 1.8-8 $Å^{-1}$ Q range. All models are trained with a batch size as 32 and it takes 53 ms per step on our computer. We used an early stopping factor of 3 so that the training process can automatically stop when the validation loss does not improve after three epochs.

Table 3.6: The influence of the training library sizes on the performance of 5-phases PQ-Net model.

library size	Epochs	MAE	Val MAE
1K	81	130.18	243.57
2K	78	76.31	164.31
5K	43	88.65	135.48
10K	53	88.29	125.91
20K	47	84.54	115.32
50K	31	71.32	99.79
100K	38	85.49	96.31
250K	34	71.94	80.57

To improve the performance and robustness of our model, necessary for complex tasks such as this, we implemented and evaluated deep ensembles of the PQ-Net [53]–[55]. Deep ensembles are based on the assumption that different initial weights of a CNN will lead to different local minima which may not necessarily be a minimum on the validation data. However, these local minima are evenly distributed in parameter space and therefore sample different possible solutions efficiently [56]. Therefore, simply retraining a CNN model several times and taking the average of the outputs can usually yield better results than using only one model.

To introduce more randomness in the training of our model, we used random initial weights and also added a dropout layer after each fully connected layer apart from the final layers. As expected, due to the dropout layers added, individual models in deep ensembles show worse performance during training compared to the models without dropout layers. However, the distribution of the predicted values is narrower and more centred around the ground truth value.

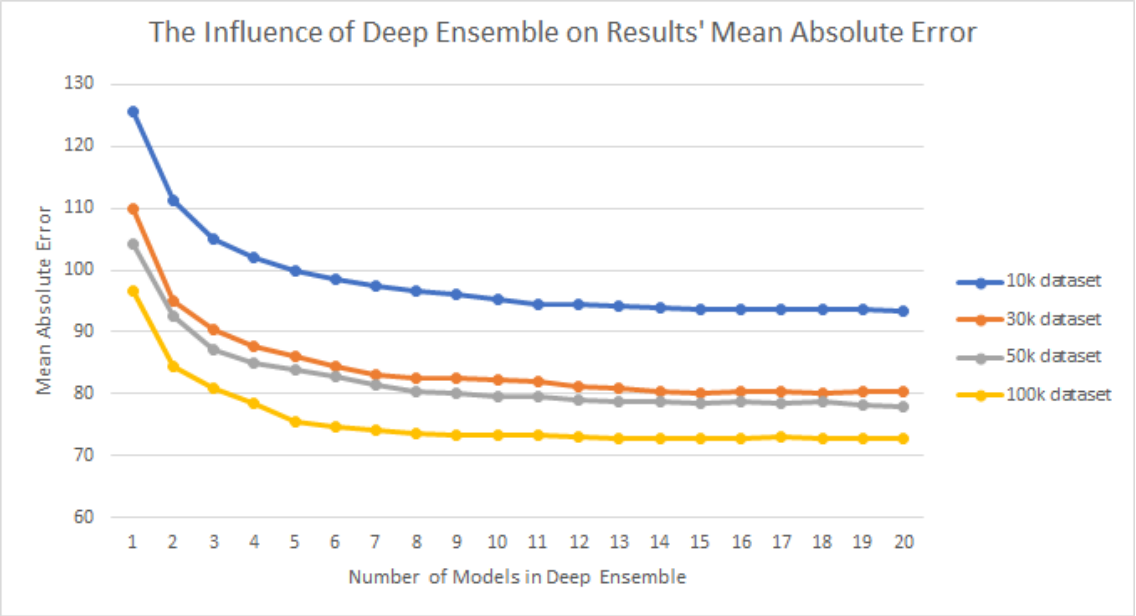

Figure 3.6: The influence of the deep ensemble sizes on models' performance. The points in the chart represent the sum of 20 outputs' MAEs in the simulated validation dataset. All curves tend to flatten out as the number of models in a deep ensemble increases; and the larger the training dataset is, the faster it flattens out. Since the uncertainties of individual models are very large, when the deep ensemble size is less than 5, the MAE shown in the figure is the average MAE among five same-sized deep ensemble models (All individual models are unique).

Figure 3.6 shows the influence of deep ensemble acting on the results mean absolute error (MAE). It can be seen that the more individual models in a deep ensemble model, the more accurate results it can generate. Another benefit of deep ensembles is that the method can give a referable uncertainty of the results. We tested twenty different deep ensemble sizes with four training dataset sizes. Because the predictions may have large standard deviations while the deep ensemble sizes are small, while the deep ensemble sizes are less than four, the mean absolute errors are the average mean absolute errors among five deep ensemble models. According to Figure 3.6, deep ensembles

can significantly reduce the mean absolute error on models with the same training dataset size.

By taking the standard deviation among all individual models' predictions on a parameter, one can easily estimate the precision of a deep ensemble result. However, because the time taken by training and merging individual models can rise proportionally as the number of models in a deep ensemble increases and the curve tends to flatten out, using a deep ensemble of size between five to ten is the most realistic option for real-world applications. It is worth mentioning that the MAEs for the validation dataset have a lower limit which cannot be surpassed by increasing the size of the training dataset or the deep ensemble because the crystallite sizes and lattice parameters are very hard to predict when scale factors are approaching zero.

The performance of the trained multi-phase PQ-Net was evaluated with a simulated XRD-CT dataset using the same intensity image as for the single-phase dataset. Here, the same strategy was followed creating five single-phase XRD-CT datasets, one for each component which was then added together to form the multi-phase simulated XRD-CT dataset. This dataset was then passed to the PQ-Net for analysis and quantification of the various parameters. Each phase's scale factor, lattice parameter, and scale factor results from the deep ensemble PQ-Net consisting of ten models and trained with 100 K patterns are presented in Figures 3.7 to 3.11. It can be clearly seen that the PQ-Net is able to create accurate phase distribution maps for all components and retain the local features and relative intensities. Importantly, it is able to also create accurate crystallite size and lattice parameter maps for all phases while the R_{wp} remains <10 % for all particles (Figure 3.12).

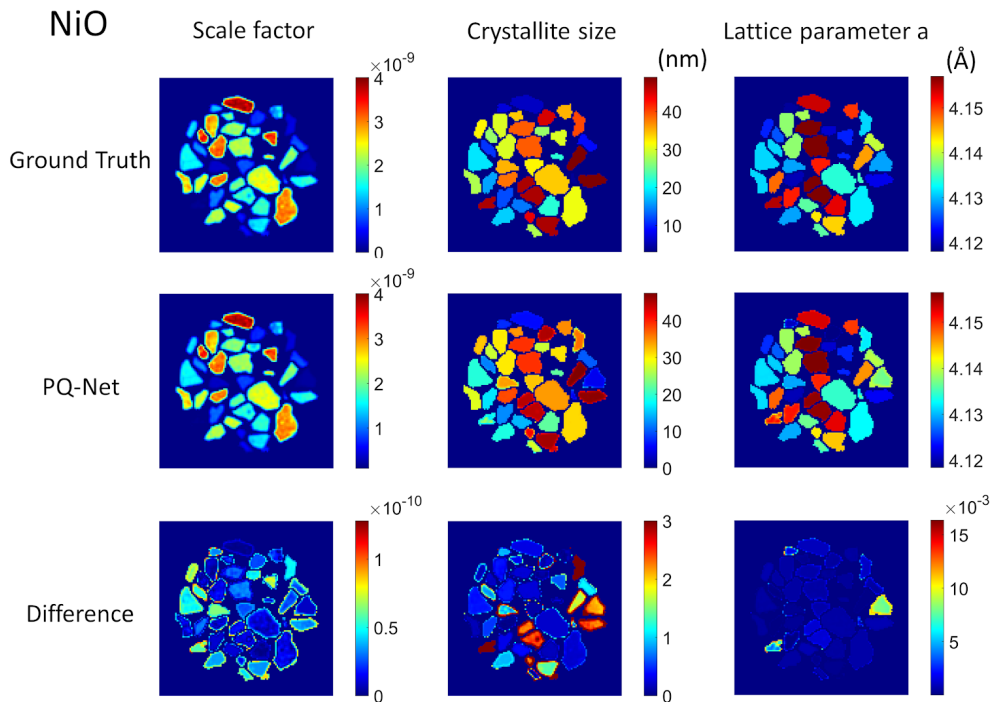

Figure 3.7: Comparison of the scale factor, lattice parameter and crystallite size maps obtained with the PQ-Net (ten model deep ensemble trained with 100 K patterns) for the NiO phase and the ground truth values.

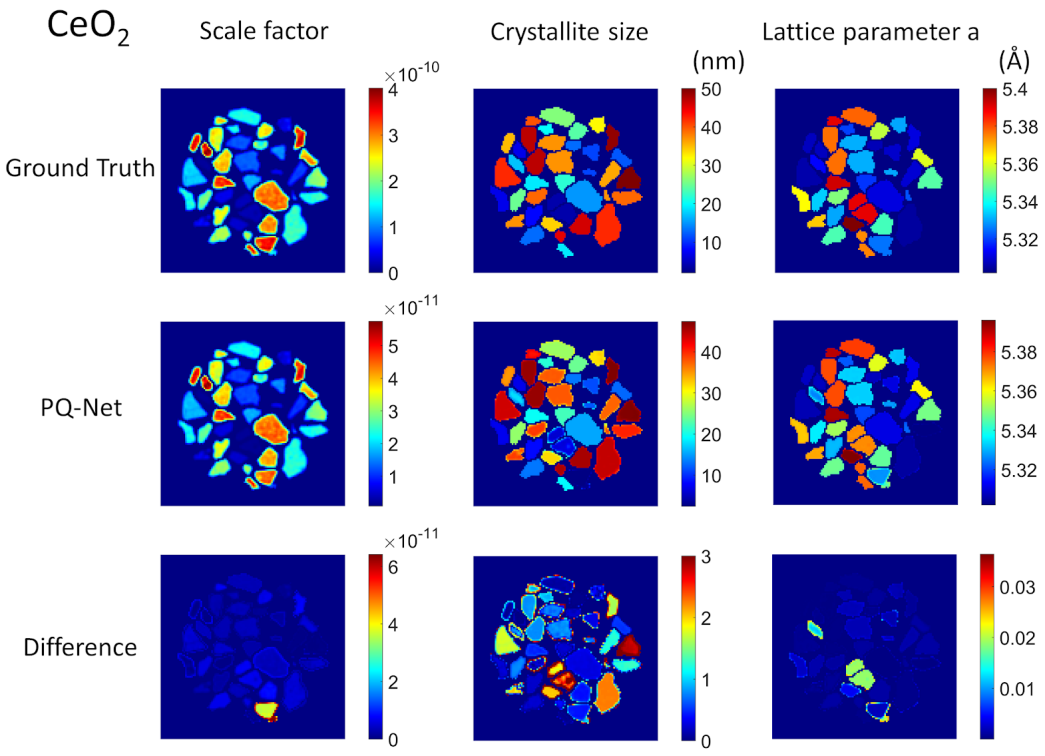

Figure 3.8. Comparison of the scale factor, lattice parameter and crystallite size maps obtained with the PQ-Net (ten model deep ensemble trained with 100 K patterns) for the CeO_2 phase and the ground truth values.

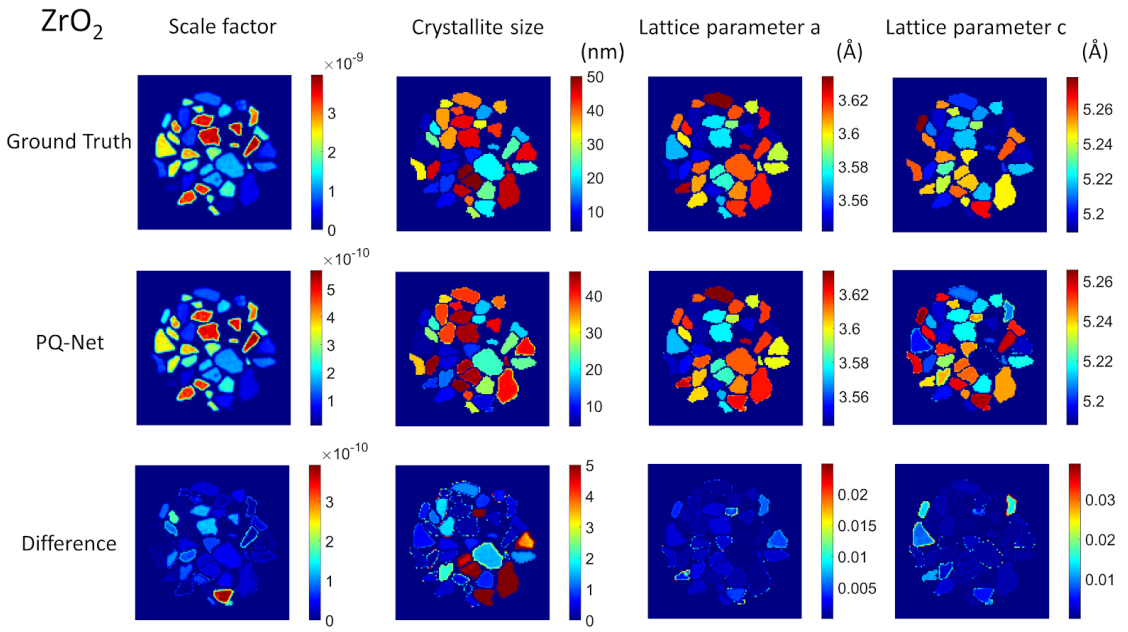

Figure 3.9: Comparison of the scale factor, lattice parameter and crystallite size maps obtained with the PQ-Net (ten model deep ensemble trained with 100 K patterns) for the ZrO_2 phase and the ground truth values.

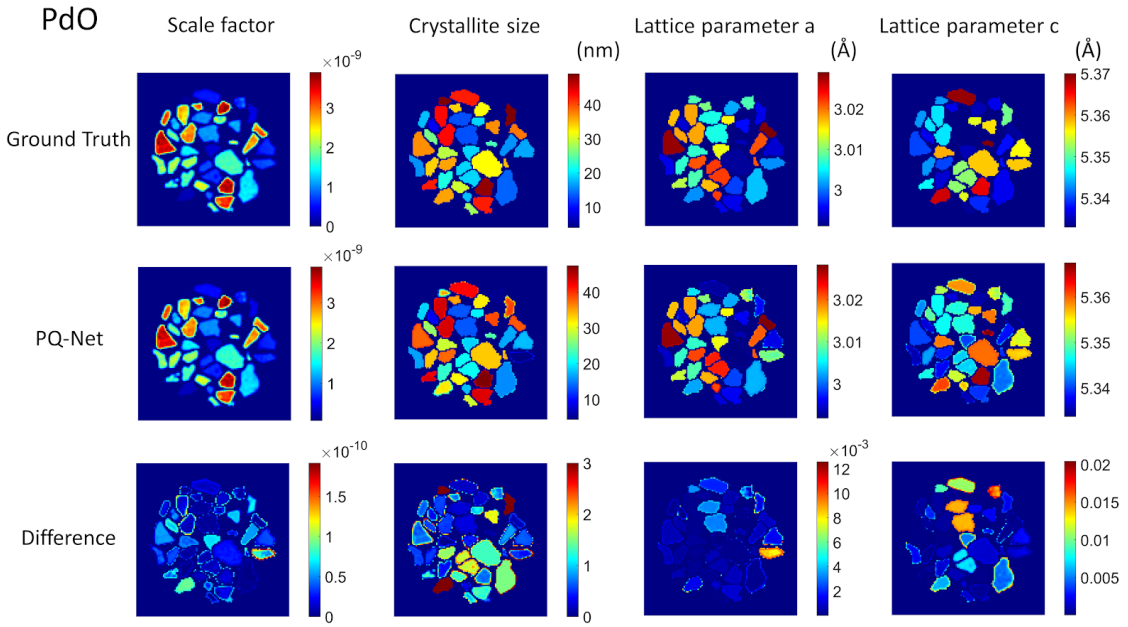

Figure 3.10: Comparison of the scale factor, lattice parameter and crystallite size maps obtained with the PQ-Net (ten model deep ensemble trained with 100 K patterns) for the PdO phase and the ground truth values.

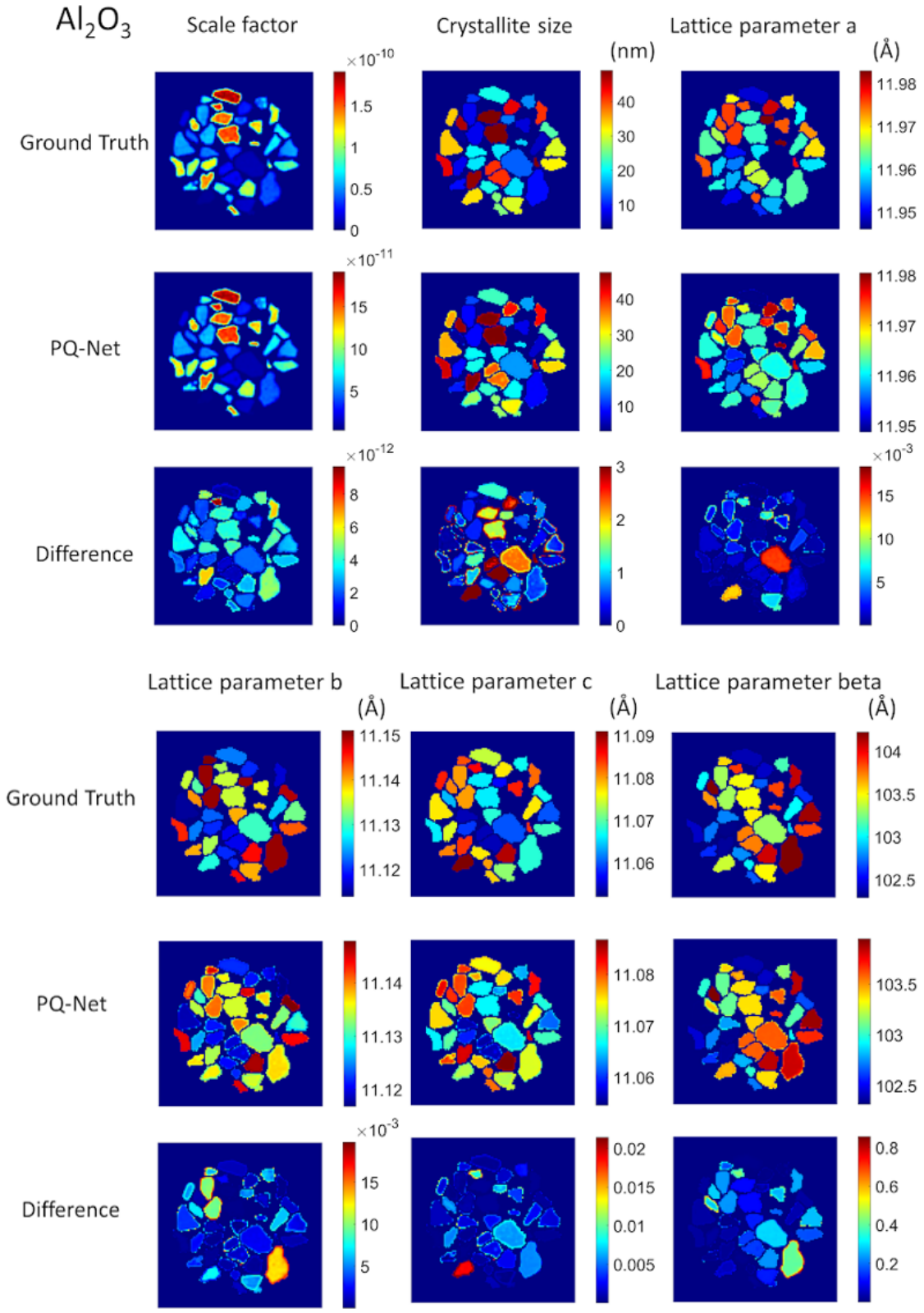

Figure 3.11: Comparison of the scale factor, lattice parameter and crystallite size maps obtained with the PQ-Net (ten model deep ensemble trained with 100 K patterns) for the Al_2O_3 phase and the ground truth values.

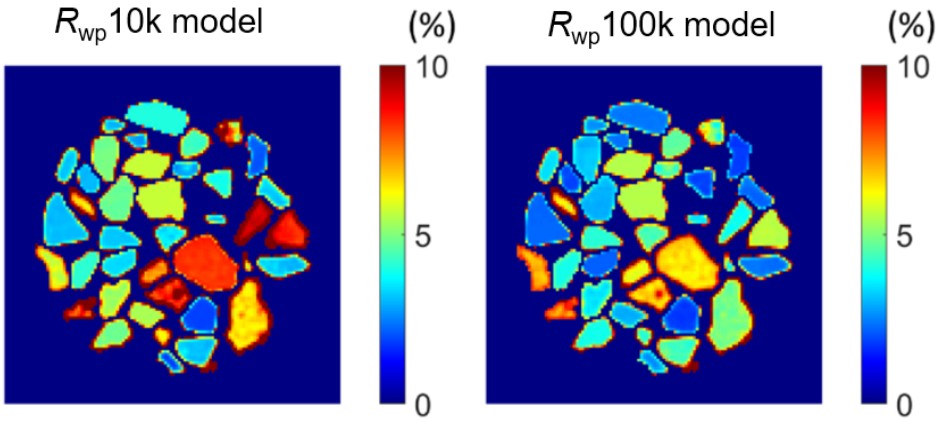

Figure 3.12: Comparison of R_{wp} maps obtained with a 10-model deep ensemble PQ-Net trained with 10 and 100 K patterns respectively.

To further illustrate the ability of the PQ-Net to handle multi-phase systems, we also extracted three diffraction patterns from the XRD-CT which correspond to the mean diffraction patterns from three particles. The results from the analysis of these three patterns are presented in Figure 3.13 where it is clearly shown that the PQ-Net with 100 K training patterns is able to model them accurately (R_{wp} of 5.024, 5.419 and 4.798 % respectively; see also Appendix 1 Tables S1.1 - S1.5).

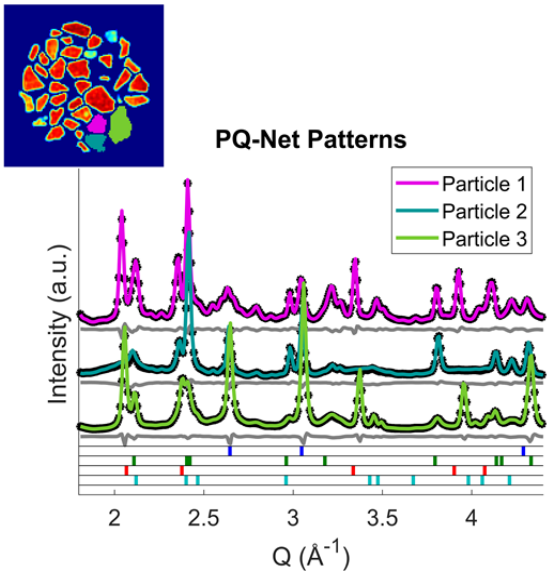

Figure 3.13: Comparison between the average diffraction patterns extracted from three particles of interest and the TOPAS generated patterns using the parameters predicted by the PQ-Net. The main reflections for NiO (blue ticks), PdO (green ticks), CeO_2 (red ticks) and ZrO_2 (cyan ticks) are also presented

3.4.4 Experimental XRD-CT dataset

The final and most challenging dataset is an experimental XRD-CT dataset acquired using the Ni-Pd/CeO_2-ZrO_2/Al_2O_3 catalyst. Details regarding its preparation and phase identification are provided in our previous work [13]. Here we use the previous deep ensemble PQ-Net consisting of 10 models and trained with 100 K patterns (training data) to analyse the 22801 XRD patterns (test data) present in the XRD-CT dataset (XRD-CT images of 151 x 151 pixels). We also perform conventional Rietveld analysis which serves as an approximation to the ground truth and the benchmark to assess the performance of the PQ-Net.

The results for the scale factors for all phases using the Rietveld method and the PQ-Net are presented in Figure 3.14. It can be seen that the PQ-Net is able to accurately predict the scale factors of both the main phases, such as Al_2O_3, NiO and ZrO_2, and also minor components such as the PdO. It should be noted here that the PdO should not have been detectable at this resolution due to its low content (<1 wt.%) and it is only possible due to areas of high concentration near the surface of some particles. Importantly, the PQ-Net results are consistent with the ones obtained with the Rietveld method showing for example that the ZrO_2 (or Zr-rich $Ce_xZr_{1-x}O_2$ phase, where $x \leq 1$) is located only at the periphery of the catalyst particles while the CeO_2 (or Ce-rich $Ce_xZr_{1-x}O_2$ phase) is present also at the interior of the catalyst particles. The Al_2O_3 distribution is correctly shown to be homogeneous in all catalyst particles while NiO, the primary catalyst active component, is also present in all particles. The uncertainty maps shown in the last row of Figure 3.14 correspond to the standard deviation of the calculated values from the 10 models of the deep ensemble PQ-Net.

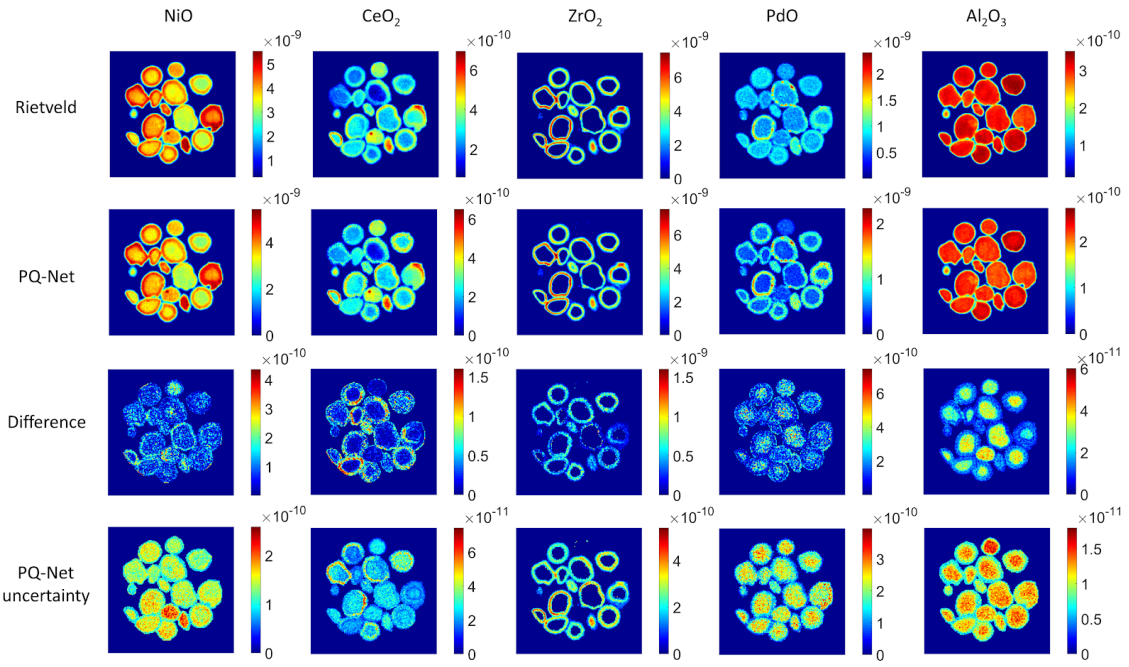

Figure 3.14: Scale factor maps obtained with the Rietveld method (top row), results obtained with the PQ-Net (second low), their absolute difference for the experimental multi-phase NiO-PdO-CeO_2-ZrO_2-Al_2O_3 system (third row) and the uncertainty maps of the deep ensemble PQ-Net (last row).

Importantly, apart from the scale factors, the PQ-Net is able to capture the chemical gradients present in this challenging XRD-CT dataset. As shown in Figure 3.15, it is able to resolve the heterogeneities in the crystallite size and lattice parameter of the CeO_2-ZrO_2 phases. As we have previously reported, the CeO_2 crystallite size and lattice parameter follow an egg-shell distribution in this catalyst with lower values for lattice parameter and crystallite size at the shell compared to the core of the particles. As expected, this result is reproducible with the Rietveld method but it can also be seen that the PQ-Net yields the same results. Moreover, the difference between the CeO_2-ZrO_2 lattice parameter maps obtained with the Rietveld method and the PQ-Net is below 2×10^{-2} Å for the majority of the particles while for the crystallite size maps the difference is in the order of 1-2 nm. The uncertainty maps presented in the last row of Figure 3.15 indicate that the error for the crystallite size for both phases is in the order of 1 nm.

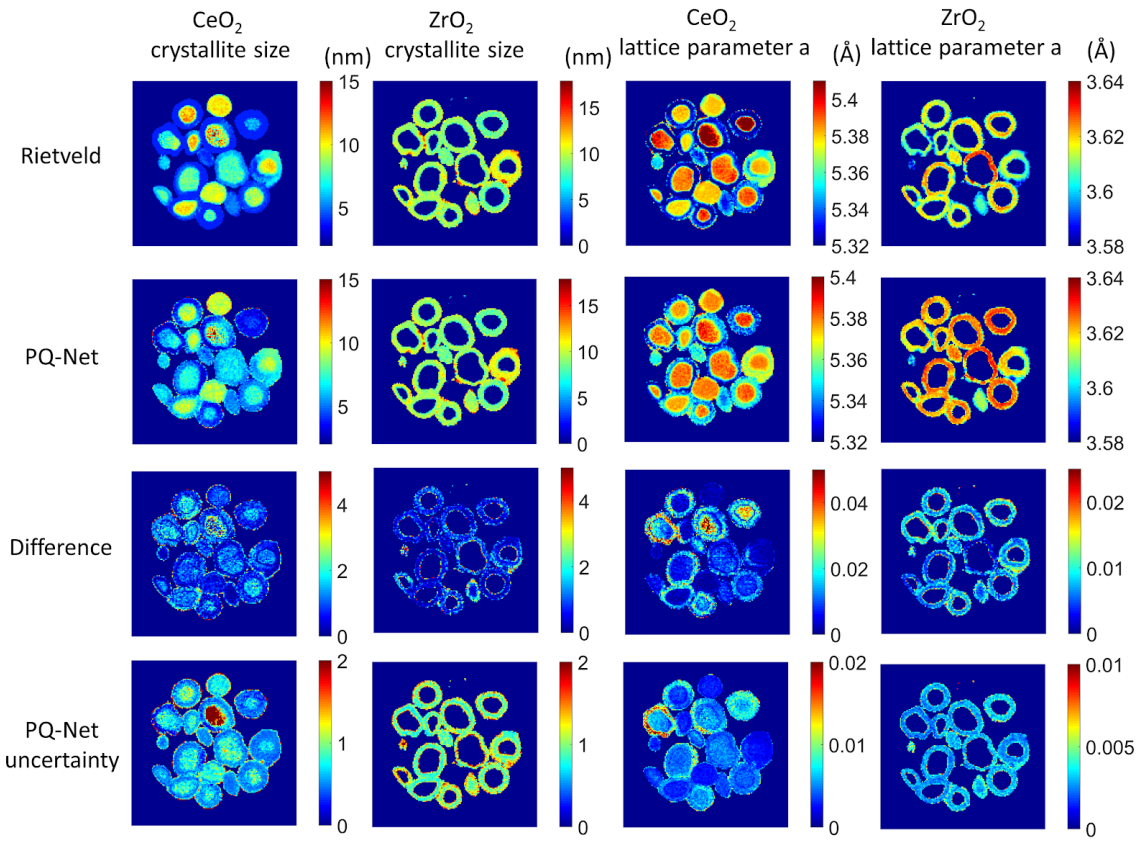

Figure 3.15: Crystallite size (columns 1 and 2 with colourbar axis corresponding to nm) and lattice parameter a (columns 3 and 4 with colourbar axis corresponding to Å) maps for CeO_2 and ZrO_2 obtained with the Rietveld method (top row), results obtained with the PQ-Net (second row), their absolute difference for the experimental multi-phase NiO-PdO-CeO_2-ZrO_2-Al_2O_3 system (third row) and the uncertainty maps of the deep ensemble PQ-Net (last row).

Diffraction patterns from two representative regions of interest were extracted from the XRD-CT for further analysis. As shown in Figure 3.16, one pattern is derived from the periphery of three particles (light blue region) while the second is derived from the whole area of two other particles (magenta region). The first region corresponds to an area where the ZrO_2 phase is present while the second is where it is absent. Both patterns are the mean XRD patterns from the respective regions in the sample. These two patterns were then analysed with the PQ-Net and the results are presented in Figure 3.16. It can be clearly seen that the PQ-Net is able to model accurately the data (R_{wp} of 8.353 and 7.749 % respectively).

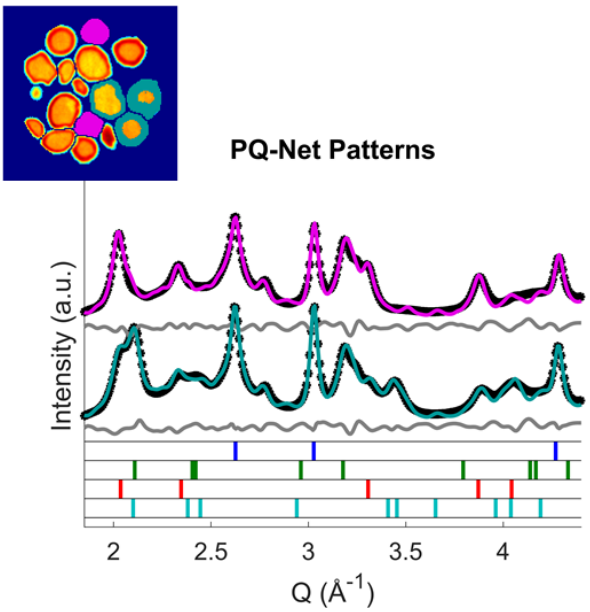

Figure 3.16: Comparison between the average diffraction patterns extracted from two regions of interest and their modeling with PQ-Net. The main reflections for NiO (blue ticks), PdO (green ticks), CeO_2 (red ticks) and ZrO_2 (cyan ticks) are also presented.

The multi-phase PQ-Net can not only predict the results precisely, but also do so efficiently. Figure 3.17 shows the time-taken difference between the multi-phase PQ-Net and the traditional Rietveld method. It shows the PQ-Net model and the PQ-Net with 10 deep-ensembled models only take 6 and 20 % of the time of the conventional Rietveld method respectively. It is worth mentioning that the time taken shown here includes the simulated library generation time and the model training time, and these procedures can be performed before the experiment. The prediction time normally takes just a few seconds for a PQ-Net model and at most a few minutes for the deep ensemble model, so the PQ-Net has a great potential to be applied to the *in situ/operando* experiments to observe the results during the experiments.

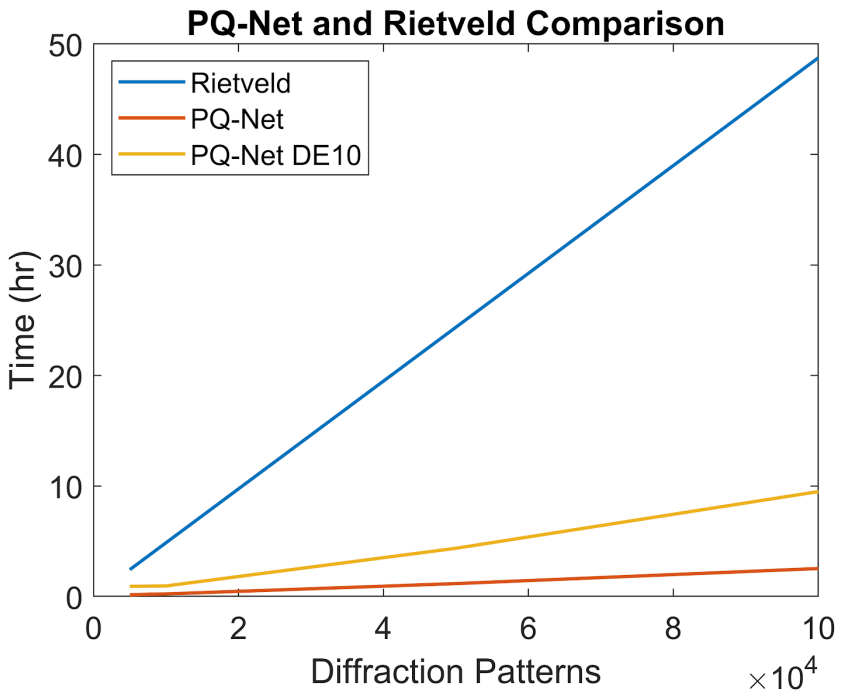

Figure 3.17: Performance comparison between the Rietveld method, single-model PQ-Net and PQ-Net Deep Ensemble with 10 models.

3.5 Extending PQ-Net: Retrieving Site Occupancy Factor (sof)

Within the chapter we have focused on the extraction of scale factors, lattice parameters and crystallite sizes which in the exemplar are sufficient to describe and understand the materials and process performance under operational conditions, and from our experience is that which is most commonly desired from XRD-CT and indeed the majority of multiphase *in situ* experiments. In principle, though PQ-Net can be extended to other parameters and to illustrate, we demonstrate its use for extracting occupancy where the architecture is illustrated in Figure 3.18 for a single-phase problem. There are only two sites in this system, and the sum of their occupancy factors is equal to 1, so there is only one site occupancy factor being predicted in the architecture to reduce the computational resources needed. 100,000 simulated NiPd patterns were used

to train the PQ-Net with a validation split rate of 10 %. And the parameter simulation ranges are set as:

- NiPd scale factor range: 0-4 × 10^{-9}
- NiPd lattice parameter range: 3.6 ± 0.2 Å
- NiPd crystallite size a range: 2-25 nm
- Ni site occupancy factor: 0-1

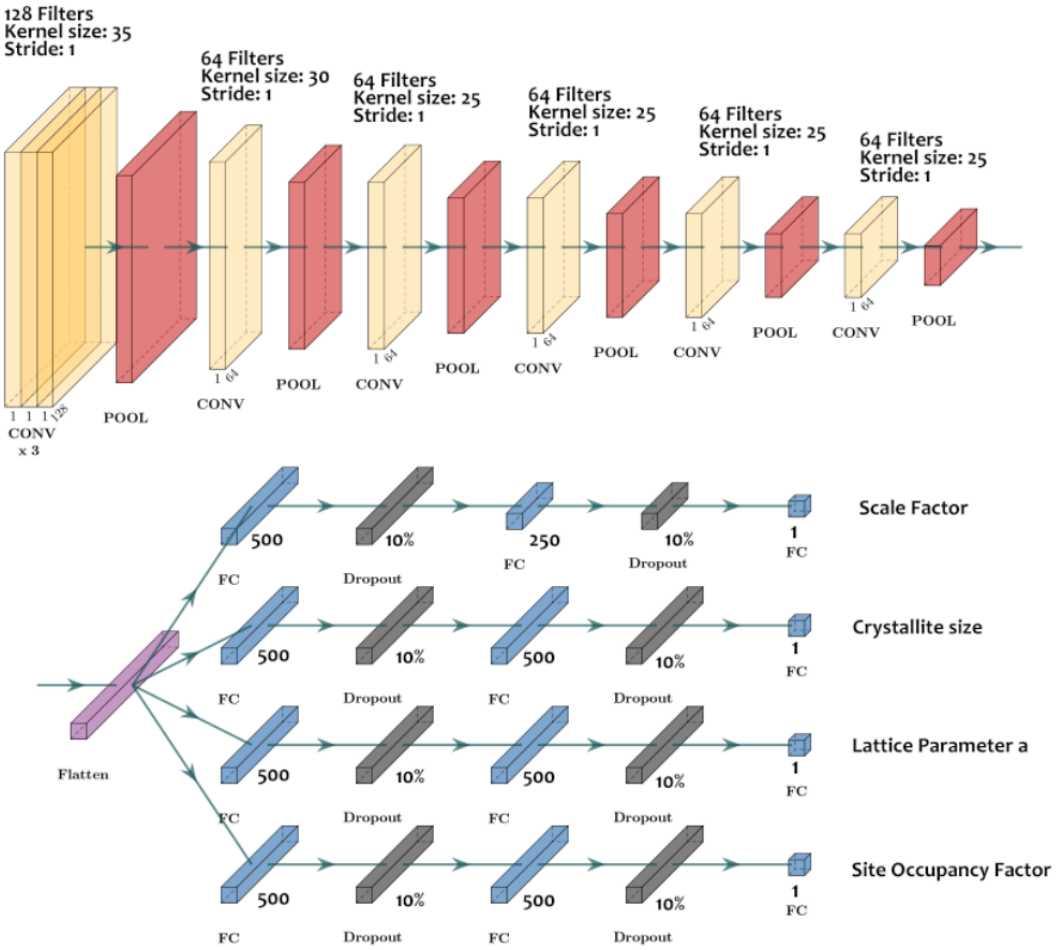

Figure 3.18: Single Phase PQ-Net architecture for NiPd crystal with an additional site occupancy factor. The uncertainty analysis of this architecture is shown in Table 3.7.

The NiPd lattice parameter range is given a much larger range than that introduced in the previous libraries to reflect real-world lattice parameter changes that would accompany the occupancy ranges introduced. As previously, all parameters are normalised to 0-100 based on their sampling

range while training. The errors before and after normalisation are shown in Table 3.7. The results validate that site occupancy can be extracted in this way.

Table 3.7: The accuracy of the single-phase PQ-Net model with site occupancy factor.

Parameter Name	Validation MAE (normalised)	MAE on original unit
Scale factor	1.62	6.48×10^{-9}
Crystallite size	2.54	0.0102 Å
Lattice parameter a	1.97	0.453 nm
site occupancy factor	2.46	0.0246

3.6 Summary and Conclusion

The results presented in this work demonstrate that the PQ-Net model is able to extract accurate physico-chemical information from XRD patterns. Its performance was evaluated with different datasets of increasing complexity, varying from simulated noiseless single-phase to experimental five-phase systems. In all cases, PQ-Net was able to quantify the scale factors, lattice parameters and crystallite sizes of the various phases providing predictions with errors within acceptable ranges (i.e. compared to the ground truth and Rietveld results for the simulated and experimental data respectively). The work presented here therefore serves also as an exemplar study demonstrating PQ-Net's flexibility and scalability as the method applied to datasets of different complexity and it can be scaled up or down depending on the data requirements. The robustness of the PQ-Net was further improved through the implementation of deep ensembles that allow for uncertainty quantification.

The optimised deep ensemble PQ-Net model was able to provide results with less than a 2 % difference in R_{wp} compared to the one obtained from the analysis of the same data using the Rietveld method. These results were obtained (i.e. analysis of ca. 50K XRD patterns) in ca. 10 s while the state-of-the-art Rietveld required ca. 4.4 h. It should be noted though that the

PQ-Net was not designed to replace the Rietveld method but to assist and provide a very fast and good approximation to the global minimum; conventional least-square minimisation techniques, such as the Rietveld method, can be used for fine parameter tuning using the PQ-Net results as the starting point. The PQ-Net in its current form, similar to the Rietveld method, requires some *a priori* knowledge of the sample and its chemistry. However, in case there is evolving/unpredictable chemistry or unknown phases form/are present in the system, it can be easily spotted through the R_{wp} and the difference between the experimental and predicted patterns.

The key advantage of the PQ-Net is that the diffraction libraries can be generated prior to an experiment and used to pre-train the model. This approach will allow for real-time assessment of diffraction data acquired during a dynamic experiment and enable the user to better guide the experiment (e.g. through the applied operating conditions) and intervene when necessary. In future, we plan to increase the complexity of the PQ-Net and explore its ability to model other parameters such as complex backgrounds, to quantify amorphous/crystalline components and to perform simultaneous calculation of weight percentages and strain analysis.

We would like to emphasise though that the benefit of the PQ-Net is really seen when one collects big datasets (i.e. fast acquisitions as we mention in the introduction). For example, the training of a PQ-Net can take tens of mins to a few hours depending on the library size but it is done only once. It is then able to yield predictions in seconds to a few minutes for 10,000 diffraction patterns. In contrast, even the fastest of the conventional full profile analysis methods using least-square minimisation approaches (employed in XRD analysis) take many hours per 10,000 datasets. So this is essentially a scale-up problem and that's where the deep learning approaches thrive. Another crucial point to make is that the trained PQ-Net can be used to fit multiple experimental data with varying chemistry. For example, one can train the PQ-Net with a dataset that contains both reduced and oxidised forms of metals and apply it to all

experimental datasets collected from an *in situ/operando* experiment where the material is exposed to varying chemical environments and changing forms (e.g. from reduced to oxidised and vice versa). In contrast, with the conventional approaches, one has to build and refine a new model every time; a model that is close to the results. Last but not least, I expect that the PQ-Net has the potential to become an essential tool for diffraction applications beyond synchrotron experiments such as real-time quality inspection in manufacturing/synthesis.

3.7 References

[1] D. Matras *et al.*, 'Emerging chemical heterogeneities in a commercial 18650 NCA Li-ion battery during early cycling revealed by synchrotron X-ray diffraction tomography', *J. Power Sources*, vol. 539, p. 231589, Aug. 2022, doi: 10.1016/j.jpowsour.2022.231589.

[2] G. B. M. Vaughan *et al.*, 'X-ray transfocators: focusing devices based on compound refractive lenses', *J. Synchrotron Radiat.*, vol. 18, no. 2, pp. 125–133, Mar. 2011, doi: 10.1107/S0909049510044365.

[3] C. G. Schroer and B. Lengeler, 'X-Ray Optics', in *Springer Handbook of Lasers and Optics*, F. Träger, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 1461–1474. doi: 10.1007/978-3-642-19409-2_22.

[4] T. Ebensperger, P. Stahlhut, F. Nachtrab, S. Zabler, and R. Hanke, 'Comparison of different sources for laboratory X-ray microscopy', *J. Instrum.*, vol. 7, no. 10, pp. C10008–C10008, Oct. 2012, doi: 10.1088/1748-0221/7/10/c10008.

[5] A. J. Morgan *et al.*, 'High numerical aperture multilayer Laue lenses', *Sci. Rep.*, vol. 5, no. 1, p. 9892, Jun. 2015, doi: 10.1038/srep09892.

[6] K. T. Murray *et al.*, 'Multilayer Laue lenses at high X-ray energies: performance and applications', *Opt Express*, vol. 27, no. 5, pp. 7120–7138, Mar. 2019, doi: 10.1364/OE.27.007120.

[7] Benjamin Hornberger, Jack Kasahara, Martin Gifford, Ronald Ruth, and Rod Loewen, 'A compact light source providing high-flux, quasi-monochromatic, tunable X-rays in the laboratory', Sep. 2019, vol. 11110. doi: 10.1117/12.2527356.

[8] C. Ponchut, N. Tartoni, and D. Pennicard, 'X-ray imaging at synchrotron research facilities', *Radiat. Meas.*, vol. 140, p. 106459, 2021, doi: https://doi.org/10.1016/j.radmeas.2020.106459.

[9] M. Xia *et al.*, 'Lab-Scale In Situ X-Ray Diffraction Technique for Different Battery Systems: Designs, Applications, and Perspectives', *Small Methods*, vol. 3, no. 7, p. 1900119, Jul. 2019, doi: 10.1002/smtd.201900119.

[10] F. H. Gjørup, J. V. Ahlburg, and M. Christensen, 'Laboratory setup for rapid in situ powder X-ray diffraction elucidating Ni particle formation in supercritical methanol', *Rev. Sci. Instrum.*, vol. 90, no. 7, p. 073902, Jul. 2019, doi: 10.1063/1.5089592.

[11] A. M. Beale, S. D. M. Jacques, E. K. Gibson, and M. Di Michiel, ‘Progress towards five dimensional diffraction imaging of functional materials under process conditions’, *Chem. Struct. Using Synchrotron Radiat.*, vol. 277–278, pp. 208–223, Oct. 2014, doi: 10.1016/j.ccr.2014.05.008.
[12] H. F. Poulsen and G. B. M. Vaughan, ‘Multigrain crystallography and three-dimensional grain mapping’, in *International Tables for Crystallography*, American Cancer Society, 2019, pp. 601–616. doi: 10.1107/97809553602060000970.
[13] A. Vamvakeros *et al.*, ‘5D operando tomographic diffraction imaging of a catalyst bed’, *Nat. Commun.*, vol. 9, no. 1, pp. 1–11, 2018.
[14] G. Pacchioni, ‘An upgrade to a bright future’, *Nat. Rev. Phys.*, vol. 1, no. 2, pp. 100–101, Feb. 2019, doi: 10.1038/s42254-019-0019-5.
[15] C. Wang, U. Steiner, and A. Sepe, ‘Synchrotron Big Data Science’, *Small*, vol. 14, no. 46, p. 1802291, Nov. 2018, doi: 10.1002/smll.201802291.
[16] F. Oviedo *et al.*, ‘Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks’, *Nat. Commun.*, vol. 5, no. 1, pp. 2057–3960, May 2019, doi: 10.1038/s41524-019-0196-x.
[17] V. Stanev, V. V. Vesselinov, A. G. Kusne, G. Antoszewski, I. Takeuchi, and B. S. Alexandrov, ‘Unsupervised phase mapping of X-ray diffraction data by nonnegative matrix factorization integrated with custom clustering’, *Npj Comput Mater*, vol. 4, p. 43, 2018.
[18] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, ‘Deep learning for visual understanding: A review’, *Neurocomputing*, vol. 187, pp. 27–48, 2016.
[19] B. Wang, K. Yager, D. Yu, and M. Hoai, ‘X-ray scattering image classification using deep learning’, in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 697–704.
[20] D. M. Pelt and J. A. Sethian, ‘A mixed-scale dense convolutional neural network for image analysis’, *Proc. Natl. Acad. Sci.*, vol. 115, no. 2, pp. 254–259, 2018.
[21] X. Liu *et al.*, ‘A comparison of deep learning performance against health-care professionals in detecting diseases from medical imaging: a systematic review and meta-analysis’, *Lancet Digit. Health*, vol. 1, no. 6, pp. e271–e297, 2019.
[22] J.-W. Lee, W. B. Park, J. H. Lee, S. P. Singh, and K.-S. Sohn, ‘A deep-learning technique for phase identification in multiphase inorganic compounds using synthetic XRD powder patterns’, *Nat. Commun.*, vol. 11, no. 1, pp. 1–11, 2020.
[23] S. Muto and M. Shiga, ‘Application of machine learning techniques to electron microscopic/spectroscopic image data analysis’, *Microscopy*, vol. 69, no. 2, pp. 110–122, 2020.
[24] G. R. Schleder, A. C. Padilha, C. M. Acosta, M. Costa, and A. Fazzio, ‘From DFT to machine learning: recent approaches to materials science–a review’, *J. Phys. Mater.*, vol. 2, no. 3, p. 032001, 2019.
[25] W. B. Park *et al.*, ‘Classification of crystal structure using a convolutional neural network’, *IUCrJ*, vol. 4, no. 4, pp. 486–494, 2017.
[26] K. Ryan, J. Lengyel, and M. Shatruk, ‘Crystal Structure Prediction via Deep

Learning', *J. Am. Chem. Soc.*, vol. 140, no. 32, pp. 10158–10168, Aug. 2018, doi: 10.1021/jacs.8b03913.
[27] A. Ziletti, D. Kumar, M. Scheffler, and L. M. Ghiringhelli, 'Insightful classification of crystal structures using deep learning', *Nat. Commun.*, vol. 9, no. 1, p. 2775, Jul. 2018, doi: 10.1038/s41467-018-05169-6.
[28] F. Oviedo *et al.*, 'Fast and interpretable classification of small X-ray diffraction datasets using data augmentation and deep neural networks', *Npj Comput. Mater.*, vol. 5, no. 1, p. 60, May 2019, doi: 10.1038/s41524-019-0196-x.
[29] C.-H. Liu, Y. Tao, D. Hsu, Q. Du, and S. J. L. Billinge, 'Using a machine learning approach to determine the space group of a structure from the atomic pair distribution function', *Acta Crystallogr. Sect. A*, vol. 75, no. 4, pp. 633–643, Jul. 2019, doi: 10.1107/S2053273319005606.
[30] H. Wang *et al.*, 'Rapid Identification of X-ray Diffraction Patterns Based on Very Limited Data by Interpretable Convolutional Neural Networks', *J. Chem. Inf. Model.*, vol. 60, no. 4, pp. 2004–2011, 2020.
[31] M. Tatlier, 'Artificial neural network methods for the prediction of framework crystal structures of zeolites from XRD data', *Neural Comput. Appl.*, vol. 20, no. 3, pp. 365–371, 2011.
[32] J. K. Bunn, J. Hu, and J. R. Hattrick-Simpers, 'Semi-Supervised approach to phase identification from combinatorial sample diffraction patterns', *Jom*, vol. 68, no. 8, pp. 2116–2125, 2016.
[33] J. K. Bunn, S. Han, Y. Zhang, Y. Tong, J. Hu, and J. R. Hattrick-Simpers, 'Generalized machine learning technique for automatic phase attribution in time variant high-throughput experimental studies', *J. Mater. Res.*, vol. 30, no. 7, p. 879, 2015.
[34] G. Barr, W. Dong, and C. J. Gilmore, 'High-throughput powder diffraction. II. Applications of clustering methods and multivariate data analysis', *J. Appl. Crystallogr.*, vol. 37, no. 2, pp. 243–252, 2004.
[35] S. M. Obeidat, I. Al-Momani, A. Haddad, and M. Bani Yasein, 'Combination of ICP-OES, XRF and XRD techniques for analysis of several dental ceramics and their identification using chemometrics', *Spectroscopy*, vol. 26, no. 2, pp. 141–149, 2011.
[36] Z. P. Chen *et al.*, 'Enhancing the signal-to-noise ratio of X-ray diffraction profiles by smoothed principal component analysis', *Anal. Chem.*, vol. 77, no. 20, pp. 6563–6570, 2005.
[37] D. Lee, H. Lee, C.-H. Jun, and C. H. Chang, 'A variable selection procedure for X-ray diffraction phase analysis', *Appl. Spectrosc.*, vol. 61, no. 12, pp. 1398–1403, 2007.
[38] H. Lee, J. Lee, H. Kim, B. Cho, and S. Cho, 'Deep-Neural-Network-Based Sinogram Synthesis for Sparse-View CT Image Reconstruction', *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 3, no. 2, pp. 109–119, Mar. 2019, doi: 10.1109/TRPMS.2018.2867611.
[39] J. A. Aguiar, M. L. Gong, and T. Tasdizen, 'Crystallographic prediction from diffraction and chemistry data for higher throughput classification using machine learning', *Comput Mater Sci*, vol. 173, p. 109409, 2020.
[40] X. Li and X. Wu, 'Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition', *2015*

IEEE Int. Conf. Acoust. Speech Signal Process. ICASSP, pp. 4520–4524, 2015.
[41] I. Martens *et al.*, 'Holistic Multi-scale Imaging of Oxygen Reduction Reaction Catalyst Degradation in Operational Fuel Cells', *ArXiv Prepr. ArXiv200804770*, 2020.
[42] S. W. Price, K. Geraki, K. Ignatyev, P. T. Witte, A. M. Beale, and J. F. W. Mosselmans, 'In situ microfocus chemical computed tomography of the composition of a single catalyst particle during hydrogenation of nitrobenzene in the liquid phase', *Angew. Chem. Int. Ed.*, vol. 54, no. 34, pp. 9886–9889, 2015.
[43] A. Coelho, 'TOPAS and TOPAS-Academic: an optimization program integrating computer algebra and crystallographic objects written in C++', *J. Appl. Crystallogr.*, vol. 51, no. 1, pp. 210–218, Feb. 2018.
[44] A. Vamvakeros *et al.*, 'Removing multiple outliers and single-crystal artefacts from X-ray diffraction computed tomography data', *J. Appl. Crystallogr.*, vol. 48, pp. 1943–1955, 2015, doi: 10.1107/S1600576715020701.
[45] D. P. Kingma and J. Ba, *Adam: A Method for Stochastic Optimization*. 2017.
[46] A. Vamvakeros *et al.*, 'Real-time multi-length scale chemical tomography of fixed bed reactors during the oxidative coupling of methane reaction', *J. Catal.*, vol. 386, pp. 39–52, 2020, doi: https://doi.org/10.1016/j.jcat.2020.03.027.
[47] A. Vamvakeros *et al.*, 'DLSR: a solution to the parallax artefact in X-ray diffraction computed tomography data', *J. Appl. Crystallogr.*, vol. 53, no. 6, Dec. 2020, doi: 10.1107/S1600576720013576.
[48] L. Bottou, 'Large-scale machine learning with stochastic gradient descent', in *Proceedings of COMPSTAT'2010*, Springer, 2010, pp. 177–186.
[49] N. Qian, 'On the momentum term in gradient descent learning algorithms', *Neural Netw.*, vol. 12, no. 1, pp. 145–151, 1999.
[50] A. Lydia and S. Francis, 'Adagrad—an optimizer for stochastic gradient descent', *Int J Inf Comput Sci*, vol. 6, no. 5, pp. 566–568, 2019.
[51] S. Ruder, 'An overview of gradient descent optimization algorithms', *ArXiv Prepr. ArXiv160904747*, 2016.
[52] L. Kovarik *et al.*, 'Unraveling the Origin of Structural Disorder in High Temperature Transition Al2O3: Structure of θ-Al2O3', *Chem. Mater.*, vol. 27, no. 20, pp. 7042–7049, Oct. 2015, doi: 10.1021/acs.chemmater.5b02523.
[53] B. Lakshminarayanan, A. Pritzel, and C. Blundell, *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles*. 2017.
[54] H.-I. Suk, S.-W. Lee, and D. Shen, 'Deep ensemble learning of sparse regression models for brain disease diagnosis', *Med. Image Anal.*, vol. 37, pp. 101–113, Apr. 2017, doi: 10.1016/j.media.2017.01.008.
[55] N. Kozlovskaia and A. Zaytsev, 'Deep ensembles for imbalanced classification', in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2017, pp. 908–913.
[56] S. Fort, H. Hu, and B. Lakshminarayanan, 'Deep ensembles: A loss landscape perspective', *ArXiv Prepr. ArXiv191202757*, 2019.

Chapter 4: Deep-Learning-Based Tomographic Image Reconstruction

4.1 Abstract

In recent years, machine-learning methods have offered the promise of transforming image reconstruction across the life and physical sciences. In this chapter, it will show how both supervised learning and self-supervised learning ways to reconstruct computed tomographic data. It will also explore the possibility of using machine-learning methods to solve the common artefacts in computed tomographic (CT) image reconstruction. It will demonstrate the effect of and present a method for avoiding, training set bias in deep neural networks for tomographic image reconstruction. This is done by firstly, trying to remove the angular undersampling artefacts with convolutional neural networks (CNNs) trained by the DIV2K dataset, however, the image-to-image cleaning is problematic since minor features are difficult to retain. Therefore, a new, lightweight and scalable artificial neural network architecture, SingleDigit2Image (SD2I), to reconstruct tomographic images directly from their sinogram domain was developed. The SD2I is a self-supervised learning approach leading to more robust results and better transferability among different experimental data frames. Because of the unique design of our generator network, this new method can reconstruct images that are more than 2k x 2k large. The network's performance on both simulated and experimental synchrotron tomographic images of X-ray diffraction CT (XRD-CT), micro-CT and neutron-CT was then tested. It is shown that the method can correctly reconstruct images with fewer projection data, and the results are more accurate than conventional reconstruction methods such as Filtered Back Projection (FBP), Simultaneous Algebraic Reconstruction Technique (SART), Simultaneous Iterative Reconstruction Techniques (SIRT), and Conjugate Gradient Least Squares (CGLS). Note that the SD2I architecture has been published in Digital Discovery as ‘A scalable neural network architecture for self-supervised tomographic image reconstruction’ on June 2023 [1]:

4.2 Introduction

4.2.1 CT Image artefacts

In the last few decades, computed tomography (CT) imaging has become increasingly popular in measuring and sampling in a non-destructive way. Many physical measurements, however, will introduce noise in CT image reconstruction, such as the high background radiation noise and software/hardware [2]–[4]. Although the noise can come from many different sources, the final cumulative noise can be accurately characterised by a Poisson distribution, because the noise on each pixel is independent [2]. Depending on the CT method, the Gaussian noise or an additive Gaussian noise can also be found in CT images [5]–[7]. Spatial filters applied directly to noisy images are traditionally used for reducing Poisson or Gaussian noise. A spatial filter, such as a mean filter or median filter, is based on the assumption that the noise occurs at a higher frequency, but it inevitably introduces blurring into the images [8]. There are several standard performance metrics used to assess the accuracy of CT image denoising algorithms, including mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), structural similarity (SSIM) and peak signal-to-noise ratio (PSNR) [9]–[11].

The angular undersampling artefact can occur when reconstructing CT images with insufficient sparsely-selected projections [12]. The primary purpose of undersampling is to reduce the size of the dataset which in turn reduces the processing time. Similar to the artefact created by reconstruction with an incomplete sinogram (a number of projections are missing), traditional methods to solve this problem focus on inpainting the missing angles/projections with interpolations [13]–[17], however, they all effect some sort of blurring on the images. Recently, deep neural network (DNN) methods are becoming more popular with supervised-learning methods shown to be very promising in completing sinograms without introducing other artefacts [18]–[20].

Metal artefacts or high Z-containing materials are another significant issue in reconstructed CT images [21]. Methods to correct metal artefacts have been researched and developed for nearly four decades [22]. Metal objects in the CT scanning area will weaken the transmitted X-ray signal or even block the X-ray from penetrating through the sample. This leads to a false projection being recorded on the sinogram resulting in a reconstructed image that will have bright or dark streaks which are called metal artefacts [22], [23]. A very common method to correct the metal artefact is to manually remove the area influenced by metal objects on the sinogram domain, and then apply sinogram inpainting methods to complete the missing data [16], [24]–[26]. More recently it has been shown that traditional supervised learning image-domain cleaning methods can also help [27].

4.2.2 Supervised-learning-based CT image reconstruction

During the past few years, machine-learning methods have been applied as a powerful new tool for removing the artefacts in CT images. A number of notable exceptions do exist, where supervised learning and generative models have been used to automatically map from sinogram to real space images [28]–[34]. Supervised learning has already been applied to all procedures in the conventional image reconstruction pipeline [35]. Many papers have already demonstrated that convolutional neural networks (CNNs) can carry out image reconstruction directly. But they normally require a large number of computing resources because they require a domain transfer [35]–[37]. For example, a stable AUTOMAP CT reconstruction network for images with sizes of 512x512 requires over 340 Gigabytes (GBs) of graphic processing unit (GPU) memory which is way beyond the capacity of modern GPUs of only tens of GBs of memory [28]. Therefore, using CNN to perform a pretreatment on sinograms [18], [19], [38], [39] or else a posttreatment on reconstructed images [40]–[42] is a more common method used to remove a certain artefact.

4.2.2.1 CT Image Reconstruction by DNN

Although deep neural networks (DNNs) have emerged as a powerful and promising new tool for CT image reconstruction, bottlenecks still exist to their application for image reconstruction due to their scalability (i.e. their ability to handle large images), their network size (large networks can be computationally very expensive) and particularly for applications where absolute values (as opposed to normalised values) are important in the reconstructed image, such as in chemical tomography and in quantitative analysis of attenuation-based tomography data [43]–[45]. For example, the AUTOMAP published in 2018 by Zhu et al and shown in Figure 4.1, is a CNN architecture that was specifically developed for tomographic reconstruction [28]. This architecture has one flatten, one fully connected layer with 2 x m x n nodes, two fully connected layers with n^2 nodes and several convolutional layers before the output of the reconstructed image is created (here m and n represent the number of projections and image size, respectively). The authors tested this architecture with Radon projection imaging [28], conjugate-gradient sensitivity encoding reconstruction [46], and more datasets with non-uniform Fourier transformation [47]. Although these works showed that AUTOMAP can yield very respectable results, it has been remarked elsewhere that the architecture has some shortcomings [36], [37]. For example, the first two fully connected layers in AUTOMAP can produce $2\times(m\times n)^2$ trainable weights [37], where m and n represent the two dimensions of sinograms. If we consider a 2D input sinogram with 128 x 512 pixels and the required reconstructed image has 512 x 512 pixels then AUTOMAP needs over 340 Gigabytes to save all weights, which is currently impossible for modern GPUs [36]. The original article only attempted to reconstruct images up to 128 x 128 pixels, which is not big enough for modern CT images.

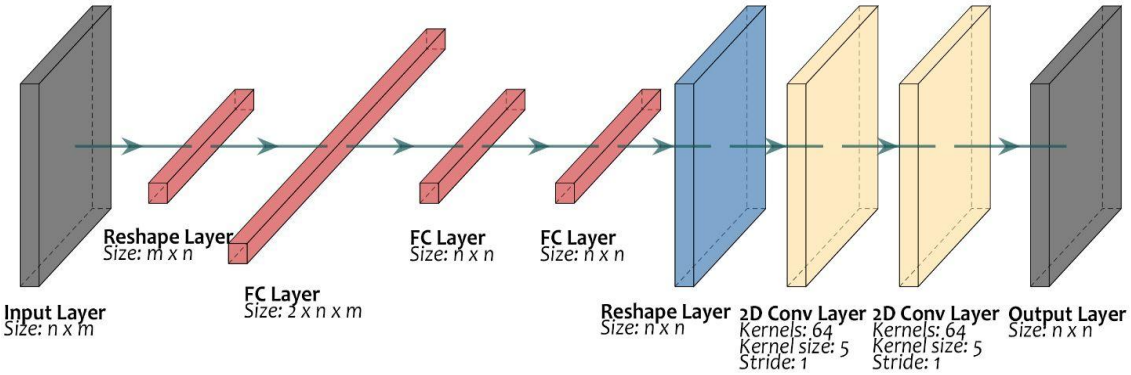

Figure 4.1: The AUTOMAP CNN architecture [28]. It starts with a 2D input layer which has the same size as the sinograms n x m, here n denotes the number of X-ray sensor pixels and m denotes the number of signals at different projection angles taken from the object. Then the 2D input sinogram is reshaped into a 1D array and sent to a fully connected (FC) layer with 2 x n x m nodes inside. It follows with two more fully connected layers n^2 nodes, and the 1D output of the final fully connected layer will be reshaped to a 2D array and sent to two convolutional layers to get the output layer or reconstructed image.

4.2.2.2 Artefact removal on image domain

Another very popular way to solve the artefact problem in CT images is to directly clean the reconstructed image. During the past few years, CNNs have become increasingly popular for image denoising [48]. While the widely used CNN models like the AlexNet [49], VGG [50], and GoogleNet [51] were applied to computer vision and image classification, image-denoising CNNs have more recently been applied to scale up the images with more sharp details on the higher resolution domain which is called image 'super-resolution'. Since there is no domain transfer involved in this way, the models are generally CNN-based and have fewer parameters than the methods that do a domain transfer. The residual learning networks have been proven to have a good performance in CT image denoising and artefact removal [52], [53]. Lee et al. used U-Net to obtain good-quality images from sparse-view medical CT images [12]. The U-Net architecture was first developed for biomedical image segmentation problems [54], [55]. Formally, a U-Net architecture contains one down-scaling part and one up-scaling part. In the up-scaling section, they do not only take the output from lower levels as input, but they also acquire output images from the same level in the down-scaling part. As a result, each level in the up-scaling part has two inputs containing the features captured in the previous level and in down-scaling convolutional layers, which makes the model more robust.

Compared to traditional CNN architectures, the U-Net can yield more precise segmentation with fewer training datasets [56], [57]. According to the original paper, segmentation of a 512 x 512 image takes less than a second on a modern GPU [54]. Different from the original U-Net architecture, Lee et al. used an overall residual learning between the input and output layers, so that U-Net can learn the artefacts caused by the sparse selected angular undersampling directly. They concluded that they can use only 20 % of X-ray projections to form a good CT image, and the results are better than combined interpolation methods [38], [58], [59]. An encoder-decoder CNN architecture proposed by Chen et al. also shows a good performance on Filtered back projection (FBP) de-noising problems [60]. In this work, they used three residual learning connections between the encoder and the decoder. As with the U-Net described above, they also claimed the necessity of residual learning to enhance the resulting quality. In 2020, Liu et al. published TomoGAN which also used a generative adversarial network (GAN) to train a U-Net that can remove the angular-undersampling artefact on the image domain [61]. In this approach, the GAN is used as a special supervised CNN training method because the generator will be used alone with a real experimental dataset after the generator is well-trained. Jelmer et al. also used a GAN in training their image denoising CNN [62]. The input to the CNN was low-dose CT images and the output was routine-dose CT images. They compared the results with and without GAN loss, and they concluded that the GAN-like architecture did not improve the results. In 2021, Sidorenko's team applied the Residual Encoder-Decoder Network (RED-Net) in micro-CT image noise removal [63]. They found that by applying the SSIM loss [9] and the perceptual loss [64], the network yields better image quality with all metrics than the traditional mean squared error (MSE) loss.

4.2.2.3 Sinogram cleaning and inpainting networks

Performing sinogram cleaning and inpainting before applying conventional methods like FBP to reconstruct images are very popular approaches to improve reconstruction quality. In 2014, Li et al. used dictionary learning to inpaint the sparse sampled Shepp-Logan sinogram into a complete sinogram

[14]. In 2013, Pelt and Batenburg developed the NN-FBP and used the DNN to learn the filters for FBP [65]. They claimed that the NN-FBP could yield more accurate results than both FBP and SIRT methods on simulated data. The same approach was subsequently applied to experimental electron tomographic data with only a few projections [66], The results showed that the angular-undersampling artefact is clearly improved using NN-FBP. Ding et al. demonstrated a CNN can inpaint the missing parts of a sinogram with a limited angle range rather than sparse-selected sinograms [67]. The CNN is trained by a GAN combined with the MSE loss between the generated sinogram and the ground truth. In 2019, Li et al. used a similar approach to perform sinogram inpainting with a U-Net [18], and they also illustrated the sinogram-inpainting-generative adversarial network (SI-GAN) which combined the image domain MAE loss in their joint loss function [68]. Although the image domain loss slows the training down because an FBP is used at each training step, the final result is an improved image. To use the U-Net as the generator, they leave the missing angles' data blank in the input to keep the dimension of the sinogram. Work carried out by Dong et al. also used a similar U-Net architecture [69] although instead of inputting the sinograms with blank on the missing angles, they applied a Radon transformation on the FBP images of the under-sampled sinograms to obtain new sinograms with the same size as the original sinograms before under-sampling. In this way, they could keep the input and output sizes of the U-Net the same without any mathematical interpolation methods.

In 2022, Xie et al. combined the sinogram inpainting and image domain artefact removal networks into a single large network called SIAR-GAN [20]. In this work, networks are trained together in two local GAN-like training loops which combine the supervised learning and unsupervised learning approaches to handle the insufficient training data. In addition, they used a new discriminator that not only outputs a binary true/false result based on the overall similarity between two images but also outputs a 59 × 19 image block. Each pixel in the block is used to calculate the individual discriminator loss in the corresponding

block of the original image. By applying the new regional discriminator, the resultant images showed higher resolution and greater details. The most recent work published by Liu et al. illustrated a new CNN-based method that combined residual learning and multi-head attention that can precisely recover the original sinogram with only 20 % of randomly sampled projection data [70]. They claimed that the trained network can also be used to reconstruct the full image in batches. However, they only showed the simulated images with different regular shapes and as such it will need some transfer learning before applying the trained CNN to real experimental data.

4.2.3 Self-supervised learning approaches

While powerful supervised learning approaches have been developed over a relatively short period of time, most of them require many matched pairs of noisy and clean images similar to the experimental data, which is not always applicable in real life. This is particularly so when it comes to solving the denoising problem where very clean images are hard to obtain due to instrument or dose limitations. To overcome this problem, self-supervised learning has become more popular recently in solving the problem of artefacts in CT images because it is more controllable, robust, and requires no pre-training [71]–[73]. Although self-supervised learning methods like the GAN and self-attention CNN focus on the experimental data itself, there are still very few networks doing the sinogram-to-image reconstruction directly because of its difficult nature.

4.2.3.1 Self-supervised CT image enhancement

As previously discussed for the supervised learning methods, deep image denoising normally uses advanced neural network architectures such as ResNet [74], U-Net [54], and their variants [75]–[77]. In 2018, Lehtinen et al. published the Noise2Noise network which is trained by image pairs of a noisy image and many noisier images by adding artificial Poisson noise [78]. The premise here is that CNN is not able to learn the noise itself because the noise does not have any features. On this basis, the nearest image that the CNN can

determine, and close to all images with artificial noise is the unknown clean image. This method can clean data with different noise distributions (Poisson, Gaussian, and Bernoulli), and even Monte Carlo synthesis noise. The Self2Self architecture also tried to generate multiple different image pairs from the same noisy image using Bernoulli sampling to randomly drop out pixels from one image [79].

While the Noise2Noise and Self2Self do not need any clean images for training, it still needs a lot of training data to maintain the stability and robustness of the trained CNN, which means they are unable to denoise one image at a time. Krull et al. moved a step forward by developing a Noise2Void architecture that requires no image pairs at all. They used the self-supervised learning approach to train a blind-spot network which is trying to predict the central pixel by looking at the surrounding pixels in an 11 x 11 square [80]. They claimed that the Noise2Void has a wider range of uses and requires fewer training resources. However, their results are not necessarily better than the Noise2Noise's results in terms of the peak signal-to-noise ratio (PSNR). Subsequently, in 2019, Laine et al. proposed an architecture that used four blind-spot networks in which receptive fields are restricted to different directions. By taking the average of the four outputs, the results are determined to be more stable than those obtained using Noise2Void. Moreover, Batson and Royer introduced the Noise2Self which introduced the blind-spot self-supervised loss into both machine-learning and traditional methods to find the most optimal result. With their method, the CNN is capable of denoising one image [81]. Hendriksen et al. also determined that the training pair do not have to be different images [82]. They proposed the Noise2Inverse architecture that is specifically designed for denoising CT images. Noise2Inverse uses image pairs of the FBP images reconstructed with a quarter of sparsely selected projections and the FBP image reconstructed by the rest of the sinograms. This method can create a large training dataset without any artificially added noise and keeps all information in the training process. The most recent paper published in 2022 introduced the Noise2Fast architecture which also uses the same phenomenon, but the method can be

applied to all kinds of images [83]. They used chequerboard downsampling to split one image into two sub-images, and each of them contains half of the original image's information. They claimed that the Noise2Fast performs very well in terms of both speed and accuracy. It is worth mentioning that the method is based on the assumption that there is much redundant information in the image, and major features occupy more than one pixel in both dimensions.

In 2017, Johnson et al. first proposed perceptual loss which used the pre-trained VGG feature maps as the loss function in comparing the similarity between two images [84], [85]. This method has then been applied to the self-supervised CT image denoising by Li et al. [86]. The self-attention convolutional neural network (SACNN) combined the perceptual loss with the GAN-like architecture to denoise the low-dose CT image. During training, an encoder-decoder type of CNN uses the same noisy image as both the training input and target. They claimed that the new loss function can yield more accurate images in terms of the SSIM and PSNR than the traditional MSE loss function, and keeps more fine structural objects. In 2021, Zhang et al. proposed the SADIR-Net which tried to denoise the image in the image domain, feature domain, and sinogram domain simultaneously [87]. The architecture pursues the three different routes and then combines later to form a denoised image. After three such learning blocks, a joint loss function with MSE and SSIM is then used to compare the third block's output and the input noisy image. Kim et al. proposed a method that denoises low-dose CT images on both image and sinogram domains [88]. By using Poisson Unbiased Risk Estimator (PURE) loss to denoise sinograms and then comparing the results with those determined in the image domain, very convincing results in terms of best RMSE among all compared methods in the paper were obtained.

CycleGAN is another popular unsupervised learning method to translate a CT image from an experimental noisy image domain to a target clean image domain without a paired dataset [89]. A CycleGAN normally needs two generators which use each other's output as the input image. Harms et al. used

the paired CycleGAN generators to correct the artefacts in cone-beam CT images and map the cone-beam CT image to traditional CT images [90]. Gu and Ye used the CycleGAN to denoise the low-dose CT image, by introducing adaptive instance normalisation (AdaIN) layers in the generator [91], so the one generator can be used for both low-dose CT image to a routine-dose CT image and high-dose to low-dose images. In this way, the CycleGAN can be trained twice on each loop, thereby reducing the total number of parameters. The CycleGAN-like architecture can also be used to remove metal artefacts [92]. They applied the convolutional block attention module (CBAM) [93] in the generators, which can effectively force them to focus on the metal artefacts. Meanwhile Yang et al. applied the CycleGAN on magnetic resonance (MR) to CT image transformation [94], [95]. They introduced the structure-consistency loss to compare the structural consistency between the MR and CT images during training. They claimed that the new structure-consistency loss can effectively improve the CT image. In 2022, Sun et al. also used CycleGAN to perform an MR-to-CT transformation. They used a very special U-Net discriminator in the CycleGAN to perform pixel-wise classifications [96]. Compared to the traditional-GAN-based MR-to-CT transformation method [97], CycleGAN's results are generally more stable and accurate.

The most recent self-supervised image-denoising methods have been highlighted above and from this it can be concluded that most of them can be applied to the sinogram domain because CNN can treat sinograms as images. However, there are very few works to modify the sinogram directly with a self-supervised learning method, because that may introduce other artefacts or blur into the reconstructed image [98]. In 2021, Yu et al. proposed an architecture to remove the metal artefact [99] by training two networks in one loop; one for the sinogram completion network, and the other one is the general CT refinement output to enhance the FBP result afterwards. To train the sinogram completion network, they used a joint loss function of the MAE on the sinogram domain and the MAE on the FBP images. Niu et al. also proposed a dual-domain network for low-dose CT image denoising [100]. Dual-domain

methods can generally generate images with more minor features because they keep full information on both domains.

4.2.3.2 Self-supervised CT sinogram-to-image reconstruction

Reconstructing images with self-supervised learning generally requires lighter generator architectures than the supervised learning method. In 2019, Zhou et al. developed a Tomo-GAN architecture which can reconstruct clean CT images from a limited number of projections [101] and which consists of two parts: The CycleGAN-based Sino-Net continues to generate simulated sinograms with limited angular input whereas the Recon-GAN takes the simulated sinogram and its FBP as input, then outputs the reconstructed image. A Long-Short-Term-Memory (LSTM) layer is used in the Recon-GAN generator to learn the features of projections from different angles. This architecture used six loss functions in total and the two parts can be trained separately. On the sinogram with a limited angle of 120°, Tomo-GAN can generate better images than all traditional methods tested in the paper. However, Tomo-GAN is a very heavy architecture and in order to achieve the results shown in the paper, 5935 real experimental sinograms and 232,228 simulated sinograms were used, which took about 48 h to finish the training.

In 2019, Yang et al. proposed the GANrec architecture which comprises a CNN-based generator that receives back projection images as inputs and outputs the generated artefact-free images. A Radon transformation is used afterwards to convert the generated image back to sinograms, and the MSE is used to compare the generated sinogram and input sinogram. This architecture is very simple and powerful, but it only works for normalized images. It means that the information on intensities will be lost with this method. Subsequently, Zang et al. developed the IntraTomo architecture which can reconstruct images with retention of the intensity information [102]. They used a generator with Fourier feature projection layers and multiple fully connected layers, in addition to a geometry refinement module, to improve the reconstruction in the image domain. This method can also be applied to reconstructing limited-angle tomographic images. Unal et al. used a similar architecture to reconstruct

low-dose CT images [103]. They combined the GAN architecture with the Noise2Self architecture to achieve a one-goal reconstruction method with image denoising. Based on the above precis of the current state of the literature, we see great potential for self-supervised learning methods for solving multiple image artefacts together. However, as outlined below there are some pitfalls that need careful consideration when the self-supervised learning methods are applied.

4.2.4 Identifying and avoiding training set bias in the supervised learning approach

In recent years neural networks have offered the promise of transforming image reconstruction across the life and physical sciences, however, there are still many concerns to be addressed before this promise can be fully realised; not least of which is the problem of biased deep neural networks resulting from biased training datasets [104].

With all supervised learning models, the issue of training set bias is of utmost importance. Typically studies of neural networks for image reconstruction have focussed on limited data domains. However, in many reports of neural networks for tomographic reconstruction, the models are trained and tested on only a limited data domain [105], [106]. The area of understanding when a new test example is outside the distribution of training examples is an active field of research in deep learning with many schemes for uncertainty quantification and out-of-distribution detection proposed [107]–[111]. A recent study in materials discovery showed how a so-called leave-one-out cross-validation can be used to assess the ability of a model to perform on previously unseen data [112].

This chapter addresses the question of biased datasets when developing DNN models for tomographic image reconstruction, focusing on the domain of materials science. It will show how biased datasets result in poor model performance on new, out-of-distribution test examples. The neural networks trained on a well-balanced dataset of diverse materials systems can match or

outperform the filtered back projection method. A representative architecture, based on convolutional layers which were trained on a diverse set of materials samples, was developed which we note has not previously been applied to XRD-CT data. It will be shown that models trained on one sample type perform very well on that sample type, but very poorly on the other types of samples. Then, it will be demonstrated that balancing the training set with different sample types can yield a DNN capable of reconstructing all sample types in the test set. Finally, and most importantly, it will illustrate how the application of manifold learning allows us to assess in advance if a sample belongs to the same set of data that was used for training the DNN, resulting in it being possible to make a statement about the reliability of the DNN for the reconstruction of a given sample without having to compare the reconstruction to a known ground-truth. A 3XS Data Science Workstation C264X2 with 2x Intel Xeon Silver 4216, 350 GB RAM and 2x Quadro RTX 8000 was used to perform the development and training of the neural networks used in this work.

4.3 Results and Discussion

4.3.1 Supervised-Learning Reconstruction CNN

The CNN architecture used for reconstructing the images from the sinograms is depicted in Figure 4.2. The architecture consists of a block of convolutional layers, followed by a block of fully connected layers followed by another block of convolutional layers. The inclusion of the initial block of convolutional layers before the dense layers means that we can reduce the parameter size of the network and therefore it can be applied to larger sinograms and images, without causing severe memory problems, as encountered when applying the AUTOMAP network [28]. The full details of the network architecture and training are provided below. It is worth mentioning that the model architecture that has been used in this work is not novel, but rather a common type of architecture and our intention is to highlight the potential pitfalls of applying this type of model without due consideration.

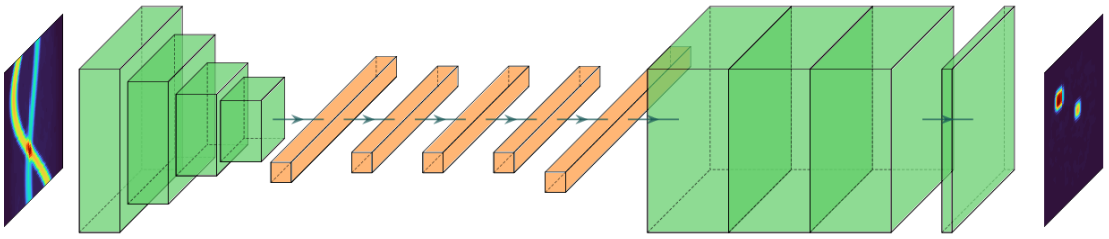

Figure 4.2: A representation of the CNN reconstruction architecture. The green blocks represent convolutional layers, the orange blocks represent fully connected layers. The first convolutional block of the CNN reconstruction network has four convolution layers, each with 16 (3 x 3) filters and a stride of two. The outputs are flattened and passed into a block of 4 fully connected layers, each with 100 nodes and 10 % dropout between dense fully connected layers. The output of the fourth fully connected layer is passed to another fully connected layer with dimensions of the number of pixels width and height of the final image. These outputs are reshaped to the final image dimensions and then passed through three convolutional layers with 16 (3x3) filters and a stride of one, finally the output of the final of these convolutional layers is passed through a convolutional layer with one (3 x 3) filter, which produces the final image.

4.3.1.1 Training/Testing Datasets

For the datasets examined in this study we have chosen a set of four diverse materials science use cases, each of which has been studied previously using X-ray diffraction computed tomography (XRD-CT). XRD-CT is particularly powerful in materials systems where the spatial and chemical composition of the system determines utility [113], [114]. The materials systems examined have just such chemical/spatial dependencies. We examine three heterogeneous catalyst systems, one for the partial oxidation of methane (POX) [115] which consists of Ni-Pd/CeO_2-ZrO_2/Al_2O_3, one for CO_2 methanation (3D printed catalyst) consisting of Ni/Al_2O_3 [116] and a dataset of a commercial AAA Li-ion battery (Battery), which contains diffraction patterns for seven phases including Fe, Al, Cu, LiC_{12}, LiC_6, and two $LiCoO_2$. We also include a dataset of a simulated pure Ni catalyst system (Ni) consisting of Ni crystallites of different sizes and lattice constants [117].

The datasets chosen are not only representative of typical systems where XRD-CT is particularly powerful but also provide a diversity of morphologies. For example, the battery dataset is arranged as spiral filaments, while the Ni catalyst consists of a range of differently sized and dispersed crystallites. This kind of structural diversity can present a particular challenge to convolutional neural networks if they have not been exposed to a particular type of feature in the training set.

The datasets are generated from the XRD-CT diffraction maps - the maps consist of 128 x 128 pixels, each pixel containing 2048 points in the 2θ dimension of the diffraction pattern. 2D maps of each point in the diffraction pattern are built, resulting in 2048 different maps per sample, each of these is then augmented by random rotations to increase the dataset and each 2D image is Radon converted to a sinogram to provide a sample (sinogram) and label (map) pair for training the networks and this process is illustrated in Figure 4.3. Before data augmentation, 10 % of the maps for each sample are removed and kept as an independent test set which is used to check the performance of the models on data not used in the training in any capacity.

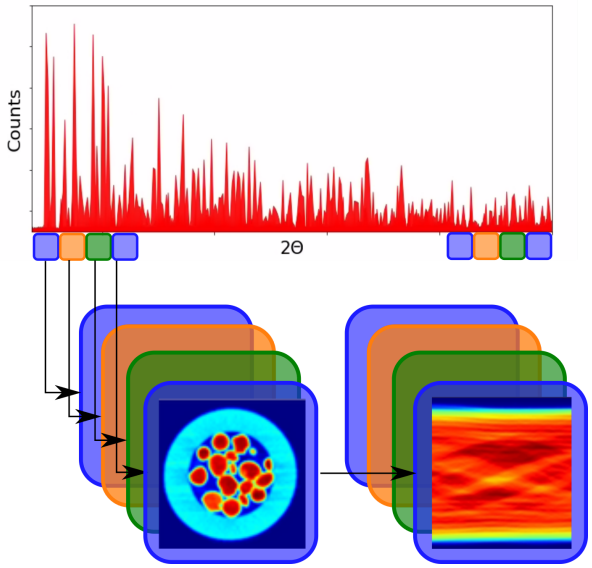

Figure 4.3: The dataset generation procedure. Diffraction patterns for each pixel in the XRD-CT data are divided into 2048 points along 2θ for each pixel in a 2D map which is produced from the diffraction intensity at that location in the sample. The 2048 maps are augmented to result in 20000 maps by applying random rotations of the data, these are then converted to sinograms by radon transform.

The images from each XRD-CT dataset (i.e. the two experimental and two simulated datasets) were resized to images with 128 x 128 pixels using bilinear interpolation. A 10 % random split was applied to each library in order to extract the test dataset. The remaining 90 % of each library was augmented to 20,000 images by applying a random rotation in the angular range of 0-360 °. The images in each library were then randomly mixed and each image was normalised. The sinogram volumes were calculated from these augmented libraries using the Astra toolbox using 128 projections covering a 0-180 ° angular range [118].

4.3.1.2 t-SNE Analysis

In order to analyse the datasets and see if it was possible to know, in advance, if a pre-trained CNN would be applicable to a previously unseen dataset, a manifold learning approach was employed. In manifold learning, the dimensionality of the data is reduced in order to better interpret them. In this case, the training sinograms are projected into two-dimensional space, in order to see if data from the same training sets tend to cluster together. If the data from the same underlying datasets cluster together and separate from the data in other datasets, then it is possible to tell in advance if a previously unseen piece of data is covered by the training set, by looking at where it lies in this two-dimensional space.

To reduce the dimensions we combine a pre-trained CNN, the ResNet-50 model [74], with the t-distributed stochastic neighbour embedding (t-SNE) [119] approach. The CNN initially reduces the dimensionality of the input data to 2048-D by successive convolution and pooling operations, this 2048-D vector is reduced to 200-D by principal component analysis (PCA) and is then used as input for tSNE, which calculates probabilities that data points in the high-dimensional space are related and then finds a lower dimension (in this case in 2-D) embedding which keeps related points together. In this way, the input data is converted into a 2-D representation which can be investigated to determine if related datasets cluster together. The advantage of this approach is

that the initial CNN reduction can pick unbiased features out of the images, which are most useful for separating between different classes, while the subsequent t-SNE reduction allows for mapping the relationship between the most important of these features in a non-linear way into a low-dimensional space. For example using just PCA to project down to 2D would miss any non-linear dependencies between the most important features of the CNN learned features.

For the t-SNE analysis, 224 projections were used covering a 0-180 ° angular range as the required image size is 224 x 224 for the ResNet. All filtered back projection (FBP) images presented in this work were also obtained using the Astra toolbox. A 10 % validation split was applied to the data for the training of the reconstruction CNN.

4.3.1.3 CNN performance influenced by the training dataset size

Before looking at the individual libraries we first investigated the amount of data required to achieve the best-performing network. Taking a mixture of all four libraries as our test case we trained the CNN with sets of 1000, 2000, 5000, 10000, 20000, 40000, and 80000 sinogram/image pairs; the results are presented in Figure 4.4. After an initial large improvement of the mean squared error (MSE) calculated between test outputs and ground-truth images the model performance begins to plateau after around 20,000 training examples and has almost stopped improving by 80,000 pairs.

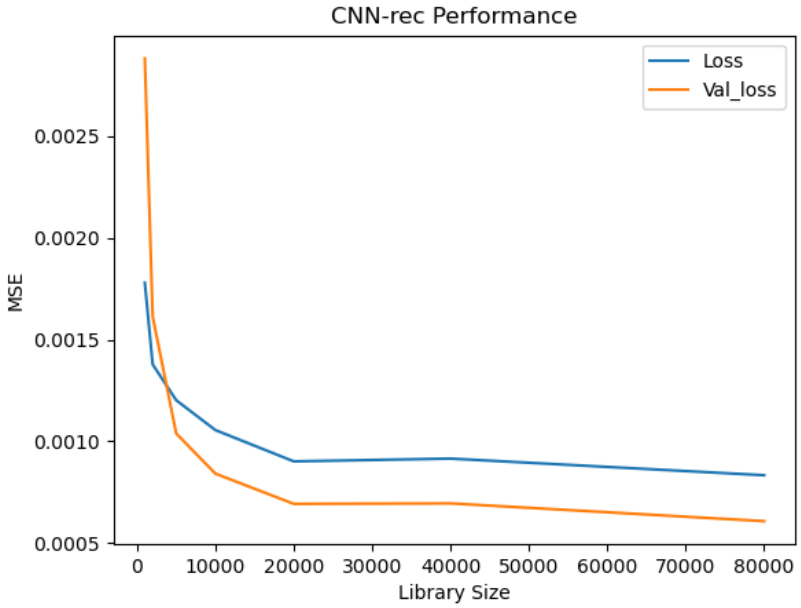

Figure 4.4: Validation performance versus training set size of the reconstruction network for the Ni system.

4.3.1.4 Individual Libraries

We begin by assessing the performance of the reconstruction CNN within a single application domain, i.e. training on a given sample type and then reconstructing on that same sample type. We compare the performance of the trained models on datasets of 18,000 sinogram/image pairs with the performance of filtered back projection (FBP). For each sample type, we have calculated three common image quality metrics to compare the reconstructions to the ground truth, i.e. the SSIM, the PSNR and the MSE on test sets of sinograms/image pairs that were not included in the training or validation sets. The values are reported in Table 4.1 and representative results are shown in Figure 4.5. From Table 4.1 we find that the CNN outperforms the FBP algorithm in all metrics for all samples. These results clearly demonstrate the ability of the CNN architecture to reconstruct images from sinograms, matching or bettering the best available physics-based reconstruction methods.

Table 4.1: Comparison of CNN reconstruction (trained on single set data) to FBP for the different materials test data sets studied. Metrics compared are the SSIM, MSE, and PSNR. The metrics are calculated on the test sets as described in the Training/Testing Datasets section.

Testing library	Algorithm	SSIM	PSNR	MSE
Ni Phantom	CNN	0.9933	38.0883	0.0002
	FBP	0.9496	32.7723	0.0005
Battery	CNN	0.9800	29.3750	0.0012
	FBP	0.9250	23.8939	0.0041
POX Experimental	CNN	0.9689	31.4884	0.0007
	FBP	0.9663	30.6442	0.0009
Printed catalyst	CNN	0.9752	30.1523	0.0009
	FBP	0.9280	27.9155	0.0016

4.3.1.5 CNN transferability

The CNNs in the previous section were applied to reconstruct sinograms from the types of the same material as those on which they are trained. The question now is: how well can a model trained on one dataset perform on another, very different dataset? In other words, how does data bias affect model performance? To assess this, we have applied the models trained on each individual material system to reconstruct each of the other systems. The full results of all tests are reported in Appendix 2, and representative results are presented in Figure 4.5.

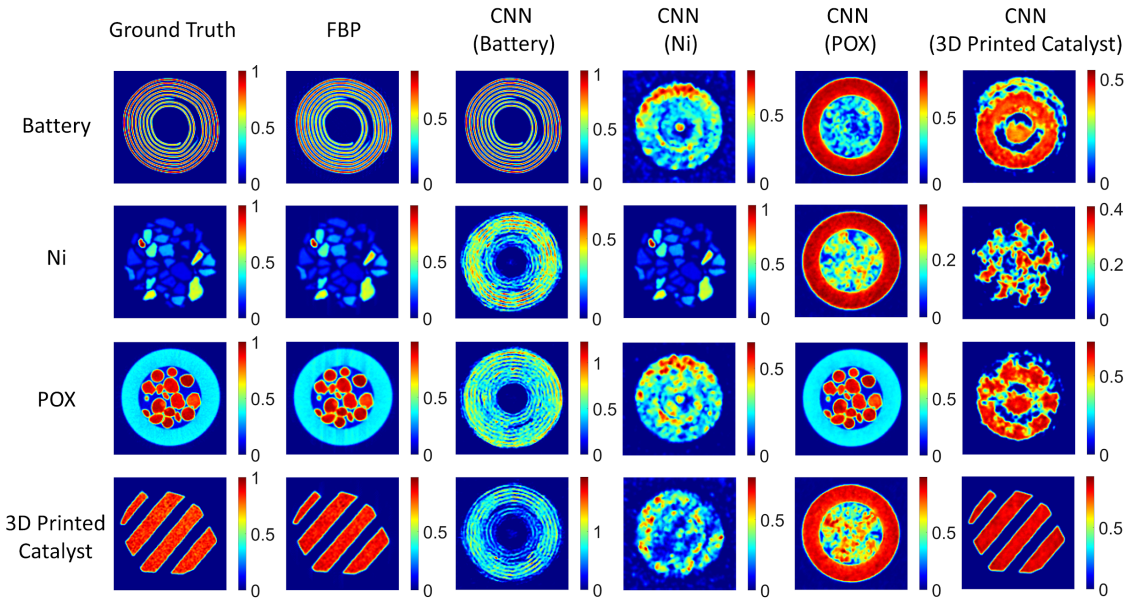

Figure 4.5: Representative examples of reconstructed images using CNNs trained on different test data sets. The reconstruction algorithm is given along the top and is constant for each column, whereas the sample used for reconstruction is constant across the rows. The ground truth image is given in the first column, the FBP reconstructed image is given in the second column and the subsequent columns are CNNs trained on different datasets. Since the signal intensities of different experimental dataset are not the same, for a better performance when training the model, all input images are normalised from 0-1.

Figure 4.5 presents randomly chosen representative examples of reconstructed images from each of the test sets; the results are striking. We can see clearly that the CNN trained for one material system does not perform well on the other systems. The training set bias means that the CNN has learned to reconstruct only images that resemble the training set so that even given a new sinogram the CNN attempts to reconstruct something similar to the training data. This represents a significant bottleneck for the routine application of CNNs for tomographic reconstruction from sinograms. Each CNN we trained here used 20,000 image pairs with 32 as batch sizes. The Adam optimizer and the MSE loss function are used for training the networks. All images are 128 x 128 large, and they are normalised, so the trained CNNs are comparable on the same order of magnitude. For training and testing the CNN, each step takes 16 ms on average.

4.3.1.6 Combined Library

We next tested how diversifying the training dataset affects the performance of the reconstruction CNN. We built libraries of 80,000 and 20,000 image/sinogram pairs, drawn uniformly from all four materials systems and retrained the CNN on these. We then applied this CNN to previously unseen examples from each of the types of material. The full results are again presented in the Appendix 2, Tables S2.2-S2.5. In Table 4.2 we present the results for the CNN trained on the combined library of 80,000 image/sinogram pairs and the FBP, representative examples are presented in Figure 4.6.

Table 4.2: Comparison of CNN reconstruction (trained on mixed data) to FBP for the different materials data sets studied. Metrics compared are the SSIM [120], the peak signal-to-noise ratio (PSNR) and the mean squared error (MSE). The metrics are calculated on the test sets as described in the Training/Testing Datasets section.

Testing Library	**Algorithm**	**SSIM**	**PSNR**	**MSE**
Ni Phantom	CNN (20K)	0.9783	33.7930	0.0004
	CNN (80K)	0.9871	35.5392	0.0003
	FBP	0.9496	32.7723	0.0005
Battery	CNN (20K)	0.9629	27.2021	0.0019
	CNN (80K)	0.9710	27.7324	0.0017
	FBP	0.9250	23.8939	0.0041
POX Experimental	CNN (20K)	0.9668	30.2381	0.0009
	CNN (80K)	0.9664	30.8215	0.0008
	FBP	0.9663	30.6442	0.0009
Printed Catalyst	CNN (20K)	0.9632	29.7243	0.0011
	CNN (80K)	0.9727	30.1083	0.0010
	FBP	0.9280	27.9155	0.0016

Table 4.2 and Figure 4.6 clearly demonstrate that diversifying the training set has, to a large extent, overcome the problems introduced by training set bias in the previous examples. The CNN now performs similarly, if not better than, the FBP algorithm for all of the datasets across all three metrics considered here. This demonstrates the importance of a dataset with sufficient diversity and moreover shows that transferable CNN models are possible, given the correct training data, therefore they can be applied as a generic tool for reconstructing images from sinograms. However, a final question remains: How can we know if the dataset we wish to apply the CNN on is covered by the data in the training set of the CNN?

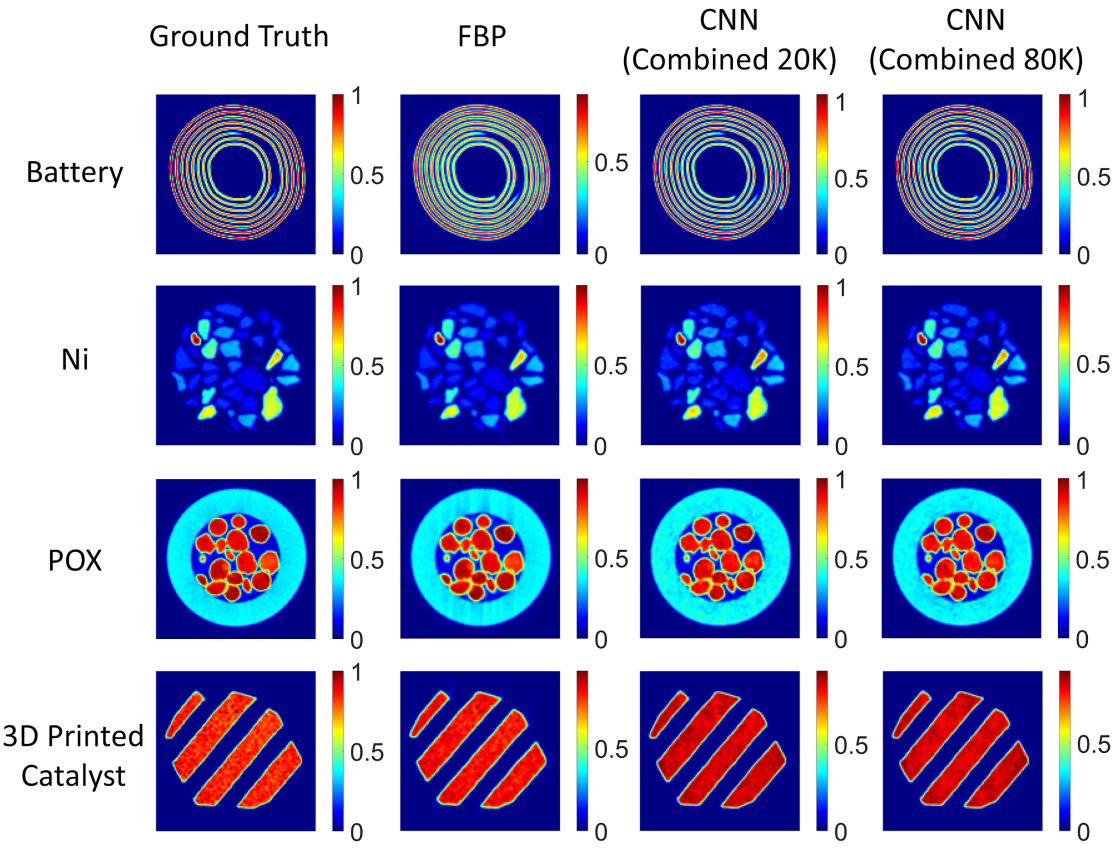

Figure 4.6: Representative examples of reconstructed images using CNNs trained on mixed data sets. The reconstruction algorithm is given along the top and is constant for each column, whilst the sample used for reconstruction is constant across the rows. The ground truth image is given in the first column, the FBP reconstructed image is given in the second column and the subsequent columns are CNNs trained on different datasets. All scale bars represent the diffraction intensities that the network predicted. For a better balance among all experimental datasets which possess different intensities, all images are normalised before conversion to the sinogram by Radon transformation.

4.3.1.7 t-SNE result map

In Figure 4.7 each of the points from each of the four datasets is coloured differently with representative examples of the images are also shown in the plot for reference. Note that we have also included a new dataset formed by blending the data from the POX and Ni-catalyst datasets (coloured purple). The 2-D map shows clear clustering of data points; moreover, the clusters are formed of data points from the same dataset, showing that it is possible with the reduction and projection approach to identify, in advance, whether a new data point is likely to be represented well by the training data.

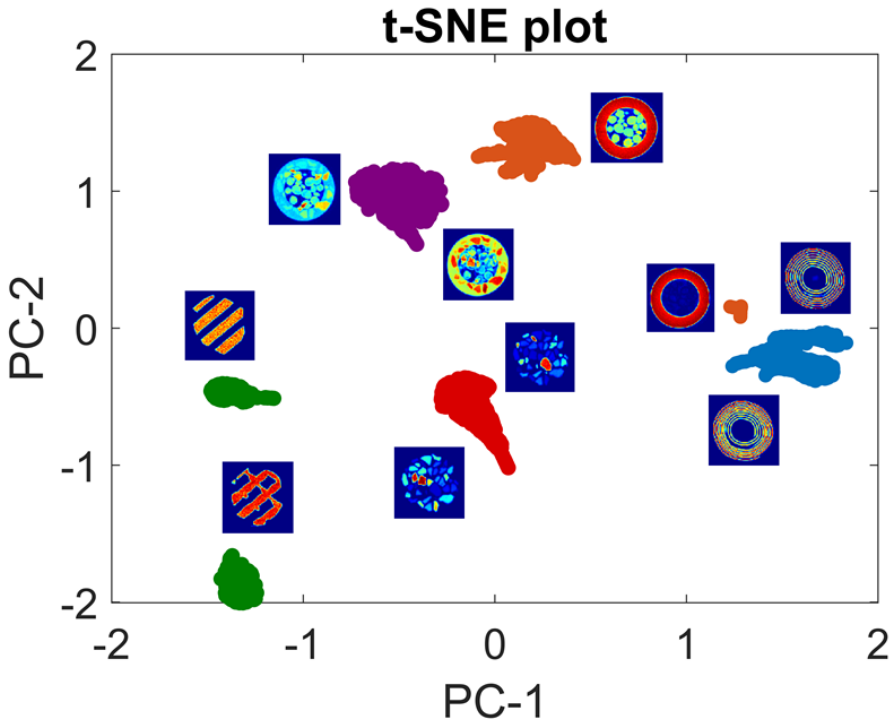

Figure 4.7: Reduced dimension projection of the different training/testing libraries. The data were passed through a CNN and then the dimensionality reduced further by the tSNE algorithm. The colours of the points indicate which original library the data were taken from; Battery (blue), Ni (red), POX (ochre), printed catalyst (green), and POX/Ni blend (purple). Representative images of the materials in the data sets are presented on the map to aid interpretation, these images are the reconstructed images, but the clustering was performed using the sinograms as input. Note that x and y coordinates are the units of the reduced dimensional space from the analysis and do not necessarily have physical meaning.

It is important not to over-interpret tSNE projections for example, while individual clusters indicate the relatedness of the constituent points, the size and shape of the clusters and the distance between clusters should not be attributed any particular meaning, so two clusters close together in the 2D projection are not necessarily more similar than two clusters far apart [121]. While the Ni and battery datasets (red and blue) lie almost exclusively in one cluster, other datasets have some clusters separated by a large distance. In those cases the separate clusters are related to some differences in the data, the two distinct clusters of the printed catalyst (green) represent scans at different sections of the catalyst system where new geometric features are present in the different sections.

We have also looked at what happens if a model is trained on two distinct datasets, and then applied to a blend of these datasets, which was not used in training. The t-SNE projection clearly indicates that this data belongs to a third cluster distinct to either of the parent structures (the purple cluster in Figure 4.7), suggesting that the CNN will not work well on this new data. When we test the reconstruction, as suggested by the clustering the results are quite poor; Figure 4.8 presents these results visually and Table 4.3 gives the numerical results. This is a further indication of the ability of the clustering approach to assess the applicability of a CNN reconstruction model in advance of use.

Table 4.3: Ni-POX test dataset images. Metrics calculated with respect to the ground truth images.

Training library	SSIM	PSNR	MSE
Combined (20 K)	0.8862	23.8185	0.0042
Combined (80 K)	0.8746	24.2387	0.0038
FBP	0.9889	37.0308	0.0002

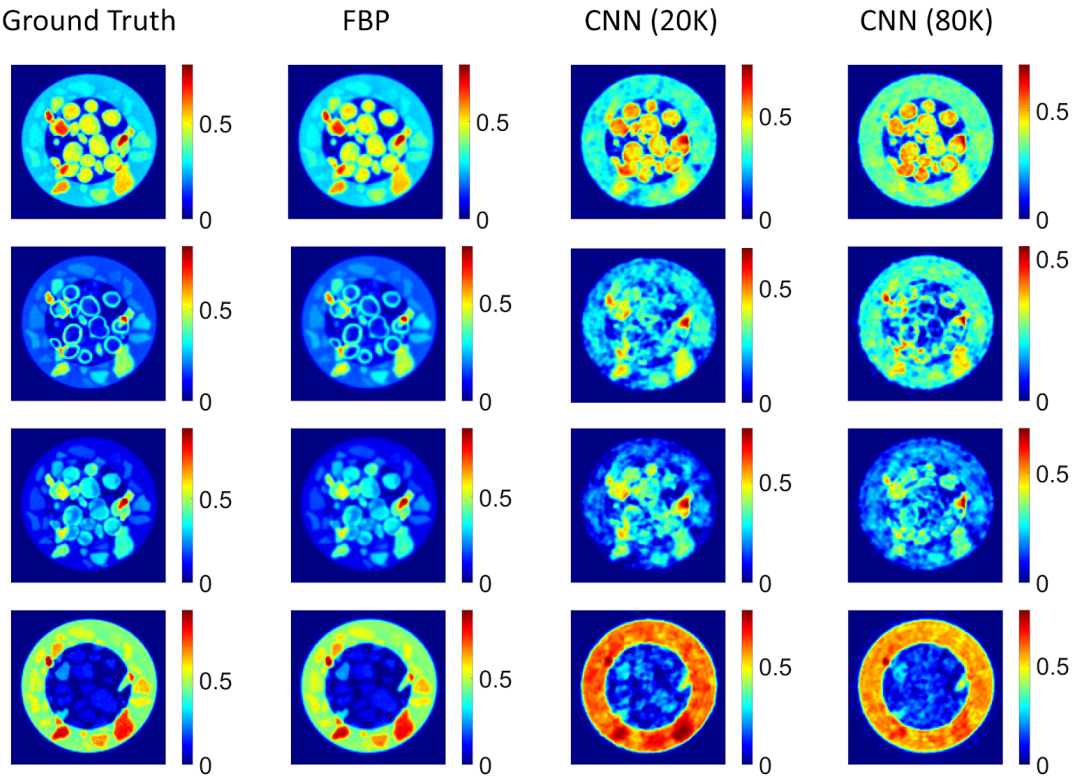

Figure 4.8: Ni-POX test dataset images. The CNN is trained with two separate datasets and then applied on composite images. CNN (20K) and CNN (80K) represent the CNNs trained with twenty thousand and eighty thousand images respectively. It is shown that even if the CNN has already seen similar structures during training, it is difficult to use the trained CNN with the new dataset directly.

The projection of the data into a lower dimensional space in this manner is therefore likely to highlight similarities and differences in the datasets. The reduced dimensionality map possibly over-emphasises differences in the data points, but we can clearly see that if a data point is part of a given cluster, then we can say with a high degree of confidence that the data point is likely to be closely related to the rest of the data in that cluster and therefore if the CNN was trained on the data in that cluster then it is also likely to perform well on the new data point. This provides a critical tool for applying CNNs in the reconstruction of tomographic data with increased confidence that the network is well suited to the given use case.

4.3.1.8 Summary

In this section, the utility of CNNs for reconstructing tomographic data in a diverse range of materials systems, with the ability to outperform traditional

direct methods like filtered back projection has been demonstrated. It has been shown that training set bias can be a serious burden to the application of CNNs for the reconstruction of images from sinograms, a point rarely discussed openly in the literature. It was then demonstrated that balancing the training dataset leads to CNNs that perform well across the range of applications while sacrificing very little in performance relative to those trained exclusively for one type of material system. In all systems tested the CNNs achieve similar or better performance than the standard FBP image reconstruction. Finally, we provide a method, based on manifold learning, which allows one to assess, in advance, whether a new data point for which we wish to apply the reconstruction CNN is likely to be well represented by the training data for that CNN. This reduced dimensional mapping approach can be a key ingredient for the widespread, trusted, generic application of CNNs in tomographic image reconstructions.

4.3.2 Image-to-image supervised artefact removal

During the past few years, CNN-based methods have previously been shown to be effective in reliably learning nonlinear relationships between high and low resolution [122]–[124]. Unlike the sinogram-to-image reconstruction networks, the image-to-image artefact removal networks do not normally require many fully connected layers and many parameters. Figure 4.9 shows the artefact removal U-Net (ARU-Net) with dilated convolutional layers that has been used to remove angular undersampling artefacts created by the FBP reconstruction with only a quarter of the sinogram.

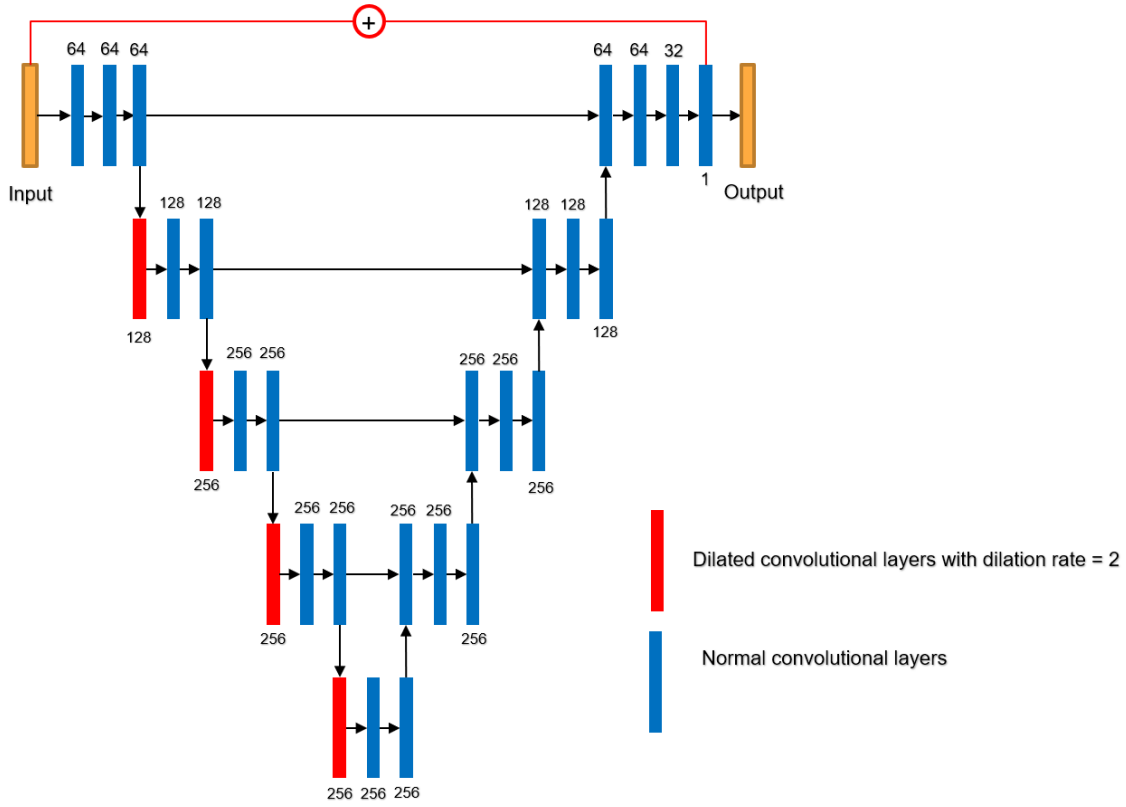

Figure 4.9: Artefact removal U-Net (ARU-Net) architecture for image artefact correction. Red layers are dilated convolutional layers with a dilation rate of 2, and blue layers represent normal convolutional layers. All filter sizes are equal to 3, and filter numbers are shown above or below each layer. A residual learning connection (marked as a red line) between the input and output layers is used to force the network to focus only on artefacts.

The architecture is trained on a combined training dataset with DIV2K images in a Python environment. The DIV2K is a very popular open-source image pack for training artificial neural networks (ANNs) [125], [126]. Since the images from this dataset are not from the same data frame, it covers and balances all the features that an image may encounter in real life, therefore, it is possible to avoid training bias and apply the trained model to more data frames. First Radon transformation was used to obtain 512 x 512 sinograms, subsequently, each sinogram is splitted into four sparsely angular-undersampled 512 x 128 sinograms so that each of them contain only 25% projections as the full sinogram. Next the order of sinograms to form 10,000 undersampled sinograms was disrupted and FBP images of the sinograms to form 10,000 training datasets together with their ground truth were created. As illustrated before, the

architecture is based on the maximum computation resources of our computer, so it takes 52 min for each training iteration. Fortunately, the model's results are rather good and it is even possible to reduce the number of parameters in the architecture to give a faster performance on training and predicting.

Figure 4.10 shows the prediction of this CNN with a test DIV2K image. It shows the ARU-Net can correctly detect the artefacts caused by the FBP method with fewer projections in sinograms. Comparing the input FBP and the ARU-Net result in the figure, the ARU-Net correctly cancels most of the straight noise lines and sharpens the image quality. To further reduce the noise level on the predicted results, it is possible to adjust the loss functions and also use a learning rate scheduler in the training process. Then it is possible to apply the trained model into an experimental dataset to test its robustness.

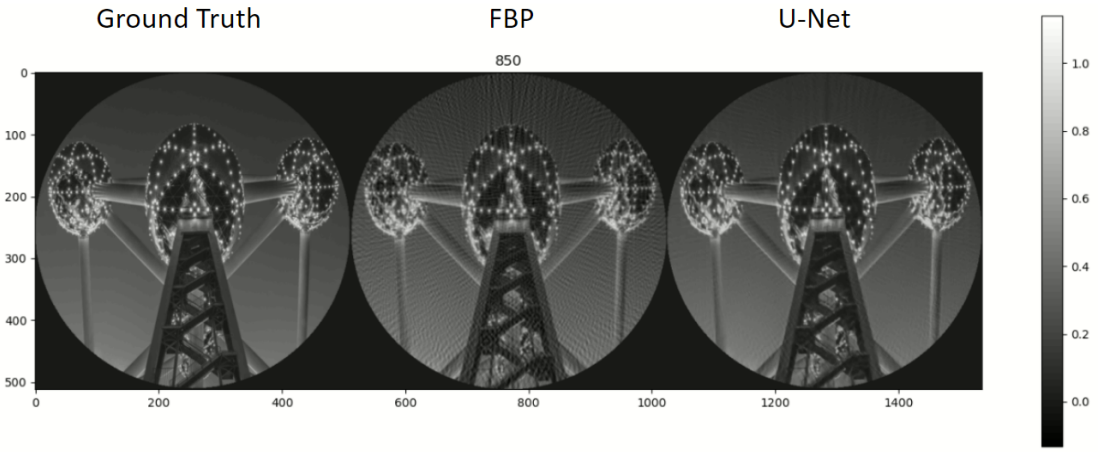

Figure 4.10: Performance of the image-to-image angular-undersampling artefact correction U-Net with dilated convolutional layers. The image on the left is the ground truth, the image in the middle the input to the model and the U-Net result shown on the right. The picture is from the test dataset from DIV2K [125], and the input is the 512 x 512 FBP image that comes from the 512 x 128 under-sampled sinogram. The grey-scale bar represents the signal intensities (brightness) at each point. The ground truth image is a real-world image normalised from 0 to 1.

Figure 4.11 shows the improved resolution of a neutron-CT image reconstructed using only 25 % of the projections after being processed using the trained ARU-Net CNN. The model never sees similar pictures in the training dataset, but it can still suppress the angular-undersampling artefact on the right. Different from the sinogram-to-image reconstruction results, because this model

is trained with generalised DIV2K images which covers all features in the real life image, the well-trained network can be applied directly on CT datasets without transfer-learning. The ARU-Net processed reconstruction shows comparable if not of qualitatively better resolution than the ground truth image. This demonstrates that it is possible to obtain satisfactory CT reconstructions with less information; in principle, this technique could be used to collect datasets faster. The current collection time for neutron tomography measurements with 70 and 100 um resolution on a sample of a few centimeters size is 4 and 6 h respectively. This could be reduced to 1 and 1.5 h respectively, increasing the amount of time available to run other samples during beamtime.

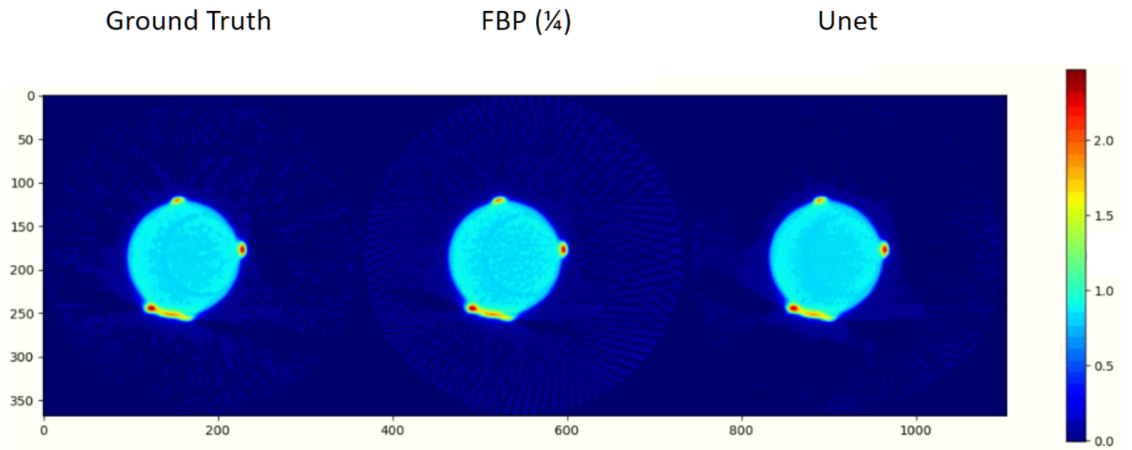

Figure 4.11: Artefact removal with ARU-Net. Ground truth neutron-CT image (i.e. the original images constructed from the complete set of sinograms) (left), the image reconstructed with 1/4 of the sinograms (centre), and the image reconstructed with 1/4 of the sinograms after being passed through the ARU-Net super-resolution CNN (right). The slice is from the metal phantom, with a metal rod (light blue) and glue residues (red). The scale bar represents the reconstructed signal intensities in each pixel.

However, we found it is not always safe to trust the results predicted by CNN. This is because the CNN processes the images directly and the angular-undersampling artefacts look similar in all images, with minor structures in the CT image likely to have been removed, especially those structures that appear to be artefacts. Figure 4.12 shows an example of the experimental XRD-CT image in which the minor structure is not reconstructed correctly.

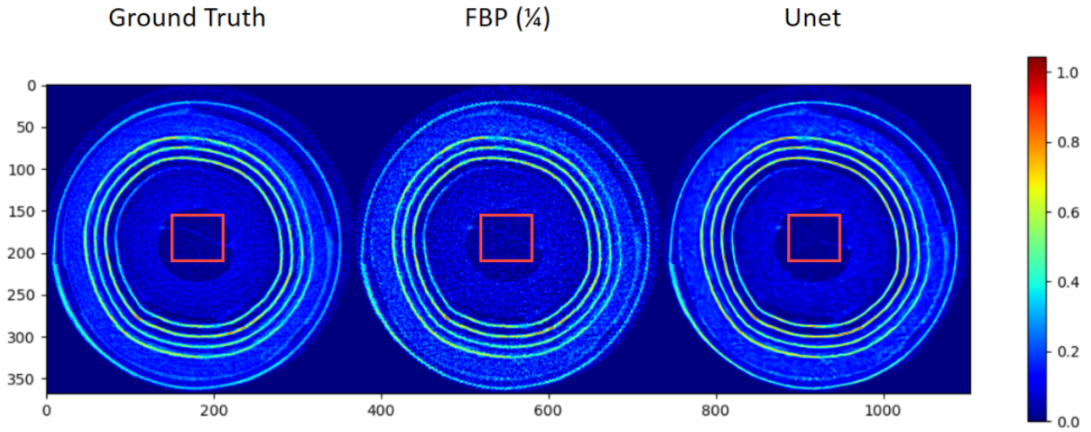

Figure 4.12: Performance of the image-to-image angular-undersampling artefact correction ARU-Net with dilated convolutional layers on real experimental data. The image on the left is the ground truth, the image in the middle is the input to the model with the ARU-Net result shown on the right. The ground truth is an XRD-CT image of a Li-ion battery reconstructed by full sinogram projections [127], and the input is an FBP image that comes from the 25 % under-sampled sinogram. The enclosed parts are not reconstructed correctly by the CNN. The scale bar represents the reconstructed signal intensities in each pixel.

To build a more robust network that can retain such minor features, more information from the sinogram domain must be considered and fed into the network. In the next section, a more robust, reliable and stable self-supervised learning architecture to directly reconstruct images from their sinograms, and remove the angular undersampling artefacts simultaneously is introduced.

4.3.3 SD2I: A flexible, scalable self-supervised learning reconstruction network

The SingleDigit2Image (SD2I) network is a simple, scalable self-supervised learning generative network that can be used for direct conversion of sinogram to image [128]. This approach utilises a GAN-like architecture, but in contrast during the tests, it was observed that the discriminator does not greatly improve the reconstruction image quality. Therefore, the discriminator was abolished, and the network is trained only based on the joint loss function with SSIM and MAE. This new approach leads to a series of tomography images of differing modality with a comparable accuracy to the FBP algorithm. Importantly the scalability of the new method is improved compared to state-of-the-art networks. In this section, it will also be shown that this method is able to deal

with a commonly encountered challenge to FBP reconstruction, specifically angular undersampling in sinograms. This approach has been verified on a Shepp-Logan phantom as well as on experimental X-ray diffraction computed tomography (XRD-CT) and micro-CT sinogram data highlighting the flexibility and applicability of the method.

4.3.3.1 Training loop

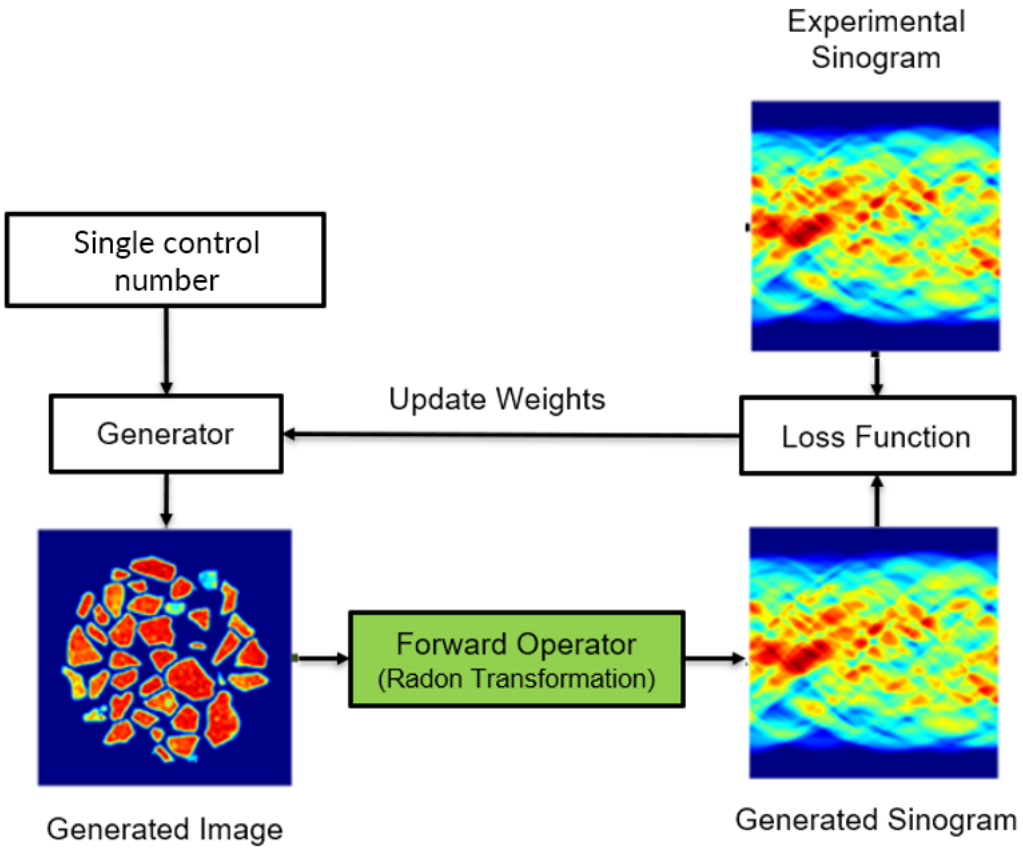

Figure 4.13: The flowchart of the SD2I training algorithm. To adapt to sinograms with different signal strengths, and make it easier for transfer learning, the input of the SD2I is a control number which preferably has a similar order of magnitude as the mean signal of the reconstructed image. The generator creates an image based on the single input; the generated image is then converted into a sinogram by the forward operator, which is compared with the sinogram from the experimental dataset. The weights of the generator are updated by minimising the joint loss function with the mean absolute error (MAE) and structural similarity index measure (SSIM).

The SD2I training loop is shown in Figure 4.13. This architecture only requires a single control number as input. Theoretically, this number can be any positive real number but is typically of a similar order of magnitude as the reconstructed image signal, which can help to use transfer learning on all image channels in a

dataset. This architecture has excellent potential to be applied to solve other image reconstruction problems by modifying the forward operator and loss functions.

4.3.3.2 Generator architecture

The architecture of the ANN used for reconstructing the images from the sinograms is depicted in Figure 4.14. Two types of SD2I generator architecture are considered that differ depending on whether the upsampling layers are used. These two architectures will be delineated as the SD2I with upsampling layers (SD2Iu) and the 'SD2I uses only full image resolution' (SD2If).

There are two novelties in this design compared to other architectures previously proposed for tomographic image reconstruction. First, the SD2I networks, as the name suggests, start from a single neuron rather than a 2D image which significantly reduces the number of parameters in the architecture. In other networks, the input is a 2D image which is either flattened and connected to a dense layer containing 100s of neurons (e.g. 256 in the GANrec) or is followed by a series of 2D convolutional and downsampling layers with the final layer being flattened and connected to the aforementioned dense layer. The second novelty is related to the large dense layer and the convolutional layers that are connected to it which dramatically reduces the number of parameters in the network's architecture. To clarify, the initial single neuron is followed by three small dense layers, each containing 64 neurons. This third small dense layer is then connected to a larger dense layer consisting of m × m × k neurons for the SD2If, and (m/4) × (m/4) × k neurons for the SD2Iu, where m is the number of pixels in one dimension of the fully reconstructed images which have a size equal to m × m. The k factor is an integer and increasing it can lead to better performance of the neural network but also increases the number of parameters.

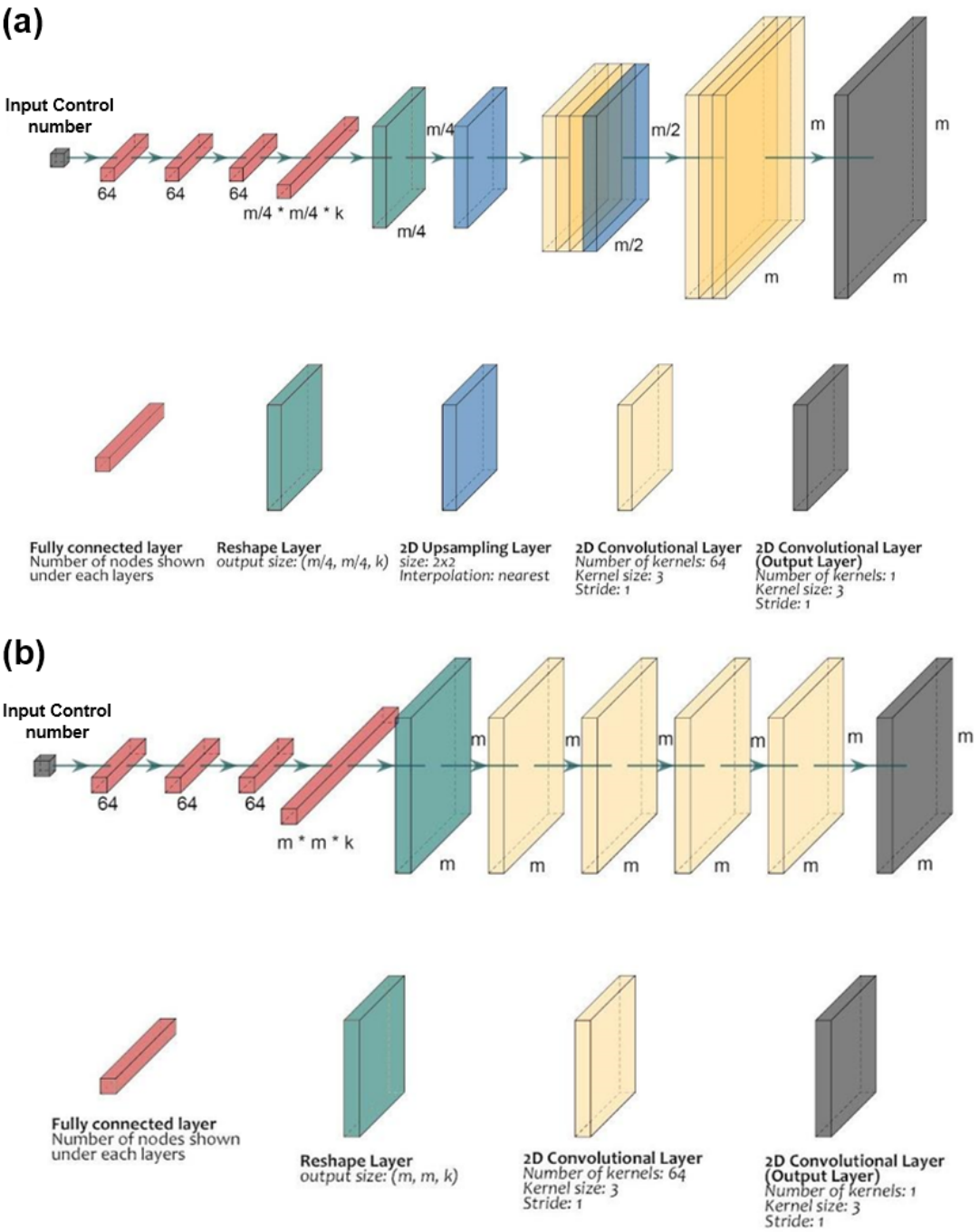

Figure 4.14: (a) A representation of the CNN reconstruction SD2I architecture with upsampling layers (SD2Iu). (b) A representation of the CNN reconstruction SD2I architecture without upsampling (SD2If). The kernel types and parameter settings are shown in the bottom part of the figure. The final fully connected layer size is adjusted by an integer k, which adjusts the number of kernels used as the input of the following reshape, upsampling and convolutional layers.

For the SD2Iu, the last large dense layer is reshaped to a 2D layer with a size of (m/4, m/4), followed by an upsampling layer resulting in an image with a size of (m/2, m/2). This is followed by three 2D convolutional layers and a second upsampling layer resulting in an image with a size of (m, m). Each of these 2D convolutional layers has 64 filters with a kernel size of 3 and a stride equal to 1. The final layer of the architecture is a 2D convolutional layer with one kernel (kernel size of 3) and stride equal to 1. Overall, the network starts with a single neuron and yields a 2D image with size of (m, m) which is equal to the image size obtained with the conventional tomographic reconstruction algorithms. The advantage of the SD2I's architecture is that it allows for a radical decrease in the number of parameters and allows it to reconstruct images that are more than 1028 × 1028 large.

While several deep learning approaches have proved very successful for CT reconstruction, a major barrier to their widespread adoption is that the number of parameters (and hence required computational resources) scales poorly as the size of the sinogram increases. The new architecture proposed here has at least an order of magnitude fewer parameters than existing deep learning approaches (e.g. Automap and GANrec), as shown in Table 4.4.

Table 4.4: Number of parameters (nop) per network as a function of image size.

Sinogram / Image size (pixels)	**Automap (nop) x10^3**	**GANrec (nop) x10^3**	**SD2If (factor 8) (nop) x10^3**	**SD2If (factor 4) (nop) x10^3**	**SD2Iu (factor 8) (nop) x10^3**	**SD2Iu (factor 4) (nop) x10^3**
64 x 64	79,356	2,444	4,361	2,308	331	263
128 x 128	1,304,960	8,698	16,910	8,583	731	462
256 x 256	-	33,815	67,370	33,813	2,328	1,261
512 x 512	-	134,477	269,741	134,998	8,718	4,456
1024 x 1024	-	537,523	1,080,283	540,269	34,277	17,235

Note that for these tests it was not possible to use Automap on images larger than 128 x 128 pixels, due to memory constraints. To find the best k factor, we tested values from 1 to 16 on reconstructing the 256 x 256 Shepp-Logan image, the results of which are shown in Figure 4.15. The figure shows that the performance of SD2I is increasing as the k factor is increased from 1 to 8, but the performance of SD2I then stops improving or even becomes worse when the k factor is larger than 8. It may be because the ratio of the number of parameters between the final two fully connected layers also increased, which makes it harder for the network to converge. In this work and after this testing, a range of k between 4 and 8 were chosen; this range providing a good balance between network size/ training speed and the quality of reconstructed images

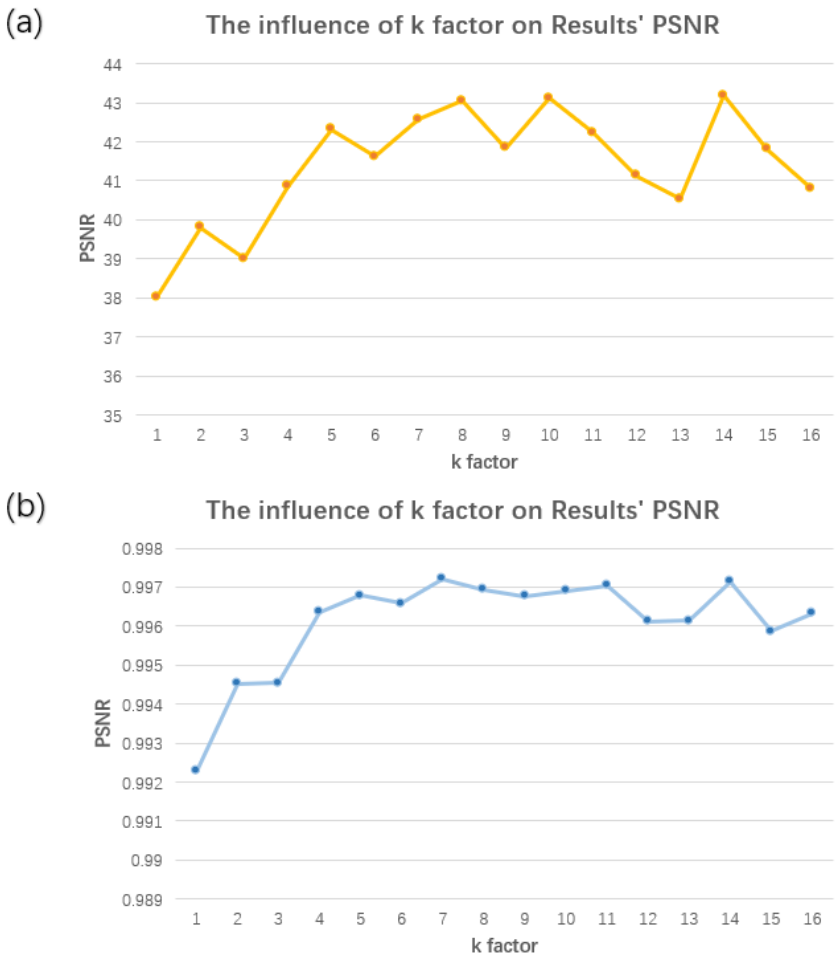

Figure 4.15: The influence of different k factors used in SD2I on the result's (a) PSNR and (b) SSIM. Using larger k factors can improve the quality of reconstructed results on both metrics. In practice, using a k factor between 4 and 8 is more appropriate for achieving a good balance between model size and accuracy.

4.3.3.3 The impact of discriminator

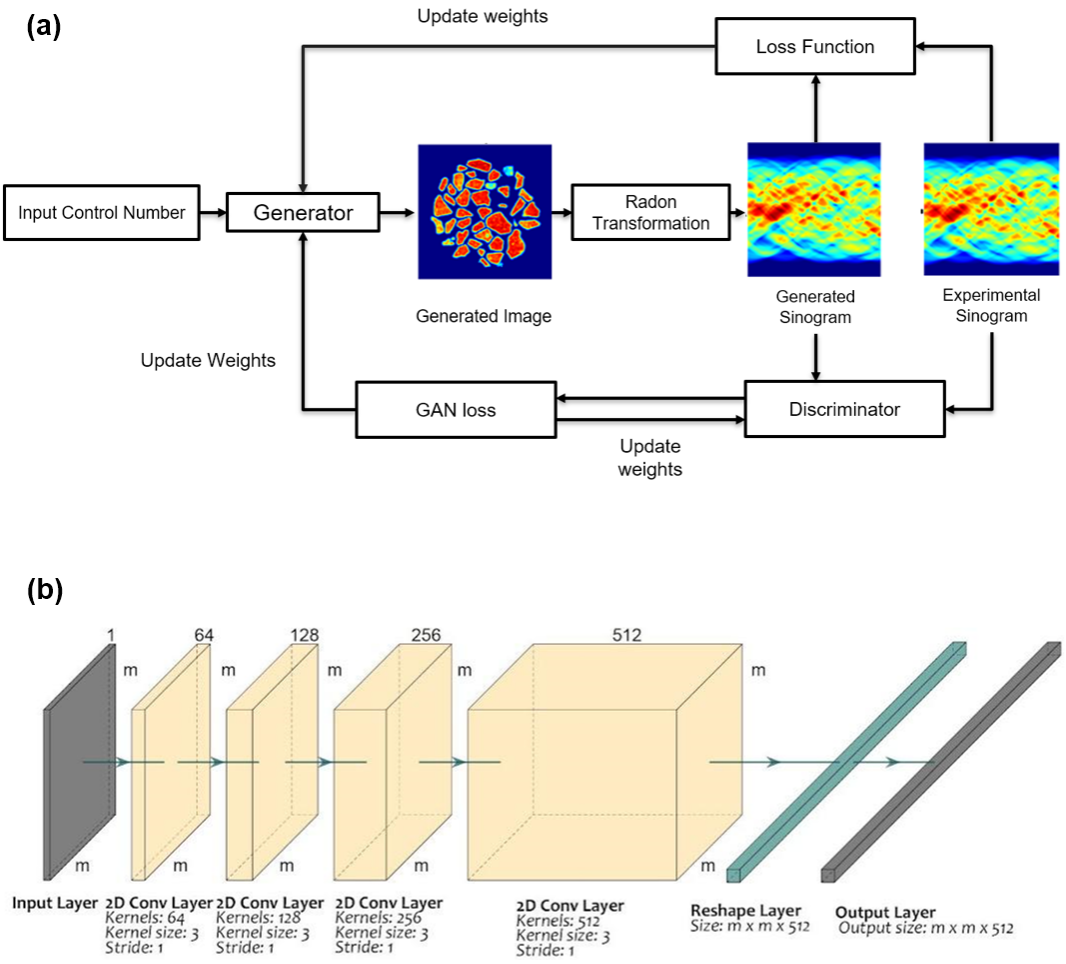

Figure 4.16: (a) The flowchart of the SD2I training algorithm with a discriminator. The input of the SD2I is a control number which should ideally have a similar order of magnitude as the reconstructed image's signal. The generator creates an image based on a single input. The generated image is converted into a sinogram by the forward operator, and then both the generated sinogram and the original experimental sinogram are sent into the discriminator for calculating the GAN loss. The weights of the generator are then updated by minimising the joint loss function with the GAN loss, MSE and SSIM while the discriminator is updated by the GAN loss only. (b) A representation of the Discriminator network used. The kernel types and parameter settings are shown in the figure. There are no fully connected (dense) layers in the discriminator so the number of parameters is very low compared to the generator networks used in this work.

A GAN-like architecture, shown in Figure 4.16(a), is also used as the SD2I training loop. Although many papers use the GAN loss to train their networks, the SD2I results seem not to improve by applying the discriminator. This may be

because the simplicity of the SD2I networks makes it easier to converge, and the joint loss function is good enough in this case.

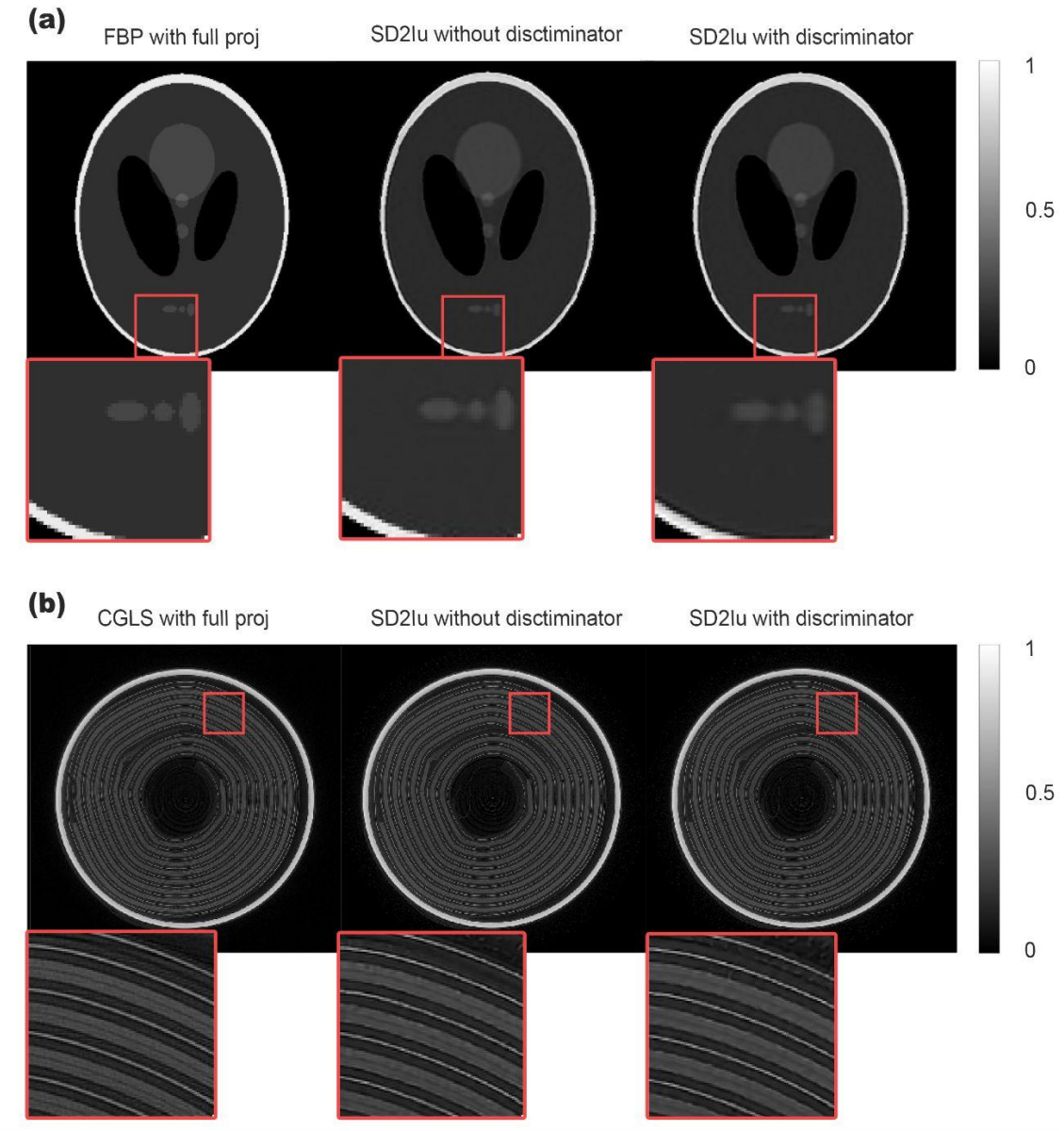

Figure 4.17: The images show the impact of the discriminator in the training loop. (a) The image is reconstructed from a Shepp-Logan simulated sinogram with 256 x 64 projections as input. The SD2Iu (k=8) is used as the generator. (b) The micro-CT experimental image with 779 x 223 sinograms as input. The SD2Iu (k=8) is used as the generator. The gray scale bars indicate the signal strength for each pixel in the reconstructed images after normalization.

The results from the testing of both loops on the simulated and experimental data are shown in Figure 4.17. The results with the discriminator not only look

worse but also are confirmed to be demonstrably worse according to all the metrics shown in Table 4.5. More importantly, the addition of a discriminator network makes the training more complex and requires more GPU memory while training. At this point we abandoned the loop with the discriminator and only applied the simpler training loop shown in Figure 4.13 in all datasets shown.

Table 4.5: Comparison of the results obtained by training loops with and without discriminator for both experimental (using CGLS of the full projection set as the ground truth image) and simulated data. The generator architecture is the SD2Iu. The metrics used to examine the quality of the reconstructed images are the MAE, MSE, SSIM and PSNR when compared to their ground truth.

	(a) Shepp-Logan		**(b) Micro-CT**	
	Without discriminator	**With discriminator**	**Without discriminator**	**With discriminator**
MAE	0.00315	0.00647	0.03191	0.03220
MSE	0.000125	0.000542	0.002306	0.002372
SSIM	0.9941	0.9853	0.7897	0.7887
PSNR	39.74	33.39	30.95	30.82

4.3.3.4 Simulated data

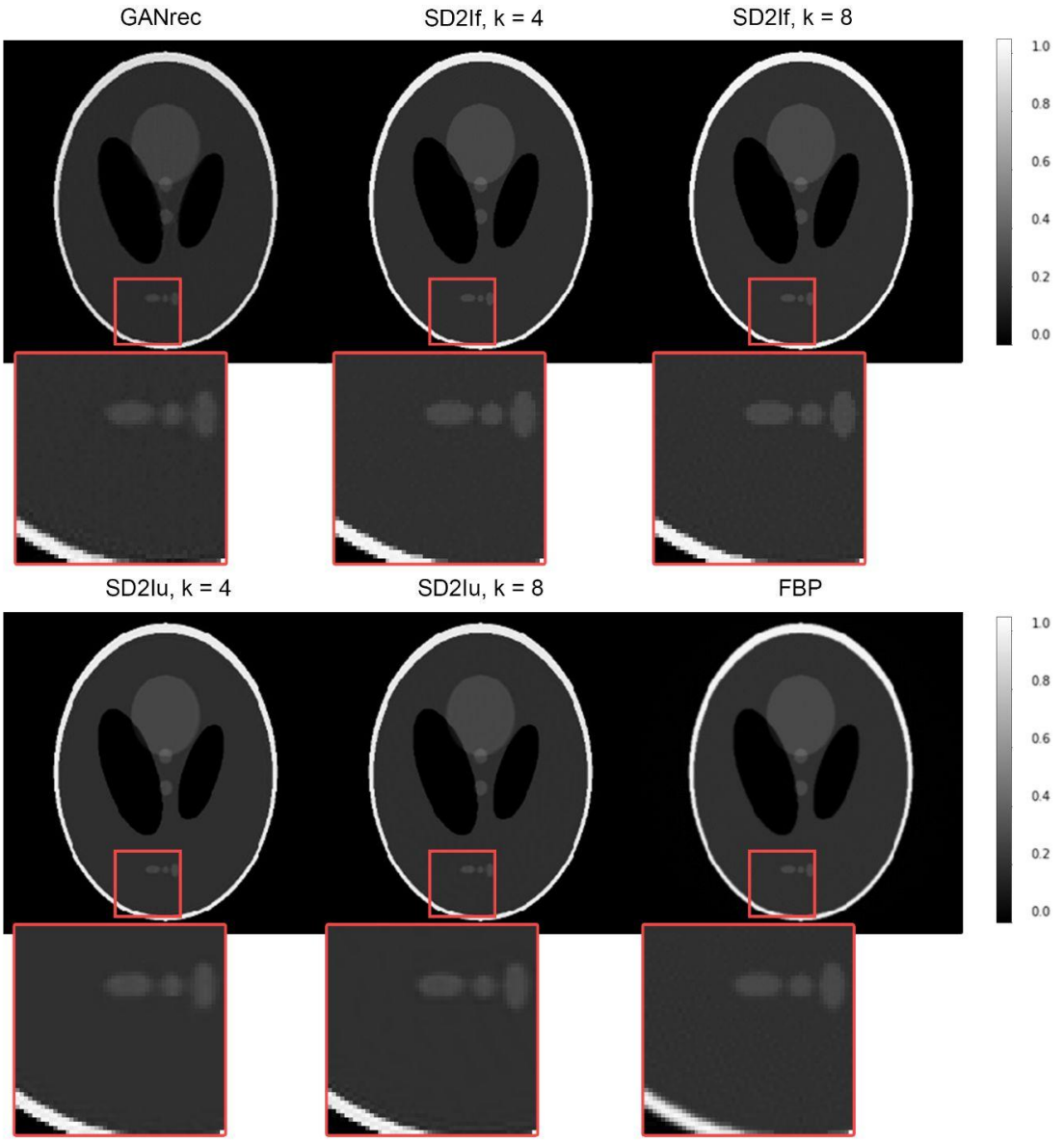

Figure 4.18: Comparison between the SD2I result and conventional reconstruction methods. The image size is 256 x 256 and reconstructed from the 256 x 400 Shepp-Logan sinogram. The gray scale bars indicate the signal strength for each pixel in the reconstructed images after normalization.

We start by comparing the performance of the new architecture against the FBP and other neural network-based reconstruction algorithms. For this comparison we use a sinogram created using the Shepp-Logan phantom with an image size of 256 x 256 pixels; the sinogram size is 256 x 400 pixels, corresponding to

detector elements and a number of projections respectively. The reconstructed images are presented in Figure 4.18 while Table 4.6 compares the results from the various reconstruction methods applied, using several common image quality metrics and specifically MAE, MSE, SSIM and PSNR. We find that all variants of our SD2I architecture outperform both GANrec and FBP across all metrics. The SD2I architectures that perform best are those where the convolutional part of the network is a single size, rather than including upsampling layers.

Table 4.6: Comparison of approaches for a 256 x 400 Shepp-Logan sinogram. Metrics were calculated using four significant figures for SSIM and PSNR and three significant figures for MAE and MSE. The bold text highlights the best scores achieved for each metric.

	GANrec	**SD2If (k = 4)**	**SD2If (k = 8)**	**SD2Iu (k = 4)**	**SD2Iu (k = 8)**	**FBP**
MAE	0.00484	0.00238	**0.00188**	0.00262	0.00247	0.00785
MSE	0.000191	0.000031	**0.000018**	0.000097	0.000087	0.000786
SSIM	0.9837	0.9933	0.9958	0.9958	**0.9968**	0.9614
PSNR	37.54	45.44	**48.26**	40.49	41.31	31.77

As we can see from the table, although the SD2Iu models perform worse on this full projection case, they still work better than the conventional FBP method. Changing the size of the final dense layer in the SD2I architecture (the factor k) has a small but appreciable effect on the image quality. Nonetheless, the main point is that all SD2I models shown in the table perform better than FBP on the standard Shepp-Logan phantom, regardless of architecture hyperparameters (within a reasonable range). The Adam optimiser was used as the optimisation algorithm [129], a combined MAE and SSIM loss function [130] was used and the learning rate was set to 0.0005 for networks presented in this work. The impact of the different loss functions is shown in Table 4.7. The learning rate was automatically reduced during training if the loss function was not decreasing after 300 iterations using a downscaling factor of 0.5 (Tensorflow ReduceLROnPlateau implementation [131]); 4000 epochs were used during the

image reconstruction process. It is important to note here that the various metrics provide only an indication of the image quality reconstruction and one should always inspect the resulting images regardless of the values of the various metrics.

4.3.3.5 Impact of loss function

The choice of loss functions has a great influence on the results' accuracy. Besides the traditional MAE or MSE function, we also introduced the SSIM loss function as many previous works suggested [5], [67]. Since the SSIM does not include the absolute difference among pixels, it is normally used as a joint loss function with MAE or MSE. Table 4.7 shows the accuracies of four reconstructed Shepp-Logan images from the same 512 x 128 sinogram with different loss functions.

Table 4.7: Comparison of approaches for a 512 x 128 Shepp-Logan sinogram with different loss functions.

	MSE	**MAE**	**MSE + SSIM**	**MAE +SSIM**
MAE	0.00346	0.00291	0.00337	0.00246
MSE	0.000241	0.000202	0.000212	0.000116
SSIM	0.9899	0.9937	0.9912	0.9952
PSNR	36.1844	36.9427	36.7385	39.3724

Our result shows that the joint loss function of MAE and SSIM works the best among all loss functions. However, the ratio between the two parts of the joint loss function should be carefully selected. Since we are keeping the relative intensities of the reconstructed images, the maximum value of the individual image and its MAE and MSE loss can be significantly different in order of magnitude. As such the ratios need to be adjusted for each reconstruction to keep a reasonable balance since whilst the SSIM loss is always between 0-1 the order of MAE is changing according to the signal strength.

4.3.3.6 Reconstruction Time

The reconstruction time is highly dependent on the number of parameters of the generator architecture and the image size. We examined the time taken by each generator on different image sizes. As shown in Table 4.8, the SD2I architectures work better than the GANrec and Automap when the image size is less than 512 x 512. However, when the image size is larger than 1024 x 1024, the improvement is not as significant. This is because the forward operator takes most of the time when the image is large, and it cannot be reduced by changing the generator architecture.

Table 4.8: Reconstruction times (time per epoch) for different image sizes (number of pixels).

Sinogram/ Image size (pixels)	**Automap (s)**	**GANrec (s)**	**SD2If (k = 8) (s)**	**SD2If (k = 4) (s)**	**SD2Iu (k = 8) (s)**	**SD2Iu (k = 4) (s)**
64 x 64	0.0131	0.0118	0.0081	0.0079	0.0110	0.0092
128 x 128	0.1285	0.0121	0.0103	0.0105	0.0136	0.0093
256 x 256	-	0.0220	0.0206	0.0176	0.0144	0.0137
512 x 512	-	0.1491	0.0940	0.0798	0.0653	0.06558
1024 x 1024	-	0.5835	0.6625	0.5962	0.5373	0.53529

The impact of the forward operator on the reconstruction time is examined in Table 4.9 where the image size is maintained whilst the number of projections in the sinogram increases. Since the generator sizes are not influenced by the sinogram size, and the time taken to calculate the loss function can be eliminated, we can clearly see that it is the Radon transformation that took most of the reconstruction time as the size of the sinogram increases.

Table 4.9: Impact of sinogram size (number of projections) for the SD2Iu (factor 8).

Sinogram size (pixels)	Number of parameters	Time per epoch (s)	Total reconstruction time (s)
512 x 64	4,455,873	0.03270	130.8
512 x 128	4,455,873	0.03518	140.72
512 x 256	4,455,873	0.04335	173.4
512 x 512	4,455,873	0.06534	261.36
512 x 1024	4,455,873	0.12548	501.92

4.3.3.7 Angular undersampling

A striking advantage of many deep learning based reconstruction approaches, when compared to traditional methods, such as FBP, is their ability to achieve high-quality reconstructions when only challenging data are practically available. These can be sinograms with angular undersampling, low signal-to-noise ratio or incomplete sinograms (e.g. not covering the full 0-180° angular range) [132], [133]. However, most of the approaches are applied on the FBP reconstructed images (i.e. post-processing of the reconstructed images) rather than performing directly the tomographic reconstruction and importantly rely on supervised learning which assumes: a) that artefact-free images (labelled data) are available and b) that the networks can generalise (e.g. train with non-scientific datasets typically used for developing neural networks and yield high-quality images when applied to experimental data). Unfortunately, these assumptions are rarely valid and the applicability of such networks to real experimental data is limited at best. Here, we show that SD2I, in addition to its ability to reconstruct large tomographic images in a self-supervised manner, is able to suppress the angular undersampling artefacts while performing the tomographic reconstruction. In Figure 4.19, we show the reconstruction of the Shepp-Logan phantom with severe angular undersampling where we have less than ¼ of the original sinogram projections (projections corresponding to ¼ of the detector elements). For comparison, also shown are the results obtained from the most often used iterative algorithms

(SART, CGLS and SIRT) using the Astra toolbox as well as from GANrec [118]. According to the determined loss functions, for all conventional reconstruction algorithms tested, SD2I produces results with significantly fewer artefacts and much closer to the ground truth reconstruction. Importantly, it is clearly shown that the SD2Iu, which corresponds to the smallest possible network in terms of the number of parameters, yields the best results. The use of the upsampling convolution layers actually improves the quality of the reconstruction, performing a function similar to denoising on the resultant images. It should be noted though that the network does not denoise the reconstructed images, it removes the angular undersampling artefacts. It, therefore, requires projection/sinogram data with a high signal-to-noise ratio; it does not lead to higher-quality reconstructed images than the FBP algorithm when the signal-to-noise ratio is low.

Table 4.10: Comparison of approaches for a 256 x 64 Shepp-Logan sinogram. 250 iterations were used for the SART, SIRT and CGLS algorithms. Metrics were calculated using four significant figures. The bold text marks the best scores achieved for each metric.

	MAE	**MSE**	**SSIM**	**PSNR**
FBP	0.01873	0.001400	0.6441	29.27
SART	0.01722	0.001834	0.7656	28.09
CGLS	0.01726	0.001693	0.7465	28.44
SIRT	0.01768	0.002298	0.8089	27.11
GANrec	0.01294	0.000912	0.8800	31.12
SD2If (k = 4)	0.00832	0.000398	0.9360	34.72
SD2If (k = 8)	0.00735	0.000318	0.9473	35.70
SD2Iu (k = 4)	0.00343	0.000171	0.9930	38.40
SD2Iu (k = 8)	**0.00315**	**0.000125**	**0.9941**	**39.74**

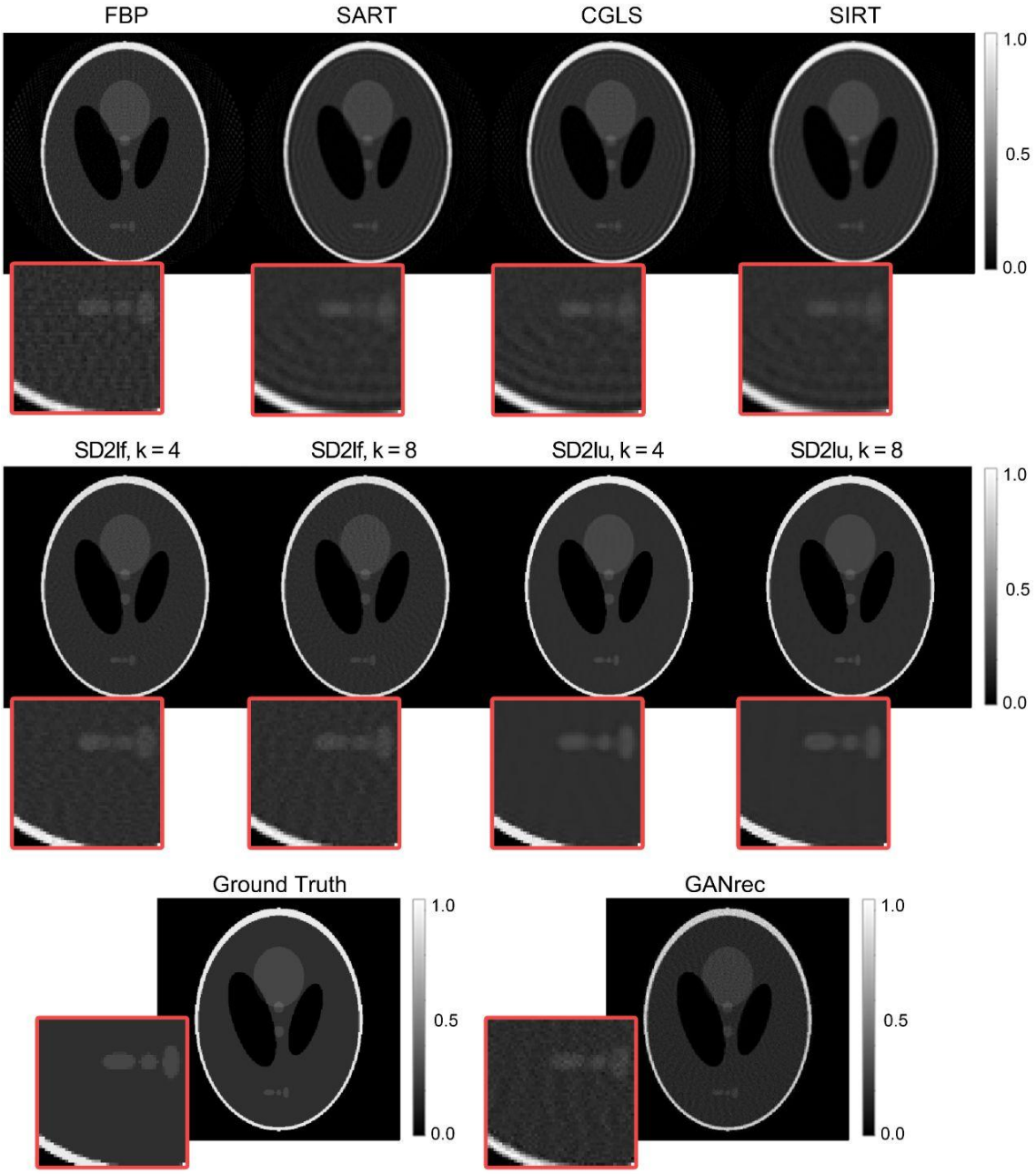

Figure 4.19: Comparison between conventional and neural network reconstruction approaches with different parameter settings. The image size is 256 x 256 and is reconstructed from the 256 x 64 Shepp-Logan sinogram. The gray scale bars indicate the signal strength for each pixel in the reconstructed images after normalization.

Table 4.10 also shows the performance of the FBP, SIRT, CGLS, SART, GANrec and various SD2I architectures on undersampled Shepp-Logan sinograms. These metrics confirm what is shown in the figures, with the SD2I outperforming other methods and the SD2I architecture with convolutional upsampling performing the best. The results in Figure 4.20 show that this

approach can be applied to larger image reconstruction tasks and the performance gains remain for SD2I. For calculating the SSIM and PSNR, we used the maximum possible pixel value as 1. A larger Shepp-Logan phantom image (512 x 512) was also tested (sinogram with size equal to 512 x 128) and the SD2I results are presented in Figure 4.20 and Table 4.11.

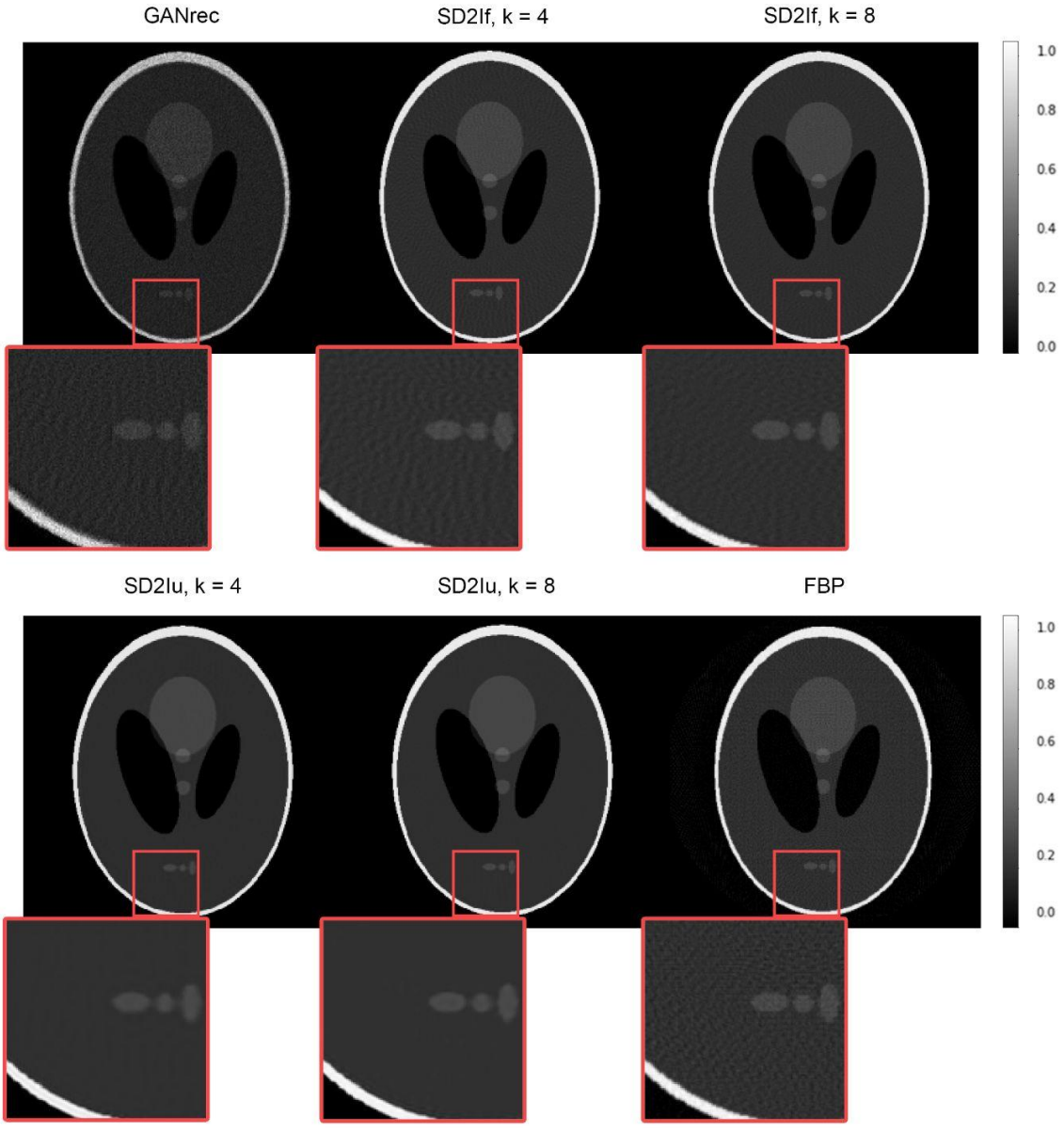

Figure 4.20: Comparison between the SD2I result and conventional reconstruction methods. The image size is 512 x 512 and reconstructed from the 512 x 128 Shepp-Logan sinogram. The gray scale bars indicate the signal strength for each pixel in the reconstructed images after normalization.

Table 4.11: Comparison of approaches for a 512 x 128 Shepp-Logan sinogram. The bold text indicates the best scores achieved on each metric.

	FBP	GANrec	SD2If (k = 8)	SD2If (k = 4)	SD2Iu (k = 8)	SD2Iu (k = 4)
MAE	0.01690	0.02589	0.00502	0.00596	**0.00246**	0.00254
MSE	0.000670	0.003531	0.000159	0.000207	**0.000116**	0.000136
SSIM	0.7123	0.6855	0.9614	0.94723	**0.9952**	0.9951
PSNR	31.5536	24.5201	37.9828	36.8326	**39.3724**	38.6760

It should be noted here that the results for the various metrics strongly depend on the choice of the ground truth image. This is not an issue for the Shepp-Logan phantom but it is a problem for the experimental data where there is no ground truth image available. This means that the quality of the reconstructed images has to be done primarily through visual inspection as the results from the various metrics can be misleading. To illustrate this problem, we measured the performance of SD2I as well as FBP, SART, SIRT and CGLS using different images as the ground truth image for the Shepp-Logan image (Figure 4.21, Tables 4.12 and S2.6 in Appendix 2).

If the FBP reconstructed image using the full projection set (400 projections) is used as the ground truth, then the metrics suggest that SIRT and CGLS outperform the SD2I. However, this is clearly not the case as shown in Figure 4.21 and from the fact that the clean (real ground truth) Shepp-Logan phantom image shows worse results for all metrics (Table 4.12). The result obtained with the CGLS method using the full projection set (400 projections) looks closer to the ground truth image compared to the FBP, SART and SIRT results obtained using the full projection set and for this reason, it is used as the ground truth for evaluating the performance of the SD2I network for the experimental data. Finally, it is important to note that when the clean Shepp-Logan image (real ground truth) or the CGLS image obtained using the full projection set is used as the ground truth, it can be seen that the SD2I with less than ¼ projections (64 projections) outperforms FBP, SART and SIRT reconstructions using the full

projection set (400 projections). This result further illustrates the accuracy of the SD2I reconstructions and the potential of this new network for data exhibiting angular undersampling.

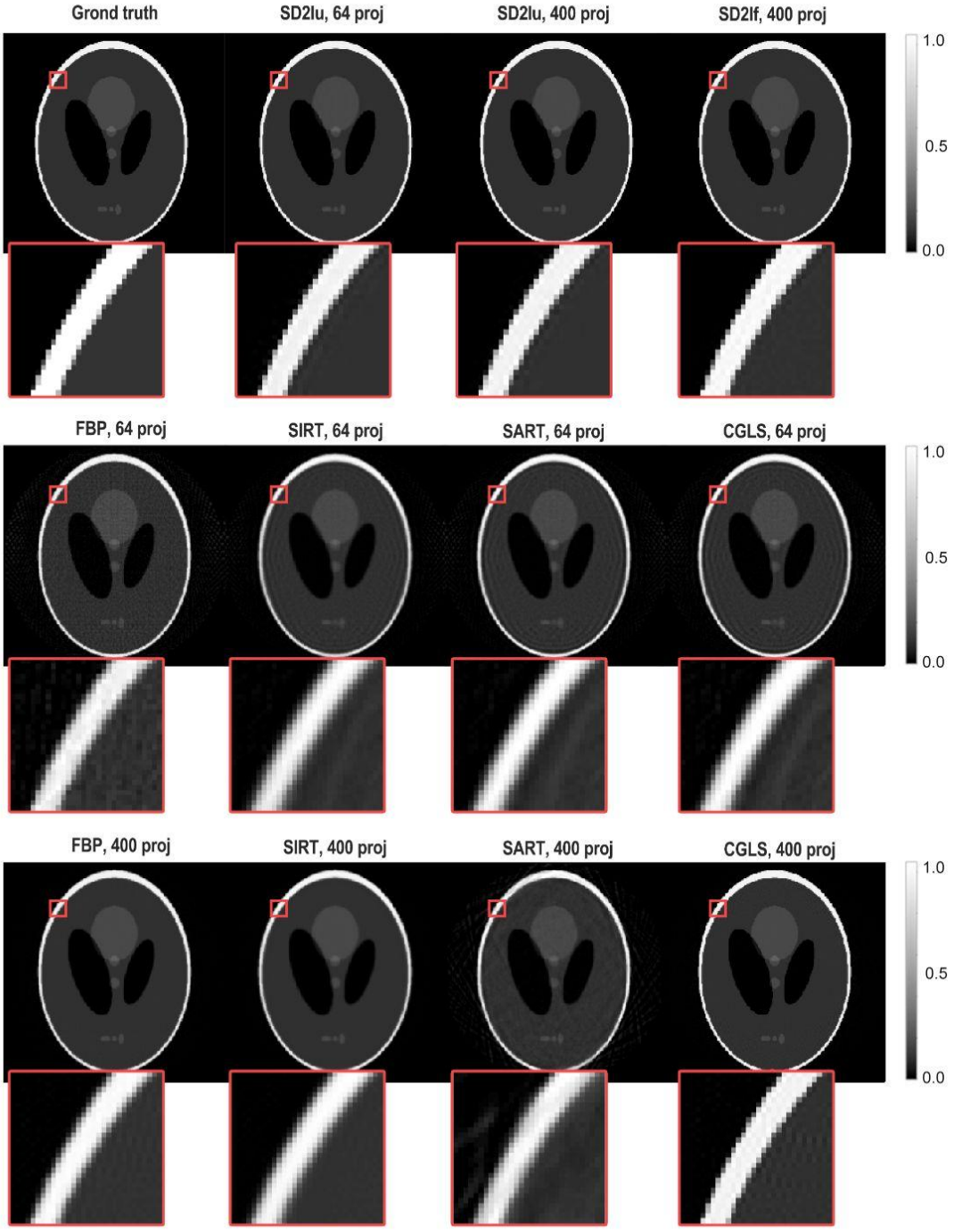

Figure 4.21: Comparison between the SD2I results and conventional reconstruction methods on 256 x 256 Shepp-Logan images with either 64 or 400 projections. All SD2I-based methods were generated using k = 8. The gray scale bars indicate the signal strength for each pixel in the reconstructed images after normalization.

Table 4.12: Comparison of the different reconstruction methods' performance with the reference of the clean Shepp-Logan image. The bold text indicates the best scores achieved for each metric.

		MAE	MSE	SSIM	PSNR
Clean Image as reference	**SD2Iu**	**0.00315**	**0.000125**	**0.9941**	**39.7403**
	FBP	0.01873	0.001400	0.6441	29.2653
	SIRT	0.01768	0.002298	0.8089	27.1113
	SART	0.01722	0.001834	0.7656	28.0914
	CGLS	0.01726	0.001693	0.7465	28.4376
FBP with 400 proj. as reference	**SD2Iu**	**0.00776**	0.000640	**0.9672**	32.6609
	FBP	0.01409	0.000598	0.6980	32.9616
	SIRT	0.01216	0.000682	0.8708	32.3845
	SART	0.01190	0.000501	0.8281	33.7261
	CGLS	0.01210	**0.000484**	0.8085	**33.8781**
	Clean image	0.00785	0.000786	0.9614	31.7733
CGLS with 400 proj. as reference	**SD2Iu**	**0.00533**	**0.000126**	**0.9498**	**39.7353**
	FBP	0.01836	0.001196	0.6908	29.9491
	SIRT	0.01787	0.002074	0.8499	27.5563
	SART	0.01709	0.001607	0.8169	28.6654
	CGLS	0.01723	0.001481	0.7960	29.0201
	Clean image	0.00444	0.000074	0.9480	42.0577

4.3.3.8 Why SD2I Works and Its Noise Performance

In this section, I will evaluate various architectural designs of SD2I and demonstrate how each component of SD2I enhances image results with angular under-sampling artifacts. Initially, I will explore the pixel learning method, which lacks neural network layers. This architecture, devoid of convolutional layers, employs the same training loop as SD2I. The map is initialized with random noise ranging from 0 to 1. The results are presented in Figure 4.22.

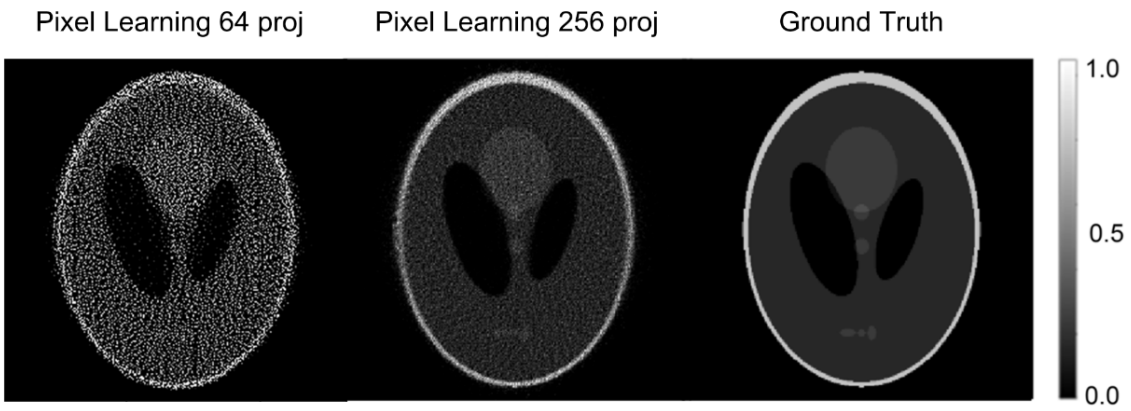

Figure 4.22: Images reconstructed from the 256 x 64 and 256 x 256 sinograms with the pixel learning method with the ground truth. The pixel learning leads to noisier images than the one reconstructed by the SD2I shown in Figure 4.19.

The architecture is trained using two sinograms: one with angularly undersampled 64 projections and the other with a full range of 256 projections. These outcomes are compared with the ground truth image. It becomes clear that the SD2I, compared to the pixel learning, can produce much more accurate and refined results as shown in Figure 4.19. This may be because the reconstructed image and the sinogram are not strictly one-to-one matched, implying numerous local minima during the training process. Therefore, certain measures must be implemented to encourage the network to view the image holistically rather than focusing on individual pixels to achieve a noise-free image.

I also tested two additional architectures on the SD2If framework. The first modification involved removing the initial few small fully connected layers, while the second modification entailed removing all convolutional layers. These two architectures are depicted in Figure 4.23 and were tested using full projection sizes from the 256 x 256 sinogram. The results of these tests are presented in Figure 4.24.

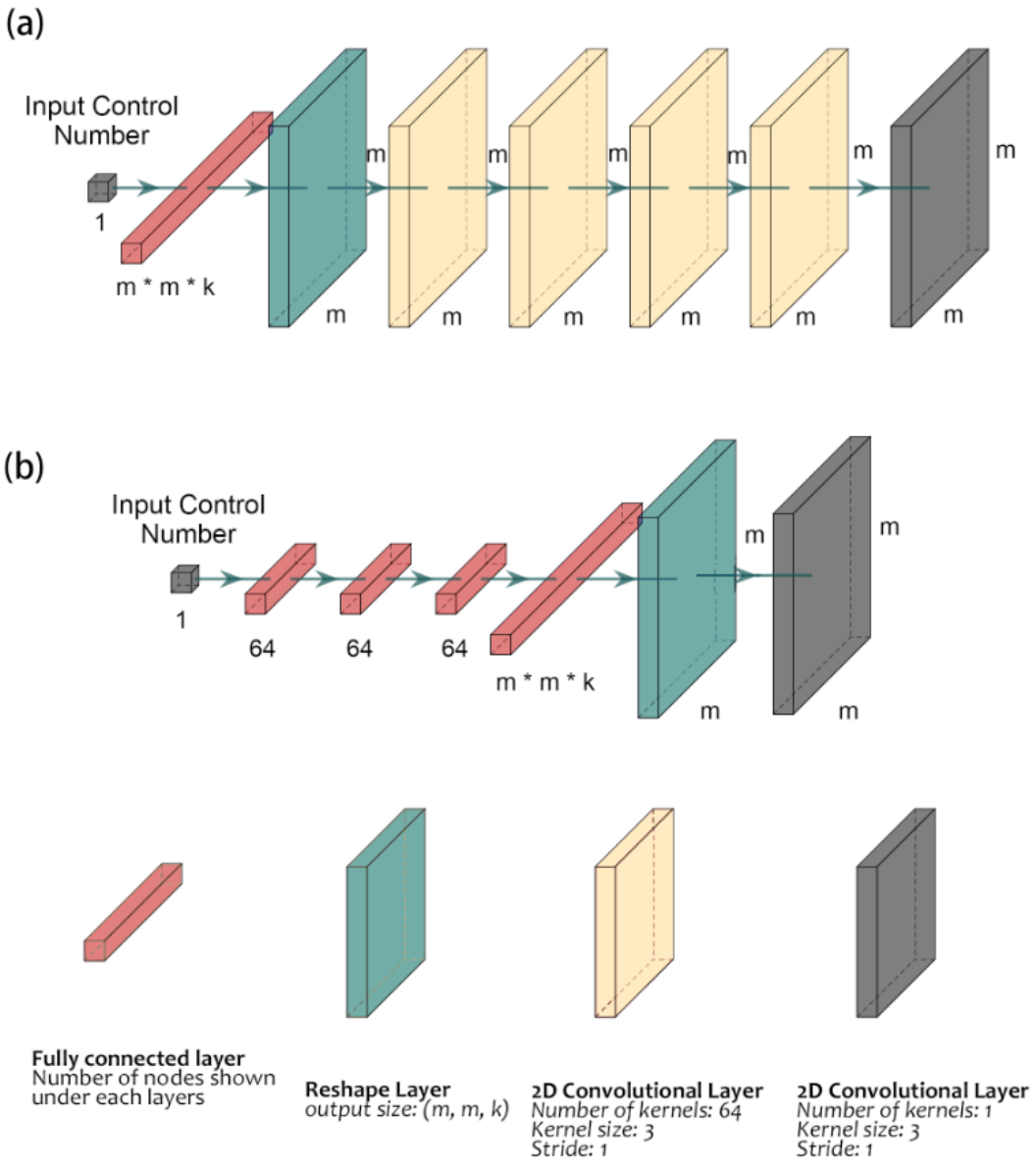

Figure 4.23: Two additional possible image reconstruction generator architectures: (a) SD2If with a single fully connected layer followed by multiple convolutional layers; (b) SD2If with only fully connected layers and a single convolutional layer as the output layer. The final convolutional layer serves to suppress the k features generated by the fully connected layer. In both architectures, k is set to 8.

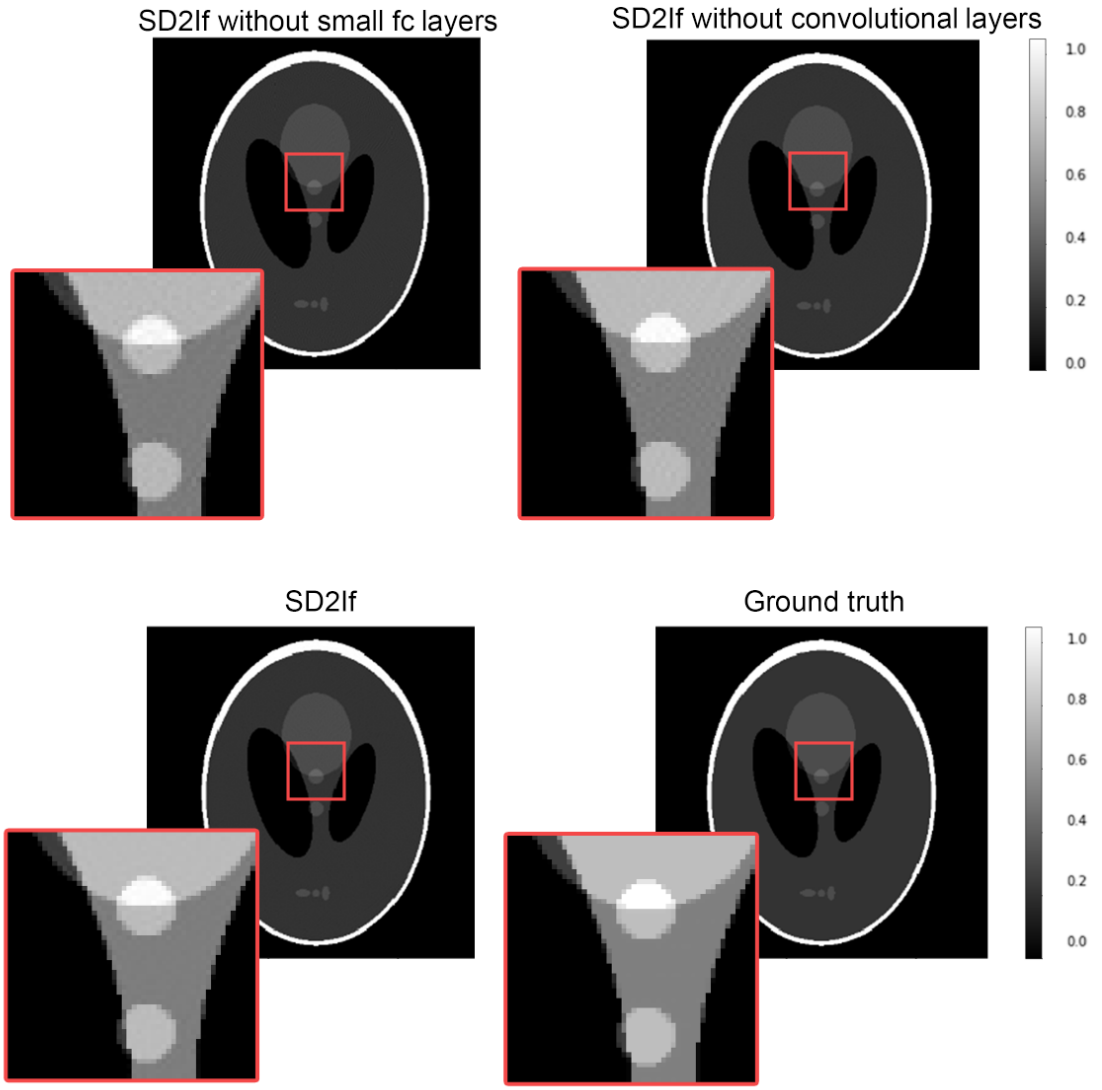

Figure 4.24: Results of the simulated Shepp-Logan images reconstructed using the two additional architectures shown in Figure 4.23, compared with the SD2If depicted in Figure 4.14(b), and the ground truth image. The zoomed-in sections are normalized to display a wider dynamic range, making more details visible. The input sinogram for all architectures is a 256 x 256 sinogram, and the reconstructed images are also 256 x 256 in size. The f factor is set to 8 for all architectures.

We can observe that, compared to Figure 4.22, both the initial small dense layers and the subsequent convolutional layers can to some extent reduce the noise in the reconstructed image. The current SD2I architecture, which utilizes both of these elements, has achieved the best results. This is because the initial few small fully connected layers ensure greater randomness in the network, where a slight change can alter all pixels. This forces the network to focus on a more general architecture and features during training. Convolutional layers are

inherently less sensitive to noise, allowing them to extract features from a noisy map and reconstruct a noiseless image. Therefore, the current SD2I architecture takes advantage of these two aspects to achieve the best results.

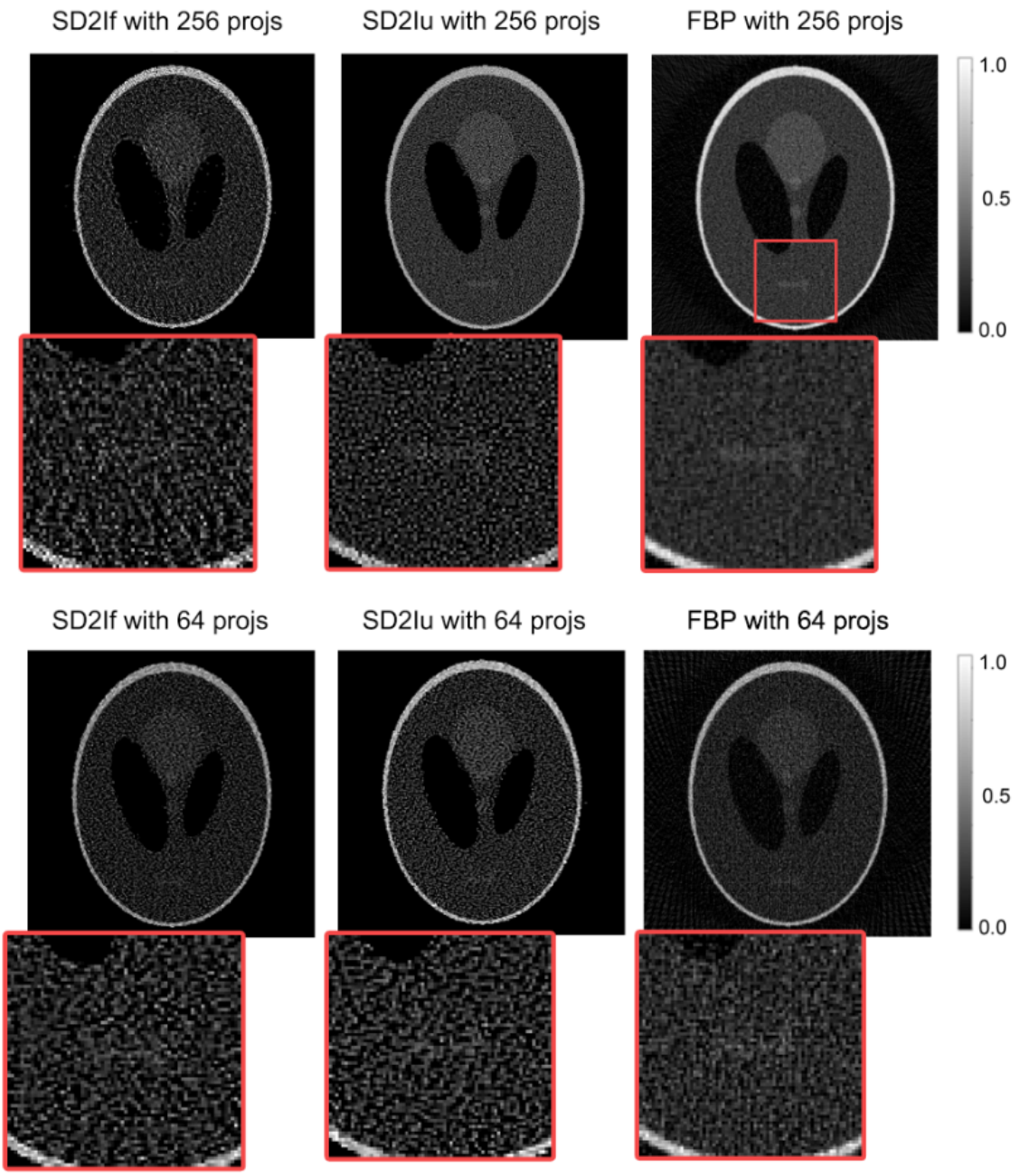

Figure 4.25: Image reconstructed by SD2If, SD2Iu, and FBP with 64 and 256 projections in the noisy sinograms with 256 translation steps. All images have the same size of 256 x 256. Both of SD2If and SD2Iu failed in this case while the FBP remained some features.

However, the biggest drawback of SD2I is its high sensitivity to noise in the sinogram. After artificially adding Poisson Noise to the sinogram, neither SD2Iu

nor SD2If can accurately reconstruct a noisy image as it should be. As previously mentioned, the design of SD2I forces the image to become easiser to converge to a local minima contains small features but not pixel-wise noise.

Figure 4.25 displays the comparison of reconstructions by SD2Iu and SD2If at 64 and 256 projections, respectively, against FBP. Since noise is added directly to the sinogram, obtaining a noisy ground truth image is impossible. Nevertheless, these images sufficiently demonstrate that SD2I loses some features and produces some unrealistic features during the reconstruction process, whereas FBP correctly reconstructs some features. Fortunately, in actual experiments, the noise levels in XRD-CT and micro-CT images are not as high as in the simulated data. With a minor noise level, SD2I architectures still produce more accurate images than traditional methods, as I will demonstrate in the next section.

4.3.3.9 Experimental data

We now turn our attention to testing the SD2I architecture on real experimental synchrotron X-ray tomography data. We obtain a ‘ground truth’ reconstruction in this case by reconstructing the images using CGLS with the full projection set. We then decrease the projection set to ¼ of the original size and compare the results of the reconstruction using CGLS, FBP, SART, SIRT and SD2I on the decreased sinogram. In Figure 4.26 we show the results for XRD-CT data, using two sinograms selected from two diffraction peaks of interest (i.e. NMC532 and Cu phases respectively). This XRD-CT dataset was acquired using a commercially available 10440 NMC532 Li-ion battery [127]. The ground truth image was obtained using the CGLS algorithm on the 547 x 400 sinograms which already have fewer projections (i.e. 400 projections) than the Nyquist sampling theorem dictates (i.e. $\pi/2$ x 547) [134]. All the reconstruction algorithms and neural networks were tested using 547 x 100 sinograms which are severely undersampled data.

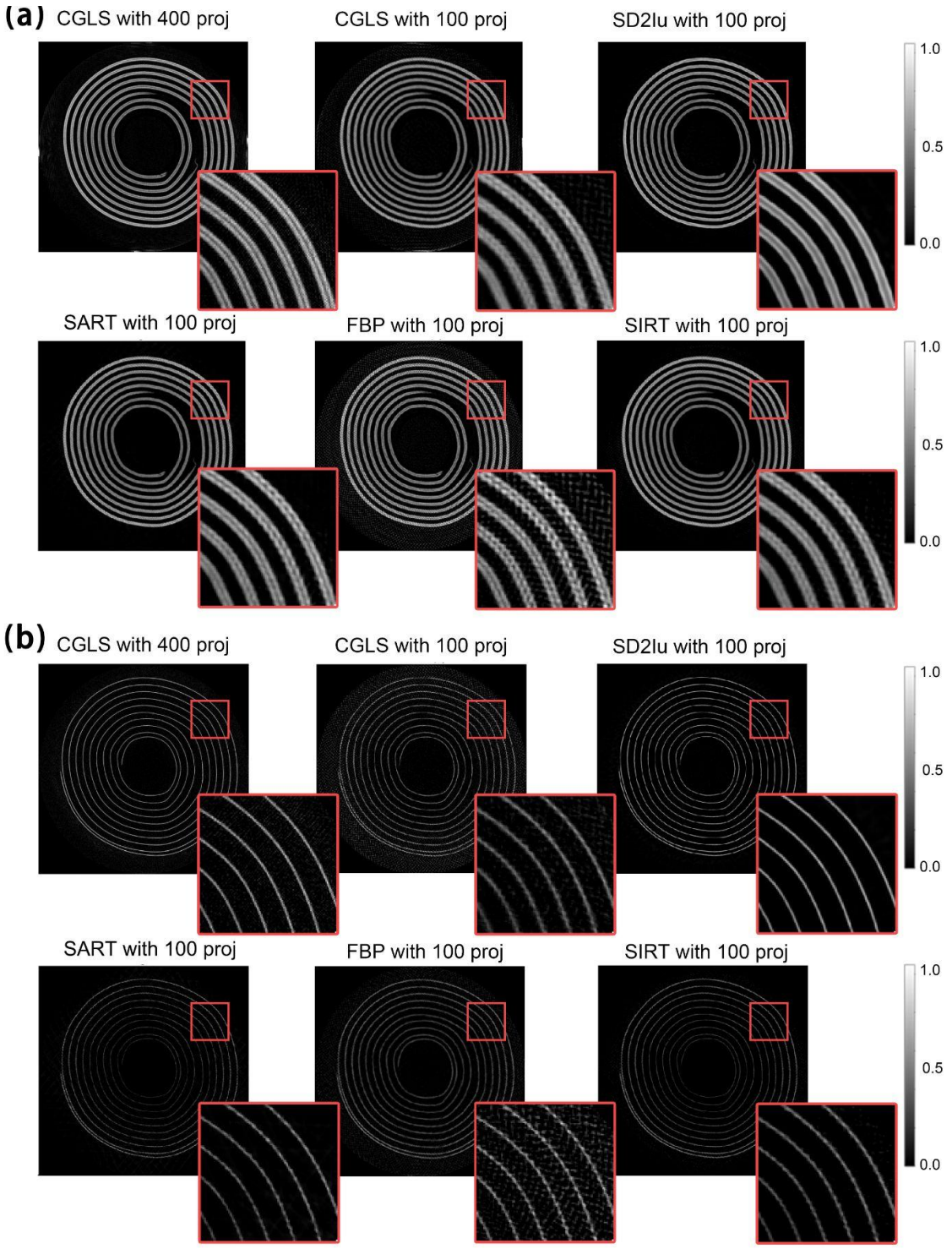

Figure 4.26: Two example XRD-CT reconstruction images: a) chemical image corresponding to the NMC532 phase, b) chemical image corresponding to the Cu phase. All SD2I results are produced using k factors equal to 8. The image sizes are 547 x 547. The SD2I and FBP results are reconstructed from the sinogram size of 547 x 100. The ground truth is obtained by the CGLS reconstruction of the 547 x 400 sinogram. The gray scale bars indicate the signal strength for each pixel in the reconstructed images after normalization.

As shown in Figure 4.26, both reconstructed images indicate that the SD2I reconstructions have suppressed the angular undersampling artefacts while these are clearly present in the traditional methods.

Table 4.13: Comparison of approaches for the example XRD-CT experimental images shown in Figure 4.26. The CGLS with 400 projections is considered as the ground truth. 250 iterations were used for the SART, SIRT and CGLS algorithms. Metrics were determined to four significant figures. The bold text marks the best scores achieved for each metric.

		FBP	SART	CGLS	SIRT	SD2Iu
(a)	**MAE**	0.3690	0.3011	0.3617	0.3270	**0.2599**
	MSE	0.4485	0.3216	0.3976	0.3648	**0.2564**
	SSIM	0.6662	0.6636	0.6841	0.6470	**0.6936**
	PSNR	23.18	24.62	23.70	24.07	**25.61**
(b)	**MAE**	0.7082	0.6030	0.6466	0.5697	**0.5694**
	MSE	1.809	1.922	1.631	1.770	**0.752**
	SSIM	0.5894	0.5460	0.6100	0.5074	**0.3251**
	PSNR	23.06	22.80	23.51	23.16	**26.88**

The metrics shown in Table 4.13 show that SD2I outperforms all other approaches but, as discussed previously, visual inspection and assessment of the reconstructed images is more important as there is no real ground truth image available for the experimental data. The hyperparameters for the SD2I networks used in this work were kept the same for all datasets and no tweaking was required (initial learning rate of 0.0005 with a decaying rate and a safe margin of 6000 epochs). This is another advantage of the network compared to iterative approaches such as SART, SIRT and CGLS where there is no standard loss function one can use to calculate the optimal number of iterations (convergence criterion), especially when trying to reconstruct different datasets. The visual results clearly demonstrate that the SD2I reconstructions are considerably better quality than all other methods (FBP, CGLS, SART and SIRT) on the undersampled sinogram. Finally, it should be noted that, although the images have been normalised for ease of visualisation/presentation, the

SD2I, in contrast to other neural network reconstruction methods such as GANrec, maintains the absolute intensity information which is essential in chemical tomography methods, such as XRD-CT.

Figure 4.26 and Table 4.13 present the results from the reconstructions of synchrotron X-ray micro-CT data acquired from the same 10440 NMC532 Li-ion battery corresponding to two different cross-sections. These two sinograms correspond to two different positions along the length of the battery (Figure 4.27); in position (a) only the Cu current collector is primarily visible in the battery jelly roll while in position (b) the NMC532 cathode can also be observed.

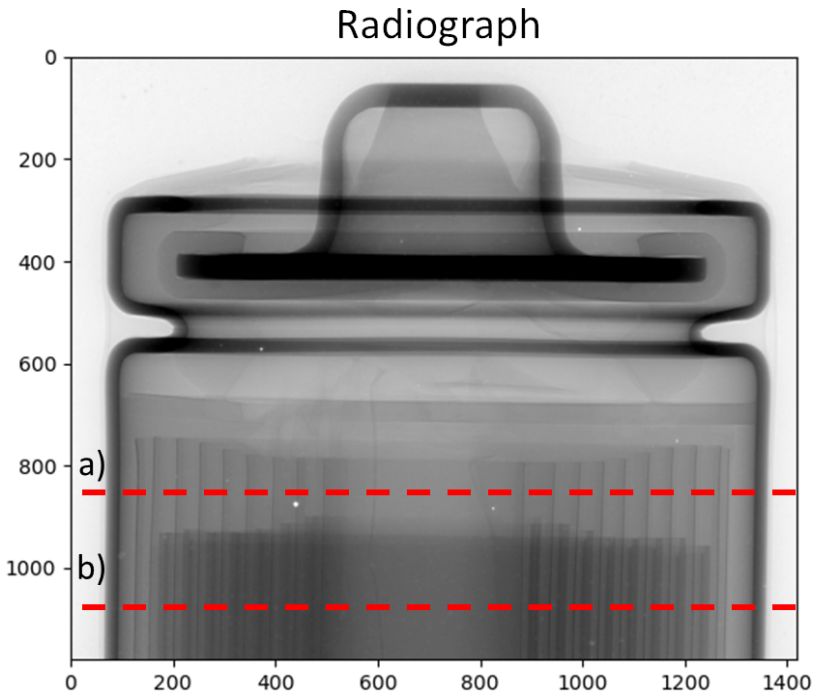

Figure 4.27. Normalised (dark current and flat field corrected) radiograph of the NMC532 Li-ion battery. Positions a) and b) correspond to the two sinograms (battery cross-sections) used in this work.

As with the XRD-CT data above, the ground truth is obtained by CGLS of a full projection and the sinogram is then decreased to ¼ of the original size and reconstructions obtained with FBP, SD2I as well as the SIRT, CGLS and SART iterative methods. As with the XRD-CT data, the SD2I reconstructions have

fewer artefacts than the images obtained with all other methods. It is important to note here the image size; the resulting images are 779 x 779 pixels. To the best of our knowledge, there is currently no other available self-supervised neural network that can perform direct reconstruction of such large sinograms/images without requiring a tremendous amount of GPU memory. Furthermore, in Figure 4.28 and Table 4.14 we also show that the SD2I is able to reconstruct images with 1559 x 1559 pixels which demonstrate the scalability of this new architecture.

Table 4.14: Comparison of approaches for the example micro-CT experimental images shown in Figure 4.28. The CGLS with 1561 projections is considered the ground truth. Metrics were calculated to four significant figures. The text in bold marks the best scores achieved for each metric.

		FBP	**SART**	**CGLS**	**SIRT**	**SD2Iu**
(a)	**MAE**	0.1668	0.1332	0.1608	0.1383	**0.0839**
	MSE	0.06497	0.06762	0.06295	0.07868	**0.01673**
	SSIM	0.6327	0.7861	0.6289	0.7643	**0.8377**
	PSNR	27.57	27.40	27.71	26.74	**33.46**
(b)	**MAE**	0.04751	0.04282	0.04880	0.04561	**0.03191**
	MSE	0.005068	0.005341	0.005101	0.006328	**0.002306**
	SSIM	0.6737	0.7614	0.6557	0.7304	**0.7897**
	PSNR	27.53	27.30	27.50	26.56	**30.95**

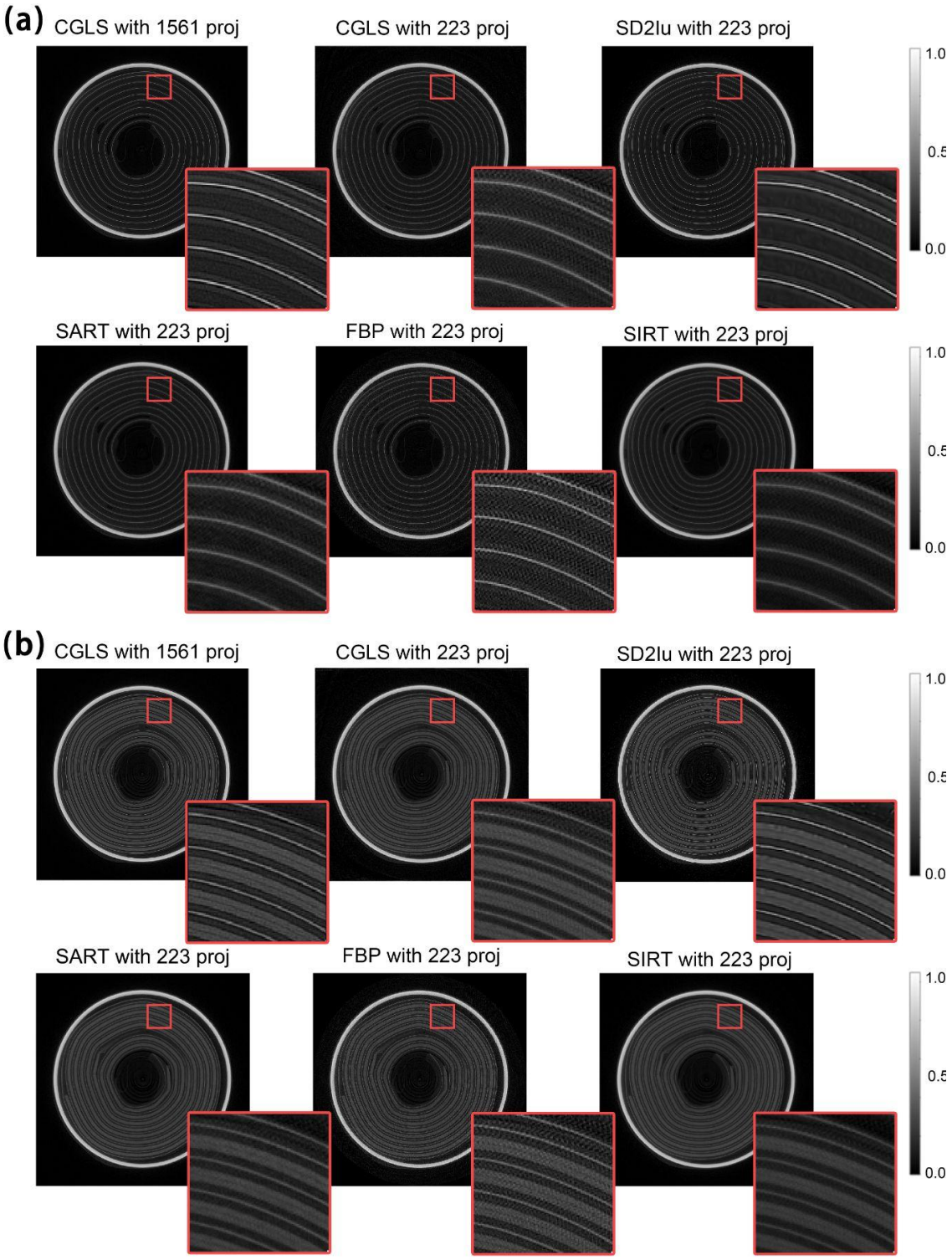

Figure 4.28: Two example micro-CT reconstruction images. All images shown in this figure comprise 779 x 779 pixels. The SD2Iu results are produced using k factors equal to 8. The SD2Iu, FBP, SART and SIRT results are reconstructed from a sinogram size of 779 x 223. The CGLS images reconstructed from 1561 projections are considered as the ground truth in calculating the metrics shown in Table 4.14. The gray scale bars indicate the signal strength for each pixel in the reconstructed images after normalization.

4.3.4 Summary and Conclusions

In this chapter, we have presented a new, self-supervised, deep-learning approach for tomographic reconstruction. The SD2I approach uses a generator network to produce a sample image, which is then converted to a sinogram via the Radon transform; the parameters of the network are updated by backpropagation to minimise the difference between the sample sinogram and the sinogram produced by the network. Similar to other deep-learning reconstruction approaches, our SD2I approach is much more robust to angular undersampling than traditional reconstruction approaches. However, SD2I is also considerably more computationally efficient than other deep-learning reconstruction methods. This means the SD2I can be applied to much larger sinograms and can produce results with a significantly lighter hardware requirement than other deep-learning approaches. The advantages of the new architecture can be summarised as the following:

- Scalability: Two new approaches in the architecture which radically reduce the number of parameters.
- Single-digit initial input Instead of using an FBP or sinogram as input to the generator, which can avoid using an encoder network and reduce the number of parameters.
- Upsampling-type architecture after the last dense layer - this allows for decreasing the number of neurons in the last dense layer by a factor of at least 16. This allows the network to reconstruct a much bigger image than the GANrec and AUTOMAP in the condition of the same number of training parameters in the network.
- Ability to suppress angular undersampling artefacts which we demonstrated using both simulated and experimental data.
- Information regarding absolute intensities are maintained; the images are not normalised.
- Simplicity: The addition of a discriminator network makes the training more complex and does not necessarily improve the resulting images.

In this chapter, both supervised learning and self-supervised learning ways to reconstruct CT images and also remove the angular undersampling artefacts with CNN and SD2I have been presented. Meanwhile, we showed how manifold learning can be applied to learn reduced dimensionality representations of the data, facilitating the generation of 2D maps of training and test data which allow us to know in advance if a new data point is likely to be well reconstructed by a pre-trained network.

Many papers have used GAN-like architectures to train both supervised and self-supervised learning models, but here we found that the discriminator does not significantly improve the results of SD2I. We also did some quick checks with the regional discriminator suggested in the paper by Liu et. al. [70], however, the results did not improve significantly. This might be because GANs and discriminators were originally invented to work with random input, without which the discriminator can easily overfit to a few pixels in the reconstructed image. As a result, after a few loops, the output of the discriminator will permanently lose its function as a loss function. The joint loss function of MAE and SSIM dominates the training process after the first few loops. Considering that the discriminator does not play any positive role in subsequent training, and in order to maintain its local minima, it introduces misleading information into the entire training process. Therefore, we found the image reconstructed by the GAN-like architecture is worse than the SD2I as shown in Figure 4.17.

However, the existing SD2I architecture is not very effective in solving the image reconstruction problem under conditions of high noise. As shown in Figure 4.25, when sinograms experience strong noise, both SD2Iu and SD2If are unable to accurately restore small features in the images, and their performance is even inferior to FBP. This may be due to its design, particularly the convolutional layers, which force the reconstructed image to have more small, block-like pixel groups, and are ineffective in expressing pixel-level features like noise. Additionally, the image and sinogram are not 1-to-1 matched, as expressed by the loss function. To effectively address this, it may

be necessary to add additional elements to the SD2I, such as a penalty function or mean kernel to reduce the noise level in the images, or to find a generator architecture that is more sensitive to noise.

The ability to accurately reconstruct images from sparsely-sampled sinograms is critical for time-resolved *in situ/operando* tomography experiments as well as for reducing X-ray doses in medical CT. In its current form, both supervised learning and self-supervised learning cannot be compared to FBP in terms of accuracy and speed, but we have demonstrated its potential to suppress angular undersampling using real experimental data. Furthermore, the SD2I network could potentially be applied to other tomographic methods and modalities, such as neutron tomography and X-ray fluorescence tomography. Last but not least, the network has been developed for tomographic image reconstruction using 2D parallel/pencil beam geometries but we can foresee its application for other inverse problems in imaging if the appropriate forward model is known, such as the parallax problem in XRD-CT.

However, the

4.4 Reference

[1] H. Dong *et al.*, ‘A scalable neural network architecture for self-supervised tomographic image reconstruction’, *Digit. Discov.*, p., 2023, doi: 10.1039/D2DD00105E.

[2] M. Diwakar and M. Kumar, ‘A review on CT image noise and its denoising’, *Biomed. Signal Process. Control*, vol. 42, pp. 73–88, Apr. 2018, doi: 10.1016/j.bspc.2018.01.010.

[3] S. Diederich and D. Wormanns, ‘Impact of low-dose CT on lung cancer screening’, *Lung Cancer*, vol. 45, pp. S13–S19, Aug. 2004, doi: 10.1016/j.lungcan.2004.07.997.

[4] S. Rob, T. Bryant, I. Wilson, and B. K. Somani, ‘Ultra-low-dose, low-dose, and standard-dose CT of the kidney, ureters, and bladder: is there a difference? Results from a systematic review of the literature’, *Clin. Radiol.*, vol. 72, no. 1, pp. 11–15, Jan. 2017, doi: 10.1016/j.crad.2016.10.005.

[5] S. Ghadrdan, J. Alirezaie, J.-L. Dillenseger, and P. Babyn, ‘Low-dose computed tomography image denoising based on joint wavelet and sparse representation’, in *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Aug. 2014, pp. 3325–3328. doi: 10.1109/EMBC.2014.6944334.

[6] W. Shao, J. Ni, and C. Zhu, ‘A Hybrid Method of Image Restoration and Denoise of CT Images’, in *2012 Sixth International Conference on Internet Computing for Science and Engineering*, Apr. 2012, pp. 117–121. doi: 10.1109/ICICSE.2012.58.
[7] H. Hanzouli *et al.*, ‘PET/CT image denoising and segmentation based on a multi observation and a multi scale Markov tree model’, in *2013 IEEE Nuclear Science Symposium and Medical Imaging Conference (2013 NSS/MIC)*, IEEE, 2013, pp. 1–4.
[8] A.-A. Zohair, A.-A. Shamil, and G. Sulong, ‘Latest methods of image enhancement and restoration for computed tomography: a concise review’, *Appl. Med. Inform.*, vol. 36, no. 1, pp. 1–12, 2015.
[9] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, ‘Image quality assessment: from error visibility to structural similarity’, *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.
[10] A. Hore and D. Ziou, ‘Image quality metrics: PSNR vs. SSIM’, in *2010 20th international conference on pattern recognition*, IEEE, 2010, pp. 2366–2369.
[11] Q. Huynh-Thu and M. Ghanbari, ‘Scope of validity of PSNR in image/video quality assessment’, *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, 2008.
[12] H. Lee, J. Lee, H. Kim, B. Cho, and S. Cho, ‘Deep-Neural-Network-Based Sinogram Synthesis for Sparse-View CT Image Reconstruction’, *IEEE Trans. Radiat. Plasma Med. Sci.*, vol. 3, no. 2, pp. 109–119, Mar. 2019, doi: 10.1109/TRPMS.2018.2867611.
[13] R. Tovey *et al.*, ‘Directional sinogram inpainting for limited angle tomography’, *Inverse Probl.*, vol. 35, no. 2, p. 024004, 2019.
[14] S. Li, Q. Cao, Y. Chen, Y. Hu, L. Luo, and C. Toumoulin, ‘Dictionary learning based sinogram inpainting for CT sparse reconstruction’, *Optik*, vol. 125, no. 12, pp. 2862–2867, 2014.
[15] H. Zhang *et al.*, ‘Computed tomography sinogram inpainting with compound prior modelling both sinogram and image sparsity’, *IEEE Trans. Nucl. Sci.*, vol. 63, no. 5, pp. 2567–2576, 2016.
[16] J. Gu, L. Zhang, G. Yu, Y. Xing, and Z. Chen, ‘X-ray CT metal artifacts reduction through curvature based sinogram inpainting’, *J. X-Ray Sci. Technol.*, vol. 14, no. 2, pp. 73–82, 2006.
[17] Y. Zhang, Y.-F. Pu, J.-R. Hu, Y. Liu, and J.-L. Zhou, ‘A new CT metal artifacts reduction algorithm based on fractional-order sinogram inpainting’, *J. X-Ray Sci. Technol.*, vol. 19, no. 3, pp. 373–384, 2011.
[18] Z. Li *et al.*, ‘A sinogram inpainting method based on generative adversarial network for limited-angle computed tomography’, in *15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, International Society for Optics and Photonics, 2019, p. 1107220.
[19] O. Elharrouss, N. Almaadeed, S. Al-Maadeed, and Y. Akbari, ‘Image inpainting: A review’, *Neural Process. Lett.*, vol. 51, no. 2, pp. 2007–2028, 2020.
[20] E. Xie, P. Ni, R. Zhang, and X. Li, ‘Limited-Angle CT Reconstruction with Generative Adversarial Network Sinogram Inpainting and Unsupervised

Artifact Removal', *Appl. Sci.*, vol. 12, no. 12, Art. no. 12, Jan. 2022, doi: 10.3390/app12126268.
[21] M. Katsura, J. Sato, M. Akahane, A. Kunimatsu, and O. Abe, 'Current and novel techniques for metal artifact reduction at CT: practical guide for radiologists', *Radiographics*, vol. 38, no. 2, pp. 450–461, 2018.
[22] L. Gjesteby *et al.*, 'Metal artifact reduction in CT: where are we after four decades?', *Ieee Access*, vol. 4, pp. 5826–5849, 2016.
[23] D. Giantsoudi *et al.*, 'Metal artifacts in computed tomography for radiation therapy planning: dosimetric effects and impact of metal artifact reduction', *Phys. Med. Biol.*, vol. 62, no. 8, p. R49, 2017.
[24] M. Bal and L. Spies, 'Metal artifact reduction in CT using tissue-class modeling and adaptive prefiltering', *Med. Phys.*, vol. 33, no. 8, pp. 2852–2859, 2006.
[25] Y. Zhang, L. Zhang, X. R. Zhu, A. K. Lee, M. Chambers, and L. Dong, 'Reducing metal artifacts in cone-beam CT images by preprocessing projection data', *Int. J. Radiat. Oncol. Biol. Phys.*, vol. 67, no. 3, pp. 924–932, 2007.
[26] H. Xue, L. Zhang, Y. Xiao, Z. Chen, and Y. Xing, 'Metal artifact reduction in dual energy CT by sinogram segmentation based on active contour model and TV inpainting', in *2009 IEEE Nuclear Science Symposium Conference Record (NSS/MIC)*, IEEE, 2009, pp. 904–908.
[27] Y. Zhang and H. Yu, 'Convolutional neural network based metal artifact reduction in x-ray computed tomography', *IEEE Trans. Med. Imaging*, vol. 37, no. 6, pp. 1370–1381, 2018.
[28] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, 'Image reconstruction by domain-transform manifold learning', *Nature*, vol. 555, no. 7697, pp. 487–492, Mar. 2018, doi: 10.1038/nature25988.
[29] A. F. Rodriguez, W. E. Blass, J. H. Missimer, and K. L. Leenders, 'Artificial neural network Radon inversion for image reconstruction.', *Med. Phys.*, vol. 28, no. 4, pp. 508–514, Apr. 2001, doi: 10.1118/1.1357222.
[30] P. Paschalis *et al.*, 'Tomographic image reconstruction using Artificial Neural Networks', *Nucl. Instrum. Methods Phys. Res. Sect. Accel. Spectrometers Detect. Assoc. Equip.*, vol. 527, no. 1, pp. 211–215, 2004, doi: https://doi.org/10.1016/j.nima.2004.03.122.
[31] M. Argyrou, D. Maintas, C. Tsoumpas, and E. Stiliaris, 'Tomographic Image Reconstruction based on Artificial Neural Network (ANN) techniques', in *2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC)*, 2012, pp. 3324–3327. doi: 10.1109/NSSMIC.2012.6551757.
[32] T. Würfl, F. C. Ghesu, V. Christlein, and A. Maier, 'Deep Learning Computed Tomography', in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds., Cham: Springer International Publishing, 2016, pp. 432–440.
[33] Y. Ge *et al.*, 'ADAPTIVE-NET: deep computed tomography reconstruction network with analytical domain transformation knowledge.', *Quant. Imaging Med. Surg.*, vol. 10, no. 2, pp. 415–427, Feb. 2020, doi: 10.21037/qims.2019.12.12.

[34] G. Ma, Y. Zhu, and X. Zhao, ‘Learning Image From Projection: A Full-Automatic Reconstruction (FAR) Net for Computed Tomography’, *IEEE Access*, vol. 8, pp. 219400–219414, 2020, doi: 10.1109/ACCESS.2020.3039638.
[35] M. T. McCann, K. H. Jin, and M. Unser, ‘Convolutional Neural Networks for Inverse Problems in Imaging: A Review’, *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 85–95, Nov. 2017, doi: 10.1109/MSP.2017.2739299.
[36] S. Bazrafkan, V. Van Nieuwenhove, J. Soons, J. De Beenhouwer, and J. Sijbers, ‘Deep Learning Based Computed Tomography Whys and Wherefores’, 2019.
[37] J. He, Y. Wang, and J. Ma, ‘Radon Inversion via Deep Learning’, *IEEE Trans. Med. Imaging*, vol. 39, no. 6, pp. 2076–2087, 2020.
[38] Y. Li *et al.*, ‘Strategy of computed tomography sinogram inpainting based on sinusoid-like curve decomposition and eigenvector-guided interpolation’, *JOSA A*, vol. 29, no. 1, pp. 153–163, 2012.
[39] Y. Chen *et al.*, ‘CT metal artifact reduction method based on improved image segmentation and sinogram in-painting’, *Math. Probl. Eng.*, vol. 2012, 2012.
[40] H. Song *et al.*, ‘Denoising of MR and CT images using cascaded multi-supervision convolutional neural networks with progressive training’, *Neurocomputing*, vol. 469, pp. 354–365, 2022.
[41] I. Domingues, G. Pereira, P. Martins, H. Duarte, J. Santos, and P. H. Abreu, ‘Using deep learning techniques in medical imaging: a systematic review of applications on CT and PET’, *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4093–4160, 2020.
[42] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, ‘Deep learning on image denoising: An overview’, *Neural Netw.*, vol. 131, pp. 251–275, 2020.
[43] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, ‘Convolutional neural networks: an overview and application in radiology’, *Insights Imaging*, vol. 9, pp. 611–629, 2018.
[44] S. M. Anwar, M. Majid, A. Qayyum, M. Awais, M. Alnowami, and M. K. Khan, ‘Medical image analysis using convolutional neural networks: a review’, *J. Med. Syst.*, vol. 42, pp. 1–13, 2018.
[45] M. M. Lell and M. Kachelrieß, ‘Recent and upcoming technological developments in computed tomography: high speed, low dose, deep learning, multienergy’, *Invest. Radiol.*, vol. 55, no. 1, pp. 8–19, 2020.
[46] K. P. Pruessmann, M. Weiger, P. Börnert, and P. Boesiger, ‘Advances in sensitivity encoding with arbitrary k-space trajectories’, *Magn Reson Med*, vol. 46, pp. 0740–3194, 2001, doi: 10.1002/mrm.1241.
[47] Q. Fan, T. Witzel, A. Nummenmaa, K. R. A. Van Dijk, and J. D. Van Horn, ‘MGH-USC Human Connectome Project datasets with ultra-high b-value diffusion MRI’, *NeuroImage*, vol. 124, pp. 1108–1114, 2016, doi: 10.1016/j.neuroimage.2015.08.075.
[48] A. E. Ilesanmi and T. O. Ilesanmi, ‘Methods for image denoising using convolutional neural network: a review’, *Complex Intell. Syst.*, vol. 7, no. 5, pp. 2179–2198, Oct. 2021, doi: 10.1007/s40747-021-00428-4.
[49] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ‘Imagenet classification with deep convolutional neural networks’, *Commun. ACM*, vol. 60, no. 6, pp.

84–90, 2017.
[50] A. Sengupta, Y. Ye, R. Wang, C. Liu, and K. Roy, ‘Going deeper in spiking neural networks: VGG and residual architectures’, *Front. Neurosci.*, vol. 13, p. 95, 2019.
[51] R. U. Khan, X. Zhang, and R. Kumar, ‘Analysis of ResNet and GoogleNet models for malware detection’, *J. Comput. Virol. Hacking Tech.*, vol. 15, no. 1, pp. 29–37, 2019.
[52] W. Yang *et al.*, ‘Improving low-dose CT image using residual convolutional network’, *Ieee Access*, vol. 5, pp. 24698–24705, 2017.
[53] X. Yin *et al.*, ‘Domain Progressive 3D Residual Convolution Network to Improve Low-Dose CT Imaging’, *IEEE Trans. Med. Imaging*, vol. 38, no. 12, pp. 2903–2913, Dec. 2019, doi: 10.1109/TMI.2019.2917258.
[54] O. Ronneberger, P. Fischer, and T. Brox, ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’, in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241.
[55] A. Cardona *et al.*, ‘An Integrated Micro- and Macroarchitectural Analysis of the Drosophila Brain by Computer-Assisted Serial Section Electron Microscopy’, *PLoS Biol.*, vol. 8, 2010.
[56] E. Shelhamer, J. Long, and T. Darrell, ‘Fully Convolutional Networks for Semantic Segmentation’, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, pp. 640–651, 2017.
[57] M. Allan *et al.*, ‘2018 Robotic Scene Segmentation Challenge’, *ArXiv*, vol. abs/2001.11190, 2020.
[58] M. Kalke and S. Siltanen, ‘Sinogram interpolation method for sparse-angle tomography’, *Appl. Math.*, vol. 2014, 2014.
[59] G. H. Weiss, A. J. Talbert, and R. A. Brooks, ‘The use of phantom views to reduce CT streaks due to insufficient angular sampling’, *Phys. Med. Biol.*, vol. 27, no. 9, p. 1151, 1982.
[60] H. Chen *et al.*, ‘Low-dose CT with a residual encoder-decoder convolutional neural network’, *IEEE Trans. Med. Imaging*, vol. 36, no. 12, pp. 2524–2535, 2017.
[61] Z. Liu, T. Bicer, R. Kettimuthu, D. Gursoy, F. De Carlo, and I. Foster, ‘TomoGAN: low-dose synchrotron x-ray tomography with generative adversarial networks: discussion’, *J. Opt. Soc. Am. A*, vol. 37, no. 3, p. 422, Feb. 2020, doi: 10.1364/josaa.375595.
[62] J. M. Wolterink, T. Leiner, M. A. Viergever, and I. Išgum, ‘Generative Adversarial Networks for Noise Reduction in Low-Dose CT’, *IEEE Trans. Med. Imaging*, vol. 36, no. 12, pp. 2536–2545, Dec. 2017, doi: 10.1109/TMI.2017.2708987.
[63] M. Sidorenko, D. Orlov, M. Ebadi, and D. Koroteev, ‘Deep learning in denoising of micro-computed tomography images of rock samples’, *Comput. Geosci.*, vol. 151, p. 104716, Jun. 2021, doi: 10.1016/j.cageo.2021.104716.
[64] Q. Yang, P. Yan, M. K. Kalra, and G. Wang, ‘CT Image Denoising with Perceptive Deep Neural Networks’. arXiv, Feb. 22, 2017. doi: 10.48550/arXiv.1702.07019.

[65] D. M. Pelt and K. J. Batenburg, ‘Fast Tomographic Reconstruction From Limited Data Using Artificial Neural Networks’, *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 5238–5251, 2013, doi: 10.1109/TIP.2013.2283142.
[66] D. M. Pelt and J. A. Sethian, ‘A mixed-scale dense convolutional neural network for image analysis’, *Proc. Natl. Acad. Sci.*, vol. 115, no. 2, pp. 254–259, 2018.
[67] G. Ding, Y. Liu, R. Zhang, and H. L. Xin, ‘A joint deep learning model to recover information and reduce artifacts in missing-wedge sinograms for electron tomography and beyond’, *Sci. Rep.*, vol. 9, no. 1, p. 12803, Sep. 2019, doi: 10.1038/s41598-019-49267-x.
[68] Z. Li *et al.*, ‘Promising Generative Adversarial Network Based Sinogram Inpainting Method for Ultra-Limited-Angle Computed Tomography Imaging’, *Sensors*, vol. 19, no. 18, p. 3941, Sep. 2019, doi: 10.3390/s19183941.
[69] J. Dong, J. Fu, and Z. He, ‘A deep learning reconstruction framework for X-ray computed tomography with incomplete data’, *PloS One*, vol. 14, no. 11, p. e0224426, 2019.
[70] Z. Liu, R. Kettimuthu, and I. Foster, ‘Masked Sinogram Model with Transformer for ill-Posed Computed Tomography Reconstruction: a Preliminary Study’. arXiv, Sep. 03, 2022. doi: 10.48550/arXiv.2209.01356.
[71] R. Krishnan, P. Rajpurkar, and E. J. Topol, ‘Self-supervised learning in medicine and healthcare’, *Nat. Biomed. Eng.*, pp. 1–7, 2022.
[72] S. Shurrab and R. Duwairi, ‘Self-supervised learning methods and applications in medical imaging analysis: A survey’, *PeerJ Comput. Sci.*, vol. 8, p. e1045, 2022.
[73] J. Xu, ‘A Review of Self-supervised Learning Methods in the Field of Medical Image Analysis’, *Int J Image Graph Signal Process.*, vol. 13, pp. 33–46, 2021.
[74] K. He, X. Zhang, S. Ren, and J. Sun, ‘Deep Residual Learning for Image Recognition’, *2016 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*, pp. 770–778, 2016.
[75] S. Laine, T. Karras, J. Lehtinen, and T. Aila, ‘High-quality self-supervised deep image denoising’, *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
[76] J. Xu, Y. Huang, L. Liu, F. Zhu, and L. Shao, ‘Noisy-asclean: Learning unsupervised denoising from the corrupted image’, *ArXiv Prepr. ArXiv190606878*, 2020.
[77] Z. Wang, N. Zou, D. Shen, and S. Ji, ‘Non-local u-nets for biomedical image segmentation’, in *Proceedings of the AAAI conference on artificial intelligence*, 2020, pp. 6315–6322.
[78] J. Lehtinen *et al.*, ‘Noise2Noise: Learning image restoration without clean data’, *ArXiv Prepr. ArXiv180304189*, 2018.
[79] Y. Quan, M. Chen, T. Pang, and H. Ji, ‘Self2Self With Dropout: Learning Self-Supervised Denoising From Single Image’, presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 1890–1898. Accessed: Nov. 04, 2022. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Quan_Self2Self_With_Dropout_Learning_Self-Supervised_Denoising_From_Single_Image_

CVPR_2020_paper.html
[80] A. Krull, T.-O. Buchholz, and F. Jug, ‘Noise2void-learning denoising from single noisy images’, in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 2129–2137.
[81] J. Batson and L. Royer, ‘Noise2Self: Blind Denoising by Self-Supervision’, in *Proceedings of the 36th International Conference on Machine Learning*, PMLR, May 2019, pp. 524–533. Accessed: Nov. 04, 2022. [Online]. Available: https://proceedings.mlr.press/v97/batson19a.html
[82] A. A. Hendriksen, D. M. Pelt, and K. J. Batenburg, ‘Noise2inverse: Self-supervised deep convolutional denoising for tomography’, *IEEE Trans. Comput. Imaging*, vol. 6, pp. 1320–1335, 2020.
[83] J. Lequyer, R. Philip, A. Sharma, W.-H. Hsu, and L. Pelletier, ‘A fast blind zero-shot denoiser’, *Nat. Mach. Intell.*, pp. 1–11, Oct. 2022, doi: 10.1038/s42256-022-00547-8.
[84] J. Johnson, A. Alahi, and L. Fei-Fei, ‘Perceptual losses for real-time style transfer and super-resolution’, in *European conference on computer vision*, Springer, 2016, pp. 694–711.
[85] K. Simonyan and A. Zisserman, ‘Very deep convolutional networks for large-scale image recognition’, *ArXiv Prepr. ArXiv14091556*, 2014.
[86] M. Li, W. Hsu, X. Xie, J. Cong, and W. Gao, ‘SACNN: Self-Attention Convolutional Neural Network for Low-Dose CT Denoising With Self-Supervised Perceptual Loss Network’, *IEEE Trans. Med. Imaging*, vol. 39, no. 7, pp. 2289–2301, Jul. 2020, doi: 10.1109/TMI.2020.2968472.
[87] Z. Zhang, S. Yu, W. Qin, X. Liang, Y. Xie, and G. Cao, ‘Self-supervised CT super-resolution with hybrid model’, *Comput. Biol. Med.*, vol. 138, p. 104775, 2021.
[88] K. Kim, S. Soltanayev, and S. Y. Chun, ‘Unsupervised Training of Denoisers for Low-Dose CT Reconstruction Without Full-Dose Ground Truth’, *IEEE J. Sel. Top. Signal Process.*, vol. 14, no. 6, pp. 1112–1125, Oct. 2020, doi: 10.1109/JSTSP.2020.3007326.
[89] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, ‘Unpaired image-to-image translation using cycle-consistent adversarial networks’, in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
[90] J. Harms *et al.*, ‘Paired cycle-GAN-based image correction for quantitative cone-beam computed tomography’, *Med. Phys.*, vol. 46, no. 9, pp. 3998–4009, 2019, doi: 10.1002/mp.13656.
[91] J. Gu and J. C. Ye, ‘AdaIN-Based Tunable CycleGAN for Efficient Unsupervised Low-Dose CT Denoising’, *IEEE Trans. Comput. Imaging*, vol. 7, pp. 73–85, 2021, doi: 10.1109/TCI.2021.3050266.
[92] J. Lee, J. Gu, and J. C. Ye, ‘Unsupervised CT Metal Artifact Learning Using Attention-Guided β-CycleGAN’, *IEEE Trans. Med. Imaging*, vol. 40, no. 12, pp. 3932–3944, Dec. 2021, doi: 10.1109/TMI.2021.3101363.
[93] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, ‘Cbam: Convolutional block attention module’, in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.
[94] H. Yang *et al.*, ‘Unpaired Brain MR-to-CT Synthesis Using a Structure-Constrained CycleGAN’, in *Deep Learning in Medical Image*

Analysis and Multimodal Learning for Clinical Decision Support, D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. S. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018, pp. 174–182. doi: 10.1007/978-3-030-00889-5_20.

[95] H. Yang *et al.*, 'Unsupervised MR-to-CT Synthesis Using Structure-Constrained CycleGAN', *IEEE Trans. Med. Imaging*, vol. 39, no. 12, pp. 4249–4261, Dec. 2020, doi: 10.1109/TMI.2020.3015379.

[96] B. Sun, S. Jia, X. Jiang, and F. Jia, 'Double U-Net CycleGAN for 3D MR to CT image synthesis', *Int. J. Comput. Assist. Radiol. Surg.*, Aug. 2022, doi: 10.1007/s11548-022-02732-x.

[97] A. Abu-Srhan, I. Almallahi, M. A. M. Abushariah, W. Mahafza, and O. S. Al-Kadi, 'Paired-unpaired Unsupervised Attention Guided GAN with transfer learning for bidirectional brain MR-CT synthesis', *Comput. Biol. Med.*, vol. 136, p. 104763, Sep. 2021, doi: 10.1016/j.compbiomed.2021.104763.

[98] L. Yu, Z. Zhang, X. Li, and L. Xing, 'Deep Sinogram Completion With Image Prior for Metal Artifact Reduction in CT Images', *IEEE Trans. Med. Imaging*, vol. 40, no. 1, pp. 228–238, Jan. 2021, doi: 10.1109/TMI.2020.3025064.

[99] L. Yu, Z. Zhang, X. Li, H. Ren, W. Zhao, and L. Xing, 'Metal artifact reduction in 2D CT images with self-supervised cross-domain learning', *Phys. Med. Biol.*, vol. 66, no. 17, p. 175003, Sep. 2021, doi: 10.1088/1361-6560/ac195c.

[100] C. Niu, M. Li, X. Guo, and G. Wang, 'Self-supervised dual-domain network for low-dose CT denoising', in *Developments in X-Ray Tomography XIV*, SPIE, Oct. 2022, pp. 65–71. doi: 10.1117/12.2633197.

[101] B. Zhou, X. Lin, and B. Eck, 'Limited Angle Tomography Reconstruction: Synthetic Reconstruction via Unsupervised Sinogram Adaptation', in *Information Processing in Medical Imaging*, A. C. S. Chung, J. C. Gee, P. A. Yushkevich, and S. Bao, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2019, pp. 141–152. doi: 10.1007/978-3-030-20351-1_11.

[102] G. Zang, R. Idoughi, R. Li, P. Wonka, and W. Heidrich, 'IntraTomo: Self-Supervised Learning-Based Tomography via Sinogram Synthesis and Prediction', presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 1960–1970. Accessed: Nov. 07, 2022. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2021/html/Zang_IntraTomo_Self-Supervised_Learning-Based_Tomography_via_Sinogram_Synthesis_and_Prediction_ICCV_2021_paper.html

[103] M. O. Unal, M. Ertas, and I. Yildirim, 'Self-Supervised Training For Low-Dose Ct Reconstruction', in *2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI)*, Apr. 2021, pp. 69–72. doi: 10.1109/ISBI48211.2021.9433944.

[104] K. Weiss, T. M. Khoshgoftaar, and D. Wang, 'A survey of transfer

learning', *J. Big Data*, vol. 3, no. 1, p. 9, May 2016, doi: 10.1186/s40537-016-0043-6.
[105] 'Deep learning for tomographic image reconstruction | Nature Machine Intelligence'. https://www.nature.com/articles/s42256-020-00273-z (accessed Dec. 12, 2020).
[106] K. S. H. Kulathilake, N. A. Abdullah, A. Q. M. Sabri, and K. W. Lai, 'A review on deep learning approaches for low-dose computed tomography restoration', *Complex Intell. Syst.*, pp. 1–33, 2021.
[107] B. Lakshminarayanan, A. Pritzel, and C. Blundell, 'Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles', in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 6402–6413. Accessed: Nov. 04, 2020. [Online]. Available: http://papers.nips.cc/paper/7219-simple-and-scalable-predictive-uncertainty-estimation-using-deep-ensembles.pdf
[108] J. Z. Liu, Z. Lin, S. Padhy, D. Tran, T. Bedrax-Weiss, and B. Lakshminarayanan, 'Simple and Principled Uncertainty Estimation with Deterministic Deep Learning via Distance Awareness', *ArXiv200610108 Cs Stat*, Oct. 2020, Accessed: Nov. 04, 2020. [Online]. Available: http://arxiv.org/abs/2006.10108
[109] J. van Amersfoort, L. Smith, Y. W. Teh, and Y. Gal, 'Simple and Scalable Epistemic Uncertainty Estimation Using a Single Deep Deterministic Neural Network', *ArXiv200302037 Cs Stat*, Mar. 2020, Accessed: Jun. 25, 2020. [Online]. Available: http://arxiv.org/abs/2003.02037
[110] Y. Gal and Z. Ghahramani, 'Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning', in *International Conference on Machine Learning*, PMLR, Jun. 2016, pp. 1050–1059. Accessed: Nov. 04, 2020. [Online]. Available: http://proceedings.mlr.press/v48/gal16.html
[111]K. Tran, W. Neiswanger, J. Yoon, Q. Zhang, E. Xing, and Z. W. Ulissi, 'Methods for comparing uncertainty quantifications for material property predictions', *Mach. Learn. Sci. Technol.*, vol. 1, no. 2, p. 025006, May 2020, doi: 10.1088/2632-2153/ab7e1a.
[112] B. Meredig *et al.*, 'Can machine learning identify the next high-temperature superconductor? Examining extrapolation performance for materials discovery', *Mol. Syst. Des. Eng.*, vol. 3, no. 5, pp. 819–825, 2018, doi: 10.1039/C8ME00012C.
[113] A. M. Beale, S. D. Jacques, E. K. Gibson, and M. Di Michiel, 'Progress towards five dimensional diffraction imaging of functional materials under process conditions', *Coord. Chem. Rev.*, vol. 277, pp. 208–223, 2014.
[114] A. M. Beale *et al.*, 'X-ray physico-chemical imaging during activation of cobalt-based Fischer–Tropsch synthesis catalysts', *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.*, vol. 376, no. 2110, p. 20170057, 2018.
[115] A. Vamvakeros *et al.*, '5D operando tomographic diffraction imaging of a catalyst bed', *Nat. Commun.*, vol. 9, p. 4751, 2018, doi: 10.1038/s41467-018-07046-8.
[116] V. Middelkoop *et al.*, '3D printed Ni/Al2O3 based catalysts for CO2

methanation - a comparative and operando XRD-CT study', *J. CO2 Util.*, vol. 33, pp. 478–487, 2019, doi: https://doi.org/10.1016/j.jcou.2019.07.013.
[117] A. Vamvakeros *et al.*, 'DLSR: a solution to the parallax artefact in X-ray diffraction computed tomography data', *J. Appl. Crystallogr.*, vol. 53, no. 6, Dec. 2020, doi: 10.1107/S1600576720013576.
[118] W. van Aarle *et al.*, 'The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography', *Ultramicroscopy*, vol. 157, pp. 35–47, 2015, doi: https://doi.org/10.1016/j.ultramic.2015.05.002.
[119] L. van der Maaten and G. Hinton, 'Visualizing Data using t-SNE', *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.
[120] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, 'Image quality assessment: from error visibility to structural similarity', *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.
[121] M. Wattenberg, F. Viégas, and I. Johnson, 'How to use t-SNE effectively', *Distill*, vol. 1, no. 10, p. e2, 2016.
[122] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, 'Deep learning for single image super-resolution: A brief review', *IEEE Trans. Multimed.*, vol. 21, no. 12, pp. 3106–3121, 2019.
[123] S. C. Park, M. K. Park, and M. G. Kang, 'Super-resolution image reconstruction: a technical overview', *IEEE Signal Process. Mag.*, vol. 20, no. 3, pp. 21–36, 2003.
[124] B. O. Leung and K. C. Chou, 'Review of super-resolution fluorescence microscopy for biology', *Appl. Spectrosc.*, vol. 65, no. 9, pp. 967–980, 2011.
[125] E. Agustsson and R. Timofte, 'Ntire 2017 challenge on single image super-resolution: Dataset and study', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135.
[126] R. Timofte *et al.*, 'NTIRE 2017 Challenge on Single Image Super-Resolution: Methods and Results', in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jul. 2017.
[127] A. Vamvakeros *et al.*, 'Cycling Rate-Induced Spatially-Resolved Heterogeneities in Commercial Cylindrical Li-Ion Batteries', *Small Methods*, vol. 5, no. 9, p. 2100512, 2021, doi: https://doi.org/10.1002/smtd.202100512.
[128] 'A flexible, scalable generative network for self-supervised tomographic image reconstruction | Materials Science | ChemRxiv | Cambridge Open Engage'. https://chemrxiv.org/engage/chemrxiv/article-details/633810080e3c6abe842f7403 (accessed May 09, 2023).
[129] D. P. Kingma and J. Ba, 'Adam: A Method for Stochastic Optimization'. arXiv, 2014. doi: 10.48550/ARXIV.1412.6980.
[130] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, 'Loss Functions for Neural Networks for Image Processing'. arXiv, 2015. doi: 10.48550/ARXIV.1511.08861.
[131] Martín Abadi *et al.*, 'TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems'. 2015. [Online]. Available:

https://www.tensorflow.org/
[132] H.-M. Zhang and B. Dong, ‘A Review on Deep Learning in Medical Image Reconstruction’, *J. Oper. Res. Soc. China*, vol. 8, no. 2, pp. 311–340, Jan. 2020, doi: 10.1007/s40305-019-00287-4.
[133] X. Yang *et al.*, ‘Tomographic reconstruction with a generative adversarial network’, *J. Synchrotron Radiat.*, vol. 27, no. 2, pp. 486–493, Mar. 2020, doi: 10.1107/S1600577520000831.
[134] H. J. Landau, ‘Sampling, data transmission, and the Nyquist rate’, *Proc. IEEE*, vol. 55, no. 10, pp. 1701–1706, Oct. 1967, doi: 10.1109/PROC.1967.5962.

Chapter 5: Self-Supervised Parallax Artefact Removal

5.1 Abstract

In this chapter, I will introduce an innovative method, ParallaxNet, designed to eliminate parallax artefacts present in X-ray powder diffraction computed tomography data acquired from large samples. Based on the SigleDigit2Comp (SD2I) architecture introduced in the previous chapter, ParallaxNet integrates a unique 3D neural network architecture with a forward projector that accounts for the experimental geometry. This self-supervised technique for tomographic volume reconstruction is designed to be chemistry-agnostic, eliminating the need for prior knowledge of the sample's chemical composition. At the time of this writing, the work presented in this chapter is prepared and ready for submission to a peer-reviewed journal.

ParallaxNet is faster and more scalable compared to the pure mathematical solution, DLSR [1]. We showcase the efficacy of this method through its application on both simulated and experimental X-ray diffraction tomography data, acquired from a phantom sample and a commercially available and industrially relevant NMC532 cylindrical Lithium-ion battery.

5.2 Introduction

Deep learning, an advanced subset of machine learning, has been a game-changer across a diverse array of fields, including image recognition and text translation [2]–[5]. Unlike traditional 'hand-crafted' algorithms that operate on fixed principles, deep learning harnesses flexible neural networks that evolve based on exposure to existing examples. This dynamic, data-driven, learning process allows deep learning models to continually refine their performance, driving significant advancements in complex tasks where flexibility and adaptability are key.

One of the key areas in which deep learning has made a significant impact is in the field of tomographic image reconstruction [6], [7]. Traditionally, tomographic image reconstruction has relied on direct methods like filtered back projection (FBP) or iterative methods that depend on prior knowledge and fine-tuning. However, these methods face their own limitations, especially when it comes to scalability, handling noise and angular undersampling data, computational demand, and the necessity for absolute values in certain applications [8]–[10]. Deep neural networks (DNNs) have emerged as a compelling solution, offering the potential to surpass the performance of these traditional physics-based approaches.

In recent years, there has been a burgeoning interest in the application of DNNs in tomography, notably in enhancing the quality of real-space reconstructed images generated from sinograms. Besides, some innovative applications even leverage supervised learning and generative models to automatically map from sinogram to real space [11]–[18]. Despite certain bottlenecks such as handling large images and the computational cost of large networks, the promise of deep learning in this sphere is quite palpable.

The advent of X-ray (powder) diffraction computed tomography (XRD-CT), a specialised form of tomography, has added a new dimension to the mix. This

technique uses a pencil beam scanning method to yield reconstructed images corresponding to a sample's cross-section [19]–[22]. What sets XRD-CT apart is its ability to resolve chemical species of similar density, a task that conventional X-ray CT often struggles with. As such, XRD-CT has found applications in various fields ranging from cultural heritage conservation to nuclear reactor materials, as well as biological samples [23]–[26]. More importantly, XRD-CT has become an invaluable tool to investigate non-destructively functional materials and devices, such as catalytic reactors [27]–[33], fuel cells [34], [35] and secondary/rechargeable batteries in custom-made laboratory cells [35]–[40] as well as in commercially available and industrially relevant cylindrical form [41]–[43], under operating conditions (*in situ* /*operando* studies). These studies have shown that the spatially-resolved diffraction patterns in XRD-CT data can yield unique physicochemical information regarding these complex materials systems and their evolving solid-state chemistry. Recently, Due to advancements in XRD-CT experimental methods, rapid XRD-CT scans have now made it possible for us to track reactions in materials within a time frame of tens to several tens of minutes [44]–[46].

Given the prowess of deep learning and the unique capabilities of XRD-CT, the combination of these two could potentially revolutionize tomographic image reconstruction. Deep learning methods could not only accelerate the XRD-CT on both data acquisition and analysis but also enhance it by addressing challenges like image super-resolution using high-resolution region-of-interest CT scans, data denoising, as well as single-crystal diffraction, self-absorption and parallax artefacts. This combination, if realised, could unlock new possibilities, including higher spatial and temporal resolution in chemical imaging and better handling of complex data sets, paving the way for breakthroughs in various fields.

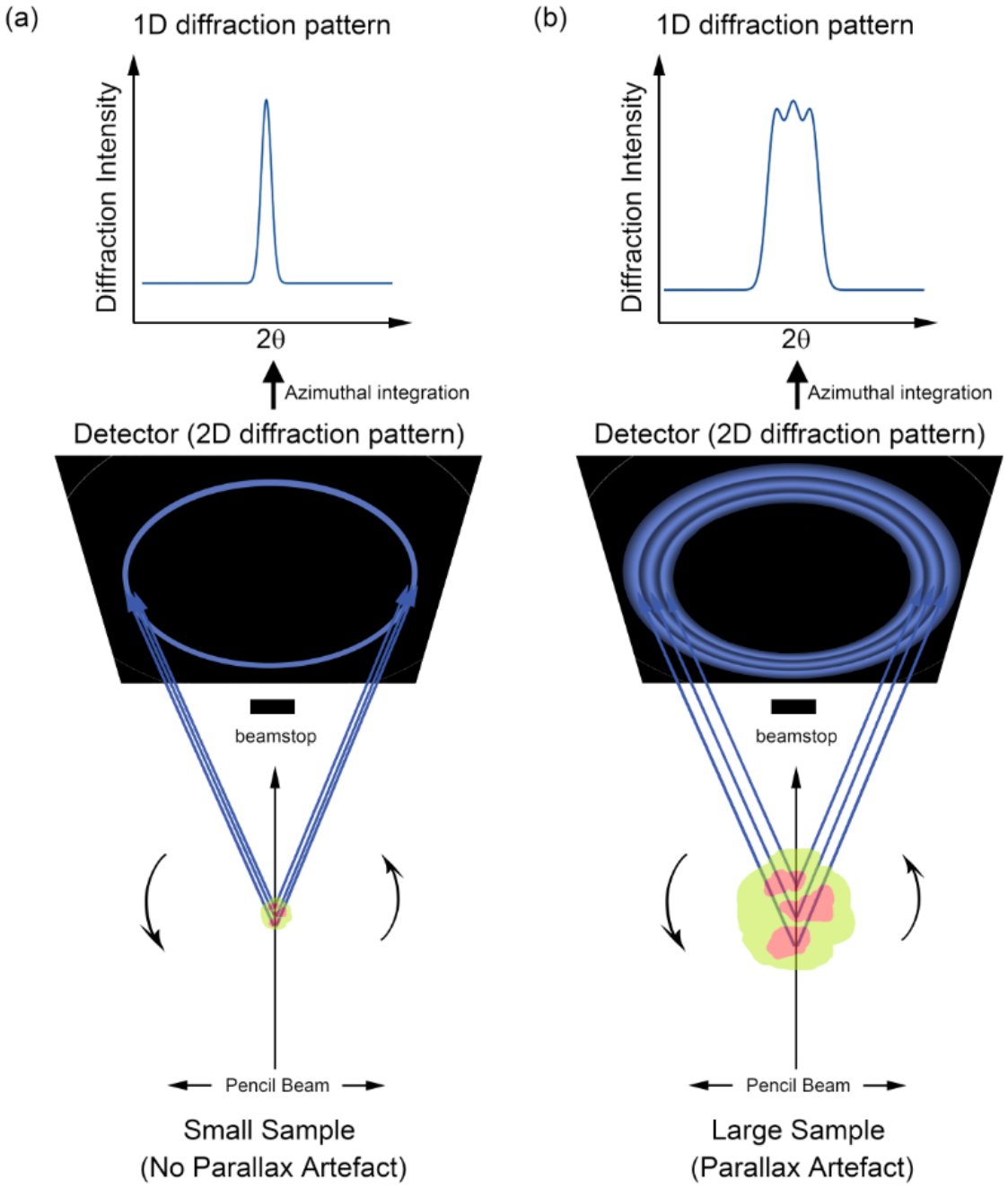

Figure 5.1: (a) Schematic representation of a 2D XRD pattern collected during the XRD-CT scanning of a small sample when there is no parallax artefact. (b) 2D XRD pattern collected during the XRD-CT scanning of a large sample with parallax artefacts present; the X-rays scattered/diffracted along the sample at certain 2θ angles arrive at different detector elements, leading to peak broadening and peak splitting.

One major obstacle that prohibits the scale-up of the XRD-CT technique and its widespread adoption to study large samples is the parallax artefact. In scattering-based CT experiments, it is generally assumed that the X-rays,

whether scattered or diffracted, arrive at the same detector element when measured at any given scattering angle 2θ across the sample's thickness, as depicted in Figure 5.1a. This assumption holds when the sample thickness is relatively small, typically on the order of a few millimetres. However, for thicker samples, this assumption becomes invalid. In these cases, diffracted X-rays measured at a specific 2θ angle are detected by multiple detector elements due to the significantly varying distances between elements within the sample and the detector. This phenomenon, known as the parallax effect, exhibits a $\tan(2\theta)$ dependency [19], [22]. The parallax effect is further illustrated in Figure 5.1b. As a result of this effect, artefacts may manifest as shifts in peak position, peak broadening, or even peak splitting [47].

In our previous work, we developed a new reconstruction approach, termed the “direct least-squares reconstruction” (DLSR) algorithm, which overcomes the parallax artefact in XRD-CT data. Conventionally, the XRD-CT sinogram data are reconstructed one by one, typically using the filtered back projection algorithm, yielding an XRD-CT reconstructed volume. The next step involves the analysis of all the local diffraction patterns in this reconstructed XRD-CT volume which can be single, multi-peak fitting or full profile analysis using methods such as LeBail or Rietveld. The DLSR was implemented using the TOPAS software [48] and combines the reconstruction and full profile analysis steps into a single step. To clarify, the sinogram XRD-CT (projection) data are fitted and the results are real-space maps corresponding to the various properties of the model that are being refined (e.g. scale factor, lattice parameter and crystallite size maps for each phase). This approach yields parallax artefact-free images but has severe limitations:

- It requires *a priori* knowledge about the chemistry of the sample before reconstruction
- It requires the construction of a robust physical model that models all chemistry accurately in the sinogram data; minor components being

overlooked during the inspection of diffraction patterns when preparing the physical model will not be part of the final results

- DLSR in its TOPAS implementation suffers from scalability; even XRD-CT images that are nowadays considered standard (e.g. 256 x 256) cannot be handled due to RAM requirements and the data have to be rebinned losing spatial resolution.
- Typically requires laborious pre-processing to decrease memory requirements (to that realistically available) and yield stable reconstructions. For example, one needs to create a separate binary mask for each crystalline phase present by analysing the FBP reconstructed XRD-CT volume (which contains parallax artefacts) and/or subtract the background from the sinogram data (in order to make it linear/ use a simple background model)

Therefore, our motivation was to develop a new method that overcomes all these limitations of the DLSR approach and yields parallax artefact-free XRD-CT images.

5.2.1 Self-supervised parallax artefact removal

We developed a self-supervised parallax XRD-CT data reconstruction architecture by integrating a forward operator that can transfer an XRD-CT volume without parallax artefacts to the sinograms with parallax artefacts. A schematic representation of the method is shown in Figure 5.2. We use an artificial neural network which acts as an XRD-CT volume generator i.e. it creates a stack of real-space XRD-CT images. The input to the generator is a non-zero constant. Next, the generated images are converted into sinograms with the addition of parallax artefacts using a differentiable forward operator. The forward operator contains two parts, the first part adds artificial parallax artefacts into the images by taking into account the geometry of the experimental setup, and the second part is the Radon transformation that can convert the images to sinograms. This generated XRD-CT sinogram volume is then compared with the experimental sinogram dataset using a designated loss

function. Based on this comparison, the weights of the generator network are updated accordingly.

Building on the innovative SD2I architecture shown in Chapter 4 [6], which is a lightweight and scalable CNN architecture that utilizes a single control number input for CT image reconstruction and addresses angular undersampling artefacts, we introduce a new generator architecture for enhanced diffraction channel stack reconstructions. By transitioning from 2D to 3D convolutional layers and reducing layer parameters, the new generator architecture offers both tailored 3D capability and greater efficiency, marking a significant step forward in CT image reconstruction. Figure 5.3 illustrates the design of the generator network employed for 3D diffraction channel stack reconstruction from the sinograms.

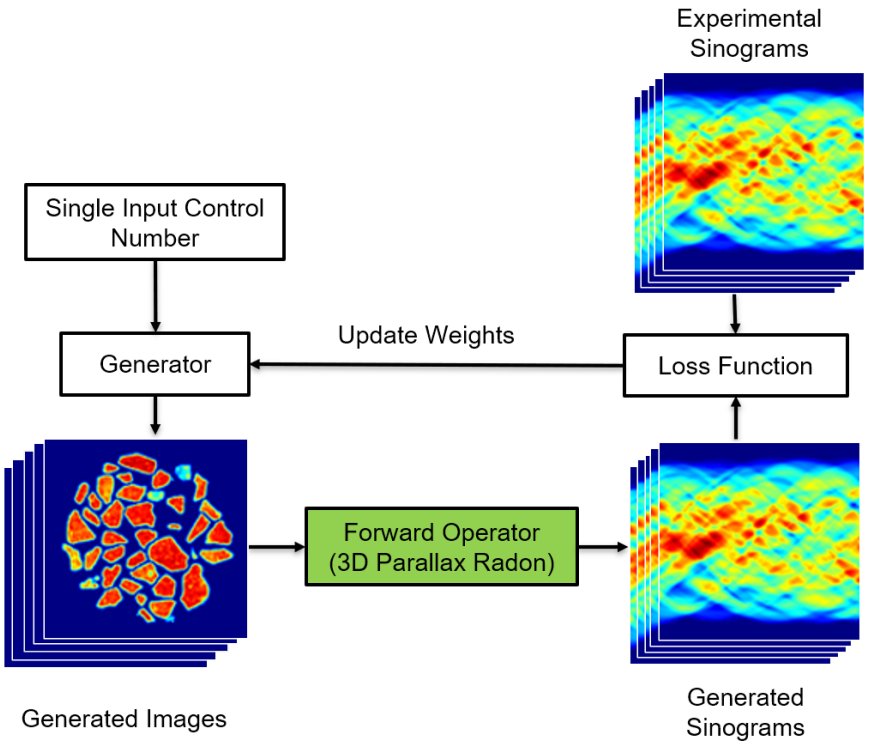

Figure 5.2: The self-supervised ParallaxNet flow chart. The generator architecture is shown in Figure 5.3. The generator takes a single-digit control number as input and outputs a full-volume size, which includes two spatial dimensions comprising the diffraction maps and a third dimension of diffraction patterns associated with each pixel. A forward operator is applied to convert these images into simulated sinograms containing parallax artefacts. A joint loss function, combining SSIM and MAE, is then used to compare the differences between the generated sinogram and the input experimental sinograms.

The generator network begins with a single input control value (specifically, 1 is used in this work), and is followed by a decoder to reconstruct the image based on the single number. After the input layer, three fully connected layers with 32 nodes and another fully connected layer where the number of nodes equal to the total number of voxels in the reconstructed volume are used. Next, the output of the final fully connected layer is reshaped to a 3D volume followed by four 3D convolutional layers. All activation functions are chosen as ReLU, except for the last layer we used the linear activation function. This architecture has a great ability to be scaled up, it allows to reconstruction of the images with m x m x c sizes reaching 550 x 550 x 51 or 100 x 100 x 4010 with parallax artefacts

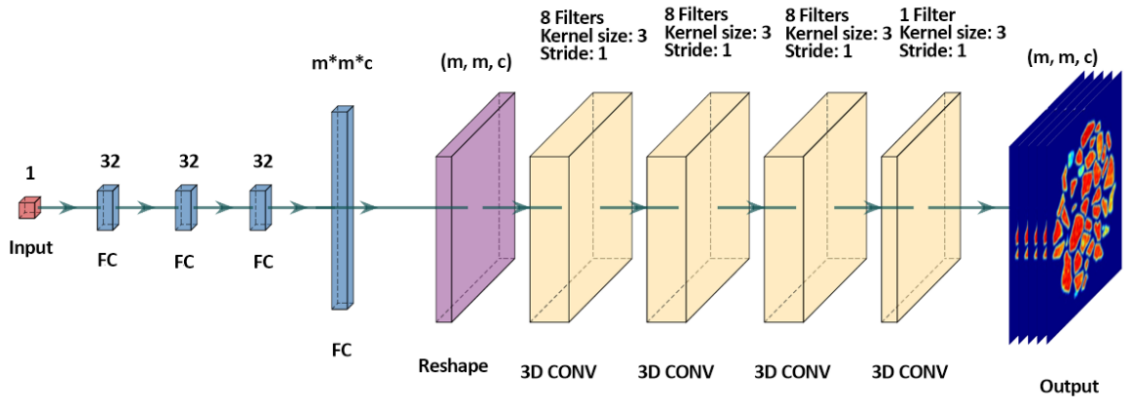

Figure 5.3: The generator architecture with a single constant as input. CONV represents 3-D convolutional layers, and FC represents fully connected layers. The filter numbers and layer sizes are shown above each layer. Here n represents the number of translation steps, and m represents the volume size of the output image. The ReLU function is used to connect the layers, except the Leaky ReLU function is used on the last fully connected layer to adapt possible negative values generated.

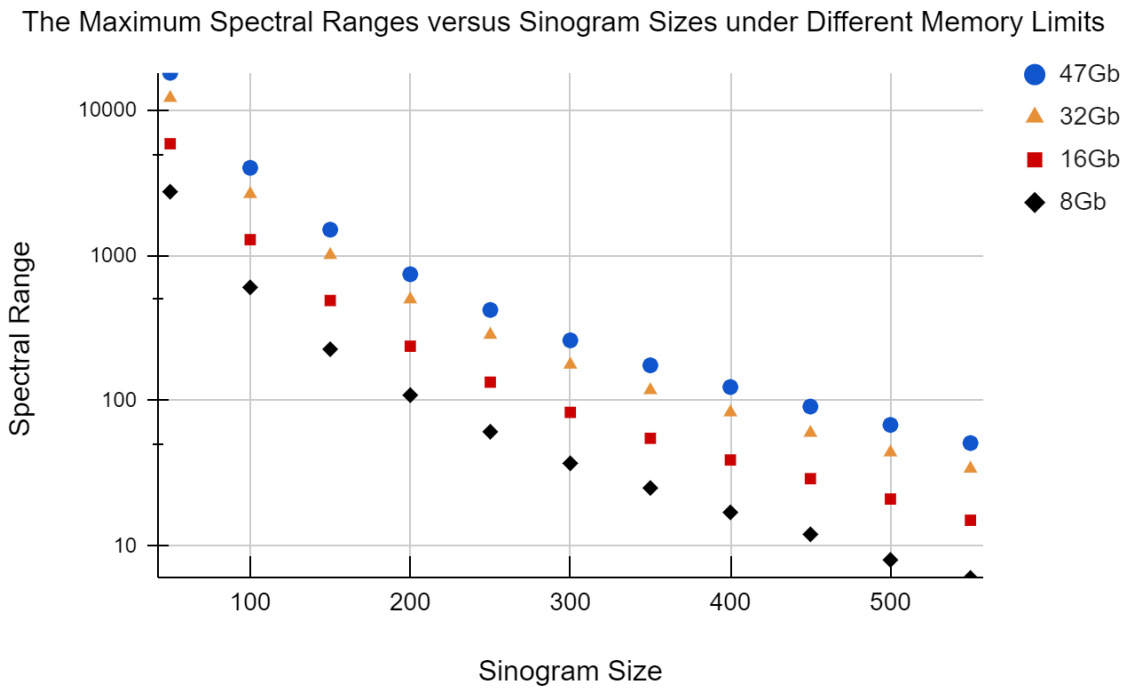

Figure 5.4: The maximum spectral range that ParallaxNet can take with different sinogram sizes under the condition of GPU memory limits of 47Gb, 32Gb, 16Gb and 8Gb respectively. This test is operated on volumes of simulated sinograms where the number of translation steps and the number of projections are equal.

The maximum volume size that ParallaxNet can reconstruct is depicted in Figure 5.4. The test was carried out under the constraints of GPU memory sizes of 47, 32, 16 and 8 GB, respectively. The tested sinograms are assumed to have the same number of translation steps and projections, referred to as the "sinogram size" in the figure. The figure shows that ParallaxNet can reconstruct a volume consisting of 200 x 200 sinograms with up to 742 channels simultaneously. Alternatively, it can also handle larger sinograms of 550 x 550 dimensions with 51 channels, covering an entire diffraction peak.

A joint loss function with the Mean Squared Error (MSE) and the Structural Similarity Indexing (SSIM) was used in training this network. The loss function compares the real experimental sinograms with the generated sinograms and updates the weights in the generator to give a generated sinogram volume that resembles better the experimental sinogram volume on the iteration. Normally, the architecture can give a very decent result after 1000 iterations. The reconstruction process of this self-supervised method can be expressed as:

$$G(\mathrm{a}) = \underset{\lambda}{arg\,min}\ L_{\mathrm{MSE}} + (1-\mathrm{r})\, L_{\mathrm{SSIM}} \tag{5.1}$$

Here, $G(\mathrm{a_c})$ is the generated reconstruction image by sending a control constant '$\mathrm{a_c}$' into the generator. The constant does not change while training. The L_{MSE} and L_{SSIM} represent the MSE loss and SSIM loss respectively, and their sum is adjusted by the fraction r. The MSE loss is defined as [49]:

$$L_{\mathrm{MSE}}(\mathbf{x},\mathbf{y}) = \frac{1}{\mathrm{N}} \sum_{i=0}^{\mathrm{N}} \left(\mathbf{x}_i - \mathbf{y}_i\right)^2 \tag{5.2}$$

Then, the SSIM loss can be expressed as [50]:

$$L_{\mathrm{SSIM}}(\mathbf{x},\mathbf{y}) = 1 - \frac{(2\mu_{\mathrm{x}}\mu_{\mathrm{y}} + \mathrm{C}_1)(2\sigma_{\mathrm{xy}} + \mathrm{C}_2)}{(\mu_{\mathrm{x}}^2 + \mu_{\mathrm{y}}^2 + \mathrm{C}_1)(\sigma_{\mathrm{x}}^2 + \sigma_{\mathrm{y}}^2 + \mathrm{C}_2)} \tag{5.3}$$

Here,

$$\mathrm{C}_1 = (\mathrm{K}_1 \mathrm{L}_{\mathrm{D}})^2 \tag{5.4}$$

$$\mathrm{C}_2 = (\mathrm{K}_2 \mathrm{L}_{\mathrm{D}})^2 \tag{5.5}$$

$$\sigma_{xy} = \frac{1}{\mathrm{N}-1} \sum_{i=1}^{\mathrm{N}} (\mathbf{x}_i - \mu_x)(\mathbf{y}_i - \mu_y) \tag{5.6}$$

The μ_x, μ_y are the average of all pixels in the input sinograms and the σ_x, σ_y represent their standard deviations. L is the dynamic range of the input images. K_1 and K_2 are two constants that are set as 0.01 and 0.03.

To adapt varying signal strengths across each chemical (diffraction/scattering angle) channel and facilitate easier training of the generator, all sinogram channels along the 2θ axis are normalised based on the maximum value of each channel. Additionally, to ensure the output images from the generator maintain a consistent relative intensity, they are divided by the same normalisation factors used for the sinograms before applying the forward operator. Then the forward operator processes images at their actual intensity

scale, yielding generated sinograms with accurate intensities. Subsequently, these generated sinograms are multiplied by the normalisation factors and the loss is calculated in comparison to the normalised input reference sinograms.

During training, a circular mask is applied to the images to filter out signals outside the CT reconstruction area. The 3D grid is calculated considering the experimental setup and specifically the 2 theta diffraction angles (1D vector), the sample-to-detector distance, the translation step size and the X-ray wavelength. Starting with a tomographic angle of 0 °, the forward operator accounts for nT voxels across the sample's thickness and simulates the parallax effect with an nT 2θ axis vector yielding the 3D grid. The modelling of the 3D grid is based on a relationship between the new 2θ axis, its offset from the centre of rotation, and the distance from the sample to the detector, as defined by Scarlett et al. [47] :

$$\sin(2\theta_{new}) = -u(y) \times \frac{\sin(2\theta_{middle})}{Rs} \tag{5.7}$$

$$Rs = \frac{\text{sample to detector size}}{\cos(2\theta)} \tag{5.8}$$

The sinogram volume with parallax can be created from parallax-free images by rotating the 3D grid, interpolating the XRD-CT data over it, and calculating the 3D Radon transform at each CT angle. The pseudocode for creating the 3D grid based on the experimental setup can be found in Algorithm 1, while the pseudocode for the forward operator is in Algorithm 2.

Algorithm 1: Pseudocode - Creating 3D grid

```
npix ← number of translation steps
npr ← number of projections
srange ← npix * translation_step_size
Rs ← sample_to_detector_size / cos(two_theta)
3D_grid ← Empty List with size (npix, npix, two_theta)
for i in range (npr) do
    xoffs ← (round(npix/2) - (i + 1)) * translation_step_size
    for j in range (npix) do
        3D_grid[i,j,:] = two_theta + arcsin(sin(two_theta) * xofs / Rs)
Return 3D_grid
```

Algorithm 2: Pseudocode - Parallax forward operating algorithm

```
Images ← Images * mask
/* Crop all values outside the circular bound */
sinograms ← Empty List
grid_3D ← create_grid(twotheta, sample_to_detector_distance, wavelength,
  translation_step_size)
for i in range (npr) do
    grid_3D_rot ← rotate(grid_3D, theta[i])
    Volume ← interpolate(Images, grid_3D_rot)
    s ← 3D_Radon_transform(Volume)
sinograms ← stack(s)
Return sinograms
```

5.3 Results & Discussion

5.3.1 Simulated XRD-CT data

To test the performance of ParallaxNet on XRD-CT images with parallax artefacts, we first use a simulated XRD-CT dataset with noiseless and zero-background using XRD patterns of a Ni fcc structure (ICSD: 64989) as presented in the DLSR paper [6]. When testing the performance of algorithms designed to solve inverse problems, it is crucial to ensure the forward projector used to generate simulated data is different from the forward projector the algorithm employs to solve the inverse problem. This differentiation helps maintain the rigour and validity of the evaluation. Being conscious of this, we coded different forward models for testing our approach with the simulated XRD-CT data. Specifically, we used an A matrix (ray tracing) calculated from astra-toolbox [51] as the forward projector to produce the simulated Ni XRD-CT dataset and a custom Radon using image rotation with bilinear interpolation for our ParallaxNet algorithm. This approach ensures a more unbiased assessment of ParallaxNet's capabilities.

We take into account the non-constant sample-to-detector distance for large samples by creating a 3D grid, where each pixel represents a distinct 2θ axis. The XRD-CT data, both the simulated and the experimental data presented in

the following sections, are interpolated using this 3D grid and subsequently, their 3D Radon transform is calculated. The simulated data were created using a sample-to-detector distance of 1000 mm, a translation step size of 0.2 mm, and a 100 keV X-ray energy. The final simulation results contain 121 translation steps, 121 projections and 2000 diffraction channels, which form a sinogram volume with the size 121 x 121 x 2000. The ParallaxNet can reconstruct the dataset in one go with 7.43 h and 5000 epochs.

The mean image and diffraction patterns, along with selected channels of maps, are shown in Figure 5.5. By visually inspecting the maps reconstructed by FBP and ParallaxNet, we can conclude that ParallaxNet accurately reconstructs the signals in the correct positions and addresses the parallax artefacts present in the simulated data.

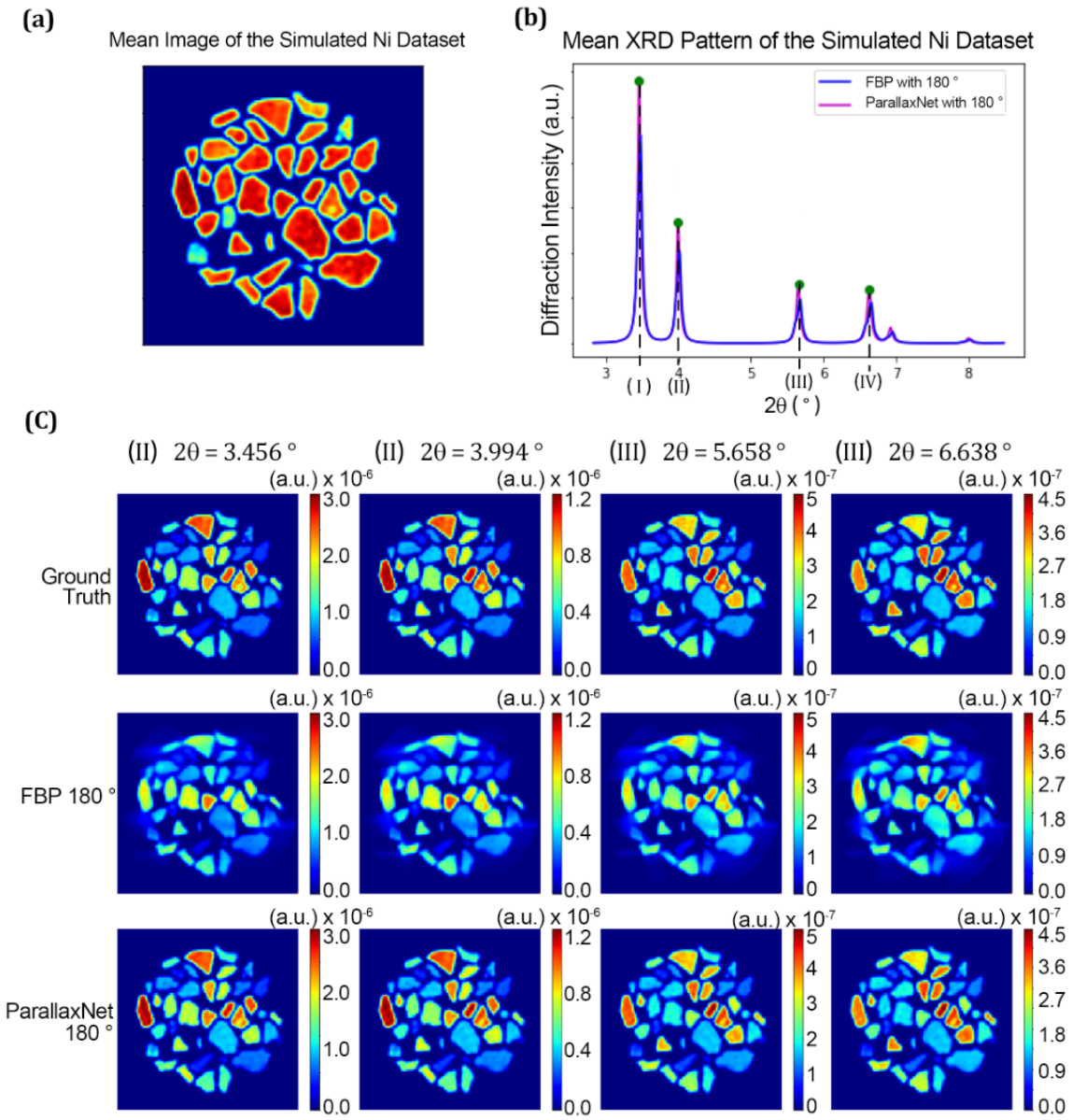

Figure 5.5: Overall result obtained with ParallaxNet on the simulated Ni dataset. (a) The mean image of the Simulated Ni dataset. (b) The average diffraction patterns of FBP and ParallaxNet were reconstructed by 0-180 ° scans. (c) Selected interesting diffraction channels which are marked in (b). The colour scale bars indicate the diffraction intensity for each pixel in both the reconstructed and ground truth images. This figure demonstrates that the diffraction signals are accurately reconstructed by ParallaxNet, whereas the FBP method exhibits significant parallax artifacts.

Figure 5.6 shows the results obtained from the sequential Rietveld analysis of both FBP and ParallaxNet reconstructed volumes using the TOPAS software which is guided by in-house developed Python scripts [48]. As presented in Figure 5.6, compared to the Rietveld analysis results of the FBP volume, the maps of Ni scale factor, crystallite size, and lattice parameter a from ParallaxNet

are almost identical to the ground truth maps while the FBP significantly diverge; this is apparent when one observes the lattice parameter and crystallite size maps. This suggests that the diffraction peak positions and shapes reconstructed by ParallaxNet closely match the ground truth patterns. The differences between the crystallite size and lattice parameter maps and their ground truth values are presented in Figure 5.7.

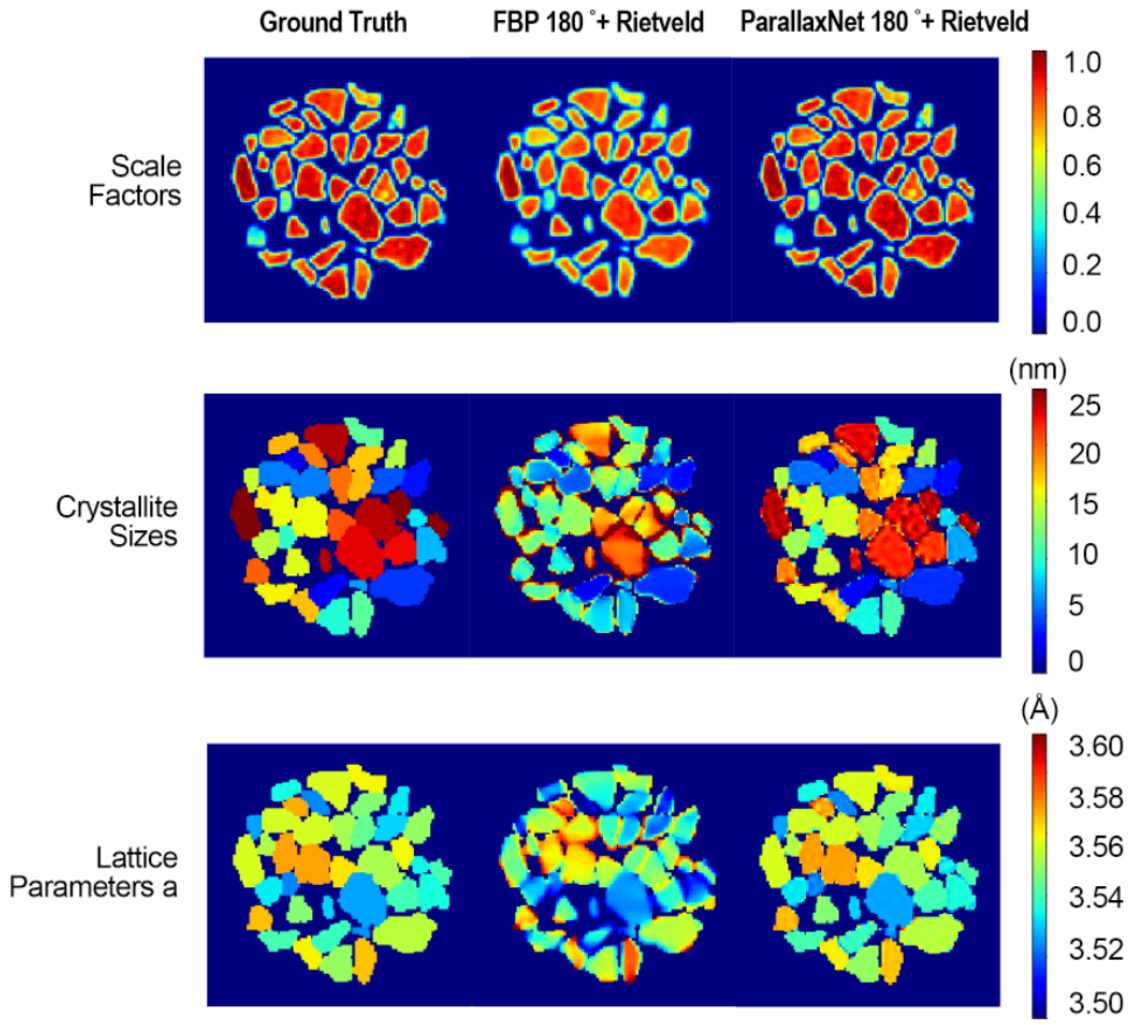

Figure 5.6: The results derived from the sequential Rietveld analysis of the reconstructed XRD-CT data using both FBP and ParallaxNet, alongside their ground truth values. The colour scale bars display the parameters refined through Rietveld analysis for each pixel in both the reconstructed and ground truth images. Additionally, the scale bar for the scale factor maps has been normalized.

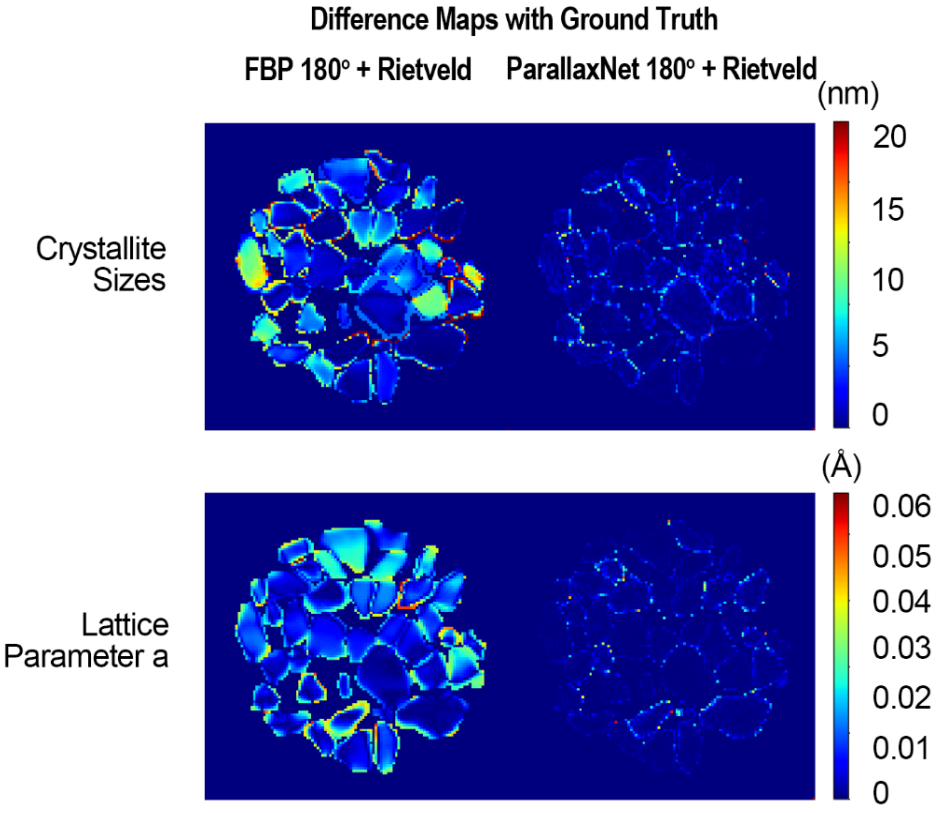

Figure 5.7: The error maps of the crystallite sizes and lattice parameters which are shown in Figure 5.6. The colour scale bars represent the absolute differences between the crystallite size and lattice parameter maps derived from Rietveld analysis compared to the ground truth values.

In a pixel-wise analysis, we select three regions of interest as depicted in Figure 5.8. It can be seen that within each region, the lattice parameter and crystallite size have the same value in the simulated dataset. As illustrated in Figure 5.8, the parallax artefact causes the FBP reconstructed patterns to exhibit significant shifts in peak positions, broadening, and some instances of splitting when reconstructed using the conventional 0-180 ° CT acquisition. However, the ParallaxNet can accurately reconstruct the volume without these artefacts and the reconstructed diffraction peaks are well aligned with the ground truth.

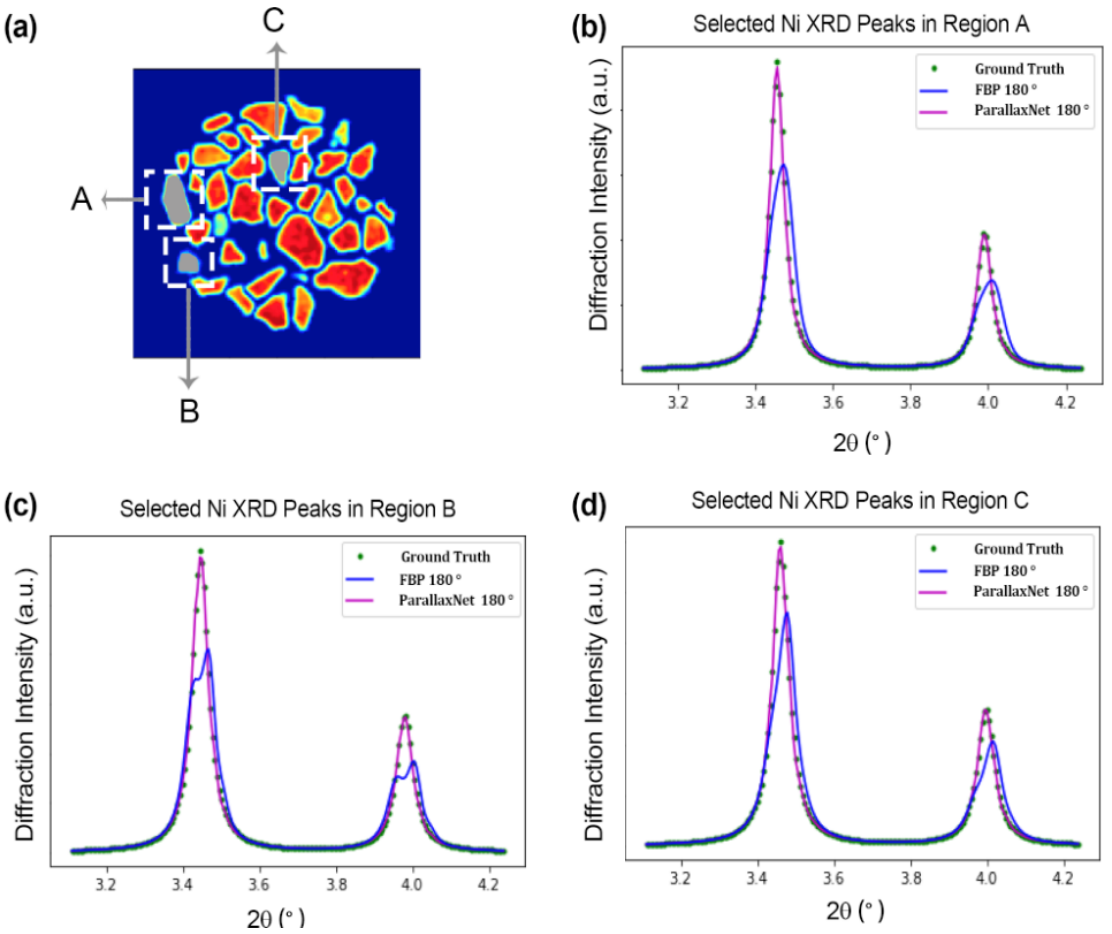

Figure 5.8: (a) The mean image of the simulated Ni XRD-CT dataset with three marked regions. (b) Selected diffraction peaks of the average diffraction pattern in Region A. (c) Selected peaks of the average diffraction pattern in Region B. (d) Selected peak from the average diffraction pattern in Region C. This figure shows the ParallaxNet correctly solving the diffraction peak splitting and shifting caused by parallax artefacts and that the diffraction pattern is aligned with the ground truth.

5.3.2 Experimental XRD-CT data

5.3.2.1 Phantom

Next, we evaluate the efficacy of the method using experimental XRD-CT data. The first dataset is a custom-made phantom consisting of four pipettes filled with different powder samples. The mean image of the phantom XRD-CT sample can be found in Figure 5.9a, which provides a view of the cross-section containing the powder samples within the four glass pipettes. This dataset contains two crystalline MgO (ICSD 9863; Sasaki et al., 1979), one SiC (ICSD 603798; Li & Bradt, 1986) and one TiO2 rutile (ICSD 33837; Sugiyama &

Take´uchi, 1991) phases respectively. The three powder samples were mounted into separate glass pipettes with an outer diameter of *ca.* 7.5 mm supported by quartz wool from both ends. Two pipettes were prepared using the same MgO powder sample. The four glass pipettes containing the powder samples were mounted onto a 3D printed sample holder designed for the parallax experiment [1].

XRD-CT measurements of the phantom sample were performed at beamline station P07 (EH2) at PETRA III, DESY, using a 103.5 keV (λ = 0.11979 Å) monochromatic X-ray beam focused to a spot size of 20 × 3 μm (H × V). 2D powder diffraction patterns were collected using a Pilatus3 X CdTe 2 M hybrid photon counting area detector. The 3D-printed sample holder was mounted directly on the rotation stage. The rotation stage was mounted perpendicularly to a hexapod; the hexapod was used to translate the sample across the beam. The XRD-CT scans were measured by performing a series of zigzag line scans in the z (vertical) direction using the hexapod and rotation steps. Two XRD-CT scans were performed, in both cases, the number of translation steps was 300 with a 80 μm step size and a 10 ms exposure time per point. The first XRD-CT scan was performed over a 0-180 ° range while the second over a 0-360 ° range, both using 300 angular steps.

Figure 5.9b presents the mean diffraction patterns derived from both FBP and ParallaxNet using 180 ° scans. It can be seen that the diffraction peaks in the FBP pattern are significantly broader compared to the ParallaxNet. Selected channels from the reconstructed XRD-CT dataset are depicted in Figure 5.9c. It can be clearly seen that the new approach reconstructs artefact-free images while the conventional FBP method displays pronounced parallax artefacts that are readily observable upon visual inspection.

Unlike the dataset presented in the DLSR paper, here we utilise the full size of the image dataset with ParallaxNet without any image rebinning/resizing. This is because the new ParallaxNet method boasts better scalability compared to the

DLSR. The XRD-CT sinogram volume dataset comprises 269 translation steps, 300 projections, and 670 selected diffraction channels. According to Figure 5.4, the 269 x 300 x 670 sinogram cannot be reconstructed as a one-go since it has exceeded the limit of our GPU memory, and the maximum spectral range we can handle is around 270-300. We then split the dataset into three batches, and each batch contains 250 channels. To mitigate the edge effect between the batches, we incorporated an overlap of 40 channels for each batch. As a result, the three batches are defined with channel numbers 0-250, 210-460, and 420-670, respectively. The three batches are merged afterwards by taking the average of the overlapped channels.

For each batch, we ran 5,000 epochs, which took 5.28 h excluding the initialization time. To improve image quality and reduce the number of required epochs, we pre-trained the generator using FBP images from 180 ° projections. This preliminary step required only 1,000 iterations and was completed in 4 min for each batch. In total, the image reconstruction with ParallaxNet took 16.43 h. The ParallaxNet training was performed using a workstation equipped with an NVIDIA Quadro RTX8000 GPU, Intel Xeon W-2155 CPU at 3.30GHz and using PyTorch version 1.13.1.

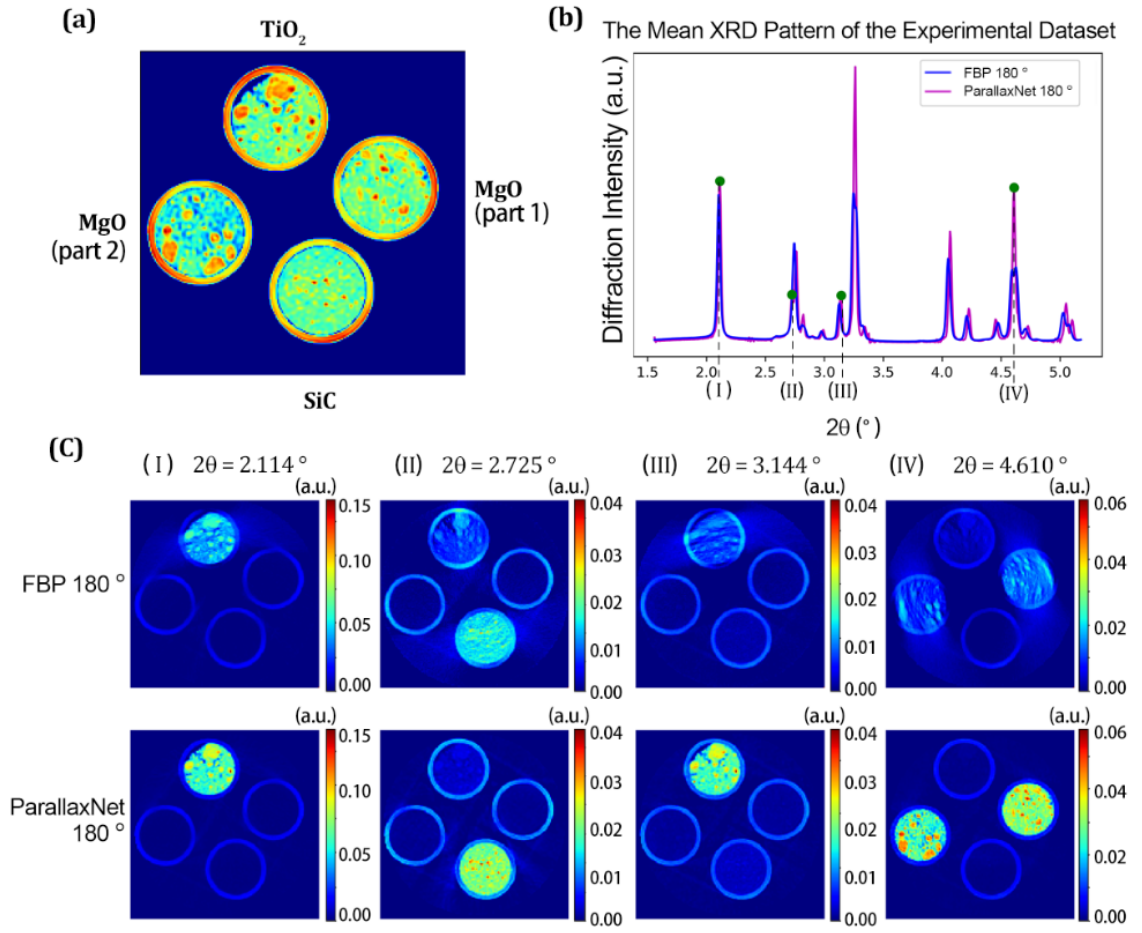

Figure 5.9: (a) The mean image of the Phantom XRD-CT dataset which contains crystalline TiO_2, MgO, and SiC. (b) The average diffraction patterns of FBP and ParallaxNet were reconstructed by 0-180 ° scans. (c) Selected interesting diffraction channels which are marked in (b). The scale bars represents the diffraction intensity of each pixel in the images reconstructed using both FBP and ParallaxNet.

Selected reflections corresponding to each of the three phases are shown in Figure 5.10. It can be clearly observed that the XRD-CT reflections reconstructed by FBP from both 180 and 360 ° scans exhibit significant peak broadening artefacts. Additionally, the diffraction peaks generated by FBP with a 180 ° scan range exhibit peak shifting artefacts. In contrast, the diffraction peaks generated with a 360 ° scan range using FBP are in good alignment with the ones obtained by the ParallaxNet with a 180 ° scan range. This demonstrates that Parallax effectively reduces various artefacts brought about by parallax and also that it simply requires a 0-180 ° scan range to reconstruct parallax artefact-free data.

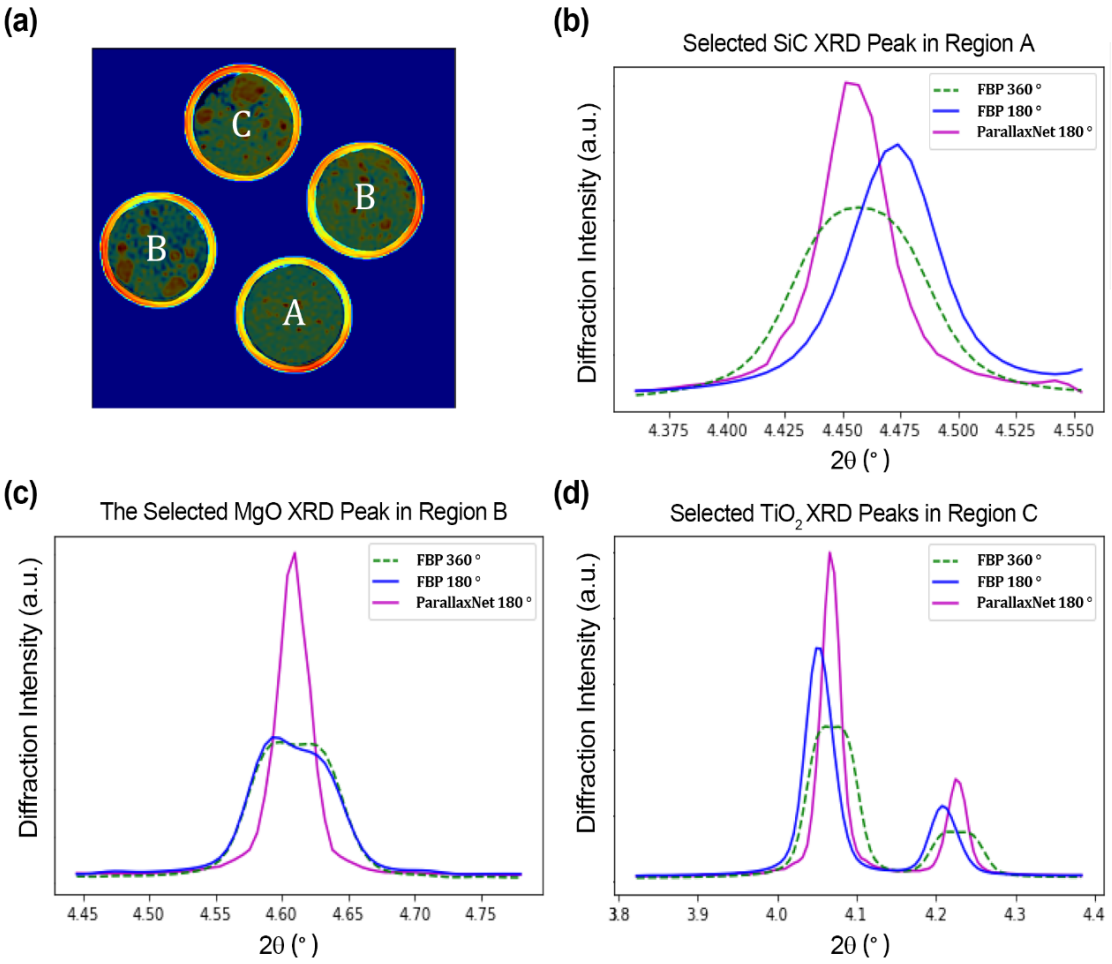

Figure 5.10: (a) The mean image of the Phantom XRD-CT dataset with three marked regions. (b) A selected peak of the average diffraction pattern in Region A. (c) A selected diffraction peak from the average diffraction pattern in Region B. (d) A selected peak from the average diffraction pattern in Region C. This figure shows the ParallaxNet can solve the peak broadening artefacts, and peak positions reconstructed by the ParallaxNet with 0-180 ° scans are aligned with the FBP reconstructed with 0-360 ° scans. In contrast, the FBP reconstructed diffraction peaks over the 0-180 ° scan range exhibit significant diffraction peak shifting which was caused by the parallax artefacts.

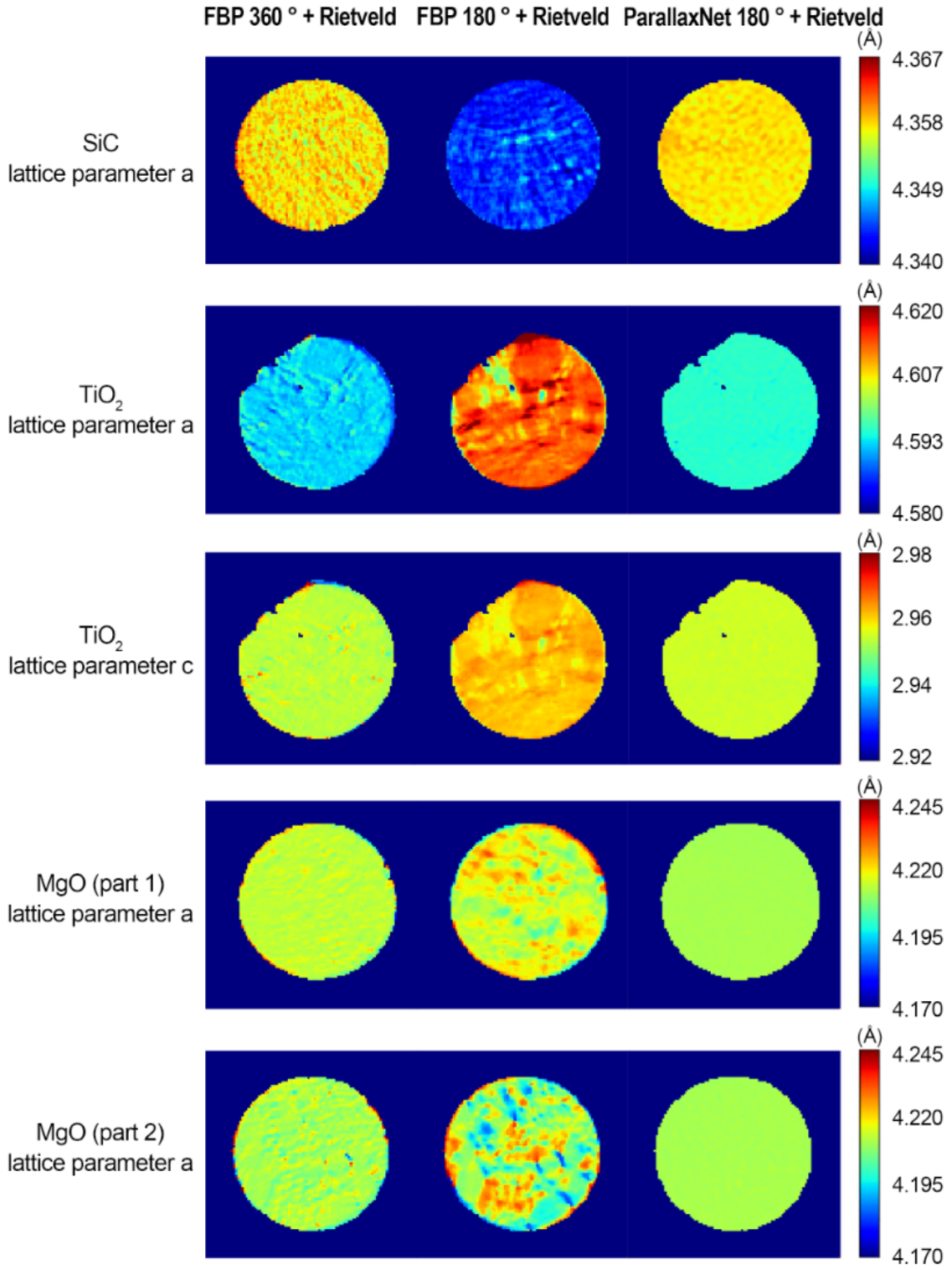

Figure 5.11: Lattice parameter maps associated with the four components are shown in Figure 5.9a. The colour scale bars represent the lattice parameters refined by the Rietveld method from the volumes reconstructed using both ParallaxNet and FBP.

The Rietveld analysis of the reconstructed volumes further demonstrates the efficacy of our method. From the lattice parameter maps shown in Figure 5.11, it can be clearly seen that the lattice parameter maps from the ParallaxNet with 180 ° scans have almost identical values with those from the maps of the FBP derived from the 360 ° scan range. FBP images reconstructed using a 360 ° scan range should not exhibit any peak shifting, even when significant parallax artefacts are present; to clarify, the centroid position of the peaks will be in the correct position as if it were a dataset without parallax. As such, the lattice parameter maps of the ParallaxNet can be considered close to the ground truth. It is also worth mentioning that the ParallaxNet maps exhibit less noise compared to those from the FBP. The distribution of the lattice parameters is shown in histograms in Figure 5.12. These results also demonstrate that ParallaxNet can yield lattice parameters that align with the results from the FBP with 360 ° scans.

Figure 5.13 presents the scale factor and crystallite size maps obtained from the Rietveld analysis. A key observation is that the parallax artefact significantly affects the crystallite sizes obtained by conventional approaches. Specifically, on both 180 ° and 360 ° XRD-CT scans, it leads to broadened diffraction peaks and reduced crystallite values when using the FBP reconstruction algorithm. The maps suggest that ParallaxNet has, to a certain extent, solved the peak broadening artefact instigated by parallax. This correction is particularly pronounced for the two MgO components, where their crystallite sizes offer mutual validation. Based on these observations, we can deduce that ParallaxNet can correctly solve the parallax artefact on the real phantom experimental dataset.

In reviewing Figure 5.10c and 5.10d, it is evident that even with a 360 ° scan, the FBP reconstructed peak shape does not conform to a Gaussian distribution. While 360 ° scans can accurately identify peak positions, the artefact of peak splitting remains unresolved. Consequently, fitting FBP images with Gaussian (or Lorentzian) models may introduce errors. Additionally, it is worth noting that

peaks at higher 2θ angles are easier to succumb peak splitting artefacts, therefore, there are more mismatched in the lattice parameter '*a*' distribution of the FBP with 360-degree scans and the ParallaxNet than in the lattice parameter '*c*' distribution for the TiO_2 phase shown in Figure 5.12.

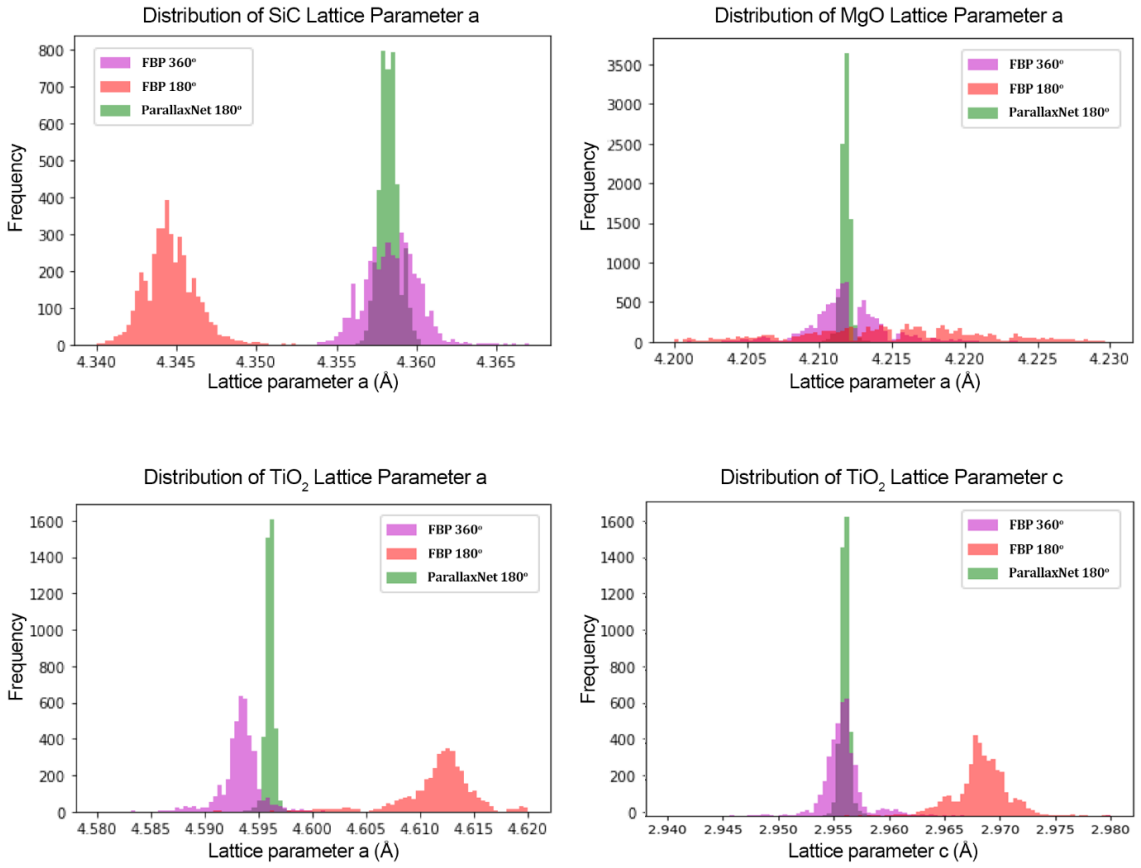

Figure 5.12: The distribution of lattice parameters for the experimental phantom dataset.

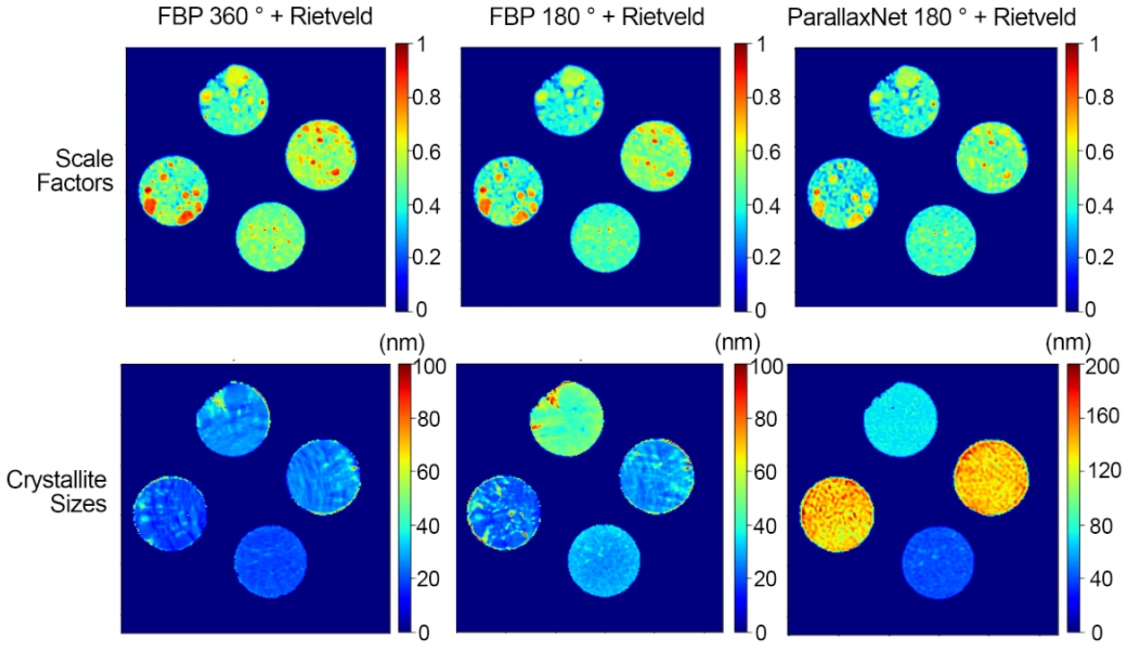

Figure 5.13: Top row: Scale factor maps (normalised) associated with four components. Button row: Crystallite size maps associated with four components. The colour scale bars represent the parameters refined by Rietveld analysis for each pixel in images reconstructed using both ParallaxNet and FBP.

5.3.2.2 NMC532 cylindrical Li-ion battery

In addition to the phantom dataset presented in the previous section, the efficacy of the method was evaluated with a second experimental dataset. Specifically, a dataset acquired from a commercially available and industrially relevant 10440 NMC532 Li-ion battery was used [41]. This dataset was scanned using the same beamline and experimental setup using a 73.89 keV (λ = 0.16779 Å) monochromatic X-ray beam focused on the same spot size of 20 × 3 μm. An XRD-CT dataset was acquired using 521 translation steps with a 20 μm step size and a 10 ms exposure time per point. The XRD-CT scan was performed over a 0-360 ° range using 1000 angles in total.

The detector calibration was performed using a CeO2 standard. Every 2D diffraction image was calibrated and azimuthally integrated to a 1D powder diffraction pattern with a 10 % trimmed mean filter using the pyFAI software package and in-house developed scripts [52]. The integrated diffraction patterns were reshaped into sinograms and centred; the air scatter signal was

subtracted from the data. For the conventional data analysis approach, the XRD-CT images (i.e. reconstructed data volume) were reconstructed using the FBP algorithm. A pseudo-voigt peak shape function was used for the refinements after the analysis of the CeO_2 pattern. Rietveld analysis was performed on the reconstructed diffraction patterns with the TOPAS software [48] on a voxel-by-voxel basis. Rietveld analysis was first performed using the summed diffraction pattern of each XRD-CT dataset (i.e. to provide a good starting model) before running the voxel-by-voxel Rietveld analysis to provide the spatially-resolved physicochemical information. The parameters refined were the scale factor, lattice parameter and crystallite size for each phase. A 2nd-order Chebyshev polynomial was used to model the background as it was fairly flat in all reconstructed patterns.

This dataset consists of 521 translation steps, 1000 projections and 1800 channels of sinograms. To train this big dataset, we divided it into batches of 55 channels, and each batch took ca. 8 h to process using 5000 epochs. To address this large dataset, we first selected an XRD diffraction peak from the Cu phase and reconstructed only the images without parallax corresponding to this peak within the 55-channel range. Then, we performed Rietveld analysis on this 521 x 521 x 55 XRD-CT dataset to get the chemical information of the Cu phase presented in this Li-ion battery dataset. The original dataset was performed using a 0-360 ° scan range, but for testing the ParallaxNet, we only used the part of the data corresponding to the 0-180 ° scan range so for each batch, the size of the reference sinogram was 521 x 500 x 55. For comparison, the XRD-CT data were also reconstructed using the FBP algorithm using both the 180 and 360 ° ranges and were analysed using the Rietveld method i.e. on the 55 selected channels of the Cu XRD peak. The ParallaxNet also utilized the FBP with 180 o projections to pre-train the generator for faster convergence.

Figure 5.14a displays the average image from the selected 55 channels of the Cu XRD peak (reflection with a *hkl* value of 111), highlighting three regions of interest. Figures 5.14b-d depict the average XRD peaks from the marked regions. As illustrated in the figure, ParallaxNet can accurately reconstruct the

Cu peak, producing a significantly sharper peak. The peak positions align with the XRD-CT data reconstructed by the FBP from the full 360 ° scan. This result shows that the ParallaxNet correctly removed both the peak shifting and broadening artefacts caused by the Parallax on this Cu XRD peak. It is important to note here that the centres of the peaks obtained from the FBP 360 ° scan align with those of the ParallaxNet-reconstructed peaks. However, it becomes evident that the peak shape cannot be effectively described using a single peak shape model. This observation is distinctly apparent across all peaks illustrated in Figure 5.14, with a particularly noticeable manifestation in the Cu diffraction peak from Region B. This observation bears significance, as attempting to fit these peaks using a single model, such as Gaussian or pseudo-Voigt models—commonly employed in XRD data analysis—can potentially yield inaccurate data interpretations. Such an approach might result in artificial shifts of the peaks, given that the employed model does not adequately capture the intricacies of the data's true behaviour. This is especially crucial when high precision is required for the calculated lattice parameter values, e.g. in the order of <10-3 Å such as when attempting to capture shifts in the Cu peak introduced by temperature gradients in these battery systems [53].

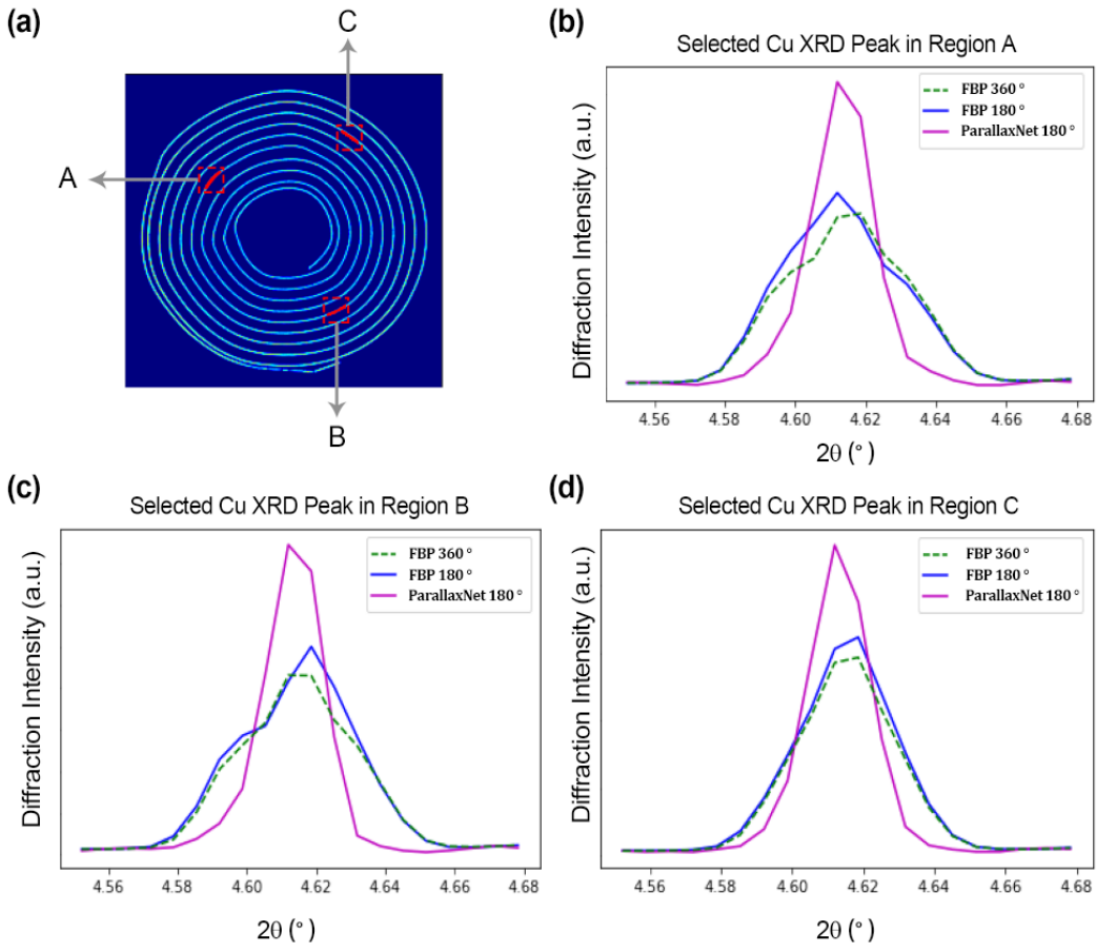

Figure 5.14: (a) The mean image of the Cu phase of Li-ion battery XRD-CT dataset with three marked regions. (b) The selected peak of the average diffraction pattern in Region A. (c) The selected peak of the average diffraction pattern in Region B. (d) The selected peak of the average diffraction pattern in Region C. This figure shows that ParallaxNet can solve the peak broadening and peak splitting artefacts.

Figure 5.15a displays the lattice parameter maps obtained through Rietveld analysis. As seen in the FBP with the 180 ° scan range lattice parameter map, the parallax artefact results in unevenly distributed lattice parameter values (as determined by the Rietveld analysis) across different positions of the same material (Cu phase). However, both the FBP with the 360 ° scan range and the ParallaxNet results with 180 ° scan range yield lattice parameter maps that are evenly distributed across all positions. The histogram depicting the distribution of lattice parameters for the three maps is shown in Figure 5.15b. The mean values of the lattice parameters for the FBP with 360 ° scan range, the FBP with

180 ° scan range, and the ParallaxNet result are 3.6047 Å, 3.6045 Å, and 3.6048 Å, respectively. The scale factor maps can be found in Figure 5.16.

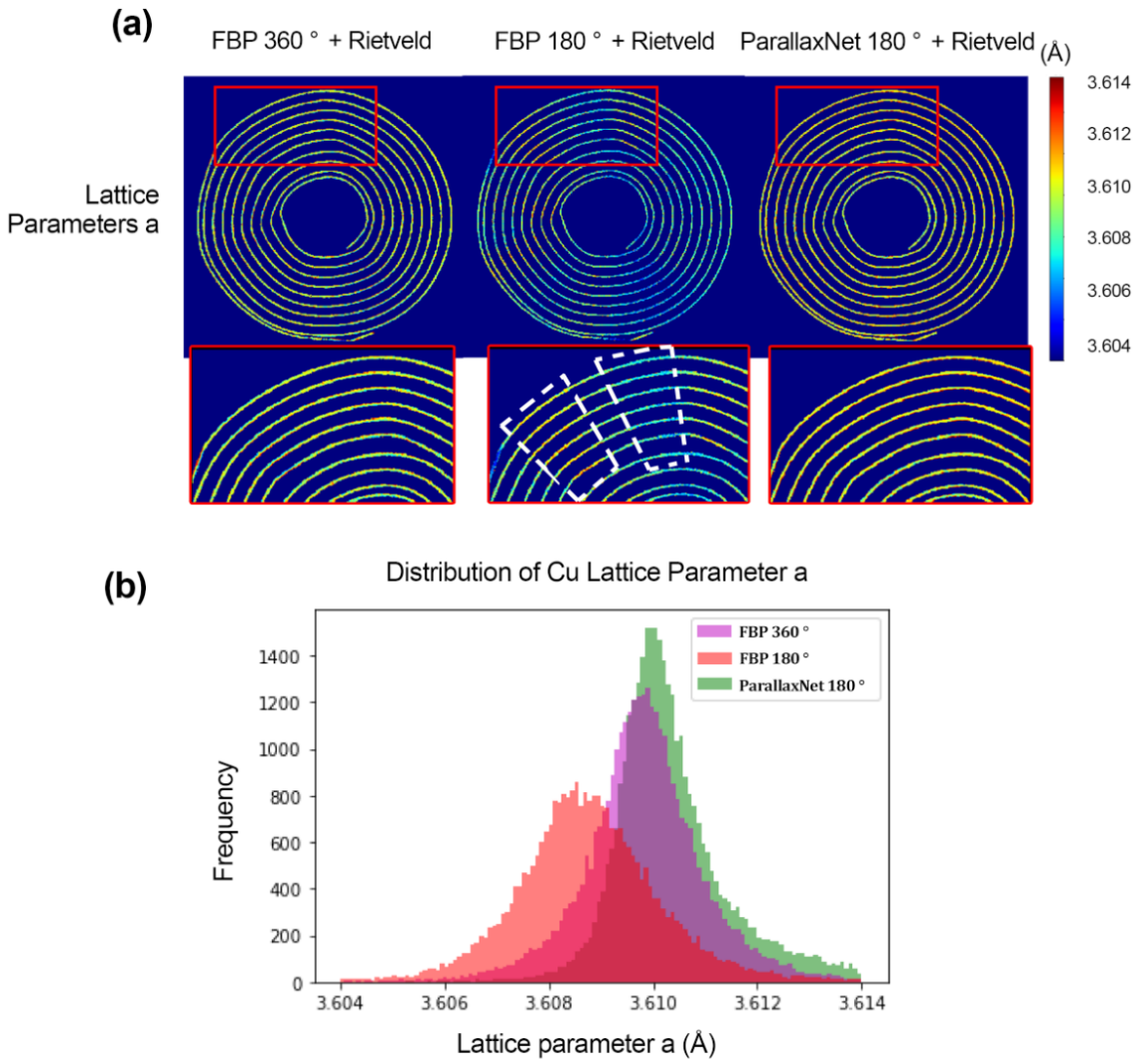

Figure 5.15: (a) The lattice parameter maps were obtained by the Rietveld method for a Li-ion battery dataset. (b) the distribution of lattice parameters for the maps shown in (a). The colour scale bar represents the lattice parameter for each pixel, refined by Rietveld analysis.

It was therefore demonstrated that ParallaxNet can accurately reconstruct XRD-CT images/diffraction patterns of this experimental Li-ion dataset and that it is possible to extract meaningful chemical information from just a single peak of the XRD pattern. Subsequently, a broader range of diffraction channels was chosen, encompassing 555 out of the 1800 channels from the original dataset. These channels span a native 2θ value range from 1.203 to 4.877 °. We divided these 555 channels into 11 segments, each containing 55 channels, consistent

with the phantom dataset approach. To mitigate edge effects between the reconstructed images of each batch, we incorporated a 5-pixel overlap on either side of each segment. We then averaged the overlapping sections to produce the final XRD-CT image volume with dimensions of 521 x 521 x 555.

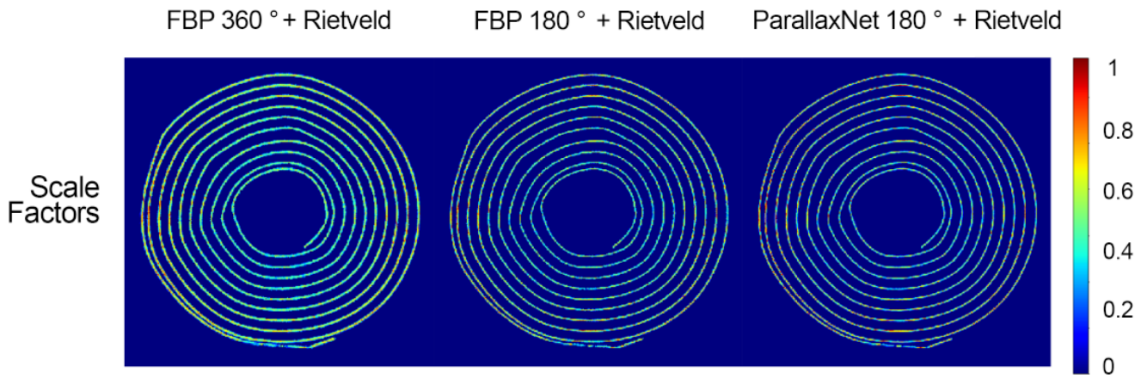

Figure 5.16: The scale factor map (normalized) of the Cu phase from an experimental AAA Li-ion battery dataset. The colour scale bar represents the value of the refined scale factor for each pixel after normalization.

To reconstruct this expanded dataset, ParallaxNet required 90 h of training time, which includes both initialization and pre-training with FBP. Since each batch is independent, we utilised two workstations with three NVIDIA Quadro RTX8000 GPUs to process these 11 batches in parallel using PyTorch. In the end, it took ca. 33 real-world h to complete this dataset. It is worth noting that this represents the most extreme scenario encountered in real-world experimental datasets, and the DLSR method cannot handle a dataset of this magnitude.

Figure 5.17 shows the NMC532 phase of the reconstructed dataset, highlighting three specific regions of interest. Additionally, this figure displays the selected average XRD peaks corresponding to the NMC532 phase. Another NMC532 peak of these three regions is also shown in Figure 5.18. These figures confirm that ParallaxNet can accurately reconstruct the same peak positions as those derived from the FBP reconstructed with the 360 ° scan range. Furthermore, ParallaxNet effectively addresses the issue of peak broadening artefacts, producing peaks that are sharper and narrower compared to those in the FBP images.

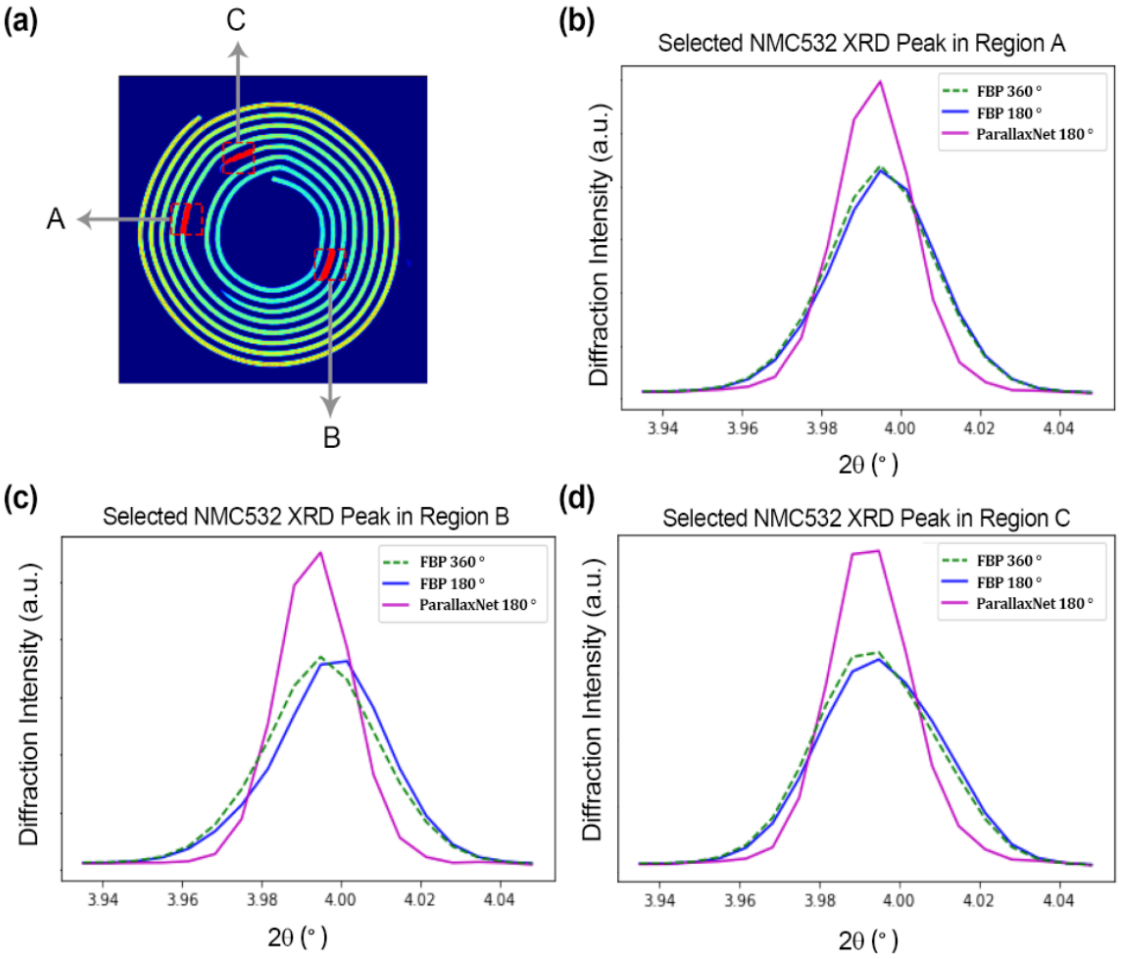

Figure 5.17: (a) The mean image of the NMC532 phase of Li-ion battery XRD-CT dataset with three marked regions. (b) The selected peak of the average diffraction pattern in Region A. (c) The selected peak of the average diffraction pattern in Region B. (d) The selected peak of the average diffraction pattern in Region C. This figure shows that ParallaxNet can solve the peak broadening artefacts, and reconstruct correct peak positions compared to the FBP with a 0-180 ° scan range.

Maps obtained from the Rietveld analysis of the NMC532 phases are shown in Figure 5.19 and Figure 5.20. The crystallite sizes obtained with the three different methods on the top line of Figure 5.20 indicate the crystallite sizes calculated from the ParallaxNet reconstructed volume are larger than both FBP methods with 180 ° and 360 ° scans respectively. The average crystallite sizes raised from ca. 91 nm (for FBP with 180 ° scans) and 92 nm (for FBP with 360 ° scans) to 137 nm (for the ParallaxNet), which also supports the conclusion we

drew from visual inspection: the diffraction peaks are sharper and narrower than those produced by conventional methods.

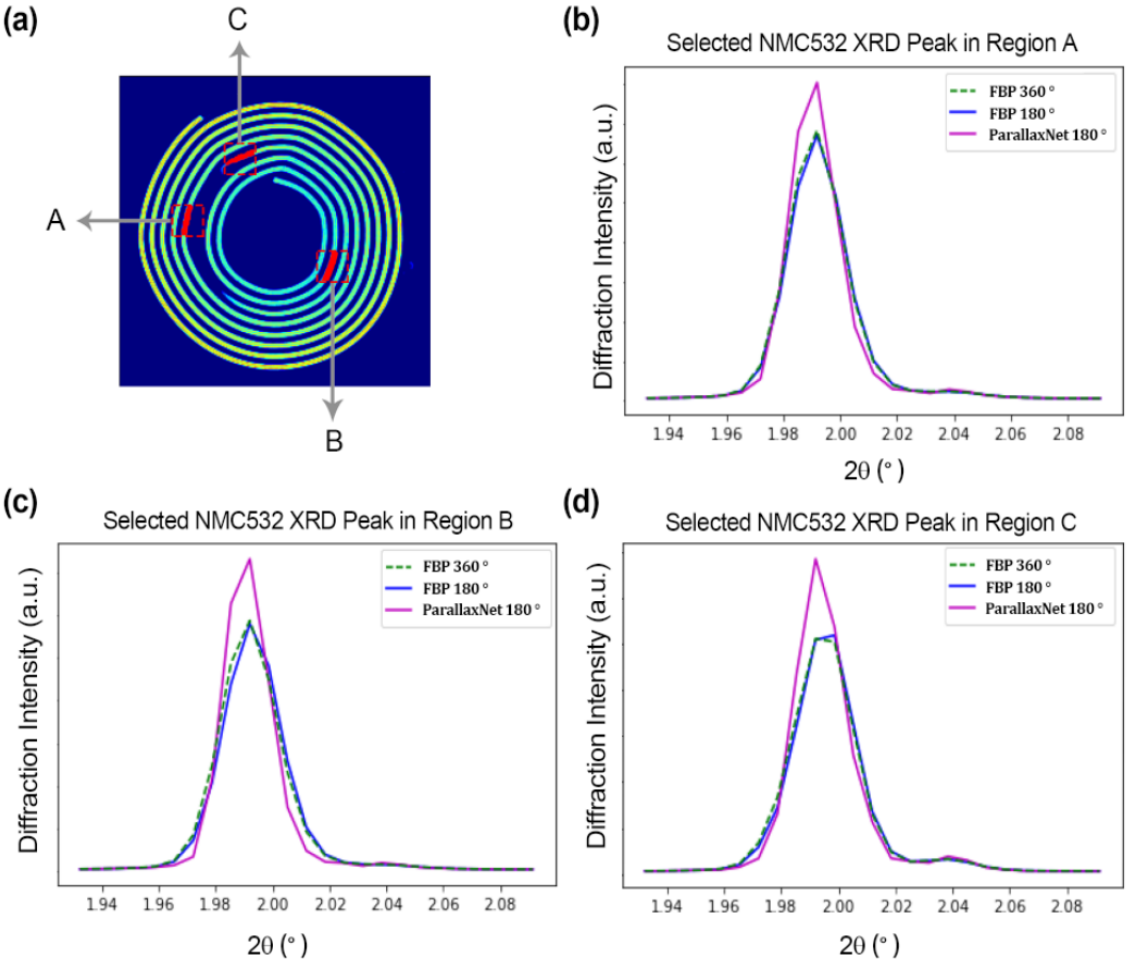

Figure 5.18: (a) The mean image of the Cu phase of Li-ion battery XRD-CT dataset with three marked regions. (b) The selected peak of the average diffraction pattern in Region A. (c) The selected peak of the average diffraction pattern in Region B. (d) The selected peak of the average diffraction pattern in Region C. This figure shows that ParallaxNet can solve the peak broadening artefacts, and reconstruct correct peak positions compared to the FBP with 0-180 ° scans.

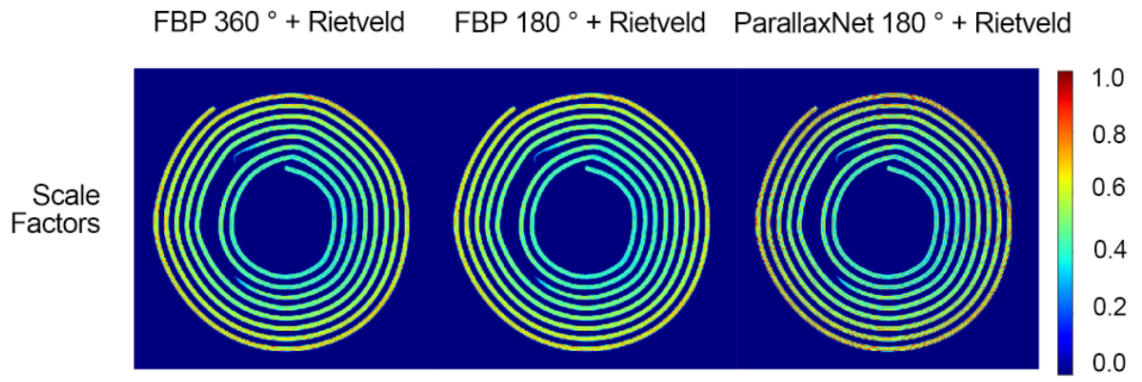

Figure 5.19: The Scale factor maps of the NMC532 phase. The colour scale bar represents the value of the refined scale factor for each pixel after normalization.

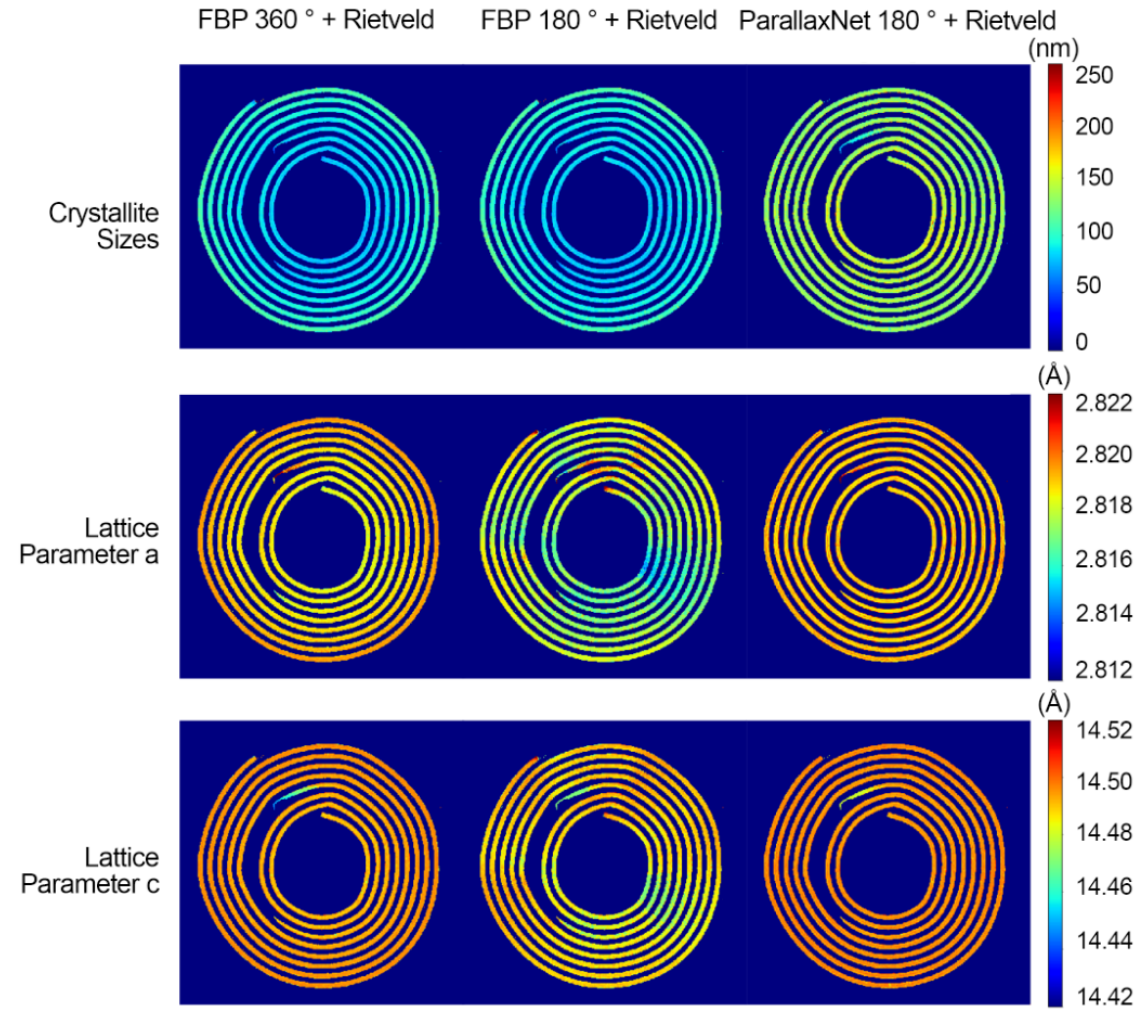

Figure 5.20: Top row: crystallite size maps of the NMC532 phase. Mid row: lattice parameter a maps of the NMC532 phase. Bottom Row: lattice parameter c maps. All maps are obtained by Rietveld analysis on the Li-ion battery XRD-CT dataset. The colour scale bars represent the values of parameters refined through Rietveld analysis for each pixel.

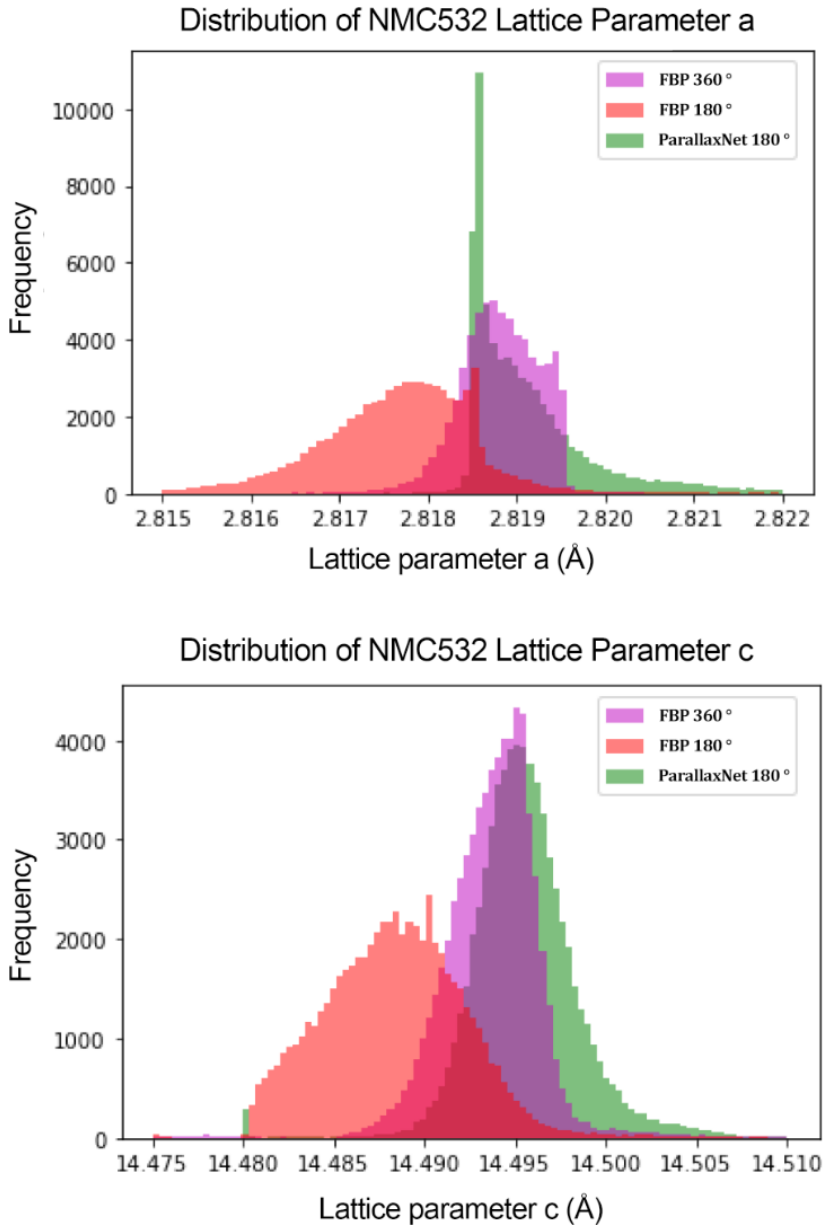

Figure 5.21: The distribution of lattice parameters a and c for the NMC532 phase. The results demonstrate that ParallaxNet can accurately reconstruct peak positions, yielding lattice parameters that align with the results from the FBP with 360 ° scans.

The lattice parameter maps of the NMC532 phase also indicate that the ParallaxNet can correctly reconstruct the evenly distributed lattice parameter maps which aligns with the FBP with 360 ° scan. The peak shifting artefact in the images reconstructed by the FBP with a 180 ° scan range has been effectively eliminated by ParallaxNet. The distribution of the lattice parameters of the NMC532 is presented in Figure 5.21. The distribution maps show the lattice parameters derived using ParallaxNet are closer to the FBP with 360 °

projections than the FBP with 180 ° projections, which is a clear indication that the parallax artefact has been successfully removed.

5.4 Summary & Conclusions

In this chapter, we introduced a novel and robust XRD-CT reconstruction approach, ParallaxNet, designed to reconstruct images from XRD-CT data containing parallax artefacts. The ParallaxNet strategy employs a 3D neural network generator framework together with a customised forward projector to produce parallax artefact-free images/diffraction patterns. This is achieved through an iterative approach by comparing the difference between the generated sinogram volume and the input reference sinogram volume. We evaluated ParallaxNet's performance using three datasets: a simulated XRD-CT dataset, a measured phantom object, and an experimental dataset recorded on an NMC532 cylindrical Li-ion battery.

For all three datasets, this new approach accurately reconstructed the peak positions using only a 0-180 ° angular range, eliminating the need for a 0-360 ° scan which halves the required acquisition time (i.e. half the number of projections). Furthermore, the reconstructed peaks were sharper and narrower than those produced by traditional FBP methods, both with 180 ° and 360 ° scans. It should also be noted that in this work it was also shown that using a 360 ° scan approach as a means to remove parallax artefacts is insufficient and should be avoided as it leads to peaks with shapes that cannot be modelled with single peak profile (e.g. Gaussian peak). This was clearly demonstrated with the experimental XRD-CT presented in this work, an example being the Cu component in the cylindrical Li-ion battery.

ParallaxNet presents distinct advantages over the previously developed DLSR methodology, addressing several inherent limitations:

- Firstly, ParallaxNet operates without requiring *a priori* knowledge about the chemical composition of the sample being measured.

- DLSR also requires the identification of all phases and the construction of a robust physical model; this can potentially lead to some minor components being overlooked during the inspection of diffraction patterns in the sinogram data.
- Furthermore, ParallaxNet is more scalable. In this work, we applied the ParallaxNet on the full-size experimental phantom XRD-CT datasets. However, the DLSR can only be applied on the scaled-down version of the same dataset(s) as shown in the DLSR paper (e.g. with 121 x 121 image sizes for the Li-ion battery). Moreover, ParallaxNet does not need as much RAM requirements as the DLSR approach, especially when DLSR is used in conjunction with TOPAS.
- ParallaxNet does not require any data preprocessing. ParallaxNet can be applied to the raw sinograms, but DLSR needs the manually created masks for each phase and background subtraction on the sinograms in order to use a simple background model.
- We have demonstrated that the conventional method of employing a 0-360 ° scan with FBP to eliminate parallax artefacts and obtain precise lattice parameter values can be precarious. For instance, the FBP reconstruction of the Cu peaks in the 0-360 ° scan revealed peaks that cannot be modelled using a single peak shape model (such as Gaussian or pseudo-Voigt). This could potentially result in wrong lattice parameter values and misinterpretation of the data, especially if lattice parameter values with high precision are required to be extracted from the data.

At this stage, while ParallaxNet presents a promising approach to XRD-CT image reconstruction with parallax artefact, it is not without its limitations. A significant constraint is the extended computational time required for large datasets. For instance, the Li-ion dataset, with sinogram dimensions of 521 x 500 x 540, demanded a staggering 90 h of computational time. Currently, due to GPU memory constraints, there's a necessity to divide datasets into smaller batches for processing. Moving forward, there is potential to explore a more streamlined generator, which could significantly minimise the computational

resources needed and address some of these challenges. Last but not least, it should be noted that the developed method can be applied to other X-ray scattering-based computed tomography data suffering from parallax artefacts, such as pair distribution computed tomography

5.5 References

[1] A. Vamvakeros *et al.*, ‘DLSR: a solution to the parallax artefact in X-ray diffraction computed tomography data’, *J. Appl. Crystallogr.*, vol. 53, no. 6, 2020.

[2] Esteva, A. et al. Deep learning-enabled medical computer vision. Npj Digit. Med. 4, 1–9 (2021).

[3] L. Jiao and J. Zhao, ‘A survey on the new generation of deep learning in image processing’, *Ieee Access*, vol. 7, pp. 172231–172263, 2019.

[4] P. N. Druzhkov and V. D. Kustikova, ‘A survey of deep learning methods and software tools for image classification and object detection’, *Pattern Recognit. Image Anal.*, vol. 26, pp. 9–15, 2016.

[5] F. Olsson, ‘A literature survey of active machine learning in the context of natural language processing’, 2009.

[6] H. Dong *et al.*, ‘A scalable neural network architecture for self-supervised tomographic image reconstruction’, *Digit. Discov.*, Jun. 2023, doi: 10.1039/D2DD00105E.

[7] X. Yang *et al.*, ‘Tomographic reconstruction with a generative adversarial network’, *J. Synchrotron Radiat.*, vol. 27, no. 2, pp. 486–493, Mar. 2020, doi: 10.1107/S1600577520000831.

[8] M. Beister, D. Kolditz, and W. A. Kalender, ‘Iterative reconstruction methods in X-ray CT’, *Phys. Med.*, vol. 28, no. 2, pp. 94–108, 2012.

[9] X. Pan, E. Y. Sidky, and M. Vannier, ‘Why do commercial CT scanners still employ traditional, filtered back-projection for image reconstruction?’, *Inverse Probl.*, vol. 25, no. 12, p. 123009, 2009.

[10] N. Chetih and Z. Messali, ‘Tomographic image reconstruction using filtered back projection (FBP) and algebraic reconstruction technique (ART)’, in *2015 3rd International Conference on Control, Engineering & Information Technology (CEIT)*, IEEE, 2015, pp. 1–6.

[11] B. Zhu, J. Z. Liu, S. F. Cauley, B. R. Rosen, and M. S. Rosen, ‘Image reconstruction by domain-transform manifold learning’, *Nature*, vol. 555, no. 7697, pp. 487–492, Mar. 2018, doi: 10.1038/nature25988.

[12] A. F. Rodriguez, W. E. Blass, J. H. Missimer, and K. L. Leenders, ‘Artificial neural network Radon inversion for image reconstruction.’, *Med. Phys.*, vol. 28, no. 4, pp. 508–514, Apr. 2001, doi: 10.1118/1.1357222.

[13] P. Paschalis *et al.*, ‘Tomographic image reconstruction using Artificial Neural Networks’, *Nucl. Instrum. Methods Phys. Res. Sect. Accel. Spectrometers Detect. Assoc. Equip.*, vol. 527, no. 1, pp. 211–215, 2004, doi: https://doi.org/10.1016/j.nima.2004.03.122.

[14] M. Argyrou, D. Maintas, C. Tsoumpas, and E. Stiliaris, ‘Tomographic Image

Reconstruction based on Artificial Neural Network (ANN) techniques', in *2012 IEEE Nuclear Science Symposium and Medical Imaging Conference Record (NSS/MIC)*, 2012, pp. 3324–3327. doi: 10.1109/NSSMIC.2012.6551757.
[15] T. Würfl, F. C. Ghesu, V. Christlein, and A. Maier, 'Deep Learning Computed Tomography', in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016*, S. Ourselin, L. Joskowicz, M. R. Sabuncu, G. Unal, and W. Wells, Eds., Cham: Springer International Publishing, 2016, pp. 432–440.
[16] Y. Ge *et al.*, 'ADAPTIVE-NET: deep computed tomography reconstruction network with analytical domain transformation knowledge.', *Quant. Imaging Med. Surg.*, vol. 10, no. 2, pp. 415–427, Feb. 2020, doi: 10.21037/qims.2019.12.12.
[17] G. Ma, Y. Zhu, and X. Zhao, 'Learning Image From Projection: A Full-Automatic Reconstruction (FAR) Net for Computed Tomography', *IEEE Access*, vol. 8, pp. 219400–219414, 2020, doi: 10.1109/ACCESS.2020.3039638.
[18] X. Yang and C. Schroer, 'Strategies of Deep Learning for Tomographic Reconstruction', in *2021 IEEE International Conference on Image Processing (ICIP)*, Sep. 2021, pp. 3473–3476. doi: 10.1109/ICIP42928.2021.9506395.
[19] A. M. Beale, S. D. Jacques, E. K. Gibson, and M. Di Michiel, 'Progress towards five dimensional diffraction imaging of functional materials under process conditions', *Coord. Chem. Rev.*, vol. 277, pp. 208–223, 2014.
[20] A. M. Beale, S. D. Jacques, and B. M. Weckhuysen, 'Chemical imaging of catalytic solids with synchrotron radiation', *Chem. Soc. Rev.*, vol. 39, no. 12, pp. 4656–4672, 2010.
[21] G. Artioli *et al.*, 'X-ray Diffraction Microtomography (XRD-CT), A Novel Tool for Non-Invasive Mapping of Phase Development in Cement Materials', *Anal. Bioanal. Chem.*, vol. 397, pp. 2131–6, 2010, doi: 10.1007/s00216-010-3649-0.
[22] G. Harding, J. Kosanetzky, and U. Neitzel, 'X-ray diffraction computed tomography', *Med. Phys.*, vol. 14, no. 4, pp. 515–525, 1987.
[23] F. K. Mürer *et al.*, '3D maps of mineral composition and hydroxyapatite orientation in fossil bone samples obtained by X-ray diffraction computed tomography', *Sci. Rep.*, vol. 8, no. 1, p. 10052, 2018.
[24] M. G. O'Brien, S. D. Jacques, M. Di Michiel, P. Barnes, B. M. Weckhuysen, and A. M. Beale, 'Active phase evolution in single Ni/Al 2 O 3 methanation catalyst bodies studied in real time using combined μ-XRD-CT and μ-absorption-CT', *Chem. Sci.*, vol. 3, no. 2, pp. 509–523, 2012.
[25] G. B. M. Vaughan *et al.*, 'ID15A at the ESRF – a beamline for high speed operando X-ray diffraction, diffraction tomography and total scattering', *J. Synchrotron Radiat.*, vol. 27, no. 2, Art. no. 2, Mar. 2020, doi: 10.1107/S1600577519016813.
[26] C. Vanhoof, J. R. Bacon, U. E. Fittschen, and L. Vincze, '2020 atomic spectrometry update–a review of advances in X-ray fluorescence spectrometry and its special applications', *J. Anal. At. Spectrom.*, vol. 35, no. 9, pp. 1704–1719, 2020.

[27] A. Vamvakeros *et al.*, ‘Real-time multi-length scale chemical tomography of fixed bed reactors during the oxidative coupling of methane reaction’, *J. Catal.*, vol. 386, pp. 39–52, 2020.
[28] V. Middelkoop *et al.*, ‘3D printed Ni/Al2O3 based catalysts for CO2 methanation-a comparative and operando XRD-CT study’, *J. CO2 Util.*, vol. 33, pp. 478–487, 2019.
[29] D. Matras *et al.*, ‘Operando and postreaction diffraction imaging of the La–Sr/CaO catalyst in the oxidative coupling of methane reaction’, *J. Phys. Chem. C*, vol. 123, no. 3, pp. 1751–1760, 2019.
[30] D. S. Wragg *et al.*, ‘Mapping the coke formation within a zeolite catalyst extrudate in space and time by operando computed X-ray diffraction tomography’, *J. Catal.*, vol. 401, pp. 1–6, Sep. 2021, doi: 10.1016/j.jcat.2021.07.001.
[31] D. Matras *et al.*, ‘Effect of thermal treatment on the stability of Na-Mn-W/SiO 2 Catalyst for the Oxidative Coupling of Methane’, *Faraday Discuss.*, 2020.
[32] D. Matras *et al.*, ‘In situ X-ray diffraction computed tomography studies examining the thermal and chemical stabilities of working Ba 0.5 Sr 0.5 Co 0.8 Fe 0.2 O 3- δ membranes during oxidative coupling of methane’, *Phys. Chem. Chem. Phys.*, vol. 22, no. 34, pp. 18964–18975, 2020.
[33] A. Vamvakeros *et al.*, ‘Real-time tomographic diffraction imaging of catalytic membrane reactors for the oxidative coupling of methane’, *Catal. Today*, 2020.
[34] I. Martens *et al.*, ‘Imaging heterogeneous electrocatalyst stability and decoupling degradation mechanisms in operating hydrogen fuel cells’, *ACS Energy Lett.*, vol. 6, no. 8, pp. 2742–2749, 2021.
[35] T. Li *et al.*, ‘Design of next-generation ceramic fuel cells and real-time characterization with synchrotron X-ray diffraction computed tomography’, *Nat. Commun.*, vol. 10, no. 1, Art. no. 1, Apr. 2019, doi: 10.1038/s41467-019-09427-z.
[36] J. Sottmann *et al.*, ‘Chemical Structures of Specific Sodium Ion Battery Components Determined by Operando Pair Distribution Function and X-ray Diffraction Computed Tomography’, *Angew. Chem. Int. Ed.*, vol. 56, no. 38, pp. 11385–11389, 2017.
[37] G. Tonin, G. B. Vaughan, R. Bouchet, F. Alloin, M. Di Michiel, and C. Barchasz, ‘Operando investigation of the lithium/sulfur battery system by coupled X-ray absorption tomography and X-ray diffraction computed tomography’, *J. Power Sources*, vol. 468, p. 228287, 2020.
[38] H. Liu *et al.*, ‘Quantifying reaction and rate heterogeneity in battery electrodes in 3D through operando X-ray diffraction computed tomography’, *ACS Appl. Mater. Interfaces*, vol. 11, no. 20, pp. 18386–18394, 2019.
[39] D. P. Finegan *et al.*, ‘Spatially resolving lithiation in silicon–graphite composite electrodes via in situ high-energy x-ray diffraction computed tomography’, *Nano Lett.*, vol. 19, no. 6, pp. 3811–3820, 2019.
[40] D. P. Finegan *et al.*, ‘Spatial quantification of dynamic inter and intra particle crystallographic heterogeneities within lithium ion electrodes’, *Nat. Commun.*, vol. 11, no. 1, pp. 1–11, 2020.

[41] A. Vamvakeros *et al.*, ‘Cycling Rate-Induced Spatially-Resolved Heterogeneities in Commercial Cylindrical Li-Ion Batteries’, *Small Methods*, vol. 5, no. 9, p. 2100512, 2021, doi: 10.1002/smtd.202100512.
[42] D. Petz *et al.*, ‘Lithium distribution and transfer in high-power 18650-type Li-ion cells at multiple length scales’, *Energy Storage Mater.*, vol. 41, pp. 546–553, Oct. 2021, doi: 10.1016/j.ensm.2021.06.028.
[43] D. Matras *et al.*, ‘Emerging chemical heterogeneities in a commercial 18650 NCA Li-ion battery during early cycling revealed by synchrotron X-ray diffraction tomography’, *J. Power Sources*, vol. 539, p. 231589, Aug. 2022, doi: 10.1016/j.jpowsour.2022.231589.
[44] A. Vamvakeros *et al.*, ‘5D operando tomographic diffraction imaging of a catalyst bed’, *Nat. Commun.*, vol. 9, no. 1, pp. 1–11, 2018.
[45] D. Matras *et al.*, ‘Multi-length scale 5D diffraction imaging of Ni–Pd/CeO2–ZrO2/Al2O3 catalyst during partial oxidation of methane’, *J. Mater. Chem. A*, vol. 9, no. 18, pp. 11331–11346, May 2021, doi: 10.1039/D1TA01464A.
[46] J. Sottmann *et al.*, ‘5D total scattering computed tomography reveals the full reaction mechanism of a bismuth vanadate lithium ion battery anode’, *Phys. Chem. Chem. Phys.*, vol. 24, no. 44, pp. 27075–27085, Nov. 2022, doi: 10.1039/D2CP03892G.
[47] N. V. Y. Scarlett, M. R. Rowles, K. S. Wallwork, and I. C. Madsen, ‘Sample-displacement correction for whole-pattern profile fitting of powder diffraction data collected in capillary geometry’, *J. Appl. Crystallogr.*, vol. 44, no. 1, pp. 60–64, 2011.
[48] A. A. Coelho, ‘TOPAS and TOPAS-Academic: an optimization program integrating computer algebra and crystallographic objects written in C++’, *J. Appl. Crystallogr.*, vol. 51, no. 1, pp. 210–218, 2018.
[49] J. Dong, J. Fu, and Z. He, ‘A deep learning reconstruction framework for X-ray computed tomography with incomplete data’, *PloS One*, vol. 14, no. 11, p. e0224426, 2019.
[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, ‘Image quality assessment: from error visibility to structural similarity’, *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004, doi: 10.1109/TIP.2003.819861.
[51] W. van Aarle *et al.*, ‘The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography’, *Ultramicroscopy*, vol. 157, pp. 35–47, Oct. 2015, doi: 10.1016/j.ultramic.2015.05.002.
[52] A. Vamvakeros *et al.*, ‘Removing multiple outliers and single-crystal artefacts from X-ray diffraction computed tomography data’, *J. Appl. Crystallogr.*, vol. 48, no. 6, Art. no. 6, Dec. 2015, doi: 10.1107/S1600576715020701.
[53] T. M. M. Heenan *et al.*, ‘Mapping internal temperatures during high-rate battery applications’, *Nature*, vol. 617, no. 7961, Art. no. 7961, May 2023, doi: 10.1038/s41586-023-05913-z.

Chapter 6: XRD-CT Image Decomposition

6.1 Abstract

Modern chemical imaging techniques generate vast amounts of high-resolution, multidimensional data, capturing both spatial and spectral information. Image clustering methods are widely applied in categorising chemical imaging datasets into meaningful groups, and reducing the amount of data that needs to be analysed. In this chapter, I will introduce the Self-to-Components (Self2Comp) algorithm, a novel neural network structure designed to decompose complex imaging datasets into component clusters without the need for predefined cluster numbers. This architecture was primarily designed for clustering X-ray diffraction Computed tomography (XRD-CT) images into clusters and corresponding X-ray diffraction (XRD) patterns. The Self2Comp utilises three Deep Neural Networks (DNNs) and essentially works like Non-negative Matrix Factorization (NMF) which factorises datasets into two matrices. The Factor-Net, particularly, controls the scale of each component, which allows the automatic selection of the necessary cluster numbers. Through a combined loss function of Mean Squared Error (MSE), Structural Similarity Index Measure (SSIM), and regularisation of the predicted factor vector using the sigmoid function, the architecture is optimised.

To evaluate its effectiveness, the architecture was tested on both simulated XRD-CT datasets and experimental Li-ion battery XRD-CT images. In the simulated environment, Self2Comp excelled in differentiating the Zn component, a feat that was challenging for conventional methods like NMF. The architecture also performed admirably on experimental datasets, showing its capability to capture phases with weak signals and determine phases like LiC_{30} and polymer separator that NMF could not.

6.2 Introduction

6.2.1 Conventional image clustering methods

Before introducing the new machine learning-based image clustering methods, I will first present the most prevalent and popular conventional techniques used for image clustering. These methods have been widely adopted in various fields due to their effectiveness in categorising datasets based on their similarities and differences.

6.2.1.1 K-means

The K-means clustering algorithm is one of the most popular unsupervised algorithms for multi-dimensional data clustering [1]. It can be used to partition the input data set into k pre-setted clusters. The initial centroid positions are randomly chosen from data points, each data point is assigned to its nearest centroid's cluster S and then the new centroid can be updated according to:

$$\mathbf{c}_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x} \tag{6.1}$$

Here $|S_i|$ represent the total number of points in the cluster S_i, and $\mathbf{c}_i$ is the new position of the centroid. To compute the best centroid positions, the sum of the distance between all points to their centroids is optimised until it is not changing any more [2]:

$$\arg\min_{\mathbf{S}} = \sum_{i=1}^{\mathrm{k}} \sum_{\mathbf{x} \in S_i} \left|\left| \mathbf{x} - \mathbf{c}_i \right|\right|^2 \tag{6.2}$$

The performance of the K-means clustering algorithm can be influenced by the choice of initial points. To counter this, Arthur et al. proposed a K-means++ algorithm that selects the initial centroids in a more intelligent way instead of choosing the random starting points [3], [4]. With K-means++, the initial points are determined as sparsely as possible on the map, and then the traditional K-means algorithm is used to learn the clusters based on these initial points. Since K-means methods generally perform badly on datasets with irregular cluster borders and are computationally difficult for a large number of clusters, the they are often used together with other image pre-processing methods such

as: wavelet transformation [5], density-based spatial clustering of applications with noise (DBSCAN) [6] and the convolutional neural networks (CNNs) [7] before applying K-means methods are applied for image segmentation.

6.2.1.2 Hierarchical Agglomerative Clustering (HAC)

HAC is a hierarchical clustering method that starts by considering each data point as a cluster, and then gradually merging the closest clusters [8]. This merging process is repeated until the desired number of clusters is obtained. The distance metric used to determine which clusters to merge can be based on various measures such as Euclidean distance, cosine similarity, or correlation coefficient [8]–[11].

HAC is widely used in areas of biology [12], social science [13], and image processing [11]. Meanwhile, this method is commonly applied to X-ray diffraction (XRD) data clustering [14]. In 2015, Sarhaddi-Dadian et al. applied HAC to analyse XRD data from archaeological property shards to draw the conclusion that a small cluster of shards had differing chemical compositions due to their being manufactured in different places [15]. Meier et al. used both HAC and principal component analysis (PCA) clustering methods on an XRD dataset with heavy mineral sands and which were revealed to contain $FeTiO_3$, TiO, $ZrSiO_4$, SiO_2 and other minor components [16]. They claimed that these clustering methods successfully identified the outliers before Rietveld analysis and provided two cluster standard XRD patterns among 64 XRD experimental patterns, which can accelerate data processing in industrial production control.

Although HAC has been widely used, its shortcomings cannot be ignored. HAC has a time complexity of $O(N^3)$ i.e. that the running time grows cubically as the size of the input increases, which can become computationally expensive for large datasets [17]. Meanwhile, it is not robust to noise and outliers, which can cause the formation of spurious clusters or the merging of unrelated clusters [18]. Therefore, some new linkage methods to select the closest clusters are implemented to improve the performance of HAC. The most commonly used

linkage method is Ward's linkage which typically calculates the distance between clusters using a minimum variance criterion, which measures the increase in the total within-cluster sum of squares when two clusters are merged [19].

6.2.1.3 Spectral Clustering

Spectral clustering is a method that groups data by utilising eigenvectors derived from a similarity matrix [20]. This technique is widely used for clustering high-dimensional and non-linear data [21]. If we represent the original dataset in the form of an adjacency matrix $\mathbf{A}$, the graph Laplacian ($\mathbf{L}$) can then be represented as [22]

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{6.3}$$

Here $\mathbf{D}$ is a diagonal degree matrix that each element is the row sum of the matrix $\mathbf{A}$:

$$\mathbf{D}_{ii} = \sum_{j} \mathbf{A}_{ij} \tag{6.4}$$

By taking the smallest eigenvectors of the Laplacian matrix first and doing the K-means analysis on their elements, the original dataset can then be clustered into *K* groups.

Due to the purpose of applying spectral clustering, the normalised graph Laplacian is also popular in dimension reduction [22]:

$$L_{\text{rw}} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A} \tag{6.5}$$

The random walk normalised Laplacian L_{rw} tends to cluster the dataset into groups where data points are evenly presented, instead of forcing all clusters to have a similar number of points.

Spectral clustering and its extensions are widely used in image segmentation and decomposition problems. Chang et al. applied the robust path-based spectral clustering measure to define the similarity between data points. They claimed that, compared with the conventional spectral clustering, k-means

clustering and the path-based spectral clustering, the new method has the smallest standard deviation of errors against the manually segmented dataset. It also indicates that the robust path-based spectral clustering becomes more accurate on both supervised and unsupervised occasions against noise and outliers [23]. In 2019, Xia et al. proposed the oriented grouping-constrained spectral clustering (OGCSC) method that takes into account the prior knowledge about the dataset in defining the similarities between points [24]. They applied the new method to medical images and showed that OGCSC could utilise semi-supervised data to fulfil the needs of image segmentation, however, due to the multiple matrix inversion used in their algorithm, this method is inefficient on large datasets like decomposing XRD-CT images. Yoon et al. applied spectral clustering to classify 7214 single-particle X-ray diffraction images [25]. Among the 24 predicted clusters, the agreement between the manually clustered images and their predicted groups was 90 % while the manually defined clusters have only 95 % confidence of correctness. They also claimed that this method can handle over 10^7 images at the same time, which has great potential to be applied to high throughput image analysis. Li et al. proposed a new combined method that uses One-versus-One (OVO) decomposition on the multi-dimensional dataset first and then applies spectral clustering on the low dimensional data points to group the data into classes [26]. They claimed that the new joint method can yield more accurate results than directly applying the spectral clustering on high dimensional datasets in terms of the precision of the cluster with the smallest size ($P_{\min}$) [27] and the average precision of all clusters (P_{avg}) [28]. Here the precision is defined as

$$P = \frac{\text{true postive rate}}{\text{true positive rate} + \text{false positive rate}} \tag{6.6}$$

6.2.1.4 Principal Component Analysis (PCA)

PCA is another commonly used method for decomposing large datasets that have many dimensions or features into several components that are sorted by

importance [29]. Its purpose is to enhance data interpretability while retaining as much information as possible and making it easier to visualise data with multiple dimensions [30]. PCA has wide usage in analysing XRD data. In 2005, Chen et al. applied the smoothed PCA (SPCA) to identify weak but important diffraction peaks that are masked by noise in individual patterns and then de-noising the patterns by combining back with the main principal components [31]. By introducing a smoothing filter before applying PCA to the real XRD patterns, SPCA becomes more robust to noise compared to traditional PCA, resulting in a higher peak ratio of the suspensions and the solvent. In 2019, Conterosito et al. used PCA in analysing *in situ* single-crystal X-ray diffraction experiment data. The utilisation of PCA made it possible to identify and remove problematic data sets while also gaining a more comprehensive understanding of the trends related to the occupancies of CO_2, Na+, and water [32]. One year later, the same group applied PCA to both the unprocessed intensities acquired from *in situ* X-ray powder diffraction experiments and to the parameters acquired through Rietveld refinement [33]. The use of PCA proved highly beneficial in recognizing and assigning all alterations that took place during the adsorption/desorption experiments. Oddershede et al. applied a combination of powder diffraction and PCA to analyse the inner radius of bulk multi-walled carbon nanotubes [34]. They claimed that by applying the PCA, the data dimensionality was strongly reduced, and it was easier to extract physical information from the enhanced principal component images.

6.2.1.5 Non-negative matrix factorization (NMF)

Non-negative matrix factorization (NMF) is an unsupervised learning dimension reduction method that was created by Lee & Seung in 1999 [35]. It is a versatile algorithm that makes a parts-based representation of input data with all positive values [36]. This method is widely used in image clustering, data compression and signal separation. The NMF can factorise the original dataset matrix **V** with two matrices **W** and **H** with the following function:

$$\mathbf{V} = \mathbf{W} \cdot \mathbf{H} \tag{6.7}$$

Where: **V** is the original matrix with m x n elements, **W** represents a matrix including a set of basis images, and **H** contains the coefficients for each basis image. **W** and **H** have m x p and p x n elements respectively, where p represents the number of clusters to be identified. After the number of clusters is determined, the two matrices can be updated simply by gradient descent with its Euclidean distance to the original matrix [37].

Apart from the basic Euclidean NMF, the Kullback-Leibler NMF (KL-NMF) that minimises the Kullback-Leibler divergence between the original matrix **V** and the reconstructed matrix **WH** tends to use only a small number of basis vectors to represent the data [38], [39]. This is beneficial for reducing the dimensionality of the data and extracting meaningful features. β-divergence NMF is another type of NMF that combines both traditional NMF and KL-NMF by introducing a parameter β that controls the ratio of the types of divergence [40]. Because this kind of NMF allows negative values on reconstruction errors, the β-divergence NMF is generally more robust to noise and outliers than other types of NMF on real experimental data [41]. The NMF with Itakura-Saito Divergence is also very popular in spectra/audio decomposing problems [42]. It has been shown to be effective in reducing noise and redundancy in the data [43]. General normalisation methods like the L1 or L2 norm can also be applied to reduce the dimension of features [44]. Adding a regularisation term to the objective function encourages the learned coefficients to be sparse, and focus on the most important and relevant features [45].

In recent years, the NMF method has shown great potential for fast analysis of X-ray diffraction (XRD) data. For example, Long et al. used NMF to extract the 9 basic component X-ray diffraction patterns in hundreds of XRD patterns from a combinatorial materials library. In this way, they dramatically reduced the time taken for patterns to be analysed in the real experimental system [46]. In 2018, Stanev et al. applied the NMF to determine the basic phases' present in a large number of diffraction patterns [47]. They showed that the NMF method can help to identify possible peak-shifted patterns and decompose both the simulated

and the real Fe-Ga-Pd experimental data. In 2021, Maffettone et al. applied constrained NMF with prior knowledge about physical components on synchrotron X-ray total scattering data [41]. They claimed that the constrained NMF can yield XRD decomposition that is more scientifically relevant than the conventional NMF method, demonstrated by lower R-weighted pattern (R_{wp}) errors in the Le Bail refinement results of the individual components. In 2021, Bordet et al. used the PCA, Multivariate Curve Resolution-Alternating Least-Squares (MCR-ALS) and NMF on 3D XRD-CT images to reduce the number of data that need to be analysed [48]. They applied the Rietveld analysis on the six clusters predicted by the NMF and found some minor phases that were not easily seen in the summed data although could be clearly identified as component patterns. However, they also mentioned that the number of components should be specified beforehand for the NMF to predict the components accurately.

6.2.2 Self to Components (Self2Comp) Architecture

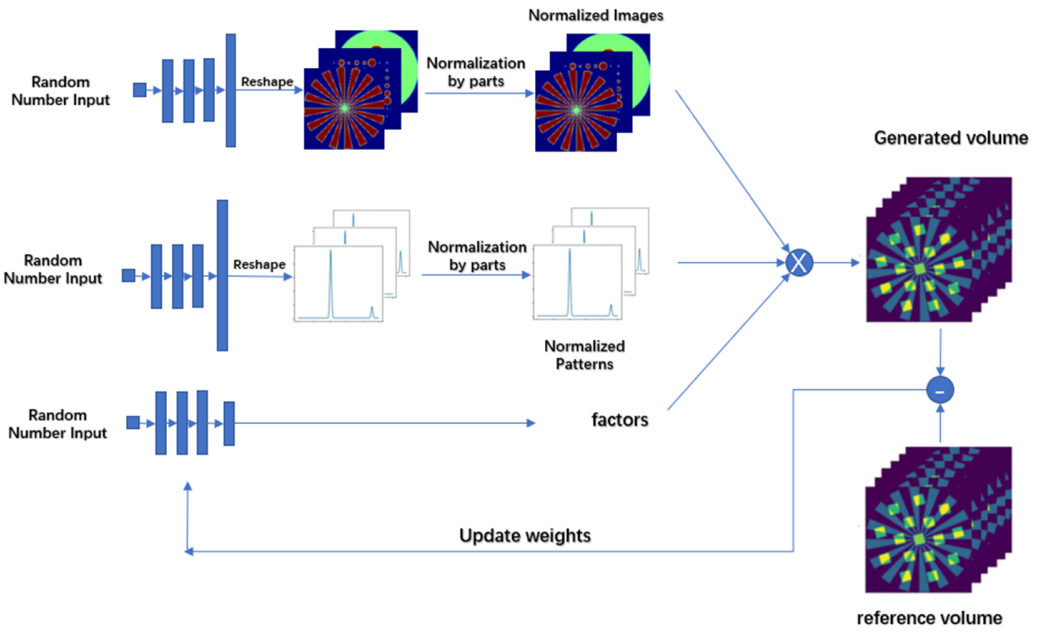

Figure 6.1: Self2Comp construction. The model consists of three networks all trained in parallel; one network creates the component images (Image-Net, top), one the component spectra/patterns (Pattern-Net, middle) and one the adjustment (scale) factors (Factor-Net, bottom). The component images and spectra are normalised before being combined with the adjustment factors to yield a simulated chemical imaging dataset; this is compared with the original one through the network's loss function. Finally, the networks' weights are updated.

The Self2Comp structure is shown in Figure 6.1. Self2Comp consists of three networks, each one having five dense layers with only a single number as input: Image-Net, Pattern-Net and Factor-Net. Specifically, regarding the three networks: (1) The output layer of the Image-Net comprises m x n x p neurons where m and n represent the image sizes and p represents the number of clusters. The output array is reshaped as a 3D matrix **W** which contains p normalised images which are the cluster maps that the architecture predicts. (2) The output of the Pattern-Net is a 1D array with c x p elements where c represents the number of diffraction channels before it is reshaped to a 2D matrix **H** with p normalised patterns corresponding to the maps predicted by the Image-Net. (3) The Factor-Net is a very small neural network which only has c outputs. The output array **f** contains the predicted adjustment (scale) factors added to combine the normalised images and the normalised patterns to match the original chemical imaging dataset. The basic idea of Self2Comp is similar to the NMF that factorises the original dataset into two matrices **W** and **H**. However, the **W** and **H** are normalised according to each component and generated by two Deep Neural Networks (DNNs). To make the Self2Comp automatically select the number of clusters needed, we introduce the Factor-Net which has only p outputs. Since each component in the **W** and **H** is normalised, the only factor controlling each component's scale is the vector **f** created by the third CNN, which can be regularised easily to use only necessary components. Each element in the n x m x c simulated matrix **S** can then be represented as:

$$\mathbf{S} = \mathbf{W} \cdot \mathbf{H} \cdot \mathbf{f} \tag{6.8}$$

To optimise the Self2Comp, we compare the loss between the simulated matrix **S** and the original dataset **V**. In this case, we use a combined loss function with MSE and SSIM and a regularisation penalty on the predicted **f** vector. In this case, we choose the sigmoid equation as the regularisation function. The impact of different regularisation functions on the performance of the network will be discussed later. Therefore, the overall loss function we use to optimise the Self2Comp can be written as:

$$L_{tot} = L_{\mathrm{MSE}}(\mathbf{S},\mathbf{V}) + L_{\mathrm{SSIM}}(\mathbf{S},\mathbf{V}) + \sum_{i=1}^{p} \mathrm{sigmoid}(\mathbf{f}_i) \tag{6.9}$$

All works shown in this chapter are performed in the Tensorflow 2.8 environment with Intel Xeon Silver 4216 CPU, 350 GB RAM and a Quadro RTX 8000 GPU. Adam was used as the optimisation algorithm with the learning rate set to 0.0005 for all case studies that will be presented in this chapter.

6.3 Results & Discussion

6.3.1 Self2Comp on Simulated XRD-CT image

To test the performance of the Self2Comp architecture, we first tried a simulated clean XRD-CT that is generated by the nDTomo package [49]. The nDTomo package is a simulation tool that was developed for the building of complex phantom data comprising 2D to 5D hyperspectral datasets. We first generated five simulated component maps and then defined five noiseless diffraction patterns corresponding to the Al, Cu, Fe, Pt, and Zn phases of the simulated maps. Each individual pattern has 250 channels. Then the five maps with the patterns are combined together to form a simulated 200 x 200 x 250 XRD-CT data volume. Therefore, each pixel of the XRD-CT image will have a corresponding composition diffraction pattern of the five phases.

For the initial testing, as a comparison, we performed the HAC, K-means, Spectral clustering and NMF together with the Self2Comp on the simulated datasets. The sci-kit-learn python-based machine learning package performs all of these conventional methods [50]. As we can see from Figure 6.2, among all conventional methods, NMF performed the best in decomposing the dataset. However, it struggles to give an accurate prediction of the Zn phase, which may be because it appears everywhere in the volume, and is hard to differentiate from the other components.

The result from the Self2Comp method is presented in the last line of Figure 6.2. It can be clearly seen that the Self2Comp outperforms the NMF in this case as it is able to capture clearly the presence of the Zn component. The quantitative analysis of Self2Comp indicated that the overall SSIM score for the

five clusters with the patterns to the ground truth is over 99.5 %, in the meanwhile, the NMF only achieved 98.3 % accuracy.

We also applied these methods to the pattern domain. Figure 6.3 shows the comparison between the ground truth patterns and the standard patterns we obtained for the clusters predicted with the different methods. Except for the NMF and Self2Comp methods, since the patterns and the clusters are not predicted at the same time with the other methods, the HAC, K-means, spectral clustering and PCA are performed solely on the pattern domain. This figure shows that the NMF and Self2Comp can also yield patterns which are matched with the ground truth, however, the NMF patterns contain more errors (minor peaks) than Self2Comp's. Since the original dataset contains no noise, those minor peaks captured by NMF on each pattern are incorrectly taken from other phases, which may weaken the strengths of the diffraction patterns of other components.

The quantitative analysis is shown in Table 6.1. Taking the maximum of each pattern as 1, the XRD-CT volume is composed of five equal-strength normalised diffraction patterns. The numbers in the table show the weight of the maximum value on the five maps shown in Figure 6.2. According to the table, the Al phase was over-predicted by the NMF when the other phases' maximum values were similar to the ground truth. This result is aligned with the dark patterns we can see from the NMF's Pt map from Figure 6.2, which means that some parts of the Zn phase are predicted incorrectly as the Al phase is with the NMF method.

Table 6.1: The quantitive analysis of the phases predicted by the NMF and Self2Vol methods. The numbers in the tables correspond to the maximum signal strength of each cluster's maps shown in Figure2.

	Al	Cu	Fe	Pt	Zn
Ground Truth	1	1	1	1	1
NMF	1.3382	0.9995	1.0306	1.0357	1.0452
Self2Comp	1.0311	1.0131	1.0381	1.0500	0.9936

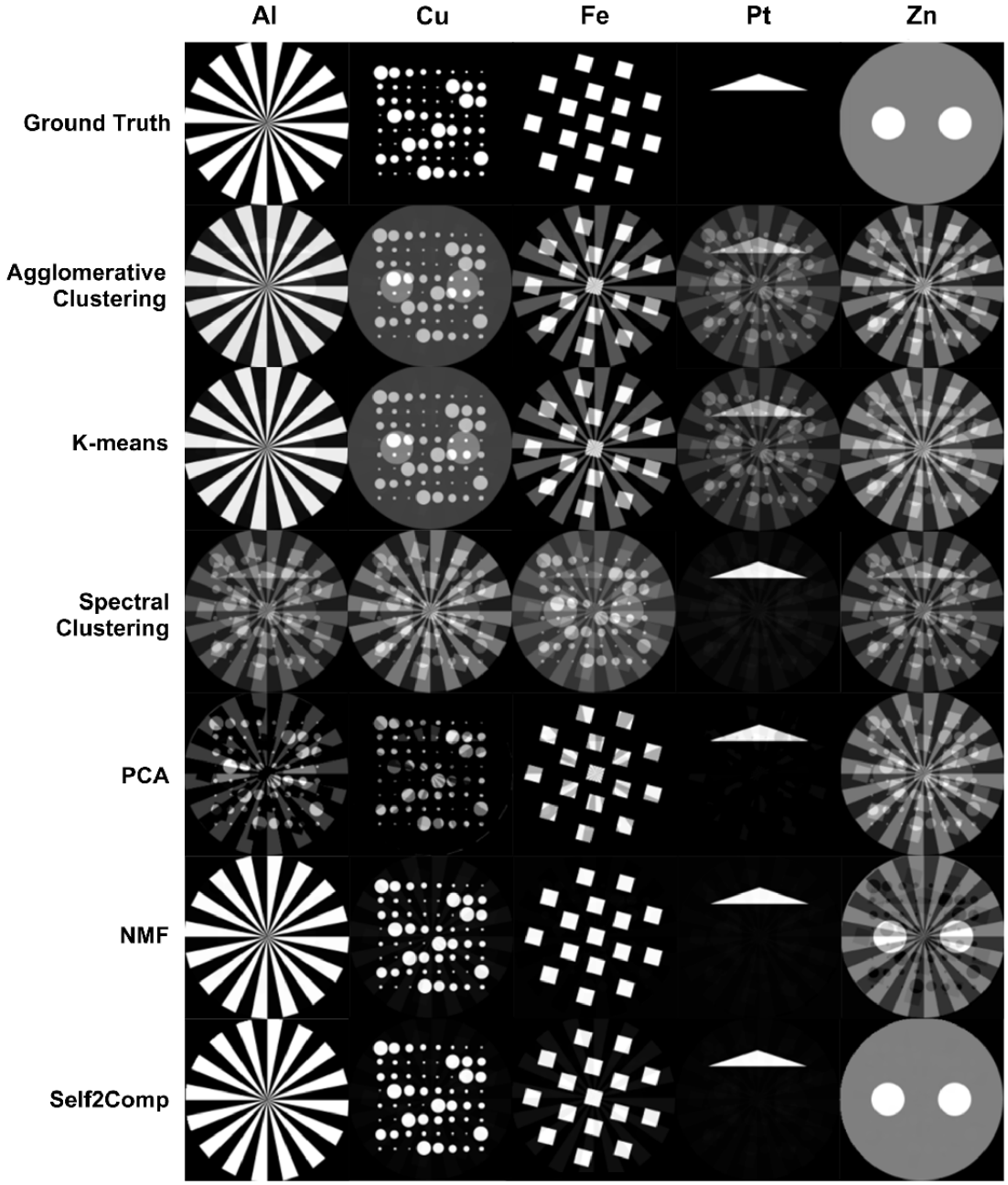

Figure 6.2: Comparison between the Self2Comp and conventional methods on the simulated noise-free dataset. The dataset contains five simulated phases with the same signal strength, and all output classes presented in the figure are normalised. The image size is 200 x 200 and each pixel contains 250 channels of diffraction patterns. The NMF and Self2Comp predicted diffraction pattern associated with the clusters is presented in Figure 6.3.

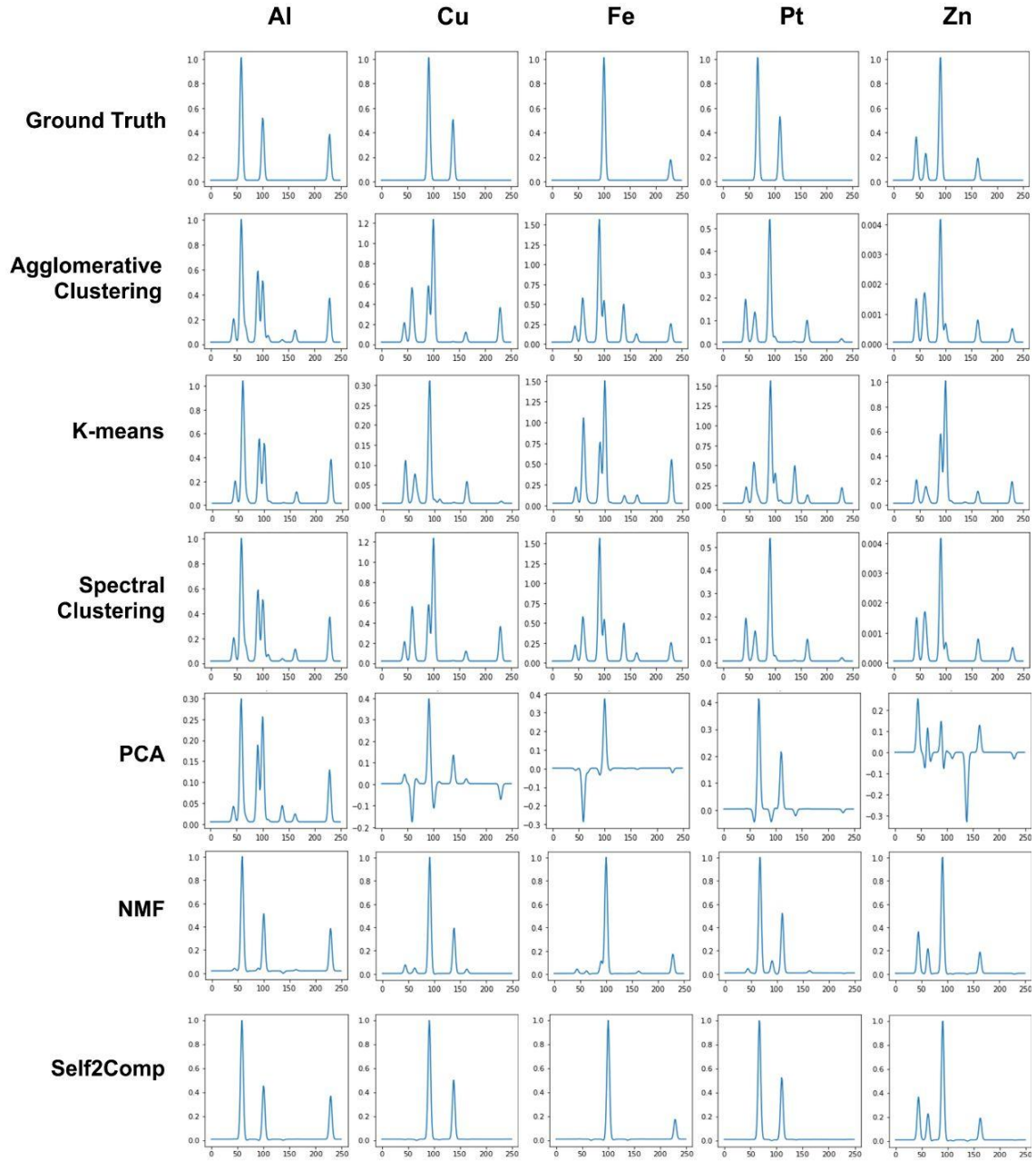

Figure 6.3: Comparison between the Self2Comp and conventional methods on the simulated noise-free dataset's diffraction patterns. The dataset contains five simulated phases with the same signal strength, and all output classes presented in the figure are normalised. The NMF and Self2Comp results are obtained simultaneously with the maps shown in Figure 6.2, The HAC, K-means, Spectral clustering and PCA results are trained separately in the pattern domain.

Furthermore, for a better representation of real-life experimental data, we then created a new composited dataset with unbalanced phases and Poisson noise. Except for the Al phase, which retained a 100 % signal strength compared to the previous dataset, the Cu, Fe, Pt and Zn phases were set to 80, 60, 40 and

20 % signal strength respectively. We then reperformed both the conventional and the Self2Comp network again on the new XRD-CT dataset and the results are shown in Figure 6.4 and Figure 6.5. These two figures indicate that the Self2Comp works well on clustering the Al, Cu, Fe and Pt maps, but it struggled with the presence of the Zn phase which is the weakest of all components. Meanwhile, the NMF can correctly predict the standard patterns and their corresponding signal strength maps for the first four components, and predict accurate diffraction patterns for the Zn phase. Because of the noise added and the more complex nature of this dataset, the SSIM score for the Self2Comp model is reduced to 99.1 %, which is calculated among all pixels on each cluster whereas NMF achieved 97.3 % accuracy.

A quantitative analysis of the results are shown in Table 6.2. It is worth mentioning that this table only represents the signal strength of the strongest component of the signal present in Figure 6.5; it does not give the general difference between the ground truth and the predicted clusters, especially for the Zn phase. According to the table, the Cu phase predicted by NMF is much stronger (0.8847) than the ground truth (0.8), which is not as good as the Self2Comp result (0.8096). Both methods' performance on the Pt clusters is worse than the other clusters and which may be because the two peaks of the Pt pattern overlap with the 2-3 peaks from the Al pattern, which is much more intense.

Table 6.2: The quantitative analysis of the phases predicted by the NMF and Self2Comp methods. The numbers in the tables correspond to the maximum signal strength of each cluster's maps shown in Figure 6.4.

	5 clusters	**10 clusters**	**15 clusters**	**20 clusters**
Ground Truth	1	0.8	0.6	0.4
NMF	1.0397	0.8847	0.6119	0.4232
Self2Comp	1.0113	0.8096	0.6019	0.4247

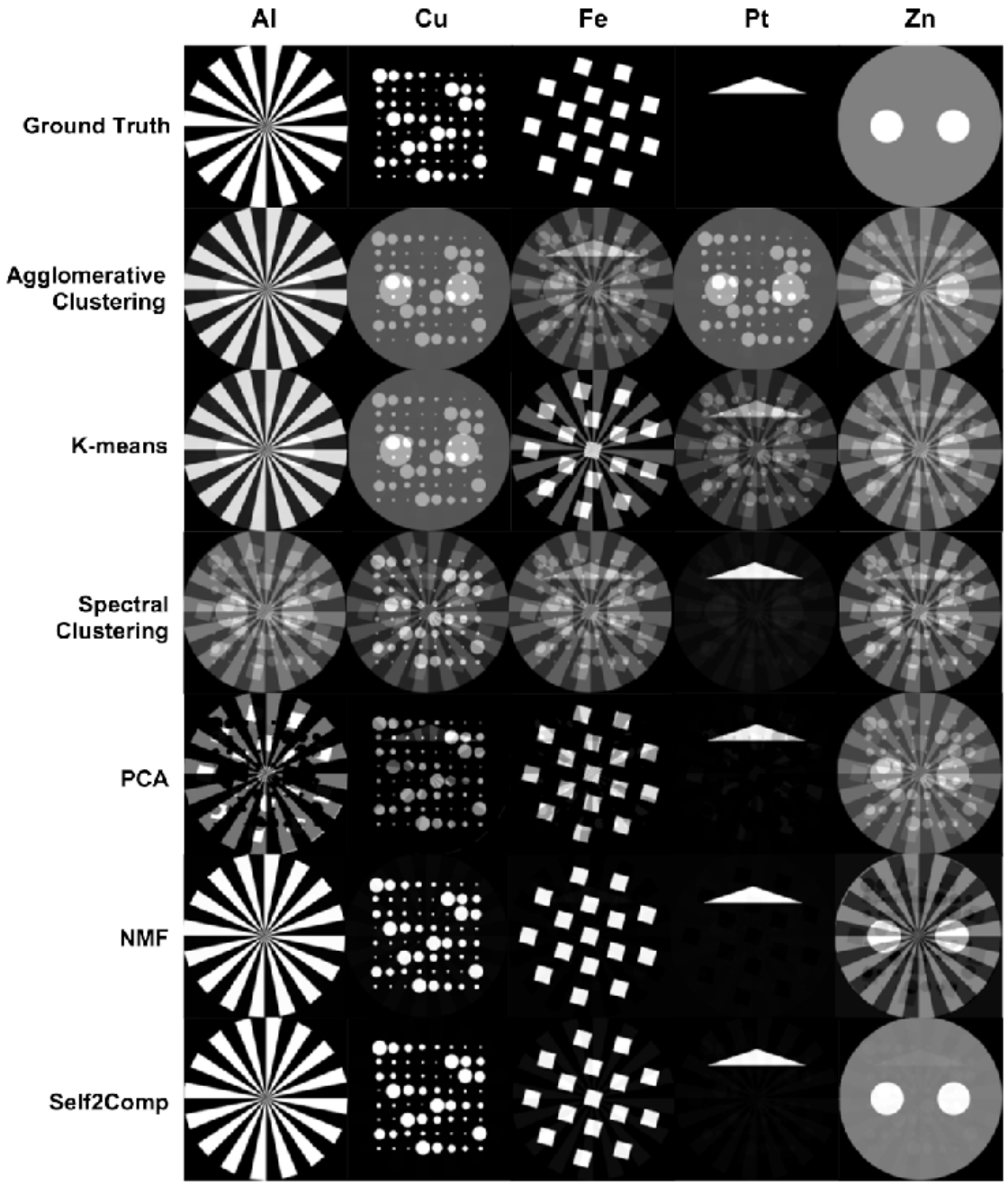

Figure 6.4: Comparison between the Self2Comp and conventional methods on the simulated noisy dataset. The relative signal strength of the metal components compared to Figure 6.2 for the Al, Cu, Fe, Pt and Zn phases were 100, 80, 60, 40, and 20 % respectively, in order to give a better simulation of real experimental data. The image size is 200 x 200 and each pixel contains 250 channels of diffraction patterns. The NMF and Self2Comp predicted diffraction patterns associated with the clusters is presented in Figure 6.5.

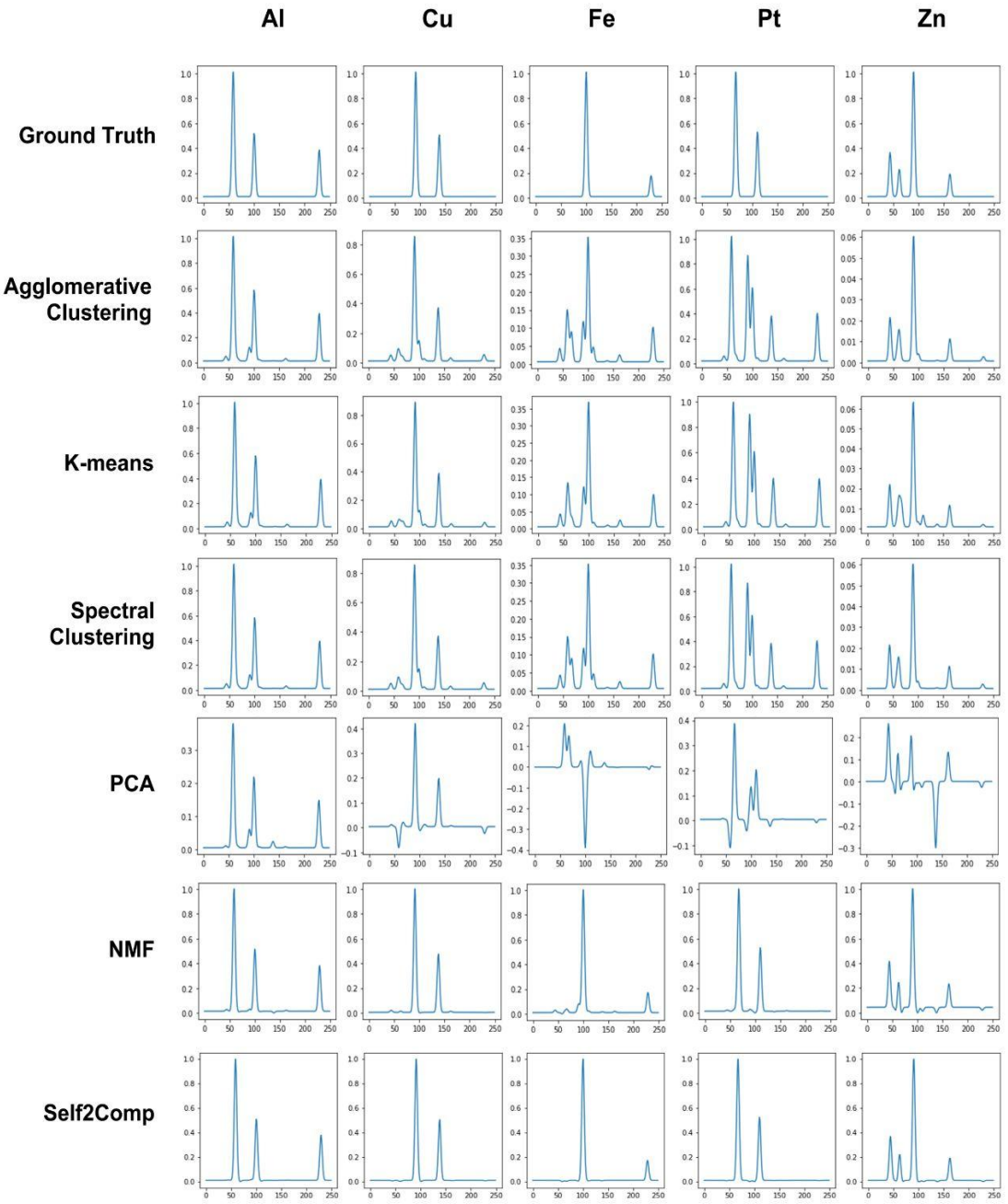

Figure 6.5: Comparison between the Self2Comp and conventional methods on the simulated noisy diffraction patterns. The relative signal strength of the metal components compared to Figure 6.2 for the Al, Cu, Fe, Pt and Zn phases were 100, 80, 60, 40, and 20 % respectively, in order to give a better simulation of real experimental data. The NMF and Self2Comp results are obtained simultaneously with the maps shown in Figure 6.4, The HAC, K-means, Spectral clustering and PCA results are trained separately in the pattern domain.

The Self2Comp clustering technique offers many advantages, one of which is eliminating the need to pre-define the number of components. By applying the sigmoid regularisation function, this network can automatically detect the number of clusters needed [51]. Figure S3.1 in Appendix 3 displays the same

scenario as the unbalanced simulated dataset, with ten components for Self2Comp to predict. Self2Comp can accurately identify the five components, while the remaining five components exhibited very low signal strength compared to the other five components.

The nature of the sigmoid function makes it more sensitive to weak signals than strong signals, which means the regularisation function does not alter the loss function too much for the correct components. Therefore, the Self2Comp can critically evaluate the dataset and determine that fewer components could more accurately represent the dataset. This approach enhances the network's robustness by preventing overfitting that might occur if all components, including those with low signal strength, were given equal consideration.

6.3.2 Self2Comp on Experimental Li-ion Battery XRD-CT Image

We further tested the Self2Comp dataset on an experimental XRD-CT image of a commercial Li-ion AAA battery during its discharge phase with a C/20 rate [52]. The dataset comprises 285 x 285 x 570 voxels and since they are real, has various artefacts such as noise, parallax, and self-absorption. We first applied the ParallaxNet to remove the parallax artefact and then used both Self2Comp and NMF to decompose this experimental dataset. We used ten components in both methods and all the results, along with their corresponding diffraction patterns, are displayed in Appendix 3. Figure 6.6 summarises the comparison between the components obtained from the two methods and the ground truth image. The ground truth is obtained by taking the average among the diffraction channels where the phase's diffraction peak exists (the background is then subtracted). Compared to the results shown in the paper, the eight components shown below cover all important phases present in the dataset. As shown in the figure, the NMF failed to detect the LiC_{30} and the polymer separator phases, and these two phases have the weakest signal strengths among the eight components.

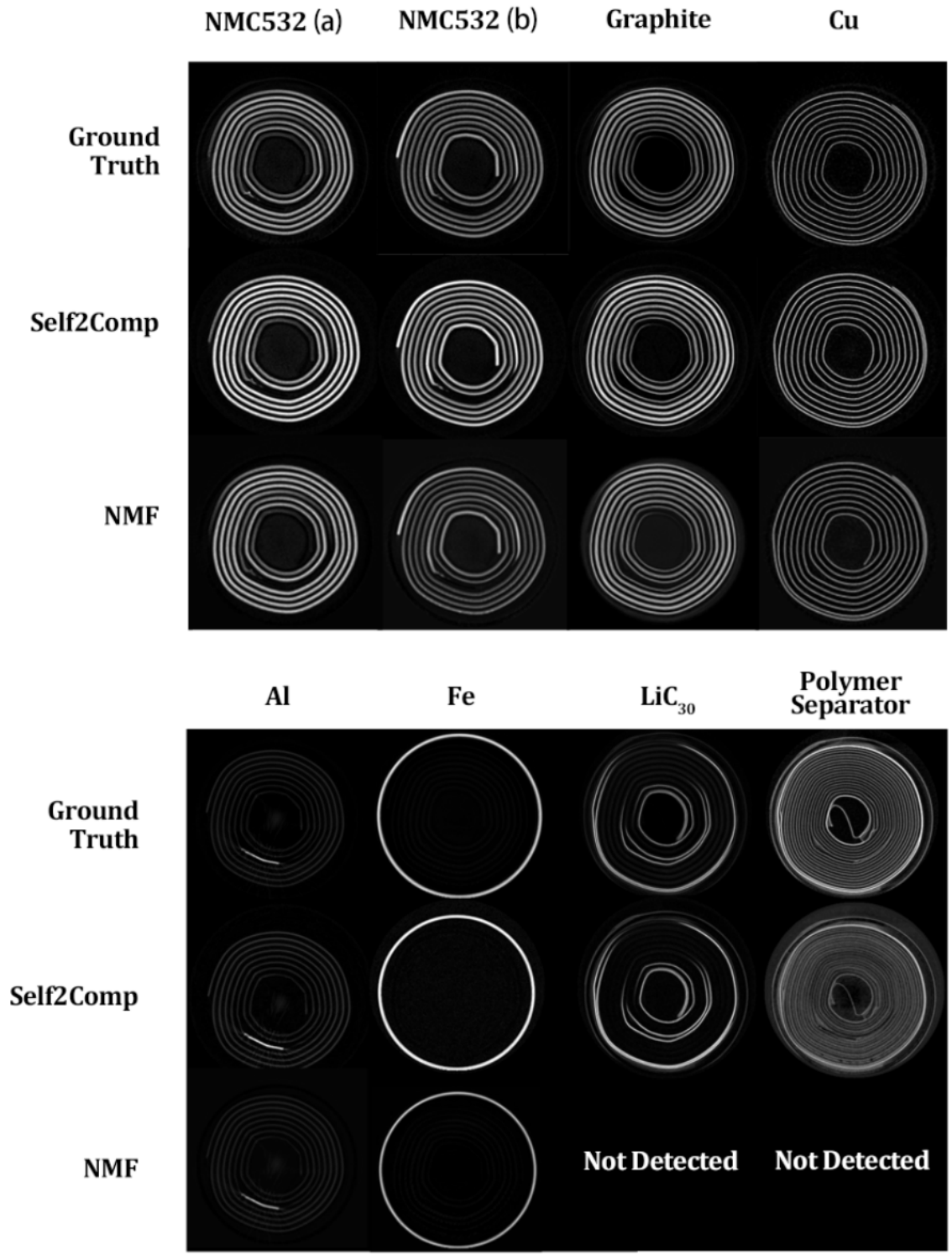

Figure 6.6: Comparison between the Self2Comp and NMF on the experimental AAA Li-ion battery XRD-CT dataset during the discharge phase at C/20 rate. The ground truth image was acquired by averaging the images around the standard peaks of the phases and subtracting the background signal. Both Self2Comp and NMF were configured to predict ten clusters for this dataset. Self2Comp accurately predicted all eight components displayed in the figure, while NMF identified only six.

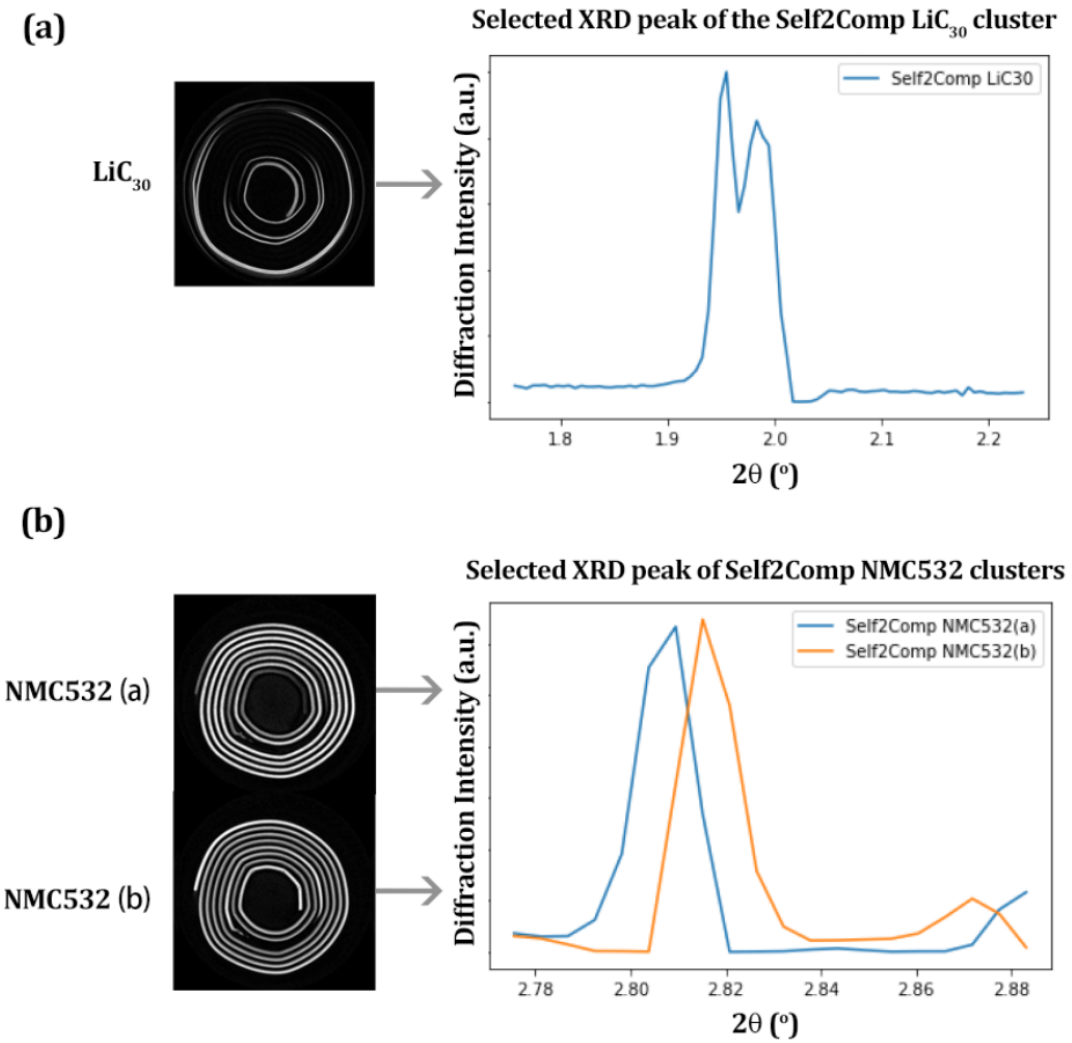

Figure 6.7: Selected XRD pattern peaks associated with the (a) LiC30 cluster (b) NMC532 clusters predicted by the Self2Comp. Both phases undergo a peak shift at various positions of the sample. The different behaviour of the Self2Comp on the number of output clusters is highly related to the phase's signal strength and the ratio of regularisation factor in the Loss function.

Since the NMC532 is a hexagonal lattice and its lattice parameters are varied, the peak shifting occurs among all positions where this phase is present. Therefore, both methods failed to summarise the NMC532 material in the same cluster, instead, they are splitting the NMC532 phase into two sub-clusters which are marked as NMC532(a) and NMC532(b). Interestingly, for another hexagonal lattice, the LiC_{30}, Self2Comp correctly identifies the phase as one cluster, however, the peak splitting is observed in its corresponding diffraction

pattern. The selected peaks of the detected NMC532 and LiC_{30} XRD patterns by Self2Comp are presented in Figure 6.7. In conclusion, for phases with weaker signals, Self2Comp and its Sigmoid function tend to group them into one cluster to seek a lower loss with regularisation. In contrast, for stronger phases, Self2Comp is more inclined to use multiple clusters to reflect different peak positions for a better representation of the original dataset.

6.4 Summary and conclusion

In this chapter, we have presented a new CNN-based image clustering method, Self2Comp. In order to evaluate the performance of the Self2Comp architecture, we first applied it to a clean, simulated XRD-CT dataset generated by the nDTomo package. We generated five simulated component maps and associated noiseless diffraction patterns corresponding to the Al, Cu, Fe, Pt, and Zn phases, each containing 250 channels. These were composited together to form a simulated 200 x 200 x 250 XRD-CT data volume.

For the initial testing, we performed HAC, K-means, Spectral clustering, and NMF, in comparison with Self2Comp on the simulated datasets using the sci-kit-learn python-based machine learning package. NMF performed the best among conventional methods in decomposing the dataset, albeit with difficulty in accurately predicting the Zn phase. However, Self2Comp outperformed NMF, as it was able to clearly capture the Zn component. The quantitative analysis indicated an overall SSIM score of over 99.5 % for Self2Comp, while NMF only achieved 98.3 % accuracy.

For a more accurate representation of real-life experimental data, we created a new dataset with unbalanced phases and Poisson noise. We then reapplied both conventional and Self2Comp networks again on this new dataset. Although Self2Comp struggled with the weakest phase (Zn), it outperformed NMF in the overall SSIM score (99.1 versus 97.3 %).

One significant advantage of Self2Comp is the elimination of the need to pre-define the number of clusters. Through the application of a sigmoid regularisation function, the network can automatically detect the number of required clusters. This prevents overfitting that might occur if all clusters, including those with low signal strength, were given equal consideration. We further tested the Self2Comp model on an experimental XRD-CT image of a commercial Li-ion AAA battery during its discharge phase at C/20 rate. Compared to the ground truth, Self2Comp accurately predicted eight components, whereas NMF only managed to identify six. The failure of NMF to detect the LiC_{30} and the polymer separator phases, both of which have the weakest signal strengths among the eight components, further highlights the robust performance of Self2Comp.

Self2Comp could potentially be applied to other data frames as an alternative choice to NMF. For faster performance, the Self2Comp can directly adapt the NMF clusters as the input and only focus on the remaining components that are missed by the NMF or other conventional techniques. However, at this stage Self2Comp may not necessarily supersede NMF since Self2Comp has more trainable parameters, it generally takes more time (For the simulated dataset NMF took 34 s, and Self2Comp took 380 s. For the experimental dataset, NMF took 13 min, and Self2Comp took 44 min.), and for some simpler problems (with fewer components and without strong artefacts), their performance is actually quite similar. In the future, we could explore potential applications of Self2Comp in various situations. For instance, we could directly cluster sinogram data to eliminate metal artefacts or denoise XRD-CT images by merging components. Furthermore, Self2Comp could have implications in other fields such as audio signal processing, as well as document clustering and topic modelling [53]–[56].

6.5 References

[1] S. Shukla and S. Naganna, ‘A review on K-means data clustering approach’, *Int. J. Inf. Comput. Technol.*, vol. 4, no. 17, pp. 1847–1860, 2014.

[2] K. Krishna and M. Narasimha Murty, ‘Genetic K-means algorithm’, *IEEE Trans. Syst. Man Cybern. Part B Cybern.*, vol. 29, no. 3, pp. 433–439, Jun. 1999, doi: 10.1109/3477.764879.
[3] D. Arthur and S. Vassilvitskii, ‘k-means++: The advantages of careful seeding’, Stanford, 2006.
[4] A. Kapoor and A. Singhal, ‘A comparative study of K-Means, K-Means++ and Fuzzy C-Means clustering algorithms’, in *2017 3rd international conference on computational intelligence & communication technology (CICT)*, IEEE, 2017, pp. 1–6.
[5] N. Gurudath and H. B. Riley, ‘Drowsy driving detection by EEG analysis using wavelet transform and K-means clustering’, *Procedia Comput. Sci.*, vol. 34, pp. 400–409, 2014.
[6] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, ‘DBSCAN revisited, revisited: why and how you should (still) use DBSCAN’, *ACM Trans. Database Syst. TODS*, vol. 42, no. 3, pp. 1–21, 2017.
[7] X. Dong, L. Qian, and L. Huang, ‘Short-term load forecasting in smart grid: A combined CNN and K-means clustering approach’, in *2017 IEEE international conference on big data and smart computing (BigComp)*, IEEE, 2017, pp. 119–125.
[8] D. Müllner, ‘Modern hierarchical, agglomerative clustering algorithms’, *ArXiv Prepr. ArXiv11092378*, 2011.
[9] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, and A. Song, ‘Efficient agglomerative hierarchical clustering’, *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2785–2797, 2015.
[10] L. Muflikhah and B. Baharudin, ‘Document clustering using concept space and cosine similarity measurement’, in *2009 International conference on computer technology and development*, IEEE, 2009, pp. 58–62.
[11] F. Murtagh and P. Contreras, ‘Algorithms for hierarchical clustering: an overview’, *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 2, no. 1, pp. 86–97, 2012.
[12] R. Nugent and M. Meila, ‘An overview of clustering applied to molecular biology’, *Stat. Methods Mol. Biol.*, pp. 369–404, 2010.
[13] J. R. Fonseca, ‘Clustering in the field of social sciences: That is your choice’, *Int. J. Soc. Res. Methodol.*, vol. 16, no. 5, pp. 403–428, 2013.
[14] Y. Iwasaki, A. G. Kusne, and I. Takeuchi, ‘Comparison of dissimilarity measures for cluster analysis of X-ray diffraction data from combinatorial libraries’, *Npj Comput. Mater.*, vol. 3, no. 1, pp. 1–9, 2017.
[15] H. Sarhaddi-Dadian, Z. Ramli, A. Rahman, and R. Mehrafarin, ‘X-ray diffraction and X-ray fluorescence analysis of pottery shards from new archaeological survey in south region of Sistan, Iran’, *Mediterr. Archaeol. Archaeom.*, vol. 15, no. 3, pp. 45–56, 2015.
[16] U. König and E. Spicer, ‘X-ray diffraction (XRD) as a fast industrial analysis method for heavy mineral sands in process control and automation—Rietveld refinement and data clustering’, in *The 6th International Heavy Minerals Conference ‘Back to Basics’; The Southern African Institute of mining and Metallurgy, Nyala Game Lodge: Natal, South Africa*, 2007.
[17] P. Franti, O. Virmajoki, and V. Hautamaki, ‘Fast agglomerative clustering

using a k-nearest neighbor graph', *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1875–1881, 2006.
[18] G. K. Chen, E. C. Chi, J. M. O. Ranola, and K. Lange, 'Convex Clustering: An Attractive Alternative to Hierarchical Clustering', *PLoS Comput. Biol.*, vol. 11, no. 5, p. e1004228, May 2015, doi: 10.1371/journal.pcbi.1004228.
[19] F. Murtagh and P. Legendre, 'Ward's hierarchical clustering method: clustering criterion and agglomerative algorithm', *ArXiv Prepr. ArXiv11116285*, 2011.
[20] A. Ng, M. Jordan, and Y. Weiss, 'On spectral clustering: Analysis and an algorithm', *Adv. Neural Inf. Process. Syst.*, vol. 14, 2001.
[21] H. Jia, S. Ding, X. Xu, and R. Nie, 'The latest research progress on spectral clustering', *Neural Comput. Appl.*, vol. 24, pp. 1477–1486, 2014.
[22] U. Von Luxburg, 'A tutorial on spectral clustering', *Stat. Comput.*, vol. 17, pp. 395–416, 2007.
[23] H. Chang and D.-Y. Yeung, 'Robust path-based spectral clustering', *Pattern Recognit.*, vol. 41, no. 1, pp. 191–203, Jan. 2008, doi: 10.1016/j.patcog.2007.04.010.
[24] K. Xia, X. Gu, and Y. Zhang, 'Oriented grouping-constrained spectral clustering for medical imaging segmentation', *Multimed. Syst.*, vol. 26, no. 1, pp. 27–36, Feb. 2020, doi: 10.1007/s00530-019-00626-8.
[25] C. H. Yoon *et al.*, 'Unsupervised classification of single-particle X-ray diffraction snapshots by spectral clustering', *Opt. Express*, vol. 19, no. 17, pp. 16542–16549, Aug. 2011, doi: 10.1364/OE.19.016542.
[26] Q. Li, Y. Song, J. Zhang, and V. S. Sheng, 'Multiclass imbalanced learning with one-versus-one decomposition and spectral clustering', *Expert Syst. Appl.*, vol. 147, p. 113152, Jun. 2020, doi: 10.1016/j.eswa.2019.113152.
[27] X. Yang, Q. Kuang, W. Zhang, and G. Zhang, 'AMDO: An Over-Sampling Technique for Multi-Class Imbalanced Problems', *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 9, pp. 1672–1685, Sep. 2018, doi: 10.1109/TKDE.2017.2761347.
[28] C. Ferri, J. Hernández-Orallo, and R. Modroiu, 'An experimental comparison of performance measures for classification', *Pattern Recognit. Lett.*, vol. 30, no. 1, pp. 27–38, Jan. 2009, doi: 10.1016/j.patrec.2008.08.010.
[29] K. Pearson, 'LIII. On lines and planes of closest fit to systems of points in space', *Lond. Edinb. Dublin Philos. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, 1901.
[30] I. T. Jolliffe and J. Cadima, 'Principal component analysis: a review and recent developments', *Philos. Trans. R. Soc. Math. Phys. Eng. Sci.*, vol. 374, no. 2065, p. 20150202, 2016.
[31] Z. P. Chen *et al.*, 'Enhancing the signal-to-noise ratio of X-ray diffraction profiles by smoothed principal component analysis', *Anal. Chem.*, vol. 77, no. 20, pp. 6563–6570, 2005.
[32] E. Conterosito, L. Palin, R. Caliandro, W. van Beek, D. Chernyshov, and M. Milanesio, 'CO2 adsorption in Y zeolite: a structural and dynamic view by a novel principal-component-analysis-assisted in situ single-crystal X-ray diffraction experiment', *Acta Crystallogr. Sect. Found. Adv.*, vol. 75, no. 2, Art. no. 2, Mar. 2019, doi: 10.1107/S2053273318017618.

[33] E. Conterosito, M. Lopresti, and L. Palin, ‘In situ x-ray diffraction study of Xe and CO2 adsorption in y zeolite: Comparison between Rietveld and PCA-based analysis’, *Crystals*, vol. 10, no. 6, p. 483, 2020.
[34] J. Oddershede, K. Nielsen, and K. Stahl, ‘Using X-ray powder diffraction and principal component analysis to determine structural properties for bulk samples of multiwall carbon nanotubes’, *Z. Für Krist. - Cryst. Mater.*, vol. 222, no. 3–4, pp. 186–192, Mar. 2007, doi: 10.1524/zkri.2007.222.3-4.186.
[35] D. D. Lee and H. S. Seung, ‘Learning the parts of objects by non-negative matrix factorization’, *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
[36] Y.-X. Wang and Y.-J. Zhang, ‘Nonnegative matrix factorization: A comprehensive review’, *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353, 2012.
[37] D. Lee and H. S. Seung, ‘Algorithms for non-negative matrix factorization’, *Adv. Neural Inf. Process. Syst.*, vol. 13, 2000.
[38] Z. Yang, H. Zhang, Z. Yuan, and E. Oja, ‘Kullback-Leibler divergence for nonnegative matrix factorization’, in *Artificial Neural Networks and Machine Learning–ICANN 2011: 21st International Conference on Artificial Neural Networks, Espoo, Finland, June 14-17, 2011, Proceedings, Part I 21*, Springer, 2011, pp. 250–257.
[39] M. Sun, Y. Li, J. F. Gemmeke, and X. Zhang, ‘Speech enhancement under low SNR conditions via noise estimation using sparse and low-rank NMF with Kullback–Leibler divergence’, *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 23, no. 7, pp. 1233–1242, 2015.
[40] C. Févotte and J. Idier, ‘Algorithms for nonnegative matrix factorization with the β-divergence’, *Neural Comput.*, vol. 23, no. 9, pp. 2421–2456, 2011.
[41] P. M. Maffettone, A. C. Daly, and D. Olds, ‘Constrained non-negative matrix factorization enabling real-time insights of in situ and high-throughput experiments’, *Appl. Phys. Rev.*, vol. 8, no. 4, p. 041410, Dec. 2021, doi: 10.1063/5.0052859.
[42] C. Févotte, N. Bertin, and J.-L. Durrieu, ‘Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis’, *Neural Comput.*, vol. 21, no. 3, pp. 793–830, 2009.
[43] C. Févotte, ‘Itakura-Saito nonnegative factorizations of the power spectrogram for music signal decomposition’, in *Machine Audition: Principles, Algorithms and Systems*, IGI Global, 2011, pp. 266–296.
[44] P. O. Hoyer, ‘Non-negative sparse coding’, in *Proceedings of the 12th IEEE workshop on neural networks for signal processing*, IEEE, 2002, pp. 557–565.
[45] S. Sra and I. Dhillon, ‘Generalized nonnegative matrix approximations with Bregman divergences’, *Adv. Neural Inf. Process. Syst.*, vol. 18, 2005.
[46] C. J. Long, D. Bunker, X. Li, V. L. Karen, and I. Takeuchi, ‘Rapid identification of structural phases in combinatorial thin-film libraries using x-ray diffraction and non-negative matrix factorization’, *Rev. Sci. Instrum.*, vol. 80, no. 10, p. 103902, 2009.
[47] V. Stanev, V. V. Vesselinov, A. G. Kusne, G. Antoszewski, I. Takeuchi, and B. S. Alexandrov, ‘Unsupervised phase mapping of X-ray diffraction data by nonnegative matrix factorization integrated with custom clustering’, *Npj Comput Mater*, vol. 4, p. 43, 2018.

[48] P. Bordet, F. Kergourlay, A. Pinto, N. Blanc, and P. Martinetto, ‘Applying multivariate analysis to X-ray diffraction computed tomography: the study of medieval applied brocades’, *J. Anal. At. Spectrom.*, vol. 36, no. 8, pp. 1724–1734, 2021, doi: 10.1039/D1JA00143D.
[49] A. Vamvakeros and H. Dong, ‘nDTomo software suite’, 2019, doi: https://doi.org/10.5281/zenodo.7139214.
[50] F. Pedregosa *et al.*, ‘Scikit-learn: Machine Learning in Python’, *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
[51] T. Doan and A. Takasu, ‘Kernel Clustering With Sigmoid Regularization for Efficient Segmentation of Sequential Data’, *IEEE Access*, vol. 10, pp. 62848–62862, 2022, doi: 10.1109/ACCESS.2022.3182345.
[52] A. Vamvakeros *et al.*, ‘Cycling Rate-Induced Spatially-Resolved Heterogeneities in Commercial Cylindrical Li-Ion Batteries’, *Small Methods*, vol. 5, no. 9, p. 2100512, 2021, doi: 10.1002/smtd.202100512.
[53] J. Le Roux, J. R. Hershey, and F. Weninger, ‘Deep NMF for speech separation’, in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 66–70. doi: 10.1109/ICASSP.2015.7177933.
[54] S. Nie, S. Liang, W. Liu, X. Zhang, and J. Tao, ‘Deep learning based speech separation via NMF-style reconstructions’, *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 11, pp. 2043–2055, 2018.
[55] N. Guan, X. Huang, L. Lan, Z. Luo, and X. Zhang, ‘Graph based semi-supervised non-negative matrix factorization for document clustering’, in *2012 11th International Conference on Machine Learning and Applications*, IEEE, 2012, pp. 404–408.
[56] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller, ‘A deep semi-nmf model for learning hidden representations’, in *International conference on machine learning*, PMLR, 2014, pp. 1692–1700.

Chapter 7: Conclusion and Future Works

7.1 Conclusion

In this thesis, I have established a pipeline for analysing the chemical imaging data, especially the X-ray diffraction Tomographic (XRD-CT) data, with the help of Convolutional Neural Networks (CNNs), supervised and self-supervised learning techniques. The code and dataset availability for this thesis can be found in Appendix 4.

The PQ-Net shown in Chapter 3 successfully extracts physicochemical data from X-ray diffraction (XRD) patterns. Its efficacy was tested across diverse datasets, ranging from simulated single-phase to complex experimental five-phase systems. The study highlighted the principal advantage of the PQ-Net: the capacity to generate diffraction libraries in advance and pre-train the model. This feature allows for real-time assessment of diffraction data during an *in situ/operando* experiment. It is important to stress that PQ-Net's benefits become particularly apparent when dealing with large datasets. The model, once trained, can rapidly generate predictions for numerous diffraction patterns, offering a time-efficient alternative to conventional full-profile analysis methods like the Rietveld method. The PQ-Net is projected to become an indispensable tool for diffraction applications beyond synchrotron experiments, potentially providing a real-time quality inspection in manufacturing or synthesis. The PQ-Net, however, is not intended to supersede the Rietveld method, because PQ-Net requires a large amount of training data, it does not have an advantage over the Rietveld method when the amount of diffraction data that needs to be analysed is small, but rather to augment it by providing a speedy and solid approximation to the global minimum.

Both supervised and self-supervised learning approaches for CT image reconstruction and angular undersampling artefact removal are shown in Chapter 4. In this chapter, a self-supervised deep-learning method SD2I for

tomographic reconstruction is presented, and this architecture has become the fundamental architecture of the ParallaxNet shown in Chapter 5. The SD2I approach exhibits remarkable resilience to angular undersampling, outperforming both conventional iterative and analytical methods. Its computational efficiency surpasses other deep-learning reconstruction techniques, enabling SD2I to handle larger sinograms with lighter hardware requirements. The simplicity of the architecture is maintained by excluding a discriminator network, which adds complexity but not necessarily image quality.

The ability to accurately reconstruct images from sparsely sampled sinograms is pivotal for time-sensitive *in situ/operando* tomography experiments and reducing X-ray doses in medical CT. Despite the current limitation in matching FBP's accuracy and speed, both supervised and self-supervised learning methods have shown potential in suppressing angular undersampling artefacts with real experimental data. The SD2I network could be extended to other tomographic methods and modalities, such as neutron and X-ray fluorescence tomography.

We also illustrated an updated version of SD2I architecture, the ParallaxNet, in Chapter 5. It successfully eliminates parallax artefacts by using a 3D neural network generator framework and a customized parallax forward projector. This approach is validated using three different datasets and offers several advantages over traditional methods, such as the ability to reconstruct accurate peak positions using only half of the CT rotation scans, thereby reducing acquisition times. The peaks are also sharper and more defined compared to those from conventional Filtered Back Projection (FBP) methods.

Compared to the Direct Least-Squares Reconstruction (DLSR), which is a purely mathematical solution for correcting parallax artefacts, ParallaxNet does not require prior knowledge about the sample's chemistry, is more scalable, has lower RAM requirements, and does not require data preprocessing. Despite its promising results, ParallaxNet does have limitations, primarily its computational

time. For example, a large dataset took up to 90 h to process due to GPU memory constraints.

In Chapter 6, we introduced the Self2Comp architecture, a new Convolutional Neural Network (CNN)-based image clustering method, and evaluated its performance against several conventional techniques. Initial tests were conducted on a simulated XRD-CT dataset with and without Poisson noise. Then the Self2Comp was applied to an experimental dataset from a commercial Li-ion AAA battery. Self2Comp was able to detect eight components, whereas the Non-negative Matrix Factorization (NMF) only identified six, failing to detect components with weak signal strengths. However, Self2Comp is computationally more intensive due to its additional trainable parameters. For example, Self2Comp took 380 s on the simulated dataset and 44 min on the experimental one, compared to NMF's 34 s and 13 min, respectively. In future research, experimenting with various optimizers and regularization techniques on Self2Comp could lead to improved speed and enhanced reliability.

7.2 Future Plans

In this thesis, I have illustrated image machine-learning methods to solve individual problems we met in the chemical imaging analysis pipeline. In the future, combined machine learning methods that can solve multiple issues we encountered in one go are worth exploring. For example, the Self2Comp method can be combined with the PQ-Net to indicate the phases that are not included in the training dataset of the PQ-Net. The ParallaxNet can be combined with PQ-Net to directly extract the chemical information from the dataset with parallax artefact. And more other advanced image-denoising methods can also be included in the SD2I and ParallaxNet loops to reconstruct noise-free CT images. For each individual method I presented in this thesis, there are also some improvements that can be made, which will be discussed in the following sections.

7.2.1 Phase Quantification Neural Network (PQ-Net)

The PQ-Net is an adjustable architecture, which can potentially be extended to solve various problems which includes measuring different materials and performing weight and parameter refinement at the same time. At the moment, PQ-Net utilises convolutional layers with filter sizes 25 to 35. However, using multiple convolutional layers with 3 filters instead of a convolutional layer with a large filter can possibly yield better results and make the CNN more flexible. We have conducted some initial tests on this and observed some improvement in the result's accuracy on the simulated test dataset. The optimal number of layers to use and the use of other kinds of feature extraction layers, like the dilated convolutional layers, are worth exploring.

Moreover, currently, PQ-Net is based on the traditional supervised learning approach, which requires a large dataset that is used to train the model, which will be used multiple times during training in order to get the best-fit minimum. This approach, in our experience, can never avoid the overfitting problem, and the dataset will introduce bias to the network. Increasing the number of patterns in the training dataset is the current solution to this. In the paper, we have illustrated how the model is improved when the training dataset size increased from 10,000 patterns to 250,00 patterns. In the future, a potential self-supervised learning approach could mitigate overfitting and bias problems caused by the training dataset. By allowing PQ-Net models to see each simulated diffraction pattern only once during training, and by using a fast, GPU-based forward operator to simulate diffraction patterns based on random input parameters, we could minimise the bias associated with a specific training dataset.

In the future, we also plan to test the PQ-Net's applicability in other analytical methods like X-ray absorption fine structure spectroscopy (XAFS), and fast analysis of powder diffraction data with more than 20 phases present. Finally, we anticipate that the PQ-Net could be a valuable resource for diffraction uses

beyond just synchrotron experiments, including real-time quality checks in manufacturing and synthesis processes.

7.2.2 Image Reconstruction Methods

In Chapter 5, we have illustrated many studies that have utilised Generative Adversarial Networks (GAN) architectures for supervised and self-supervised learning models, but we found that the discriminator did not significantly enhance the final SD2I results. However, we also found that discriminators do help the convergence speed, especially at the early stage of training. Therefore, we are considering the use of a pre-trained discriminator for future studies to potentially accelerate the convergence of the joint loss to a local minimum.

At this stage, the SD2I has shown its ability to reconstruct clean images with only a few parameters. In the future, the pixel learning methods combined with reliable denoising methods may be explored to further reduce the number of parameters needed for running the self-supervised image reconstruction.

7.2.3 XRD-CT Image Decomposition

Currently, the Self2Comp algorithm is only applied to XRD-CT datasets. Future applications of Self2Comp could extend to other data types and fields like audio signal processing and document clustering, because these data are naturally non-negative, like the XRD-CT dataset, and the interpretability of the decomposed components is important [1]–[4]. Meanwhile, a more optimised architecture design is also worth exploring. Now the Self2Comp consists of three neural networks that share the same loss function. However, the different convergence rates of the three networks render the results easier to fall into local minima. In the future, using only one big neural network to replace the three small neural networks is a possible way to improve the current Self2Comp architecture.

7.3 References

[1] J. Le Roux, J. R. Hershey, and F. Weninger, ‘Deep NMF for speech separation’, in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Apr. 2015, pp. 66–70. doi: 10.1109/ICASSP.2015.7177933.

[2] S. Nie, S. Liang, W. Liu, X. Zhang, and J. Tao, ‘Deep learning based speech separation via NMF-style reconstructions’, *IEEEACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 11, pp. 2043–2055, 2018.

[3] N. Guan, X. Huang, L. Lan, Z. Luo, and X. Zhang, ‘Graph based semi-supervised non-negative matrix factorization for document clustering’, in *2012 11th International Conference on Machine Learning and Applications*, IEEE, 2012, pp. 404–408.

[4] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. Schuller, ‘A deep semi-nmf model for learning hidden representations’, in *International conference on machine learning*, PMLR, 2014, pp. 1692–1700.

Appendix 1

No-Splitting PQ-Net Architecture for Multi-Phase System

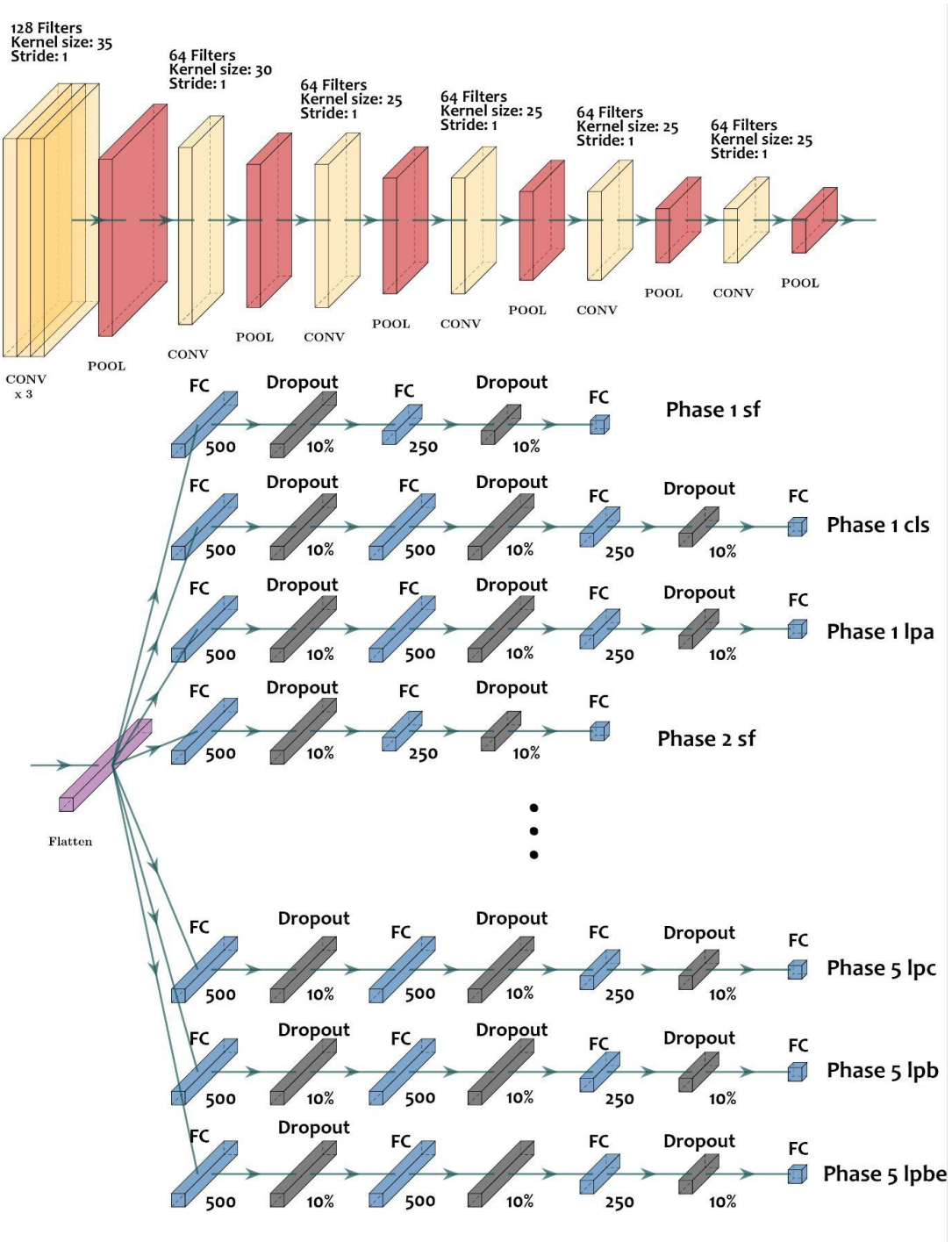

Figure S1.1: Regression CNN architecture for POX catalyst experimental system with five phases. CONV represents 1-D convolutional layers, POOL represents max-pooling layers, FC represents fully connected layers, and FC represents the fully connected layers. This architecture contains 20 outputs which are the same as the outputs in Chapter 3, Figure 5.

Phase parameters for particles-of-interest

Table S1.1:

NiO		Lattice Parameter a (Å)	Crystallite Size (nm)
Particle 1	Ground Truth	4.153	22
	PQ-Net	4.1525	22.2
Particle 2	Ground Truth	4.144	47
	PQ-Net	4.1441	45.9
Particle 3	Ground Truth	4.132	31
	PQ-Net	4.1328	31.2

Table S1.2:

CeO2		Lattice Parameter a (Å)	Crystallite Size (nm)
Particle 1	Ground Truth	5.345	46
	PQ-Net	5.346	45.6
Particle 2	Ground Truth	5.325	2
	PQ-Net	5.329	2.4
Particle 3	Ground Truth	5.305	42
	PQ-Net	5.305	43.1

Table S1.3:

ZrO2		Lattice Parameter a (Å)	Lattice Parameter c (Å)	Crystallite Size (nm)
Particle 1	Ground Truth	3.609	5.217	26
	PQ-Net	3.610	5.217	26.8
Particle 2	Ground Truth	3.618	5.267	21
	PQ-Net	3.624	5.263	8.4
Particle 3	Ground Truth	3.62	5.245	47
	PQ-Net	3.621	5.245	42.9

Table S1.4:

PdO		Lattice Parameter a (Å)	Lattice Parameter c (Å)	Crystallite Size (nm)
Particle 1	Ground Truth	2.998	5.366	48
	PQ-Net	2.999	5.366	47.4
Particle 2	Ground Truth	2.995	5.339	42
	PQ-Net	2.996	5.339	42.5
Particle 3	Ground Truth	3.003	5.337	14
	PQ-Net	3.003	5.338	15.2

Table S1.5:

Al2O3		Lattice Parameter a (Å)	Lattice Parameter b (Å)	Lattice Parameter c (Å)	Lattice Parameter beta (°)	Crystallite Size (nm)
Particle 1	Ground Truth	11.962	11.146	11.089	104.068	19
	PQ-Net	11.964	11.146	11.087	103.838	19.8
Particle 2	Ground Truth	11.957	11.14	11.072	103.501	30
	PQ-Net	11.963	11.137	11.070	103.382	31.8
Particle 3	Ground Truth	11.963	11.15	11.068	104.219	9
	PQ-Net	11.959	11.139	11.074	103.870	9.3

Appendix 2

CNN-rec: Impact of training data type

Library sizes: 20,000

Learning rate : 0.000025

Loss function: mean squared error (mse)

Image Size: 128 × 128 pixels

Batch size: 32

Training time: 16ms/step

Early stopping (patience) : 5

The mse for the validation data is lower due to the dropout layers in the CNN-rec

Table S2.1: The details of training details of reconstruction networks with single training data type

Training library	Epochs	MSE	Val MSE
Ni Phantom	56	2.24×10^{-4}	1.24×10^{-4}
Battery	200	7.50×10^{-4}	6.04×10^{-4}
POX Experimental	117	5.74×10^{-4}	4.25×10^{-4}
3D printed catalyst Experimental	67	8.19×10^{-4}	7.30×10^{-4}
Combined (20 K)	160	9.01×10^{-4}	6.92×10^{-4}
Combined (80 K)	75	8.33×10^{-4}	6.07×10^{-4}

CNN-rec: Metrics and Comparison with FBP

Table S2.2: Ni test dataset images. Metrics calculated with respect to the ground truth images.

Training library	SSIM	PSNR	MSE
Ni Phantom	0.9933	38.0883	0.0002
Battery	0.3677	13.4364	0.0453
POX Experimental	0.4072	13.8146	0.0415
3D printed catalyst	0.5744	16.9376	0.0202
Combined (20 K)	0.9783	33.7930	0.0004
Combined (80 K)	0.9871	35.5392	0.0003
FBP	0.9496	32.7723	0.0005

Table S2.3: Battery test dataset images. Metrics calculated with respect to the ground truth images.

Training library	SSIM	PSNR	MSE
Ni Phantom	0.1133	11.7616	0.0667
Battery	0.9800	29.3750	0.0012
POX Experimental	0.3555	12.1973	0.0603
3D printed catalyst	0.3740	12.2542	0.0595
Combined (20 K)	0.9629	27.2021	0.0019
Combined (80 K)	0.9710	27.7324	0.0017
FBP	0.9250	23.8939	0.0041

Table S2.4: POX test dataset images. Metrics calculated with respect to the ground truth images.

Training library	SSIM	PSNR	MSE
Ni Phantom	0.1510	9.3692	0.1156
Battery	0.5225	13.3934	0.0458
POX Experimental	0.9689	31.4884	0.0007
3D printed catalyst	0.3886	8.6631	0.1360
Combined (20 K)	0.9668	30.2381	0.0009
Combined (80 K)	0.9664	30.8215	0.0008
FBP	0.9663	30.6442	0.0009

Table S2.5: 3D printed catalyst test dataset images. Metrics calculated with respect to the ground truth images.

Training library	SSIM	PSNR	MSE
Ni Phantom	0.3337	11.0777	0.0780
Battery	0.3324	8.5462	0.1398
POX Experimental	0.3450	8.7414	0.1336
3D printed catalyst	0.9752	30.1523	0.0009
Combined (20 K)	0.9632	29.7243	0.0011
Combined (80 K)	0.9727	30.1083	0.0010
FBP	0.9280	27.9155	0.0016

Table S2.6. The influence of the choice of different ground truth values on the metrics. All the images are reconstructed from the same 256x400 sinogram.

		MAE	MSE	SSIM	PSNR
Clean Image as reference	SD2Iu	0.00247	0.000087	**0.9968**	41.3097
	SD2I	**0.00188**	**0.000018**	0.9958	**48.2559**
	FBP	0.00785	0.000786	0.9614	31.7733
	SIRT	0.01020	0.001441	0.9670	29.1397
	SART	0.01799	0.001886	0.7677	27.9702
	CGLS	0.00444	0.000074	0.9480	42.0577
FBP with 400 proj as reference	SD2Iu	0.00775	0.000685	0.9643	32.3659
	SD2If	0.00794	0.000737	0.9624	32.0493
	SIRT	**0.00476**	**0.000186**	**0.9868**	**38.0200**
	SART	0.01327	0.000696	0.8171	32.3003
	CGLS	0.00807	0.000613	0.9731	32.8483
	Clean image	0.00785	0.000786	0.9614	31.7733
CGLS with 400 proj as reference	SD2Iu	0.00503	0.000107	0.9478	40.4165
	SD2If	**0.00454**	**0.000070**	**0.9484**	**42.2571**
	FBP	0.00807	0.000613	0.9731	32.8483
	SIRT	0.01134	0.001249	0.9519	29.7601
	SART	0.01595	0.001380	0.8271	29.3253
	Clean image	0.00444	0.000074	0.9480	42.0577

Appendix 3

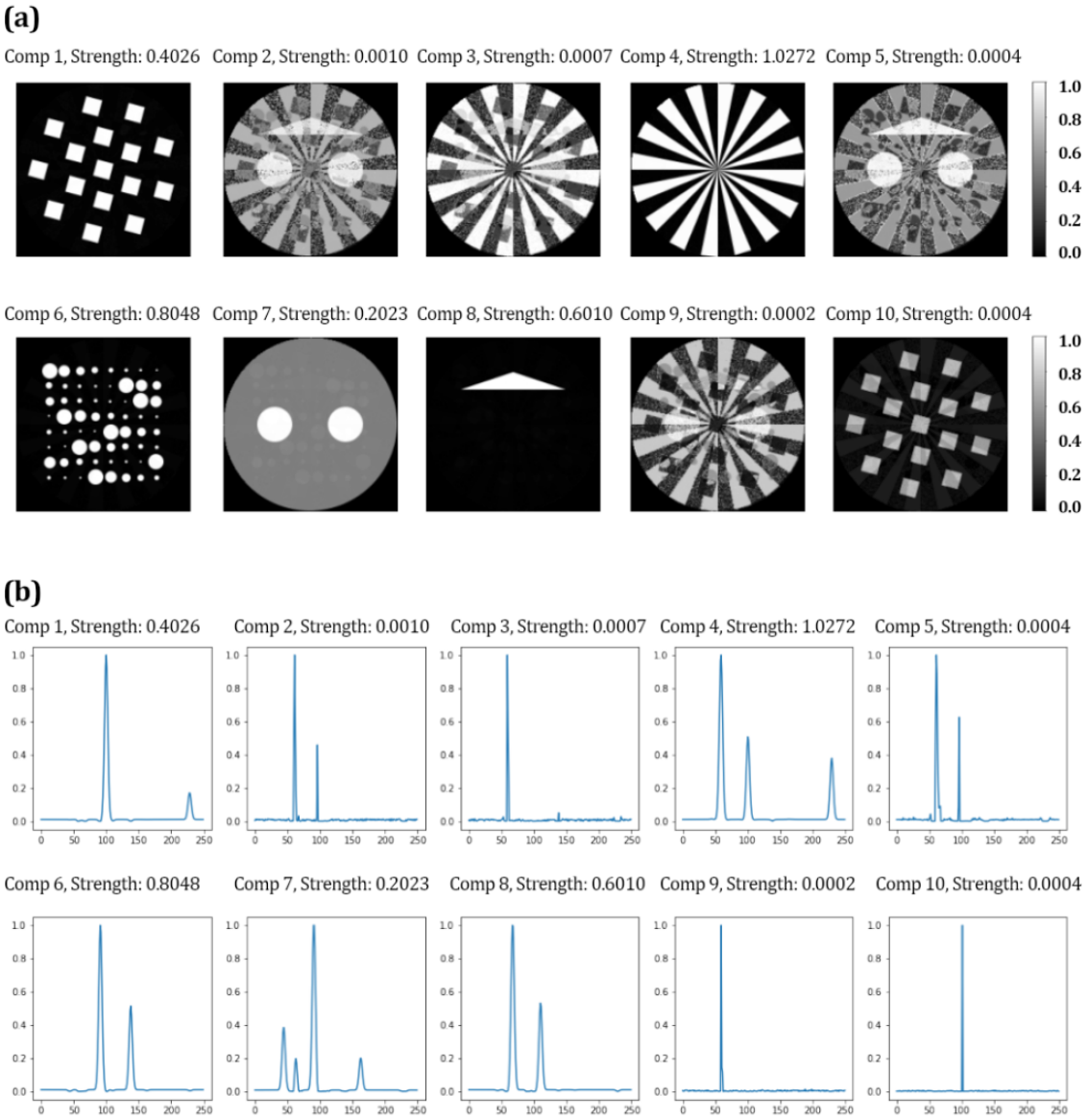

Figure S3.1: Result (a) clusters (b) XRD patterns of the Self2Comp on the simulated unbalanced dataset. All signal strengths are shown above the clusters. It indicates that the Self2Comp correctly located the five components, and eliminates the other five which are not necessary. This result shows that the Self2Comp can automatically select the number of components needed for representing the original dataset.

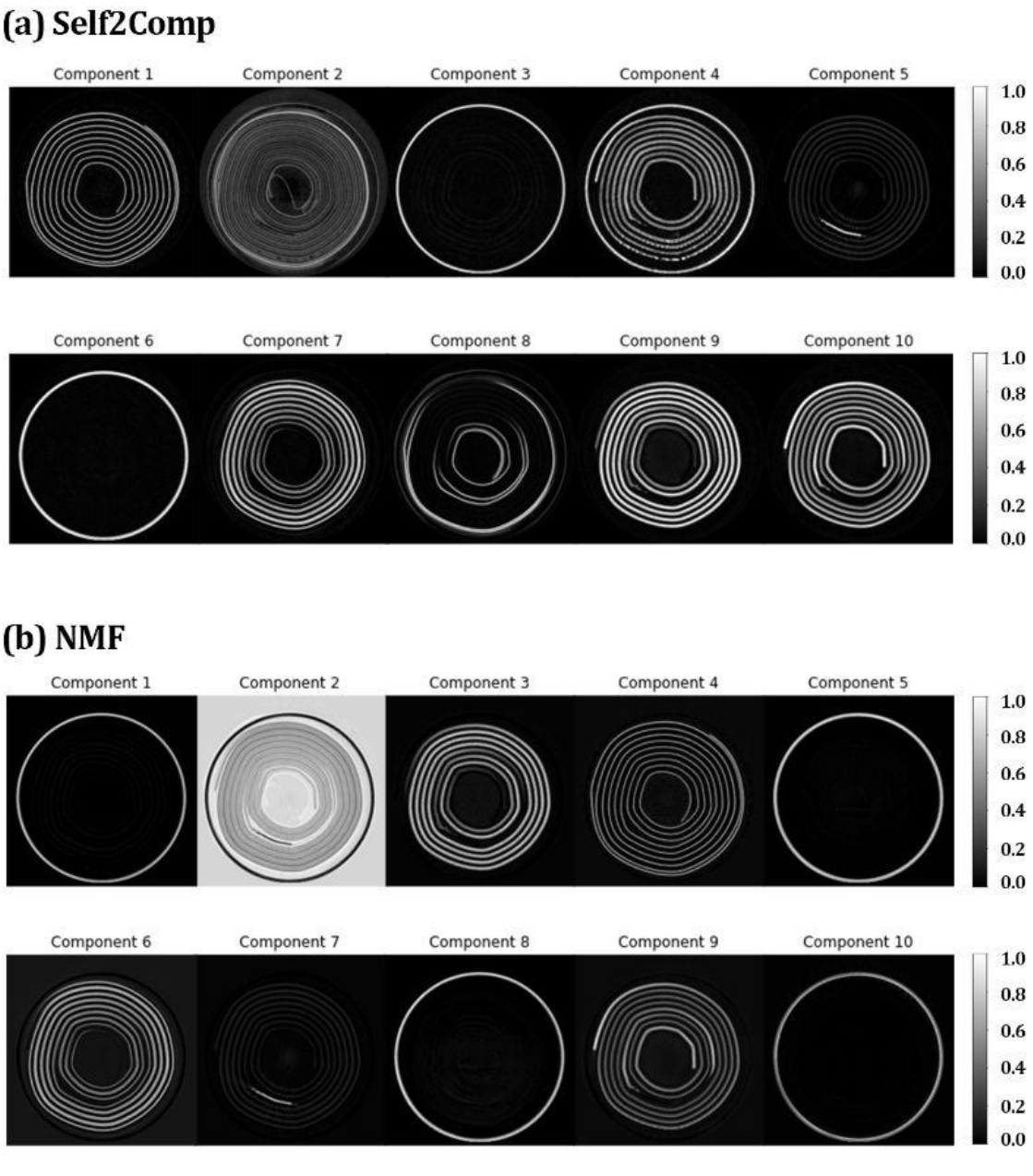

Figure S3.2: Clusters obtained by (a) the Self2Comp algorithm, (b) NMF on experimental AAA Li-ion battery XRD-CT image. Both methods require ten components to represent the original dataset.

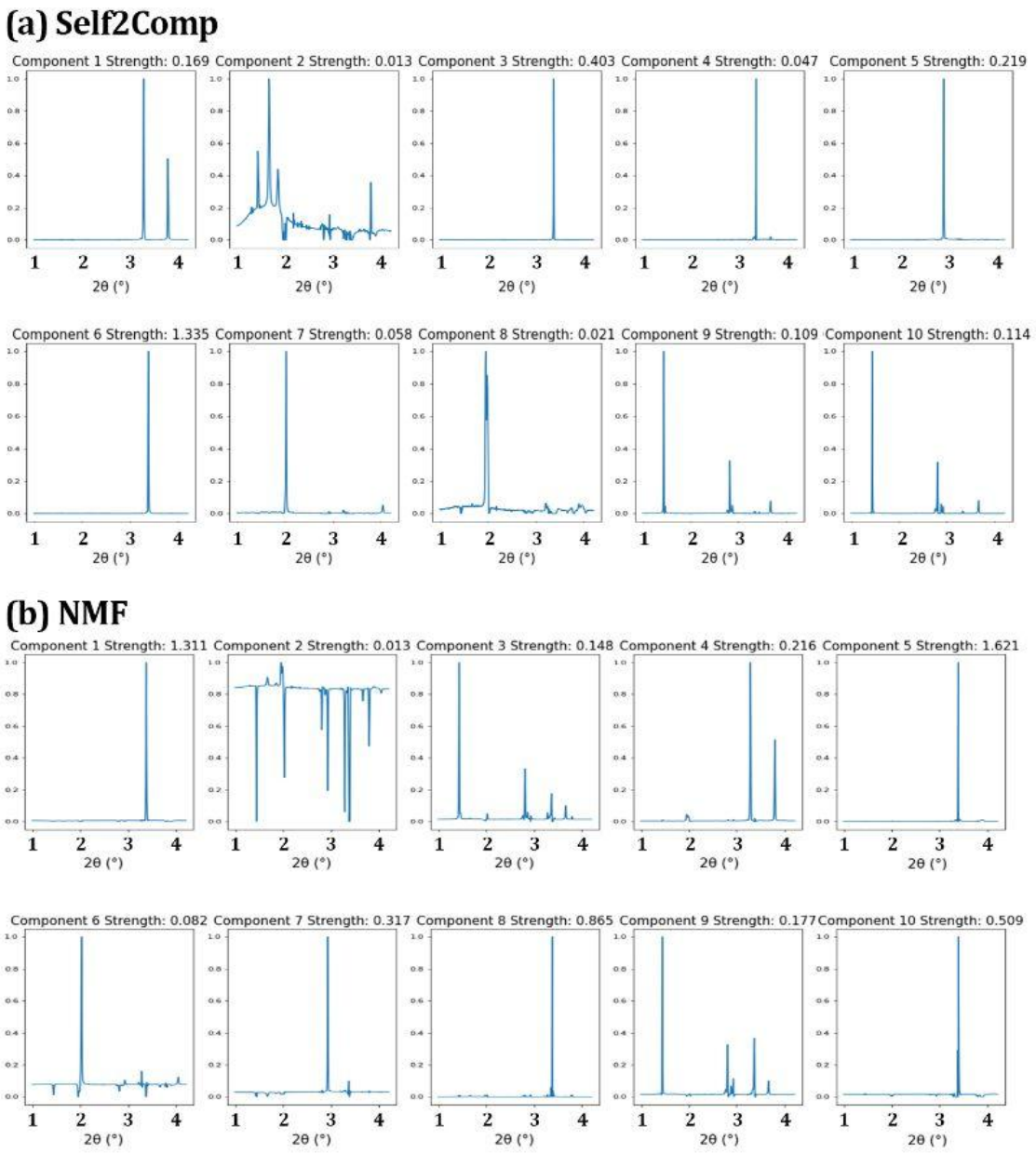

Figure S3.3: Corresponding XRD patterns relate to the clusters shown in Figure S2.1. The patterns are obtained by (a) the Self2Comp algorithm, (b) NMF on an experimental AAA Li-ion battery XRD-CT image. The strength of each pattern is shown above the plots, which represents the signal strength of the maximum value predicted.

Appendix 4

Code / Data Availability For PQ-Net:

The training and test datasets used in Chapter 3 are publicly available through https://doi.org/10.5281/zenodo.4664597. The codes for generating the PQ-Net architecture is co-developed with Finden Ltd. There will be a 12 month moratorium on the publication of the code due to commericial sensitivities. The situation will be reviewed at the 12 month mark.

Code / Data Availability For SD2I:

All codes and data for reproducing the SD2I results shown in Chapter 4 can be found at: https://github.com/robindong3/SD2I

Code / Data Availability For ParallaxNet:

The integrated XRD-CT data presented in Chapter 5, both simulated and experimental, have been made publicly available through an open access repository and can be found here: https://zenodo.org/record/8344637

The ParallaxNet shares the same training loop as the SD2I, but the parallax forward operator and the code for generating the grid of example are co-developed with Finden Ltd. There will be a 12 month moratorium on the publication of the code due to commericial sensitivities. The situation will be reviewed at the 12 month mark.

Code / Data Availability For Self2Comp:

All codes and data for reproducing the Self2Comp results shown in Chapter 6 can be found at: https://github.com/robindong3/self2comp