

# **CONTENTS**

<b>1</b>	<b>Motivation and outlook</b>	<b>1</b>
1.1	Preface . . . . .	1
1.2	Context, PhD objectives and manuscript outline . . . . .	1
<b>2</b>	<b>Bragg Coherent Diffraction Imaging</b>	<b>5</b>
2.1	Foreword on typical assumptions and approximations in BCDI . . . . .	5
2.2	Coherent X-ray scattering from crystalline structures . . . . .	6
2.3	Practical BCDI at ID01 - ESRF . . . . .	21
<b>3</b>	<b>The Phase Problem in BCDI</b>	<b>31</b>
3.1	Oversampling . . . . .	32
3.2	Phase Symmetries . . . . .	34
3.3	Alternating projections algorithms . . . . .	35
3.4	BCDI Phase Retrieval in presence of high-strain . . . . .	39
<b>4</b>	<b>Keynotes on Convolutional Neural Networks</b>	<b>43</b>
4.1	Artificial Neural Networks (ANNs) . . . . .	43
4.2	Convolutional Neural Networks (CNNs) . . . . .	49
<b>5</b>	<b>Deep Learning for Detector Gaps Inpainting</b>	<b>55</b>
5.1	The “Gap Problem” . . . . .	55
5.2	State of the art . . . . .	57
5.3	Model design: 2D case . . . . .	59
5.4	3D case - Patching approach . . . . .	65
5.5	3D model architecture . . . . .	67
5.6	Results in detector space . . . . .	68
5.7	Performance assessment . . . . .	72
5.8	Results in real space . . . . .	74
5.9	Fine-tuning . . . . .	80
5.10	Conclusions . . . . .	82
<b>6</b>	<b>Deep Learning for Phase Retrieval</b>	<b>85</b>
6.1	State of the art . . . . .	86
6.2	Reciprocal space phasing . . . . .	89
6.3	Dataset creation . . . . .	90
6.4	2D case low strain . . . . .	91
6.5	2D high strain case . . . . .	102
6.6	Phasing patches: 3D case low strain . . . . .	106
6.7	Patches: 3D case high strain . . . . .	117
6.8	Final model design: 3D case high-strain for central patch only . . . . .	123
6.9	Refinement with iterative algorithms . . . . .	130
6.10	Performance assessment . . . . .	138
6.11	Other model test . . . . .	147
6.12	Conclusion . . . . .	150

---

42	<b>7 Automatic Differentiation for Phase Retrieval</b>	<b>153</b>
43	7.1 State of the Art . . . . .	154
44	7.2 Model implementation . . . . .	154
45	7.3 Results: simulated data . . . . .	160
46	7.4 Results: experimental data . . . . .	163
47	7.5 Conclusions . . . . .	170
48	<b>8 Conclusions and Perspectives</b>	<b>173</b>
49	<b>Bibliography</b>	<b>191</b>

---

<sup>51</sup> MOTIVATION AND OUTLOOK

<sup>52</sup> 1.1 Preface

<sup>53</sup> In this manuscript, the use of Deep Learning methods, and more generally of GPU accelerated  
<sup>54</sup> optimizations, for the advance of data analysis in Bragg Coherent Diffraction Imaging (BCDI)  
<sup>55</sup> will be presented. However, before delving into the study's developments, I would like to share  
<sup>56</sup> with the reader a reflection that has taken shape over the course of this PhD, serving as a kind  
<sup>57</sup> of preface. In particular, I have come to observe that, unlike other more fundamental scientific  
<sup>58</sup> investigations, this work originates from the practical limitations of the technique in question.  
<sup>59</sup> It is indeed because the detectors are unable to record flawless images due to gaps, or incapable  
<sup>60</sup> of capturing phase information because its oscillations are too rapid—that one is compelled to  
<sup>61</sup> manipulate the available data with sophisticated algorithms. And, as often happens in science,  
<sup>62</sup> compensating for these technical shortcomings leads to the development of tools rooted in  
<sup>63</sup> the most abstract realms of mathematics and information theory. It is here the field of *inverse*  
<sup>64</sup> *problems*, i.e. the study of algorithms and numerical methods for the handling of incomplete  
<sup>65</sup> data. How much information can one extract from a signal? How can it be extracted, and under  
<sup>66</sup> what conditions? In which circumstances is it easier, and why? Thus, a fascinating world opens  
<sup>67</sup> up not when we directly investigate the foundations of matter, but when we examine *how* we  
<sup>68</sup> go about investigating them.

<sup>69</sup> Although this manuscript is ultimately focused on the specific cases of BCDI gap inpainting  
<sup>70</sup> and phase retrieval, I hope to convey at least a bit of the wonder and awe that comes from  
<sup>71</sup> knowing that such applications draw their roots from far deeper, more general, complex, and  
<sup>72</sup> abstract themes.

<sup>73</sup> 1.2 Context, PhD objectives and manuscript outline

<sup>74</sup> Over the past two decades the field of Coherent X-ray Diffraction Imaging (CXDI or CDI) has  
<sup>75</sup> combined the study of microscopic structures enabled by X-rays with the imaging world. This  
<sup>76</sup> powerful connection has been enabled by the availability of high-brilliance and coherent X-ray  
<sup>77</sup> sources and the developments of computer algorithms for Phase Retrieval (PR) [1–3]. In fact,  
<sup>78</sup> unlike conventional imaging methods, instead of relying on high-quality optics to form an

---

79 image, CDI records only the intensity of the diffracted waves on a detector and then reconstructs  
80 the object by iterative algorithms. This approach removes the limitations imposed by imperfect  
81 lenses, enabling spatial resolutions set primarily by the wavelength and the numerical aperture  
82 of the scattering geometry. Since the first work by Miao [4] the CDI technique has been widely  
83 employed in material science for characterization at nanometric scale of different functional  
84 materials. [5–8]

85 When performed in Bragg geometry (Bragg CDI or BCDI), the technique enables precise 3D  
86 visualization and investigation of the internal atomic structure of single-crystal nanoparticles.  
87 Since its first demonstration in 2001 by Robinson et al. [9] for imaging gold nanoparticles, BCDI  
88 has proven to be a powerful tool for studying strain distributions [10–12], defect populations  
89 [13–15], and particle morphologies under different physico-chemical conditions [16–19]. Over  
90 the years, it has been applied to a broad range of technologically relevant systems, including  
91 nanotechnology and electronics [13], Li-ion and Na-ion batteries [20, 21], catalysis [22], and,  
92 more recently, biological materials [23].

93 In typical experiments, the recorded diffraction patterns are processed by PR computer  
94 algorithms returning 3D complex arrays, the modulus of which represents the electron density  
95 of the sample, while the phase is associated to a component of the strain distribution inside  
96 the sample. Unlike other X-ray diffraction techniques, BCDI enables the study of isolated  
97 single particles, down to a few tens of nanometers in size [24, 25], with spatial resolution on  
98 the electron density of the order of 10 nm [26] and sensitivity to strain in the order of a few  
99 picometers [14].

100 BCDI experiments are carried out at beamline laboratories of synchrotron facilities or free  
101 electron lasers (FELs) capable of delivering intense X-ray beams with exceptional coherence  
102 properties. Some examples are the ID01 beamline at the European Synchrotron Radiation  
103 Facility (ESRF) [27], the 34ID beamline at Advance Photon Source (APS) [28], the P10 beamline  
104 at PETRA III, NanoMax at MAX IV [24] or the CARNAÚBA beamline at SIRIUS [29].

105 **Chapter 2** introduces the fundamental theoretical background of this technique and de-  
106 scribes the practical conditions for a BCDI experiment at ID01.

107 The successful application of BCDI critically depends on computational algorithms that  
108 transform measured diffraction patterns into real-space reconstructions of particle shape and  
109 strain fields. From the early developments, significant efforts have been devoted to improv-  
110 ing the speed, robustness and reliability of these algorithms. These algorithms are usually  
111 available in Python or MATLAB based softwares, including Bonsu [30], the widely used PyNX  
112 [31] with related toolkits [32, 33] and the more recent SPRING [34]. This field has gained  
113 further momentum with the advent of machine learning (ML). In parallel, the recent upgrade  
114 of numerous synchrotron facilities to fourth-generation light sources – including MAX IV  
115 (2017), ESRF-EBS (2020), Sirius (2020) and APS (2025) – has dramatically increased coherent  
116 flux, boosting the potential of crystalline nano-imaging techniques such as BCDI and Bragg  
117 ptychography [27, 35–37]. These advances also pose new challenges, particularly the need for  
118 faster and more efficient data processing pipelines capable of handling the rapidly growing  
119 volume of experimental data.

120 This PhD project, as part of the ENGAGE doctoral program<sup>1</sup>, and in collaboration with

---

<sup>1</sup>This project has been partly funded by the European Union’s Horizon 2020 Research and Innovation Pro-  
gramme under the Marie Skłodowska-Curie COFUND scheme with grant agreement No. 101034267 and the

121 ID01 beamline and the Algorithms and Data Analysis group of the ESRF, was conceived in  
122 this context, with the goal of exploring how ML algorithms can address specific computational  
123 challenges in BCDI. **Chapter 3** addresses the main aspects of the Phase Problem in BCDI and  
124 provides the necessary theoretical background on conventional phase retrieval algorithms and  
125 **Chapter 4** introduces key ML concepts tailored to BCDI data analysis conducted throughout  
126 the PhD. Namely, two central problems are addressed in this work.

127 First, due to manufacturing constraints, detectors are often made of arrays of chips, butted  
128 together in a way that leads to gaps of data within the diffraction patterns. **Chapter 5** discusses  
129 how convolutional neural networks (CNNs) can be employed to restore these missing data,  
130 thereby improving the reliability of the reconstructed images.

131 Second, X-ray detectors cannot directly measure the phase of the scattered wave-field,  
132 making its retrieval an inherently challenging computational task. **Chapters 6** and **7** present  
133 how ML-based approaches developed during this PhD can assist or complement conventional  
134 phase retrieval methods.

---

European Research Council (ERC) under the European Union’s Horizon 2020 Research and Innovation Programme (grant agreement No. 818823).



---

<sup>136</sup> BRAGG COHERENT DIFFRACTION IMAGING

<sup>137</sup> In this chapter some basic theoretical insights about the BCDI technique are provided, with the  
<sup>138</sup> aim to highlight the key concepts, assumptions and physical interpretations. More thorough  
<sup>139</sup> descriptions can be found in papers, textbooks and PhD manuscripts. I will adopt the formalism  
<sup>140</sup> of Als-Nielsen and McMorrow in [38] but similar derivations and complementing observations  
<sup>141</sup> can be found in [39, 40] as well as some more recent papers [41] and PhD theses [42–44].

<sup>142</sup> **2.1 Foreword on typical assumptions and approximations  
143 in BCDI**

<sup>144</sup> In order to keep the dissertation short and targeted to the BCDI case, I will start considering  
<sup>145</sup> some observations on this technique that will lead to some preliminary assumptions and  
<sup>146</sup> simplifications. First, the word “Bragg” suggests that crystalline specimens are involved. As  
<sup>147</sup> discussed later in the text, Bragg’s law applies to periodic structures, therefore we will focus  
<sup>148</sup> our discussion to this specific case of crystalline samples in Bragg geometry.

<sup>149</sup> The word “Coherent” implies that samples are probed with coherent beams (in our case X-rays).  
<sup>150</sup> This fundamental property of electromagnetic radiation will be briefly discussed later on.  
<sup>151</sup> For the moment, this ingredient enables us to approximate the probing radiation with plane  
<sup>152</sup> electromagnetic waves.

<sup>153</sup> The word “Diffraction” refers to the type of mechanism describing the interaction between  
<sup>154</sup> the X-rays and the samples. Paraphrasing [39] at page 4, this mechanism can be divided into  
<sup>155</sup> two main phenomena, namely (i) the scattering of the radiation by each individual atom in the  
<sup>156</sup> sample and (ii) the interference between the waves scattered by these atoms. The interference  
<sup>157</sup> mechanism, in turn, is enabled because these scattered waves are coherent with the incident  
<sup>158</sup> radiation and therefore between themselves. In other words, the information of each scatterer  
<sup>159</sup> is shared with the other scatterers as the diffracted waves “talk to each other”. The complete  
<sup>160</sup> mathematical description of these two phenomena without approximations is prohibitive,  
<sup>161</sup> hence some simplifications are usually adopted:

- <sup>162</sup> **• No refraction, no absorption:** Scattering is the only mechanism considered. Because  
<sup>163</sup> of their short wavelengths (0.5 - 2.5 Å), X-rays refraction is very small, hence for small  
<sup>164</sup> samples the deviation of the outgoing rays with respect to the incident ones is negligible.

165 Moreover, we assume the absence of any absorption effect. While inherently present, it  
166 can be neglected for small enough samples.

167 • **Elastic scattering:** The interaction between the incoming X-rays and the atom is  
168 considered only elastic, meaning that no energy is transferred to the atom, which instead  
169 bounces off the photons with unaltered energy and momentum. This is again necessary  
170 for the scattered waves to interfere, as any difference in wavelength would prevent any  
171 coherent interaction. This description, also called Thomson scattering, considers the  
172 interaction with a free charge, and it also shows that cross-section of the scattering of  
173 electrons is much higher than that of protons ( $\sim 3 \times 10^6$  larger). For this reason, only  
174 the scattering from electrons is considered.

175 • **Weak diffraction (Born approximation):** This assumption implies that each scattered  
176 wave does not interact further with the sample, therefore neglecting any possible multiple  
177 scattering event. The consequence of this assumption is that the overall diffracted wave  
178 can be approximated by the linear superposition of the contributions of each scattering  
179 site. This approximation, in crystallography, is called *kinematical approximation*. Dealing  
180 with crystalline samples, this assumption breaks for relatively thick samples ( $> 1\mu m$ )  
181 in which the light travels through the sample for longer distances before exiting, there-  
182 fore bouncing off several atoms. Diffraction of larger samples requires more complex  
183 theory of the so-called *dynamical regime* [45–47]. However, in our case, the size of the  
184 typical samples studied with BCDI rarely exceeds  $1\mu m$  size, making the kinematical  
185 approximation suitable.

186 • **Far-field approximation:** Here, the distance between the scattering atoms and the  
187 detector is assumed to be much larger than the distance among the scatterers themselves.  
188 One can intuitively see that this approximation turns the spherical waves created by the  
189 scatterers, interfering with each other, into plane waves when these are evaluated far  
190 from the sources (in this case the sample's atoms). This assumption is always respected in  
191 the BCDI technique as the sample-detector distance is in the order of tens of centimeters  
192 (practically from 30 cm to 2 m).

193 The last word “*Imaging*” tells us that the format of the data is by nature, multidimensional  
194 (2D - 3D). It will be shown later in the chapter that 3D diffracted signal is recorded stacking 2D  
195 images captured by the detector, and therefore the results after the data analysis are 3D images  
196 of the sample.

197 Given this set of assumptions and approximations we can proceed with our simplified  
198 derivation of the equation governing the coherent X-ray scattering from a crystal and its  
199 interpretation.

## 200 2.2 Coherent X-ray scattering from crystalline structures

201 Let us consider an X-ray beam, represented by a perfectly monochromatic plane wave with  
202 linear polarization in the horizontal plane, scattering with a single free electron (see Fig. 2.1).  
203 In this simple case we can imagine the X-ray electromagnetic field exerting a force (Lorentz  
204 force) onto the electron placed at the origin of the reference frame. In turn, this force will  
205 accelerate the electron accordingly, therefore inducing an oscillation motion generating an

206 electromagnetic wave. If the scattering is assumed to be elastic, the radiation produced by  
 207 the oscillating electron (*electric dipole approximation*) will have the same wave-vector as the  
 208 incoming X-ray. Moreover, the solution of Maxwell equations for this specific case shows that  
 209 this dipole radiation propagates in the form of a spherical wave. At this point we ask ourselves  
 210 what is the amplitude of this scattered wave when evaluated in a generic point  $\mathbf{r}$  on the vertical  
 211 plane, far from the origin (*far-field approximation*). The result was achieved by Thomson in  
 212 1906 and is here reported without the full detailed derivation which can be found in the cited  
 213 textbooks [38, 39, 48].

$$\mathbf{E}_{\text{dip}}(\mathbf{r}, t) = -r_e \frac{e^{ikr}}{r} e^{-i\omega t} E_0 \hat{\mathbf{z}} \quad (2.1)$$

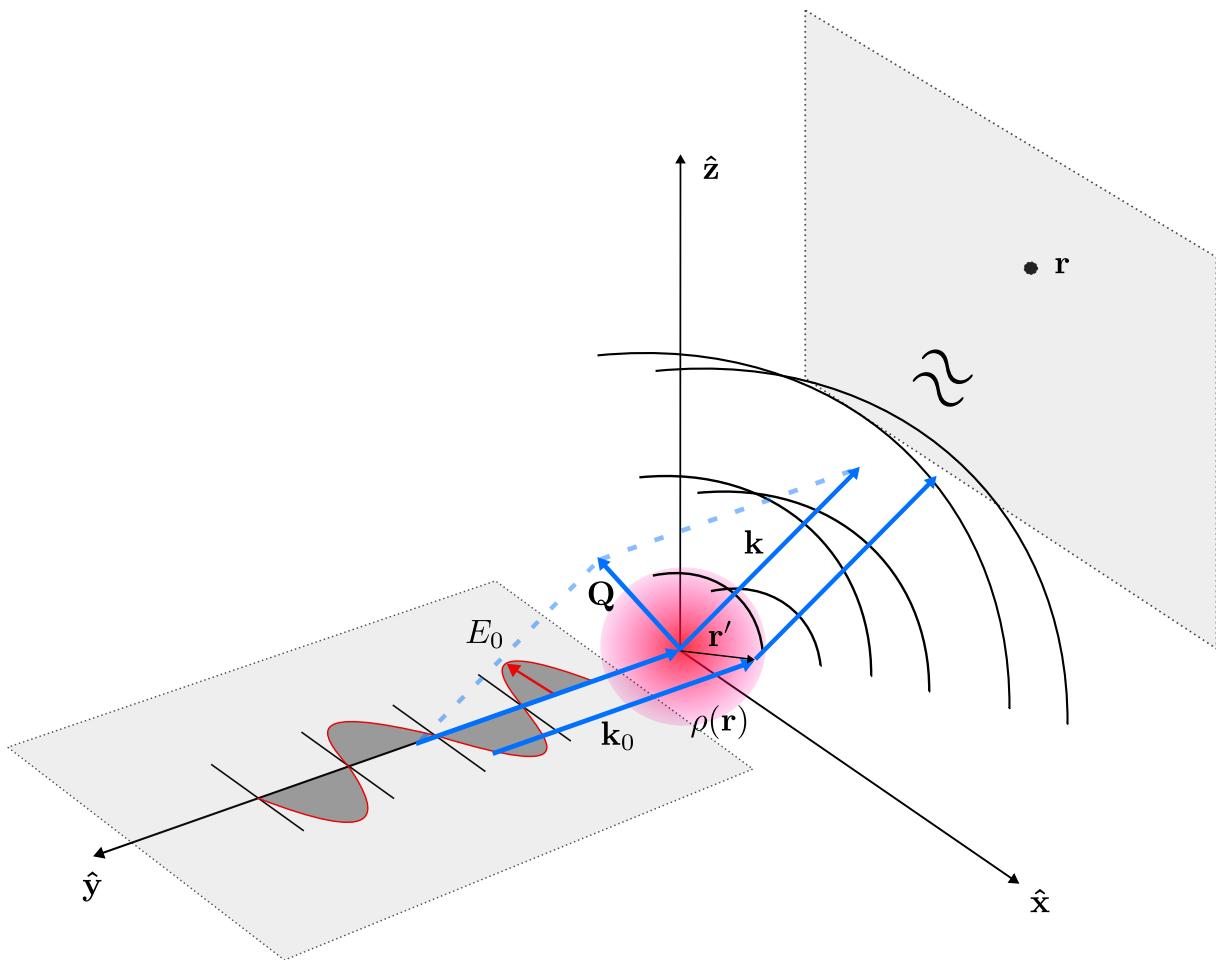
214 where  $r_e$  is the classical radius of the electron, or Thomson scattering length,  $k$  is the  
 215 outgoing wave-vector,  $\omega$  is the pulsation of the X-ray beam (incoming and outgoing),  $E_0$  is the  
 216 electric field of the incoming radiation.

217 In this case we cannot talk about diffraction as there is no interference of the outgoing  
 218 wave with other scattered waves. In order to have a diffraction pattern we need to have at  
 219 least a second charge scattering, from which a phase delay with respect to the first one can be  
 220 calculated. For instance, if we consider  $N$  electrons being illuminated by the same radiation,  
 221 each of them placed in  $\mathbf{r}'_n$  we could evaluate the contribution to the overall scattering wave-field  
 222 for each electron. The simplest way is to make use of the *kinematical approximation* and sum  
 223 linearly all the contributions. However, we must take into account the phase delays between  
 224 the scattering from different positions in space. This phase delay can be calculated, and it turns  
 225 out to be  $\Delta\phi(r) = (\mathbf{k} - \mathbf{k}_0) \cdot \mathbf{r}' = \mathbf{Q} \cdot \mathbf{r}'$  where we have expressed the difference between the  
 226 incoming and outgoing wave vectors,  $\mathbf{k}_0$  and  $\mathbf{k}$  respectively, with  $\mathbf{Q}$  often called *scattering  
 227 vector* (see Fig.2.1). The equation can thus be rewritten like:

$$\mathbf{E}_{\text{atom}}(\mathbf{r}, t) = r_e \frac{e^{ikr}}{r} e^{-i\omega t} E_0 \hat{\mathbf{z}} \sum_{n=1}^N e^{i\mathbf{Q} \cdot \mathbf{r}'_n} \quad (2.2)$$

228 In the continuum limit, replacing the  $N$  point-like charges with an overall electron density  
 229 distribution  $\rho(\mathbf{r})$  the above equation takes the form:

$$\mathbf{E}_{\text{atom}}(\mathbf{r}, t) = r_e \frac{e^{ikr}}{r} e^{-i\omega t} E_0 \hat{\mathbf{z}} \int_{\mathbb{R}^3} \rho_a(\mathbf{r}') e^{i\mathbf{Q} \cdot \mathbf{r}'} d^3 \mathbf{r}' \quad (2.3)$$



**Figure 2.1:** Sketch of the scattering process evaluated in the far-field on the vertical plane for an electron density irradiated by a monochromatic X-ray beam with linear polarization along the  $\hat{x}$  direction.

230 It is clear now that the information regarding the physical system of interest is embedded  
 231 in the integral term. In fact, this is often called “*form factor*” -  $F(\mathbf{Q})$  - and it plays an important  
 232 role in the interpretation of the scattering equations.

$$F(\mathbf{Q}) = \int_{\mathbb{R}^3} \rho(\mathbf{r}') e^{i\mathbf{Q} \cdot \mathbf{r}'} d^3 \mathbf{r}' \quad (2.4)$$

233  $F(\mathbf{Q})$  represents the Fourier transform of the electron density, and it is the main result of  
 234 this paragraph as it links the charge distribution of the sample in real space with the quantity  
 235 measured, in reciprocal space.

236 To continue, we should bear in mind that X-ray photon counting detectors are sensitive  
 237 to the time-averaged intensity of the signal as their time response is much slower than the  
 238 oscillating frequency of X-rays ( $\sim 10^9$  Hz for typical read-out limited frame rates of the Maxipix  
 239 [49] against the  $\sim 10^{18}$  Hz for X-rays at 10 keV). This limitation is also at the core of the “Phase  
 240 Problem” that we will see later on, for which the phase information of the complex-valued  
 241 wave-field is lost in the measurement. In order to do so, the time-averaged Poynting vector is  
 242 calculated.

$$\langle \mathbf{S}(\mathbf{r}) \rangle = r_e^2 \frac{1}{r^2} J_0 \left| \int \rho(\mathbf{r}') e^{i\mathbf{Q} \cdot \mathbf{r}'} d^3 r' \right|^2 \hat{\mathbf{r}} \quad (2.5)$$

243 where  $J_0 = |\mathbf{E}_0|^2 / 2\mu_0 c$  is the incident intensity. To conclude we consider the power  
 244 delivered on the detector. For a pixel with area  $d\mathbf{a} = r^2 d\Omega \hat{\mathbf{r}}$  the radiation power is equal to:

$$P(\mathbf{Q}) = r_e^2 J_0 \left| \int \rho(\mathbf{r}') e^{i\mathbf{Q} \cdot \mathbf{r}'} d^3 r' \right|^2 d\Omega \quad (2.6)$$

245 Eq.2.6 shows that the signal captured by the detectors is now in  $\mathbf{Q}$  space, and it is proportional  
 246 to the square modulus of the Fourier transform of the electron density of the sample.  
 247 The square modulus operation also shows how the phase of the Fourier transformed scattering  
 248 complex-amplitude is lost. For this reason, in order to reconstruct the electron density, it is not  
 249 possible to simply use an inverse Fourier transform, but iterative methods are required.

### 250 2.2.1 One atom

251 If now we were to consider an atom, far from resonance, we could assume the electron density  
 252 being the main responsible for the scattering. It is known indeed that protons, because of the  
 253 larger mass, have a much smaller cross-section for the scattering with photons. Using Eq.2.4  
 254 we would therefore have the *atomic form factor* -  $f_l(\mathbf{Q})$  being defined as:

$$f_l(\mathbf{Q}) = \int_{\mathbb{R}^3} \rho_l(\mathbf{r}) e^{i\mathbf{Q} \cdot \mathbf{r}} d^3 \mathbf{r} \quad (2.7)$$

255 We now need to study the specific case in which the collection of atoms is ordered into a  
 256 periodic structure.

### 257 2.2.2 Ensemble of ordered atoms: a Crystal

258 Perfect crystals are constructed by a basic structural arrangement of atoms (*motif*) repeated  
 259 periodically on a *lattice* of one or more dimensions. The regularity of the lattice is such that,  
 260 for the 3D case, any of its nodes can be located in space by the formula:

$$\mathbf{R}_n = n_1 \mathbf{a}_1 + n_2 \mathbf{a}_2 + n_3 \mathbf{a}_3 \quad (2.8)$$

261 where  $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$  constitutes the basis vectors of the primitive unit cell and  $n_1, n_2, n_3$  are  
 262 integer numbers. It follows that the information can be condensed in the unit cell, i.e. the  
 263 orientation in space of the basis vectors as any region of the lattice can be seen as the same  
 264 unit cell, translated from the origin by the amount given by  $\mathbf{R}_n$ .

265

266 The overall crystal is then constructed positioning on each of the nodes of the lattice the  
 267 same motif, or basis.

268 In another more elegant way, we could say that, being the lattice  $\mathcal{L}(\mathbf{r})$ , the motif  $\mathcal{B}(\mathbf{r})$ , the  
269 crystal  $\mathcal{C}(\mathbf{r})$  is given by the convolution of  $\mathcal{L}(\mathbf{r})$  with  $\mathcal{B}(\mathbf{r})$ :

$$\mathcal{C}(\mathbf{r}) = \mathcal{L}(\mathbf{r}) * \mathcal{B}(\mathbf{r}) \quad (2.9)$$

270 For simplicity, we will consider from now on a single atom motif. At this point, when  
271 evaluating the scattering amplitude of the crystal we have to deal to an assembly of atoms, and  
272 we may want to exploit the regular structure we have just described. First, we can assume that,  
273 similarly to the case of many scattering electrons, in the kinematical approximation the overall  
274 scattering factor is given by the sum of the contributions of each atom, weighted by a phase  
275 factor that accounts for their positions in space.

$$F_{\text{crystal}}(\mathbf{Q}) = \sum_{l=1}^{\text{All atoms}} f_l(\mathbf{Q}) e^{i\mathbf{Q} \cdot \mathbf{r}_l} \quad (2.10)$$

276 Secondly, observing that the position of each atom is given by the sum of the position of  
277 the atom inside the unit cell and the lattice vector  $\mathbf{r}_l = \mathbf{R}_n + \mathbf{r}_j$ , we can separate Eq.2.10 in  
278 two terms:

$$F_{\text{crystal}}(\mathbf{Q}) = \sum_{\mathbf{R}_n + \mathbf{r}_j}^{\text{All atoms}} f_l(\mathbf{Q}) e^{i\mathbf{Q} \cdot (\mathbf{R}_n + \mathbf{r}_j)} = \underbrace{\sum_n e^{i\mathbf{Q} \cdot \mathbf{R}_n}}_{\text{Lattice}} \underbrace{\sum_j f_j(\mathbf{Q}) e^{i\mathbf{Q} \cdot \mathbf{r}_j}}_{\text{Unit cell}} \quad (2.11)$$

279 The first summation extends over all lattice points, while the second covers all atoms within  
280 the unit cell. Here we can already wrap the second sum into a more practical term expressing  
281 the *unit cell form factor*

$$F_{\text{crystal}}(\mathbf{Q}) = F_{\text{u.c.}}(\mathbf{Q}) \sum_n e^{i\mathbf{Q} \cdot \mathbf{R}_n} \quad (2.12)$$

282 The term  $F_{\text{u.c.}}(\mathbf{Q})$  can be easily calculated as typical unit cells contain a small number of  
283 elements. On the contrary we need to exploit the properties of the lattice periodicity to evaluate  
284 the large summation over all lattice points.

### 285 2.2.3 Laue condition and Bragg's Law

286 The term we want to calculate is the sum of complex exponential, meaning that if the phases  
287  $\mathbf{Q} \cdot \mathbf{R}_n$  are misaligned the sum will mutually cancel out the contributions and the resultant  
288 will be very small, in the order of unity. On the contrary, when phase offsets are equal to an  
289 integer multiple of  $2\pi$  the *phasors*, complex vectors  $e^{i\mathbf{Q} \cdot \mathbf{R}_n}$  for each  $n$ , will add *in phase*. The  
290 problem is thus to find those  $\mathbf{Q}$  values for which

$$\mathbf{Q} \cdot \mathbf{R}_n = 2\pi \times \text{integer} \quad \forall n \quad (2.13)$$

291 In order to do than we need to construct a reciprocal space lattice with a set of basis  
 292  $\mathbf{a}_1^*, \mathbf{a}_2^*, \mathbf{a}_3^*$  which fulfill:

$$\mathbf{a}_1 \cdot \mathbf{a}_1^* = 2\pi h \quad \mathbf{a}_2 \cdot \mathbf{a}_2^* = 2\pi k \quad \mathbf{a}_3 \cdot \mathbf{a}_3^* = 2\pi l \quad (2.14)$$

293 where  $h, k, l$  known as Miller indices, are integer. Having a set of basis vectors and the  
 294 Miller indices, the resulting reciprocal space lattice lies in those points found by the vector  
 295  $\mathbf{G}_{hkl}$

$$\mathbf{G}_{hkl} = h\mathbf{a}_1^* + k\mathbf{a}_2^* + l\mathbf{a}_3^* \quad (2.15)$$

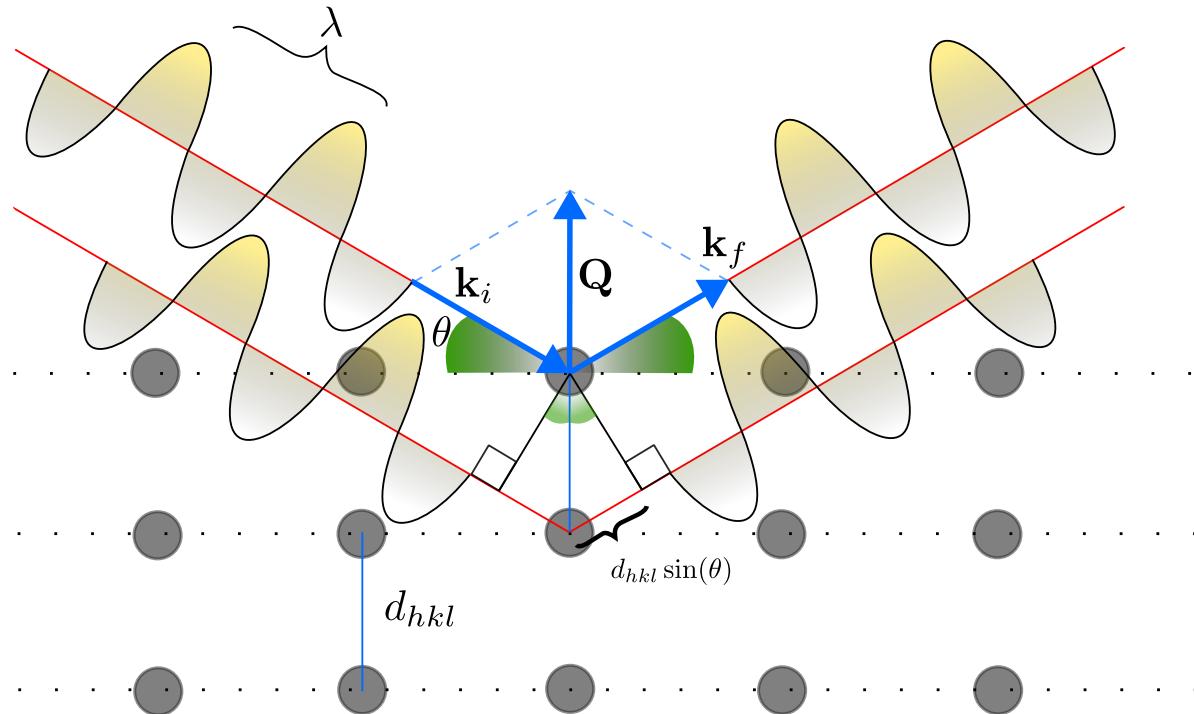
296 This result is telling us that the scattering amplitude of a diffracting crystal is detectable  
 297 only in those points in space for which the wave-vector  $\mathbf{Q}$  coincides with a point ( $hkl$  node) of  
 298 the reciprocal lattice, hence  $\mathbf{Q} = \mathbf{G}_{hkl}$ . These isolated points are called *Bragg peaks*.

299 This is known as the Laue condition for diffraction as it was discovered by Max von Laue  
 300 in 1912 [50].

301 A different but equivalent interpretation of the diffraction of a crystal was given by William  
 302 Lawrence Bragg in 1913 [51]. Here, the crystal lattice is seen as a stack of parallel planes  
 303 (see Fig.2.2) and the condition for constructive interference of the waves scattered by planes  
 304 of the same family is found as follows. Let us consider an X-ray beam of wavelength  $\lambda$  and  
 305 propagation vector  $\mathbf{k}_i$  impinging with an angle  $\theta$  on a crystal. We call  $d_{hkl}$  the distance between  
 306 the planes of the crystal, along a specific  $hkl$  direction. The scattered beam is leaving the crystal  
 307 with the same angle  $\theta$  and with a propagation vector  $\mathbf{k}_f$  equal in magnitude to the incident one  
 308 (*elastic scattering*). At this point one can find the relationship between  $\theta, d_{hkl}, \lambda$  that allows for  
 309 a constructive interference of the waves diffracted from the series of planes by evaluating the  
 310 optical path length difference induced by the spacing. Reminding that  $|k_i| = |k_f| = 2\pi/\lambda$  and  
 311 with the help of Fig. 2.2 we can observe that this difference is  $\Delta l = 2d_{hkl} \sin(\theta)$  and therefore  
 312 the phase offset between two waves is  $\Delta\phi = |k| \Delta l = 4d_{hkl}\pi \sin(\theta)/\lambda$ . We have seen above  
 313 that the condition for constructive interference requires the phase differences to be equal to a  
 314 multiple of  $2\pi$ , therefore:

$$2d_{hkl} \sin(\theta) = n\lambda \quad \text{where } n \in \mathbb{Z} \quad (2.16)$$

316 Moreover, it can be shown that the family of planes is defined by the vector  $G_{hkl}$  which  
 317 points at the reciprocal space node that is collecting the scattering from those planes, and  
 318 defines as well the spacing  $d_{hkl}$  between each of these planes.



**Figure 2.2:** Illustration of Bragg's law. The family of crystal planes given separated by  $d_{hkl}$  produces constructive interference when the incident radiation illuminates it with an angle  $\theta$  given by Eq.2.16.

319     Equation 2.16 is known as Bragg's law, and it can be derived from the Laue condition.

320     We can now rewrite Eq.2.12 for the case of an infinitely extended perfect 3D crystal as:

$$F_{\text{crystal}^\infty}(\mathbf{Q}) = F_{\text{u.c.}}(\mathbf{Q}) \sum_h \sum_k \sum_l \delta(\mathbf{Q} - \mathbf{G}_{hkl}) \quad (2.17)$$

321     where the term relative to the lattice is expressed as the sum of Dirac deltas in  $Q$  space  
322     centered in each  $hkl$  node.

### 323     2.2.4 Finite size crystals

324     We are now ready to treat the case of finite size crystals. In the above description the lattice  
325     was assumed to extend infinitely along all the 3 dimensions. This, we have seen, results in a  
326     scattering signal that lives on a perfect reciprocal lattice made of point-like nodes. Because of  
327     the Fourier transformation that links the real space scattering object and the reciprocal space  
328     diffracted signal, one can intuitively deduce that if for an infinite crystal we obtain point-like  
329     Bragg peaks, we could expect a broadening of the peaks as the crystal size is reduced.

330     Mathematically we can derive the result for crystal of shape  $S(\mathbf{r})$  considering the function  
331      $S$  a window cropping a finite portion of the 3D infinite crystal. This function can also be  
332     referred to as an Ewald function [52] or *support*. In real space this corresponds to the product:

$$\rho_{\text{fin}}(\mathbf{r}) = \rho_\infty(\mathbf{r})S(\mathbf{r}) \quad (2.18)$$

333 where

$$S(\mathbf{r}) = \begin{cases} 1, & \text{inside the crystal region,} \\ 0, & \text{otherwise.} \end{cases} \quad (2.19)$$

334 From the convolution theorem we have now that the Fourier transform  $\mathcal{F}\{\rho_{\text{fin}}(\mathbf{r})\}$  is equal  
 335 to the convolution  $\mathcal{F}\{\rho_{\infty}(\mathbf{r})\} * \mathcal{F}\{S(\mathbf{r})\}$ . Using Eq.2.12 for the Fourier transform of the infinite  
 336 crystal and calling  $\widehat{S}(\mathbf{Q}) = \mathcal{F}\{S(\mathbf{r})\}$  we can therefore write:

$$F_{\text{fin. crystal}}(\mathbf{Q}) = \left[ F_{\text{u.c.}}(\mathbf{Q}) \sum_h \sum_k \sum_l \delta(\mathbf{Q} - \mathbf{G}_{hkl}) \right] * \widehat{S}(\mathbf{Q}) \quad (2.20)$$

337 From the commutative property of convolutions, we can now rearrange the terms as follows:

$$\begin{aligned} F_{\text{fin. crystal}}(\mathbf{Q}) &= \widehat{S}(\mathbf{Q}) * \left[ F_{\text{u.c.}}(\mathbf{Q}) \sum_h \sum_k \sum_l \delta(\mathbf{Q} - \mathbf{G}_{hkl}) \right] \\ &= \int \widehat{S}(\mathbf{Q} - \mathbf{Q}') F_{\text{u.c.}}(\mathbf{Q}') \sum_h \sum_k \sum_l \delta(\mathbf{Q}' - \mathbf{G}_{hkl}) d^3 \mathbf{Q}' \end{aligned} \quad (2.21)$$

338 At this point, we can bring the integral inside the summation and calculate:

$$F_{\text{fin. crystal}}(\mathbf{Q}) = \sum_h \sum_k \sum_l \int \widehat{S}(\mathbf{Q} - \mathbf{Q}') F_{\text{u.c.}}(\mathbf{Q}') \delta(\mathbf{Q}' - \mathbf{G}_{hkl}) d^3 \mathbf{Q}' \quad (2.22)$$

339 We observe now that the Dirac delta function is non-zero only in the  $hkl$  nodes defined by  
 340  $\mathbf{G}_{hkl}$ . This leads to the final equation:

$$F_{\text{finite}}(\mathbf{Q}) = \sum_h \sum_k \sum_l \widehat{S}(\mathbf{Q} - \mathbf{G}_{hkl}) F_{\text{u.c.}}(\mathbf{G}_{hkl}) \quad (2.23)$$

341 Here we see that the Bragg peaks, from point-like spots in reciprocal space are subjected  
 342 to a broadening that shapes them according to the structure of the Fourier transform of the  
 343 crystal shape, centered at  $\mathbf{G}_{hkl}$ .

344

345 Let us now consider the specific case of a cubic crystal aligned with the direct-lattice basis  
 346  $\mathbf{a}_1, \mathbf{a}_2, \mathbf{a}_3$ . The crystal contains  $N$  cells along each basis vector  $\mathbf{a}$ . Each edge vector is therefore  
 347  $\mathbf{L}_j = N\mathbf{a}_j$ . Being the center of the cubic crystal placed in the origin of the reference frame we  
 348 can now write the support function as:

$$S(\mathbf{r}) = \prod_{j=1}^3 \mathcal{R}\left(\frac{\mathbf{r} \cdot \hat{\mathbf{a}}_j}{L_j}\right), \quad \mathcal{R}(x) = \begin{cases} 1, & |x| \leq 1/2, \\ 0, & \text{else.} \end{cases} \quad (2.24)$$

349 where  $\hat{\mathbf{a}}_j = \mathbf{a}_j/|\mathbf{a}_j|$  and  $L_j = |\mathbf{L}_j|$ , and the 1D rectangular function  $\mathcal{R}(x)$  has been  
 350 defined. Here we have defined the cube as the product of the three 1D rectangular function

351 on each of the 3 dimensions. We can therefore calculate the Fourier transform of  $S(\mathbf{r})$  as the  
 352 product of the Fourier transform of each  $\mathcal{R}\left(\frac{\mathbf{r} \cdot \hat{\mathbf{a}}_j}{L_j}\right)$ .

353 It follows that:

$$\widehat{S}(\mathbf{Q}) = \prod_{j=1}^3 L_j \operatorname{sinc}\left(\frac{\mathbf{Q} \cdot \hat{\mathbf{a}}_j L_j}{2}\right), \quad (2.25)$$

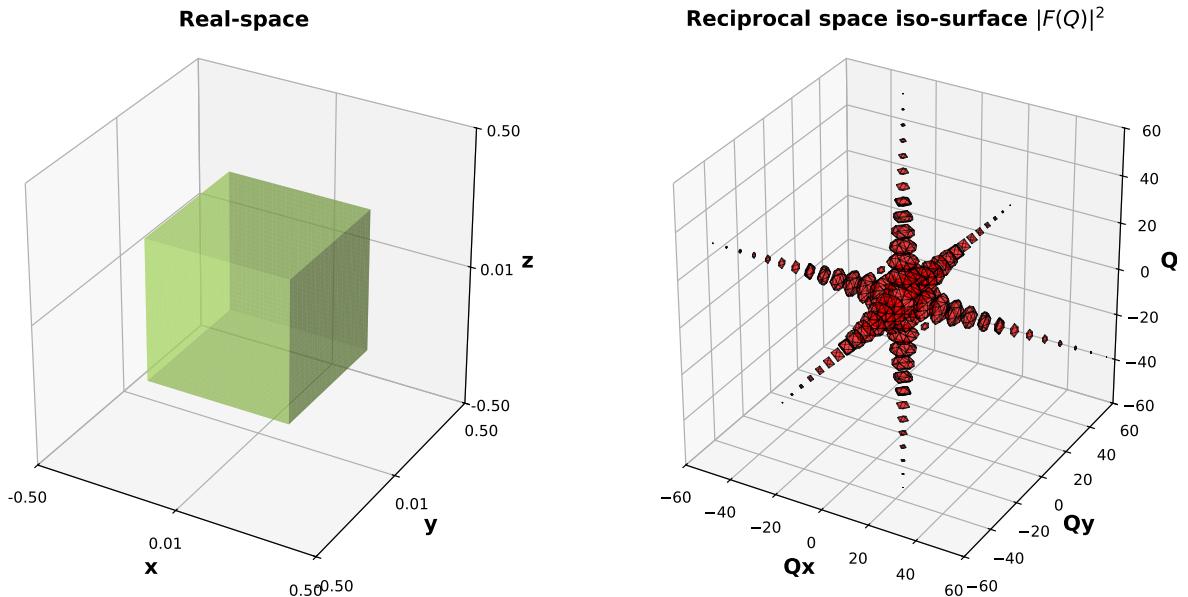
354 where we made use of the cardinal sine function defined as  $\operatorname{sinc}(x) = \frac{\sin x}{x}$ . Putting this  
 355 result in Eq. 2.23 we obtain:

$$F_{\text{cube}}(\mathbf{Q}) = \sum_h \sum_k \sum_l F_{\text{u.c.}}(\mathbf{G}_{hkl}) \prod_{j=1}^3 L_j \operatorname{sinc}\left(\frac{(\mathbf{Q} - \mathbf{G}_{hkl}) \cdot \hat{\mathbf{a}}_j L_j}{2}\right), \quad (2.26)$$

356 At this point of the derivation we can introduce another simplification, directly linked to  
 357 the BCDI technique. Up to now we have been considering the scattering amplitude in the full  
 358 reciprocal space. This means that Eq. 2.26 is encompassing all the  $hkl$  nodes in which we could  
 359 find the diffracted signal. However, in typical BCDI experiments only a small region around a  
 360 single Bragg peak is measured. This means that we can from now on focus on one  $\mathbf{G}_{hkl}$  vector  
 361 and drop the summation. Moreover, defining  $\mathbf{q} = \mathbf{Q} - \mathbf{G}_{hkl}$  and recalling the result obtained  
 362 in Eq. 2.6 one can find that the intensity of the diffraction pattern of the  $hkl$  Bragg peak is :

$$I_{\text{cube}}(\mathbf{q}) \propto F_{\text{u.c.}}(\mathbf{G}_{hkl}) \left| \prod_{j=1}^3 L_j \operatorname{sinc}\left(\frac{\mathbf{q} \cdot \hat{\mathbf{a}}_j L_j}{2}\right) \right|^2, \quad (2.27)$$

363 The diffracted signal for a cubic crystal in the vicinity of the  $hkl$  Bragg peak has therefore  
 364 the shape of a *squared* 3D cardinal sine function as represented in Fig. 2.3

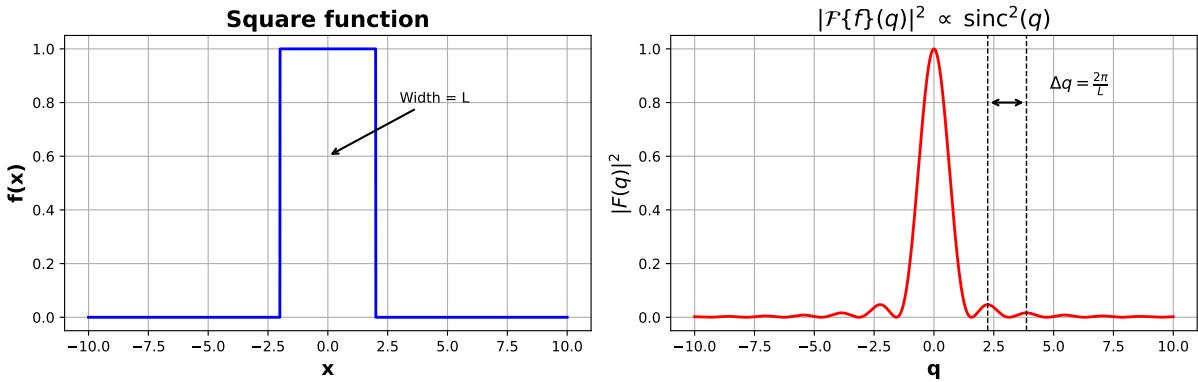


**Figure 2.3:** Illustration of the squared modulus of the Fourier transform of cube as representation of the diffracted intensity around a Bragg peak for a cubic crystal.

365 An interesting observation concerns the spacing between the fringes of the interference  
 366 pattern. Studying Eq.2.27 we can infer that:

- 367 • The maximum of the peak is concentrated at  $\mathbf{q} = 0 \Rightarrow \mathbf{Q} = \mathbf{G}_{hkl}$  as expected.
- 368 • The intensity decays as  $\sim 1/q^2$  along the intensity streaks (see Fig. 2.4), also known as  
 369 crystal truncation rods (CTR)<sup>1</sup>. This decay depends on the shape. For a spherical crystal  
 370 it follows a  $\sim 1/q^4$  law.
- 371 • The intensity drops to zero every time the argument  $(\mathbf{q} \cdot \hat{\mathbf{a}}_j L)/2 = n\pi$  with  $n$  integer.  
 372 This results in the presence of fringes that have a thickness as a function of the  $\mathbf{q}$  direction  
 373 given by:  $\Delta q_j = 2\pi/L$  (see Fig.2.4).
- 374 • Given the above relationship between the thickness of the fringes and the size of the  
 375 crystal, one can intuitively imagine extending  $L$  to infinity, thus narrowing the fringes  
 376 to zero width and the central lobe to a Dirac delta in  $\mathbf{q} = 0$  as in the infinite lattice case.

<sup>1</sup>More in general, CTRs appear as lines of more intense diffraction, perpendicular to facets of the diffracting nano-crystal [53].



**Figure 2.4:** 1D example of the squared modulus of the Fourier transform of a square function. The sharp intensity decay outside the central peak is evident. The spacing between fringes of constructive interferences is also related to the width of the square.

377 For the understanding and visualization of Fourier transforms in crystallography I recom-  
378 mend the pedagogical work of Aubert and Lecomte [54].

### 379 2.2.5 Real crystals

380 We are now ready to deal with non-perfect crystals. The presence of surfaces, substrates,  
381 physio-chemical reaction with the environment induce deformations of the perfect crystalline  
382 structure. In the most general case the deformation is modeled by a displacement field  $\mathbf{u}(\mathbf{r})$  that  
383 shifts the position of each atom of the crystal off the position of the perfect lattice. However, for  
384 simplicity the displacement field is often assumed to act on the whole unit cell rather than each  
385 single atom. One can therefore write the scattering amplitude as the sum over the unit cells.  
386 Defining  $\mathbf{u}(\mathbf{R}_n)$  the displacement of the  $n$ -th cell with respect to the perfect lattice position we  
387 can calculate the structure factor of the infinite crystal summing over all the unit cells the form  
388 factor of each cell weighted by their displaced positions:

$$F_\infty^{\text{strain}}(\mathbf{Q}) = \sum_n F_n(\mathbf{Q}) e^{i\mathbf{Q} \cdot (\mathbf{R}_n + \mathbf{u}(\mathbf{R}_n))} \quad (2.28)$$

389 To lighten up the notation we can introduce another vector  $\mathbf{R}'_n = \mathbf{R}_n + \mathbf{u}(\mathbf{R}_n)$  that points  
390 at each displaced unit cell. As this lattice is no longer perfectly periodic we cannot any more  
391 simplify the sum into a sum of delta functions calculated in the  $hkl$  nodes of the reciprocal  
392 lattice. At this point we can however repeat the calculations for the case of a finite size crystal  
393 of shape  $S(\mathbf{r})$ . For the convolution theorem we can write:

$$F_{\text{fin}}^{\text{strain}}(\mathbf{Q}) = \sum_n F_n(\mathbf{Q}) e^{i\mathbf{Q} \cdot \mathbf{R}'_n} * \hat{S}(\mathbf{Q}) \quad (2.29)$$

394 Developing the convolution integral we obtain:

$$\begin{aligned}
F_{\text{fin}}^{\text{strain}}(\mathbf{Q}) &= \int \sum_n F_n(\mathbf{Q} - \mathbf{Q}') e^{i(\mathbf{Q}-\mathbf{Q}') \cdot \mathbf{R}'_n} \widehat{S}(\mathbf{Q}') d^3 \mathbf{Q}' \\
&= \sum_n \int F_n(\mathbf{Q} - \mathbf{Q}') e^{i(\mathbf{Q}-\mathbf{Q}') \cdot \mathbf{R}'_n} \widehat{S}(\mathbf{Q}') d^3 \mathbf{Q}' 
\end{aligned} \tag{2.30}$$

where, being the sum over a finite number of unit cells converging, we moved the integral sign inside the sum. Now, we can assume to restrict the field of view in the vicinity of the  $\mathbf{G}_{hkl}$  where the unit cell form factor varies slowly compared to the  $\widehat{S}(\mathbf{Q}')$ . Intuitively we could understand this step considering that typical BCDI crystals contain a large number of unit cells, therefore in Fourier space, the features relative to the crystal shape vary much faster than the features relative to the unit cell. This approximation allows us to write:

$$\begin{aligned}
F_{\text{fin}}^{\text{strain}}(\mathbf{Q}) &= \sum_n F_n(\mathbf{G}_{hkl}) \int e^{i(\mathbf{Q}-\mathbf{Q}') \cdot \mathbf{R}'_n} \widehat{S}(\mathbf{Q}') d^3 \mathbf{Q}' \\
&= \sum_n F_n(\mathbf{G}_{hkl}) \int e^{i\mathbf{Q} \cdot \mathbf{R}'_n} e^{-i\mathbf{Q}' \cdot \mathbf{R}'_n} \widehat{S}(\mathbf{Q}') d^3 \mathbf{Q}' \\
&= \sum_n F_n(\mathbf{G}_{hkl}) e^{i\mathbf{Q} \cdot \mathbf{R}'_n} \int e^{-i\mathbf{Q}' \cdot \mathbf{R}'_n} \widehat{S}(\mathbf{Q}') d^3 \mathbf{Q}' \\
&= \sum_n F_n(\mathbf{G}_{hkl}) e^{i\mathbf{Q} \cdot \mathbf{R}'_n} S(\mathbf{R}'_n)
\end{aligned} \tag{2.31}$$

where we have recognized the inverse Fourier transform of the shape function calculated in  $\mathbf{R}'_n$ . If we now assume that the displacement affecting each unit cell does not influence the form factor, we can replace  $F_n(\mathbf{G}_{hkl})$  with  $F_{\text{u.c.}}(\mathbf{G}_{hkl})$  and take it out of the sum. Additionally, considering the relations  $\mathbf{Q} = \mathbf{G}_{hkl} + \mathbf{q}$  and  $e^{i\mathbf{G}_{hkl} \cdot \mathbf{R}_n} = 1$  we can rearrange into:

$$\begin{aligned}
F_{\text{fin}}^{\text{strain}}(\mathbf{Q}) &= \sum_n F_n(\mathbf{G}_{hkl}) e^{i\mathbf{Q} \cdot \mathbf{R}'_n} S(\mathbf{R}'_n) \\
&= F_{\text{u.c.}}(\mathbf{G}_{hkl}) \sum_n e^{i(\mathbf{G}_{hkl} + \mathbf{q}) \cdot (\mathbf{R}_n + \mathbf{u}(\mathbf{R}_n))} S(\mathbf{R}'_n) \\
&= F_{\text{u.c.}}(\mathbf{G}_{hkl}) \sum_n e^{i\mathbf{G}_{hkl} \cdot \mathbf{R}_n} e^{i\mathbf{G}_{hkl} \cdot \mathbf{u}(\mathbf{R}_n)} e^{i\mathbf{q} \cdot (\mathbf{R}_n + \mathbf{u}(\mathbf{R}_n))} S(\mathbf{R}'_n) \\
&= F_{\text{u.c.}}(\mathbf{G}_{hkl}) \sum_n S(\mathbf{R}'_n) e^{i\mathbf{G}_{hkl} \cdot \mathbf{u}(\mathbf{R}_n)} e^{i\mathbf{q} \cdot \mathbf{R}'_n}
\end{aligned} \tag{2.32}$$

The formula we obtained tells us that the scattering amplitude for the finite strained crystal is proportional to the sum over all the unit cells of the shape function evaluated in each displaced unit cell multiplied by two phase factors. The first phase factor is given by the projection of the displacement of each unit cell (relative to the perfect lattice) onto the scattering vector of the given  $hkl$  node. This term is relevant in BCDI experiments as it is directly linked to the strain and more general information on the internal lattice displacements of the particle. The

411 second complex exponential evaluates the phase delay associated to each displaced unit cell  
412 with respect to the center of Bragg peak.

413 Despite the close similarity to a classical discrete Fourier transform (DFT), one should notice  
414 that the variable  $\mathbf{R}'_n$ ) is not uniform as the displacements disrupt the regular periodicity of the  
415 grid, required for a standard Fourier transform. It is therefore common to approximate this  
416 non-uniform discrete Fourier transform with a standard discrete one by assuming:

- 417 •  $\mathbf{q} \cdot \mathbf{u}(\mathbf{R}_n) = 0$  which means that the projection of the displacements on the vector  $\mathbf{q}$   
418 originating in the  $hkl$  and exploring the vicinity of the Bragg peak, is small. Given the  
419 relatively reduced volume of  $\mathbf{q}$ -space around the Bragg peak that is usually probed in BCDI  
420 this approximation is also reasonable. This is also known as the Takagi approximation  
421 [45]. When this approximation is applied Eq.2.33 takes the form of the so-called *kinematic*  
422 *sum* and it is implemented in software for simulation of diffraction patterns starting from  
423 atomic positions like, for instance, the scattering module of PyNX [55].
- 424 •  $S(\mathbf{R}'_n) = S(\mathbf{R}_n)$ , which means that the shape function is not altered by the internal  
425 or superficial displacements. This is the case for displacements much smaller than the  
426 lattice parameter, which is easily fulfilled in typical BCDI samples. In practice, this  
427 approximation allows to describe the electron density on a periodic grid.

428 At this point, embedding the  $e^{i\mathbf{G}_{hkl} \cdot \mathbf{u}(\mathbf{R}_n)}$  in a more general *complex-valued shape function*  
429  $\tilde{S}$  and considering now the intensity of the signal impinging on the detector, we can write:

$$F_{\text{fin}}^{\text{strain}}(\mathbf{q}) = F_{\text{u.c.}}(\mathbf{G}_{hkl}) \sum_n \tilde{S}(\mathbf{R}_n) e^{i\mathbf{q} \cdot \mathbf{R}_n} \quad (2.33)$$

430 where we have now put the reference frame in the center of the Bragg peak at the  $hkl$   
431 position, therefore expressing the scattering factor as function of  $\mathbf{q}$ .

432 Eq.2.33 is now a discrete Fourier transform (DFT) of  $\tilde{S}(\mathbf{R}_n)$  calculated over the  $n$  points in  
433 coordinates  $R_n$  of the perfect crystal lattice and is the most important result of the chapter.  
434 Lastly, it is important to mention that when dealing with numerically computed DFT, in 3D,  
435 instead of the point-like  $\mathbf{R}_n$  coordinate, a general coordinate  $\mathbf{r}$  pointing at voxels of size  
436  $\delta r_i = 2\pi/q_i^{\max}$  (where  $q_i^{\max}$  is the extent of the Fourier window along the  $i$ -th dimension) is  
437 used. Moreover, later in the manuscript we will always refer to this complex-valued shape  
438 function as the “*complex object*” or “*reconstructed particle*” and will often deal with its complex  
439 phase.

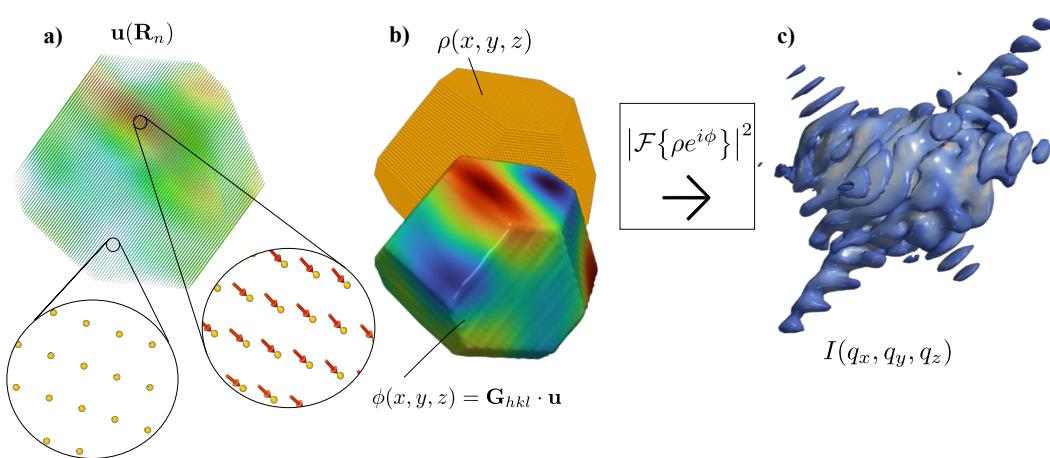
440 Specifically it will be used for simplicity the following relationship, which is derived like  
441 2.6 and drops all the constant terms:

$$I(\mathbf{q}) = |\mathcal{F}\{\rho(\mathbf{r})e^{i\phi(\mathbf{r})}\}|^2 \quad (2.34)$$

442 Eq.2.34 has the advantage of expressing the diffraction pattern with a DFT, which allows for  
443 fast computations via efficient algorithms, rather than a sum evaluated in each individual unit  
444 cell position. This represents a key ingredient for the most used phase retrieval algorithms in  
445 BCDI (as we will see in the next Chapter), due to the large number of DFT operations required.

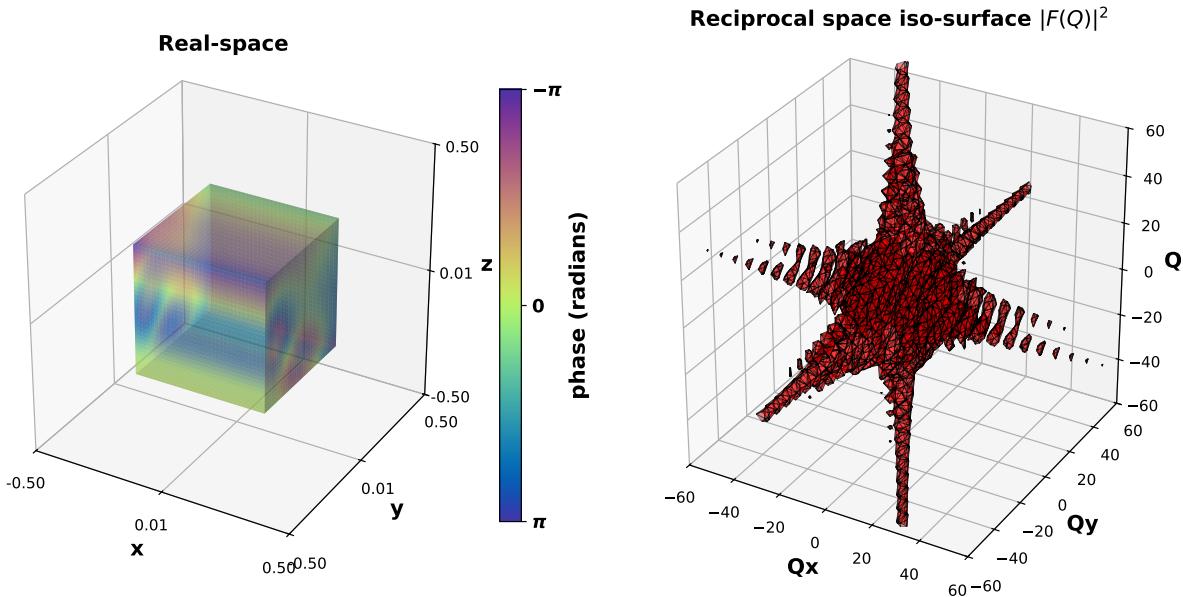
446 Interesting works have evaluated the validity of these approximations [56] and the discrepancy  
 447 between diffraction patterns calculated with the DFT and kinematic sum [57, 58].

448 A schematic representation of the link between the lattice displacements, the complex  
 449 object function and the corresponding diffraction pattern is provided in Fig. 2.5.



**Figure 2.5:** Schematic of the relationship between displacement field and diffraction pattern. **a)** Simulated displacement field obtained from the subtraction of the atomic positions of a simulated strained gold crystal with the relative perfect lattice copy. Close-up on two different regions where the displacement is small (left) and large (right). **b)** Formulation of the scattering object as a complex object. The modulus being interpreted as the electronic density and the phase being the projection of the displacement on the scattering vector pointing at the probed  $hkl$  node. **c)** Corresponding diffraction pattern proportional to the square modulus of the Fourier transform of the complex object.

450 It is interesting here to analyze what the effect of the displacement is on the shape of  
 451 the diffraction pattern. Intuitively we can think that the complex phase term is altering the  
 452 interferences of the scattered waves. If, for a perfect lattice, we had Dirac deltas broadened  
 453 and modulated by the Fourier transform of the shape function, here we have to consider an  
 454 additional broadening due to the strain. Moreover, we know from Friedel's law [59] that the  
 455 diffraction pattern of a real valued function (shape function for perfect crystals, i.e. without  
 456 strain) is always centro-symmetric while here, having introduced a complex term, we expect  
 457 this symmetry to be broken. Fig. 2.6 shows the effect of the displacement field on the diffraction  
 458 pattern.



**Figure 2.6:** Illustration of the squared modulus of the Fourier transform of a cube with an applied displacement field. The phase derived from the projection of the displacement on the scattering vector is represented in on the left. On the right the corresponding diffraction pattern. It is visible the deformation of the Bragg peak with respect to the one in Fig.2.3 for the perfect lattice case.

459 From Eqs.2.33–2.34 we can intuitively imagine that the distortion of the diffraction pattern is  
 460 linked to (i) the *magnitude* of the displacements, (ii) the *number* of lattice sites involved, (iii) the  
 461 *coherence* of each local contribution  $S(\mathbf{R}_i)e^{i\phi(\mathbf{R}_i)}$  with respect to its neighbors. The interplay  
 462 of these parameters gives rise to many possible distortions of the ideal crystal diffraction.

463 Lattice sites that are displaced with respect to the perfect lattice scatter at slightly different  
 464 Bragg angles. This produces intensity shifted to different  $\mathbf{q}$  values around the Bragg peak. The  
 465 displacements are proportional to the complex phases in the kinematic sum. If the phases are  
 466 the same across many sites, the sum is constructive, and the resulting diffraction signal remains  
 467 localized and peaked. On the other hand, if the phases vary from site to site, the constructive  
 468 interference is lost, and the diffraction pattern becomes more irregular.

469 It follows that the more voxels are involved in incoherent phase variations, the more  
 470 speckled the diffraction pattern becomes, since each contribution departs differently from the  
 471 original Bragg condition. This is often called *heterogeneous* strain. Conversely, in the case  
 472 of equal displacements across many sites, the contributions remain coherent, leading to the  
 473 so-called *homogeneous* strain. The diffraction signal in this case tends to split into two peaks:  
 474 one corresponding to the pristine lattice and one to the uniformly strained lattice.

475 An important case, discussed later in the text, is the high-strain regime, in which a significant  
 476 fraction of lattice sites undergo large heterogeneous displacements. This should be contrasted  
 477 with localized defects such as dislocations or grain boundaries (not treated here), where  
 478 relatively large displacements are confined to only a few lattice sites.

---

### 479 2.2.6 Noise

480 We should now introduce the last ingredient to our derivation, the presence of noise. When  
 481 dealing with real world experiments we have to consider that photons are quantum particles,  
 482 thus their arrivals on the detector surface are random independent events. Here, because of the  
 483 discrete nature of photons, a statistical approach based on the average arrival rate is required.  
 484 It follows that an uncertainty is intrinsically associated to the estimate, hence the emergence  
 485 of a source of noise. The statistical model describing these phenomena is known as Poisson  
 486 statistics [60]. In particular, given a constant *average* rate of arrival  $\eta$  over a unit of time, the  
 487 *expected* number of photons after an exposure time  $\tau$  is  $\mu = \eta\tau$ , and the probability of detecting  
 488  $N$  photons over an observing time  $\tau$  is given by the formula:

$$p(N) = \frac{(\eta\tau)^N e^{-\eta\tau}}{N!} \quad (2.35)$$

489 also called *Poisson probability distribution*. It can be proven that for the Poisson distribution  
 490 the variance equals the mean  $\mu = \sigma^2 = \eta\tau$ . Moreover, we can intuitively think the expected  
 491 number of photons collected during a time  $\tau$  by a pixel in position  $\mathbf{q}_i$  to be proportional to  
 492 the intensity in Eq.2.34 evaluated in  $\mathbf{q}_i$  and integrated over a time  $\tau$ . At this point, if we  
 493 were to make a measurement, in absence of background, we would have at position  $\mathbf{q}_i$  an  
 494 expected integrated intensity  $\mu = \bar{I}(\mathbf{q}_i)$  with an uncertainty of  $\sigma = \sqrt{\bar{I}(\mathbf{q}_i)}$ . Being the  
 495 Signal-to-Noise-Ratio (SNR) defined as  $SNR = \mu/\sigma$  we obtain in our case:

$$SNR = \frac{1}{\sqrt{\bar{I}(\mathbf{q}_i)}} \quad (2.36)$$

496 Though very simple, this equation is important and deserves some comments. First, we  
 497 can observe that in the limit of infinitely long integration times or infinite intensity (infinite  
 498 flux), the  $SNR \rightarrow 1$ , meaning that we would in principle able to deterministically measure  
 499 the diffraction pattern, in this case noiseless. Secondly, we know that most of the intensity  
 500 of a diffraction pattern is concentrated in the center of the Bragg peak (Fig.2.4), thus we can  
 501 deduce from Eq.2.36 that Poisson noise is affecting regions at high  $\mathbf{q}$  much more than at the  
 502 center. During PR, this uncertainty in high  $\mathbf{q}$  ranges reflects, in real space, to a lower accuracy  
 503 on fine resolution features of the reconstructed objects. The SNR, by mostly affecting lower  
 504 photon counts regions, typically located at high  $\mathbf{q}$  values, sets a limit to the actual attainable  
 505 direct-space resolution.

## 506 2.3 Practical BCDI at ID01 - ESRF

507 We have so far discussed the theoretical foundations for the understanding of the BCDI tech-  
 508 nique. It is now time to see the practical aspects.  
 509 Starting from the source of the probing radiation downstream to the detectors we will focus on  
 510 the most important stages, necessary to envision the experimental conditions as well as typical  
 511 numbers (resolution, energies, sizes).

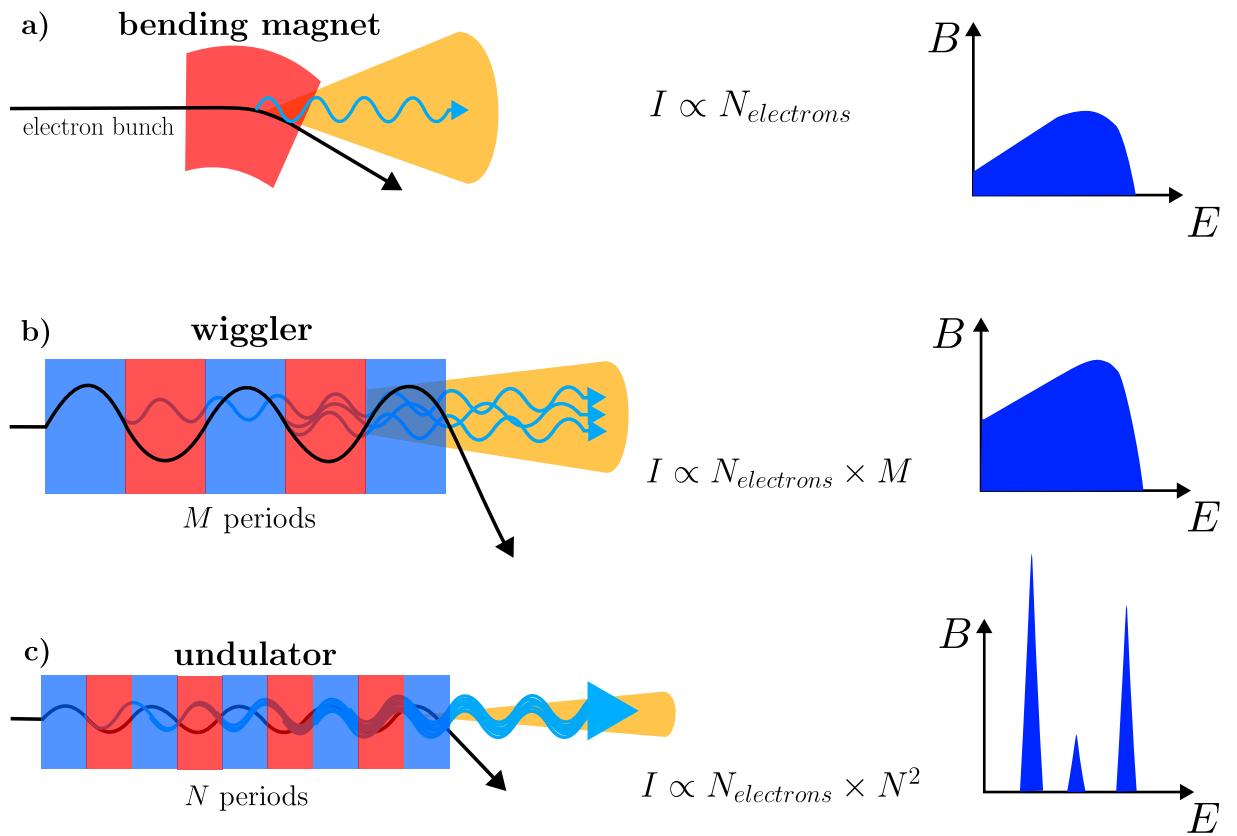
512 BCDI experiments require a *coherent* X-ray beam focussed onto a volume of a few cubic  
 513 microns or less. Moreover, such beams also need to deliver a sufficient amount of photons to

514 the sample, which, being in the micrometer range as well, possesses a limited scattering power.  
515 Such properties today can only be achieved in large scale facilities like synchrotrons and free  
516 electron lasers (FELs).

517 We know that electromagnetic waves can be generated by accelerating charges [61]. In syn-  
518 chrotron facilities electrons travel in a large circular ring (*storage ring*) at relativistic speeds  
519 and X-rays are produced deflecting their trajectories with the help of magnets. Depending on  
520 the magnet size, magnetic field and spatial configuration the X-ray beam is generated with  
521 different properties. Typically, these configurations are grouped into two main categories,  
522 namely: *bending magnets* and the more modern Insertion Devices (ID) like *w wigglers* and *undu-*  
523 *lators*. In synthesis, while bending magnets produce X-rays in a broad energy spectrum with  
524 relatively low intensity, wigglers exploit a series of smaller magnetic dipoles to increase the  
525 intensity of the radiation. Undulators on the contrary leverage the periodicity of the dipoles to  
526 stimulate coherent emission, hence sharpening the energy bandwidth around tunable peaks  
527 and increasing the intensity as well. The radiation generated by the latter is the most suited  
528 for coherent diffraction experiments because of its inherent higher coherence. An illustrative  
529 explanation is provided by Fig. 2.7 while an exhaustive description of synchrotron radiation  
530 can be found in the notable book from Prof. Giorgio Margaritondo [62].

531 The X-rays produced in these sections travel along straight lines, tangential to the storage  
532 ring, where they go through a number of optical elements dedicated to the spatial and spectral  
533 filtering as well as focusing and collimation, down to the experimental hutch where they finally  
534 radiate the samples. These are called *beamlines*, and they are usually designed and equipped  
535 for a specific class of techniques.

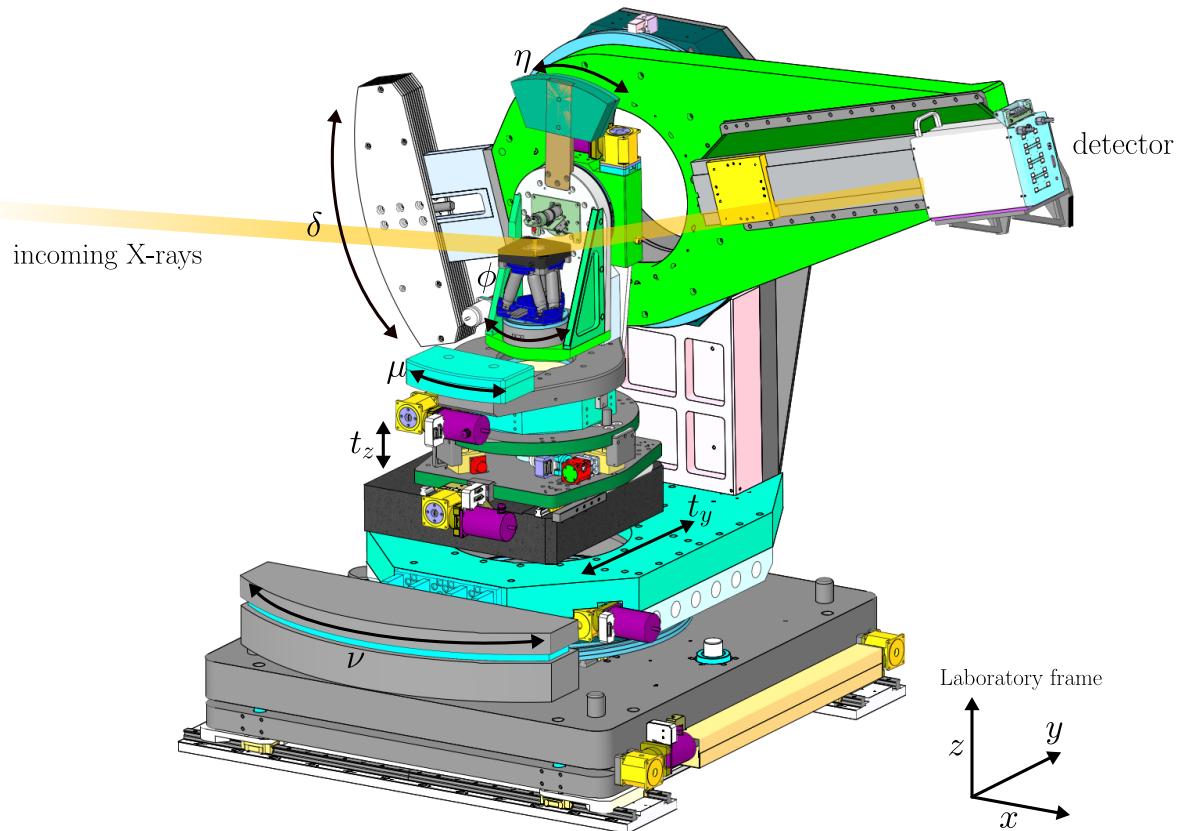
536 At the European Synchrotron Radiation Facility (ESRF) the ID01 beamline was conceived to  
537 combine the lattice parameter resolution provided by X-ray diffraction with spatial resolution  
538 of imaging techniques [27] and is able to offer various techniques for the investigation of strain  
539 at nanoscale, including BCDI.



**Figure 2.7: Illustration of the three main synchrotron X-ray sources.** **a)** The intensity of the bending magnet radiation scales with the number of electrons being deflected. The beam is emitted with a broad horizontal divergence and the spectrum covers multiple wavelengths. **b)** The wiggler exploits a series of magnets to increase the intensity of the emitted radiation. **c)** In the undulator the periodicity of the poles is such that the photons are emitted with higher temporal coherence. The X-ray beam shows narrower emission cone (increased transverse coherence) as well as higher intensity and narrower bandwidths. Adapted from [63].

### 540 2.3.1 Beam size

541 Typical BCDI experiments are focused on single nanoparticles. The diffraction pattern is the  
 542 result of the scattering of a single crystal in Bragg condition. It is therefore necessary to have  
 543 a beam size of the order of the size of the crystal. At ID01 the beam size can scale down to  
 544  $35 \times 35\text{nm}$  [64], but typical sizes range between  $0.5 - 1\mu\text{m}^2$ . A too large beam could shine on  
 545 some neighbor particles which, if in Bragg condition, could contaminate the diffraction pattern.  
 546 These often appear as isolated bright spots, called *aliens* [65]. On the contrary, regions of the  
 547 same particle that are not illuminated do not contribute to the diffraction pattern, therefore are  
 548 not imaged in the reconstructed objects.



**Figure 2.8:** Computer Aided Design of the diffractometer at the ID01 beamline with the corresponding degrees of freedom. Adapted from [44]

### 549 2.3.2 Energy and flux

550 An important figure of merit for the characterization of synchrotron radiation is *brilliance*. It  
 551 measures the number of photons populating the beam per unit of time, area and solid angle  
 552 within a 0.1 % fraction of bandwidth relative to a certain energy. It is therefore defined as:

$$B = \frac{\text{nb photons}}{dt \cdot dA \cdot d\Omega \cdot 0.1\%dE/E} \quad (2.37)$$

553 ID01's radiation is generated by up to three undulators, one of which is tuned to deliver  
 554 the optimum brilliance in the range of 6 - 11.5 keV. This energy range is particularly suited for  
 555 coherent diffraction experiments on crystals . In fact, at lower energy absorption effects from  
 556 optical elements and air start being important, hence reducing the net flux available on the  
 557 sample. On the other hand, at higher energies other problems arise, namely: (i) the degree of  
 558 coherence drops significantly (see next paragraph) and (ii) the diffraction patterns, for fixed  
 559 detector distance, shrink in a lower number of pixels causing problems for PR.<sup>1</sup>

<sup>1</sup>This is related to the concept of the *oversampling condition* discussed in the next chapter

560 **2.3.3 Coherence**

561 This fundamental property of electromagnetic radiations is probably the most important for  
 562 CDI. In the derivation of the scattering equation Eq.2.1 and following, we have always assumed  
 563 the incident radiation to be described by a plane wave, perfectly monochromatic ( $\Delta\lambda \neq 0$ ) and  
 564 with no angular spreading ( $\Delta k = 0$ ). In the ideal case, the wave-field is perfectly coherent,  
 565 meaning that the relative phase between any two points in space is completely well-defined. In  
 566 reality, electromagnetic waves have finite temporal duration and therefore, by Fourier transform  
 567 relations, a finite spectral bandwidth. Likewise, they are produced by extended sources and  
 568 observed at finite distances, which implies a finite angular spread of the wave-vectors. These  
 569 limitations introduce phase decorrelation between fields evaluated at two different points, either  
 570 *along* the propagation direction (due to bandwidth) or *across* it (due to angular dispersion). For  
 571 this reason, it is useful to introduce quantities that characterize the “degree of coherence” of  
 572 the radiation. An intuitive approach is to define a *longitudinal coherence length*, associated with  
 573 phase delays from spectral bandwidth, and a *transverse coherence length*, associated with phase  
 574 delays from angular spread. A pictorial view of these characteristic lengths is provided by Fig.  
 575 2.9

576

577 **Longitudinal Coherence Length**  $L_l$ : is defined as the distance after which two beams  
 578 originated in the same point and characterized by a wavelength difference  $\Delta\lambda = 0$  have a  
 579 phase difference of  $\pi$ , being therefore in phase opposition. This means that after a distance  $2L_l$   
 580 the two waves are back in phase. If the first wave of wavelength  $\lambda$  needs  $N$  cycles to cover  $2L_l$ ,  
 581 it follows that the second wave with wavelength  $\lambda + \Delta\lambda$  will need  $N - 1$  cycles. Therefore:

$$N\lambda = (N - 1)(\lambda + \Delta\lambda) \quad (2.38)$$

thus,  $\Delta\lambda(N - 1) = \lambda$

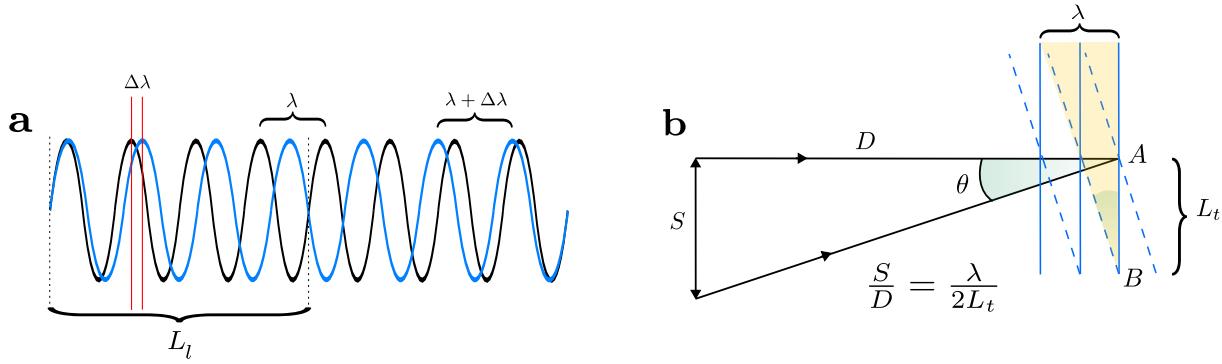
582 In case of small  $\Delta\lambda/\lambda$  we can approximate  $N - 1 \approx N$ , and obtain  $N \approx \lambda/\Delta\lambda$ . This leads  
 583 us to the expression for the longitudinal coherence length:

$$L_l \approx \frac{\lambda^2}{2\Delta\lambda} \quad (2.39)$$

584 As expected the longitudinal coherence increases with the monochromaticity of the beam.  
 585 On the contrary it significantly worsens for high energies, as anticipated above.

586 **Transverse Coherence Length**  $L_t$  is defined as the distance between two points  $A$  and  
 587  $B$  sitting on a plane at distance  $D$  from a source of size  $S$  for which both  $A$  and  $B$  are out of  
 588 phase. Fig. shows geometrically that this distance, when  $S \ll D \rightarrow \theta \approx 0 \rightarrow \cos(\theta) \approx 1$ , the  
 589 transverse coherence length is given by:

$$L_t \approx \frac{\lambda D}{2S} \quad (2.40)$$



**Figure 2.9:** **a** Sketch of the longitudinal coherence length  $L_l$ . Two plane waves with slightly different wavelength originating in phase are out of phase after propagating over a distance equal to  $L_l$ . **b** Sketch of the transverse coherence length  $L_t$ . Two plane waves emitted at slightly different angles from an extended source are evaluated on a distant plane. They become out of phase when separated by  $L_t$ . Being the highlighted triangle similar to the bigger one in black, an expression for  $L_t$  can be derived from simple geometrical considerations. Adapted from [43]

590 Also in this case we see the coherence decreasing for high energies. Moreover, the dependence  
 591 on  $S$  and  $D$  explains the challenges to build long beamlines (118 meters for ID01) and  
 592 reduce the electron beam size and angular divergence. The transverse coherence length can be  
 593 easily lifted to a coherence surface for a source extended in 2D ( $L_t^{hor} \times L_t^{ver}$ ).  
 594

595 Given these premises we can now calculate the coherence volume ( $L_l \times L_t^{hor} \times L_t^{ver}$ )  
 596 available ID01 for a typical X-ray energy used in BCDI. Given the source sizes of  $60\mu m$  hori-  
 597 zontally and  $15\mu m$  vertically, at 8 keV ( $\lambda = 1.55 \text{ \AA}$ ) we obtain a transverse coherent surface  
 598 of  $152\mu m \times 610\mu m$ . The longitudinal coherence length is found to be approximately  $0.8\mu m$   
 599 considering the  $\Delta\lambda/\lambda \sim 10^{-4}$  typical of the double Si (111) crystal monochromator. It follows  
 600 that often, in BCDI experiments, the limitation on the particle size is given by the longitudinal  
 601 coherence [66].  
 602

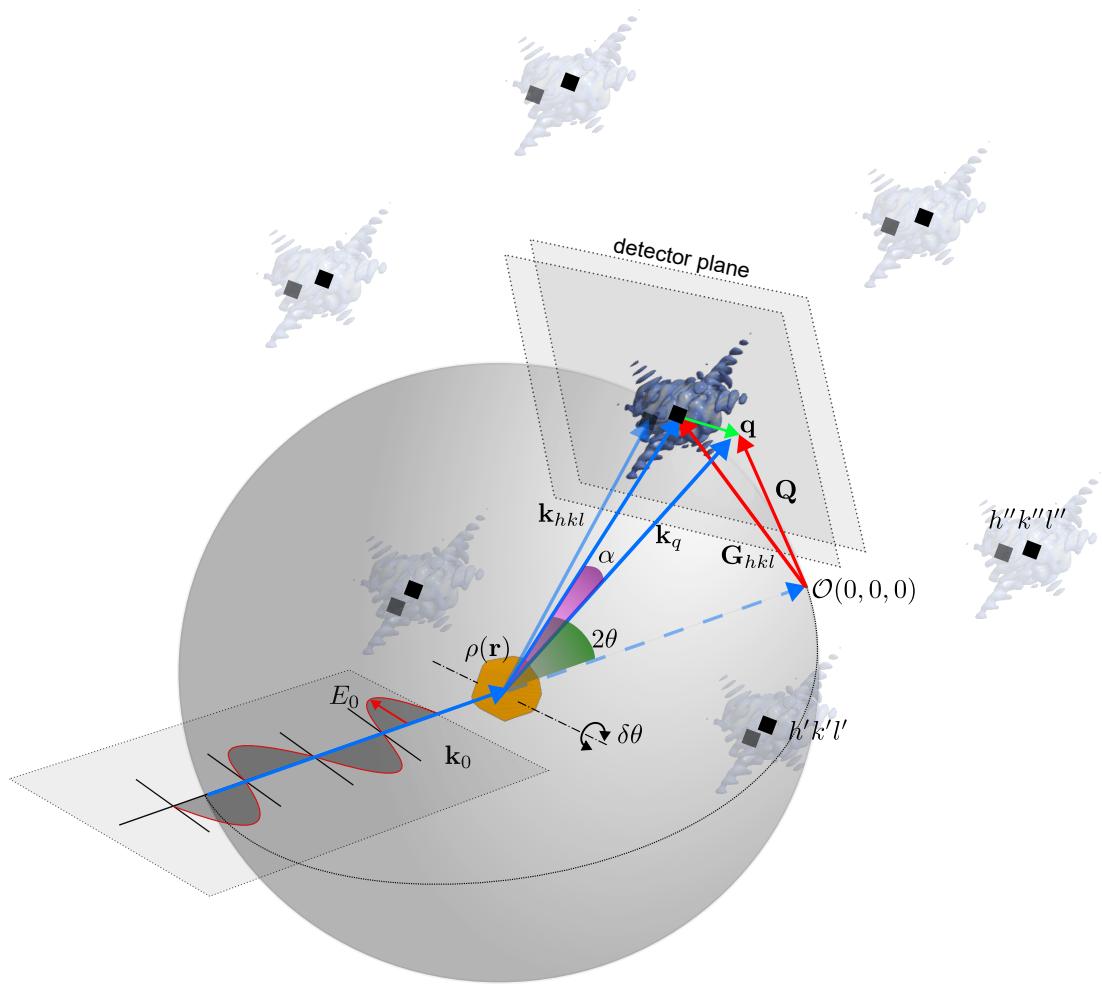
603 The coherence volume specifies the maximum spatial separation between two scatterers for  
 604 which their scattered waves remain mutually coherent and can interfere. For this reason, while  
 605 classical X-ray diffraction is limited to the interference of a few lattice points, in the order of  
 606 few  $\text{\AA}$ , the X-ray beam at ID01 can enable interference from objects separated by the order of  
 607 few  $\mu m$ . In BCDI, the sample size is usually smaller than the coherence volume, meaning that  
 608 scattering from all parts of the crystal, even from opposite surfaces, remains mutually coherent.  
 609 This results in the characteristic interference pattern observed.

### 610 2.3.4 Ewald sphere and Rocking curves

611 At this point of the dissertation we should address the procedure related to data acquisition. In  
 612 particular, in order to understand the generation of 3D diffraction patterns from the 2D images  
 613 collected by the detector we need to introduce the concept of the Ewald sphere.  
 614

615 We have said that in BCDI we focus on a single  $hkl$  node of the reciprocal lattice, and we  
 collect the diffraction pattern of the full diffracting crystal around that Bragg peak. Given the

exit wave-vector  $\mathbf{k}$  pointing at the Bragg peak, we could draw a sphere of radius  $k$  centered in the sample and find the diffraction pattern of interest at the intersection of such sphere with the reciprocal lattice in the  $hkl$  node. Such a sphere is called the *Ewald sphere* and it is illustrated in Fig.2.10. More generally, one can state that the diffraction from a wave-vector  $\mathbf{k}$  occurs only in those reciprocal lattice points that intersect the Ewald sphere of radius  $k$ . Moreover, the surface of the sphere around the Bragg peak tells us the signal captured by the 2D detector. In order to collect the 3D pattern one usually needs to apply a small rotation the sample along the axis that crosses the sample horizontal and perpendicular to  $\mathbf{k}$  such that the Bragg condition is found for the same  $hkl$  node in a slightly shifted position in space. The Ewald sphere is no longer cutting the diffraction pattern in the precise  $hkl$  node but in the extended peak broadened by the finite size effect. When this rotation, called *rocking curve*, is performed along both directions from one extremity of the pattern to the opposite, a 2D image is recorded by the detector at each angular step. The full stack of 2D images creates the full 3D diffraction pattern.



**Figure 2.10:** Illustration of the typical geometry for a BCDI experiment and the Ewald sphere (not to scale). A monochromatic plane wave approximates a coherent X-ray beam that fully illuminates a sample with a wave-vector  $\mathbf{k}_i$ . A diffraction pattern given by Eq. 2.33 is formed around each  $hkl$  node. Of the different reciprocal lattice nodes a single one is selected for the measurement, depending on the detector. The Ewald sphere, of radius  $\mathbf{k}_{hkl}$  shows that the diffraction condition is met at the intersection with the  $hkl$  node. The detector plane, perpendicular to the  $\mathbf{k}_{hkl}$  vector, captures a swathe of Ewald sphere extending of an amount  $\mathbf{q} = \mathbf{k}_q - \mathbf{k}_{hkl}$  around the  $hkl$  node. Different slices of the diffraction pattern are acquired rotating the sample at each angle  $2\theta \pm N\delta\theta$ . Notice that this angle is equivalent to the  $\eta$  angle in Fig. 2.8, typically tuned during the rocking curve.

This approach assumes that the surface of the Ewald sphere in the vicinity of the Bragg peak can be approximated to be flat. In reality onto the detector is projected the slice of diffraction pattern extending over a swathe of the Ewald sphere of radius  $r = k$ . It follows that for higher energies this approximation holds better as the contact angle is decreasing. Additionally, another consequence of the reduction of the contact angle at high energies is that the reciprocal space is shrinking. In order to understand this one can consider that: geometrically the contact angle is equivalent to the angle  $\alpha$  between the vectors  $\mathbf{k}_q$  and  $\mathbf{k}_{hkl}$ . Being now  $\delta q = |\mathbf{k}_q - \mathbf{k}_{hkl}| = |\mathbf{k}_q| \sin(\alpha)$  one could infer that for higher energies (larger  $\mathbf{k}_{hkl}$ ) the portion of diffraction pattern explored by  $\delta q = |\mathbf{k}_q| \sin(\alpha) \approx |\mathbf{k}_q|\alpha$  is now shrunk into a

639 smaller  $\alpha$ . This effect compresses the whole reciprocal lattice and thus the diffraction pattern  
640 as well, around the  $hkl$  node. This issue causes constraints in the sampling as will be discussed  
641 in the next Chapter.



## 643 THE PHASE PROBLEM IN BCDI

644 This chapter is dedicated to the discussion of the *Phase Problem* in BCDI and the main computational methods that are currently adopted to solve it. This problem arose in the beginning in  
645 the field of X-ray crystallography since the first measured diffraction patterns, but similarly  
646 affects other domains like astronomical and seismic imaging as well as the coherent diffraction  
647 imaging of our interest. As anticipated briefly in the preface, the phase problem arises from  
648 a technical limitation. In X-rays, the fast oscillations of the electromagnetic fields induce  
649 detectors to only measure a time-averaged intensity (Eq.2.5) with the consequent loss of the  
650 phase information in the measurement.

652 The Fourier phase problem is therefore the impossibility to compute the complex-valued  
653 signal  $\tilde{\rho}(\mathbf{r})$  from the intensity measurement of  $I(\mathbf{q}) = |\mathcal{F}\{\tilde{\rho}(\mathbf{r})\}|^2$  with a simple inverse Fourier  
654 transform of the type  $\mathcal{F}^{-1}\{\sqrt{I(\mathbf{q})}e^{i\varphi(\mathbf{q})}\}$  because of the lost reciprocal space phase  $\varphi(\mathbf{q})$ . It is  
655 thus necessary to find alternative strategies, often based on iterative algorithms, to perform  
656 the Phase Retrieval (PR) and recover the signal. However, as one can suspect, the modulus  
657 operation applied to the Fourier transform allows in principle an infinite variety of complex  
658 functions to be solution. For this reason the problem is said to be *ill-posed*. Consequently,  
659 the modulus operation renders the optimization landscape *non-convex*, so that phase retrieval  
660 algorithms must search for the desired solution (the *global minimum*) within a space populated  
661 by multiple “valleys” (*local minima*), where they may become trapped. In this context, the  
662 search for the solution to the problem has fascinated (and still does) scientists for decades,  
663 contributing to an extensive production of works in literature.

664 The first published studies date back to 1951 when Sayre, in a comment [67] to the paper  
665 by Shannon *Communication in the presence of Noise* [68], proposed a condition on the sampling  
666 of the diffraction pattern for the restoration of the unit cell extent. Later in 1972 Gerchberg  
667 and Saxton [69] developed an algorithm capable of inverting the diffraction pattern that is  
668 nowadays at the basis of currently used standard PR algorithms. However, a theoretical analysis  
669 of the uniqueness conditions was provided only later by Bruck and Sodin (1979) [70]. The  
670 authors showed that, for 2D and 3D problems, the phase retrieval has unique a solution except  
671 for rare cases, therefore conferring the mathematical solidity to the algorithm’s results. Later in  
672 1982 Bates draws the link between uniqueness and the Sayre sampling intuition, as necessary  
673 condition for 2D case [71]. A refined version of the Gerchberg - Saxton algorithm was proposed  
674 by J.R. Fienup in 1978 [1] who named it Error Reduction (ER). In [2], published in 1982, the  
675 same author developed the Hybrid-Input Output (HIO) algorithm, able to outperform ER, and

676 compared gradient-descent methods as well. In 1987 again Fienup showed the possibility  
677 of reconstructing *complex-valued* objects if the constraints on the object support are “tight”,  
678 i.e. the shape of the object is known [72]. This result is particularly interesting for BCID  
679 since, as we have seen in Eq.2.34, the object to be retrieved is complex-valued. Based on  
680 the suggestion of Sayre in 1991 [73] the works of Miao and coauthors from 1998 opened the  
681 X-ray coherent diffraction imaging field by exploiting phase retrieval, combining the sampling  
682 proposed by Sayre and iterative algorithms developed by Fienup [4, 74, 75]. Among the  
683 many works published later it is worth mentioning the Difference Map algorithm [76], which  
684 generalized Fienup’s HIO introducing an additional non-linear term and the (Relaxed) Averaged  
685 Alternating Reflections algorithms developed by Luke [3, 77] with better convergence properties  
686 guaranteed by reflection operations. Later in 2004 the work of Rodenburg and Faulkner [78]  
687 showed improved PR for diffraction patterns obtained illuminating the sample from multiple and  
688 partially overlapping regions. It was the birth of ptychography, later successfully demonstrated  
689 in Bragg condition as well [79]. Though not usually employed in CDI because of the large  
690 computational costs, it is worth mentioning the more recent works of Candès and coauthors  
691 who have developed, the PhaseLift algorithm [80] and the “Wirtinger flow” [81]. While the  
692 former turns the PR into a *convex* problem, thus improving the stability and convergence  
693 guarantees, in the latter the PR is solved as least square problem with gradient descent using  
694 the Wirtinger derivatives for complex functions and a *spectral initialization* method that enables  
695 the start of the PR near the global minimum.

696 Further details on the applications of PR algorithms can be found in the review by Shechtman  
697 [82] while recent developments and theoretical insights can be found in the work of Fannjiang  
698 and Strohmer [83]. Of great help for a unified overview of the main PR algorithms with  
699 comparisons is the work of Marchesini [84].

700 Here we will present first the sampling condition, the phase ambiguities and the main  
701 alternating projections algorithms that will be encountered later in the chapters dedicated to  
702 the results.

### 703 3.1 Oversampling

704 Let us consider a direct space complex object  $O(x)$  extended over a region of space  $R$ , and its  
705 Fourier transform  $\tilde{O}(q) = \mathcal{F}\{O(x)\}$  in 1D defined as:

$$O(x) = \rho(x)e^{i\phi(x)} \quad \tilde{O}(q) = A(q)e^{i\varphi(q)} \quad (3.1)$$

706 The measurement of the diffracted intensity of the object would be equal to, barring  
707 constants,  $I(q) = |A(q)|^2$ . We should consider now that we are measuring  $I(q)$  on a finite size  
708 detector made of discretized pixels. It thus follows the question: how finely in space should we  
709 sample the signal such that we can recover  $O(x)$ ?

710 If  $O(x)$  is a square of size  $R$ , Nyquist theorem states that each point sampling  $\tilde{O}(q)$  should  
711 have a spacing  $\Delta q = 1/R$ . In our case we measure  $|\tilde{O}(q)|^2$  which, corresponds to the Fourier  
712 transform or the so-called *autocorrelation function* (or *Patterson map* [85]) of the object  $(O(-x)*$   
713  $O(x))$ , which extends over a size  $2R$ . Hence, the sampling should happen every  $\Delta q = 1/2R$   
714 to recover the autocorrelation of  $O(x)$  without aliasing. This should in principle contains the  
715 information necessary to recover  $O(x)$ . This intuition was proposed by Sayre in 1952.

Following this idea a more rigorous explanation was given by Miao *et al.* in [4] in which the definition of *oversampling condition* is given. The salient ideas can be summarized as follows. With a hypothetical detector of  $N$  pixels on a line the extent of reciprocal space measured is  $\Delta q = N\delta q$  where  $\delta q$  is the extent of a single pixel. The  $\mathbf{q}$  vector of Fig.2.10 is now discretized in a  $q_k$  where  $k \in [0, N - 1]$ . In the direct space as well the coordinate  $x$  is now discretized into  $N$  values  $x_n$  where  $n \in [0, N - 1]$  and the extent of direct space is  $\Delta x = N\delta r$ . According to the relationship between direct and reciprocal space the pixel size  $\delta q = \frac{2\pi}{\Delta x} = \frac{2\pi}{N\delta x}$  which implies a Nyquist sampling. Hence, we can write the diffracted amplitude impinging on the detector as a discrete Fourier transform in each pixel.

$$\tilde{O}(q_k) = \sum_{n=0}^{N-1} O(x_n) e^{i \frac{q_k r_n}{N}} = \sum_{n=0}^{N-1} \rho(x_n) e^{i\phi(x_n)} e^{i \frac{q_k r_n}{N}} \quad (3.2)$$

Observing the above equation we can notice  $N$  variables but  $N \times 2$  unknowns ( $\rho(x_n), \phi(x_n)$ ), hence making the system under-determined. Now, by using the Sayre condition of sampling at double the frequency  $\delta q = \frac{2\pi}{2N\delta x}$  the system becomes solvable. In practice the size of the measured array is fixed by the detector, therefore one can reduce the number of unknown variables in the direct space to ensure a good sampling. In other words the object array is padded with a number of zeros determined by the oversampling condition defined as:

$$\sigma = \frac{\text{total pixel number}}{\text{unknown-valued pixel number}} \quad (3.3)$$

In the 2D or 3D case the same factor 2 needs to be fulfilled in order to have an (over)determined system of equations. However, one can calculate an oversampling ratio along each dimension  $d$ , resulting to be  $\sigma \geq 2^{1/d}$  [86]. Nevertheless, it is preferable to ensure a larger value for  $\sigma$  along each dimension for better reconstructions [87, 88].

735

Another interesting remark is that the oversampling condition can vary depending on the energy of the beam and on the distance of the detector with respect to the sample. In fact, we have seen that at high energy the reciprocal space shrinks (Fig.2.10), meaning that the same  $\Delta q$  is compressed into less detector pixels. Considering the detector positioned at distance  $D$  with respect to the sample, having a pixel size  $p_{ix} \ll D$  we can approximate the angle subtended by the pixel as  $\alpha = \frac{p_{ix}}{D}$ . This angle is also approximated to be the angle subtended by  $\delta\mathbf{q} = \mathbf{k}_q - \mathbf{k}_{hkl}$  as in Fig.2.10. We can therefore write:

$$\delta q = |k_q| \frac{p_{ix}}{D} \quad (3.4)$$

From which we see that to explore the same extent in  $q$  whilst fulfilling the oversampling condition, at high energies we need to have smaller pixel sizes or move the detector further away from the sample.

Beside this formal definition of the oversampling condition, it is useful to introduce a more practical one in terms of number of pixels. In particular, one can derive that fulfilling the oversampling condition is equivalent to a sampling in reciprocal space that is smaller than the distance between fringes by a factor of at least two. In other words, in the detector frame, at least two pixels are required between two fringes.

$$\delta q \leq \frac{\delta q_{\text{fringe}}}{2} \quad (3.5)$$

Ensuring a spacing of at least two pixels between fringes along all directions, naturally confines the object in direct space into an array which is at least half the size of the window containing the diffraction pattern in detector space. In practice a spacing of at least three pixels is always aimed during experiments to ensure easier invertibility.

## 3.2 Phase Symmetries

We have seen how the oversampling condition is fundamental for the uniqueness of the solution of the Phase Problem. However, it is important to point out that this uniqueness is valid except for some trivial reciprocal space phase ambiguities [2, 89], or symmetries, that correspond to equivalently good solutions in direct space. They are here listed:

- **Object translation - Phase ramp:** A translation of  $\mathbf{x}$  from the center  $\mathbf{x}_0$  of the object in direct space will correspond in Fourier space to the addition of a linear phase ramp to the reciprocal space phase  $\varphi(\mathbf{q}) = \varphi(\mathbf{q}_0) + \mathbf{q} \cdot \mathbf{r}$ . The physical meaning is unaltered since the object itself has not changed but its absolute position in space has. Equivalently, a translation of the Bragg peak in reciprocal space yields a phase ramp applied to the object in direct space. Practically, if during PR the diffraction pattern was not centered, the reconstructed object presents a non-physical phase ramp artifact that if not removed can induce a misinterpretation of the strain field.
- **Global phase shift - Constant offset:** The addition of a constant offset to the reciprocal space phase  $\varphi(\mathbf{q}) = \varphi(\mathbf{q}) + \psi$  corresponds to a global phase shift applied to the object  $e^{i\psi} O(\mathbf{r})$ , which is physically equivalent to  $O(\mathbf{r})$ . In fact, the global phase shift does not affect the strain, as this is calculated from the gradient of the phase. This equivalence is expressed in Fourier space by the unaltered squared modulus  $|A(\mathbf{q})e^{i\varphi(\mathbf{q})}|^2 = |A(\mathbf{q})e^{i(\varphi(\mathbf{q})+\psi)}|^2 = |A(\mathbf{q})|^2$ .
- **Object inversion - Sign symmetry:** The geometric inversion of the complex conjugate of the object  $O = \rho(\mathbf{r})e^{i\phi(\mathbf{r})}$  given by  $O' = \rho(-\mathbf{r})e^{-i\phi(-\mathbf{r})}$  corresponds to a sign flip of the reciprocal space phase  $\varphi(\mathbf{q}) = -\varphi(\mathbf{q})$ . The overall diffracted intensity (information is lost in the modulus squared operation), the object  $O$  and its *twin image*  $O'$  are equivalently good solutions as the information about their absolute orientation in space is lost. It can however happen, especially for nearly centro-symmetric objects that this symmetry is not broken automatically during PR and that the algorithm stagnates on the superposition of, the object and its twin. This combination of  $O$  and  $O'$  is not a solution to the phase problem [90] and therefore de-twinning strategies have to be employed [91].

It follows that, during multiple PR runs of the same BCDI data, equivalent reconstructed objects can differ between each other by additional linear phases, as well as spatial translations, shifted phase ranges and inverted complex conjugation.

### 786 3.3 Alternating projections algorithms

787 In this section the class of algorithms known as “alternating projections” (AP) mentioned above  
 788 is presented, and the three most used algorithms in BCDI are described in more detail. We  
 789 invite the reader to refer to the more exhaustive lecture notes by Cegielski [92] or the review  
 790 written by Marchesini, from which the following paragraphs take inspiration [84].

791

792 Before delving into the details of each algorithm is important to clarify some fundamental  
 793 concepts.

794 The goal of the Phase Retrieval is to reconstruct the complex object in direct space  $O^*(\mathbf{r}) =$   
 795  $\rho(\mathbf{r})e^{i\phi(\mathbf{r})}$  given the intensity measurement  $I(\mathbf{q}) = |\mathcal{F}\{\rho(\mathbf{r})e^{i\phi(\mathbf{r})}\}|^2 = |A(q)e^{i\varphi(q)}|^2 = |A(q)|^2$ .

796 The solution space is therefore a Hilbert space  $\mathcal{H} \in \mathbb{C}^N$  where  $N$  is the number of complex-  
 797 valued pixels, limited by typically two constraint sets  $\mathcal{C}_s$  and  $\mathcal{C}_m$ , defined as:

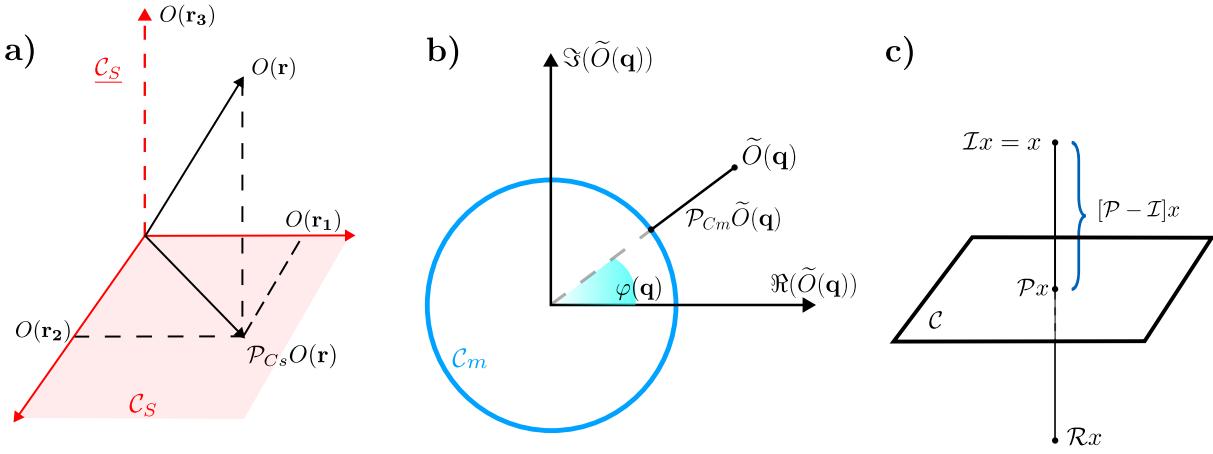
- 798 •  $\mathcal{C}_s = \{O(\mathbf{r}) \in \mathcal{H} : O(\mathbf{r}) = 0 \quad \forall \mathbf{r} \notin \mathcal{S}\}$  Often called “support constraint” is the set  
 799 containing all objects with zero amplitude outside the *support*  $\mathcal{S}$ . This last, in BCDI,  
 800 coincides with the shape function encountered in Eq.2.18 and it is in principle unknown.
- 801 •  $\mathcal{C}_m = \{O(\mathbf{r}) \in \mathcal{H} : |\mathcal{F}\{O(\mathbf{r})\}| = m\}$  Often called “modulus constraint” is the set  
 802 containing all objects with Fourier transform of modulus  $m$ . This set is however “non-  
 803 convex” as  $|\mathcal{F}\{O(\mathbf{r})\}| = m$  is fulfilled for any reciprocal space phase. This poses  
 804 challenges for deriving the convergence criterion of AP operating on this set [93].

805 The Phase Problem is then formulated as a *feasibility problem*:

$$\text{find the object } O(\mathbf{r})^* \in \mathcal{C}_s \cap \mathcal{C}_m \quad (3.6)$$

806 Moreover, we can define operators  $\mathcal{T}$  which transform the object according to the constraint  
 807 set they are operating in, and are used to bring the current object estimate closer to the solution  
 808 at each iteration. More precisely we can define:

- 809 • **Projector** onto the set  $C$  as  $\mathcal{P}_C(x) = \arg \min_{y \in C} \|y - x\|$ . It maps  $x$  to the nearest point  
 810  $y$  on the constraint set  $C$  in the Euclidean norm. The projector produces a *feasible point*,  
 811 as the mapped point belongs to the constraint set  $C$ .
- 812 • **Reflector** with respect to the set  $C$  as  $\mathcal{R}_C(x) = 2\mathcal{P}_C(x) - x$ . It maps  $x$  to a point  $y$   
 813 across the constraint set  $C$  by applying two times the projector. Since  $y$  does not belong  
 814 to  $C$  the reflector does not produce a feasible point.
- 815 • **Identity**  $\mathcal{I}$  as the operator that leaves unaltered the operated estimate.



**Figure 3.1:** Geometrical representation of (a) the projection of the object onto the support constraint set, (b) the projection of the object onto the Fourier modulus constraint set, (c) the identity, projector and reflector operators. Inspired by [84]

816 In our case we have that the projector onto  $C_s$  applied to the object sets to zero the values  
 817 outside the support and does not alter the values inside:

$$\mathcal{P}_{C_s}(O(\mathbf{r})) = \begin{cases} O(\mathbf{r}), & \mathbf{r} \in \mathcal{S} \\ 0, & \mathbf{r} \notin \mathcal{S} \end{cases} \quad (3.7)$$

818 On the other hand the projector onto  $C_m$  applied to the object replaces the modulus of its  
 819 Fourier transform with the squared root of the measured intensity  $m = \sqrt{I(\mathbf{q})}$ .

$$\mathcal{P}_{C_m}(\tilde{O}(\mathbf{q})) = \mathcal{P}_{C_m}(A(q)e^{i\varphi(q)}) = \sqrt{I(\mathbf{q})}e^{i\varphi(q)} \quad (3.8)$$

820 This step forces the modulus of the Fourier transform of the object at the iteration  $k$  to be  
 821 exactly the measured magnitude.

822 Since  $\mathcal{P}_{C_m}$  and  $\mathcal{P}_{C_s}$  operate in two conjugate spaces (direct-reciprocal), when used in  
 823 sequence a direct or inverse Fourier transform is implied in between. The symbol will be  
 824 omitted to simplify the notation.

825 At this point we have all the necessary ingredients to introduce the three main AP algorithms  
 826 used for BCDI PR.

### 827 3.3.1 Error Reduction (ER)

828 If we consider as a starting point the object  $O^0(\mathbf{r}) = \mathcal{F}^{-1}\{\sqrt{I(\mathbf{q})}e^{i\varphi^0(\mathbf{q})}\}$ , obtained by the  
 829 inverse Fourier transform of the squared root of the measured intensity with a random complex  
 830 phase array  $\varphi^0(\mathbf{q})$ , we can express the object at the  $k$ -th iteration of the *Error Reduction* (ER)  
 831 algorithm as:

$$O^{k+1}(\mathbf{r}) = \mathcal{P}_{C_m}\mathcal{P}_{C_s}(O^k(\mathbf{r})) \quad (3.9)$$

832 ER is the simplest and most intuitive AP algorithm as it only projects back and forth the  
 833 object between the two sets. Although it guarantees linear convergence, ER is not optimal since  
 834 it only converges to the nearest local minimum, and it is unable to escape it. For this reason in  
 835 typical BCDI it is used to at the end of the PR, when the current estimate is close enough to the  
 836 final solution. Additionally, one can define the magnitude error functional  $\varepsilon_m(O)$  as:

$$\varepsilon_m(O) = \|\mathcal{P}_{Cm}(O) - O\|_{L2} \quad (3.10)$$

837 i.e. the Euclidean distance between the current estimate  $O$  and its projection onto the Fourier  
 838 modulus constraint set. Differentiating this functional with respect to  $O$  yields the gradient  
 839  $\nabla \varepsilon_m(O)$ , which points in the direction that reduces the Fourier magnitude mismatch, or in  
 840 other words, the natural descent direction of the error. If this gradient is further restricted to  
 841 the support constraint set, one obtains a *projected gradient-descent step*  $\nabla_s \varepsilon_m(O)$  which is the  
 842 part of the descent direction that lies inside the feasible object region.

843 Marchesini shows that the ER step can thus be rewritten as:

$$O^{k+1}(\mathbf{r}) = \mathcal{P}_{Cs}(O^k(\mathbf{r})) - \frac{1}{2} \nabla_s \varepsilon_m^2(O^k(\mathbf{r})) \quad (3.11)$$

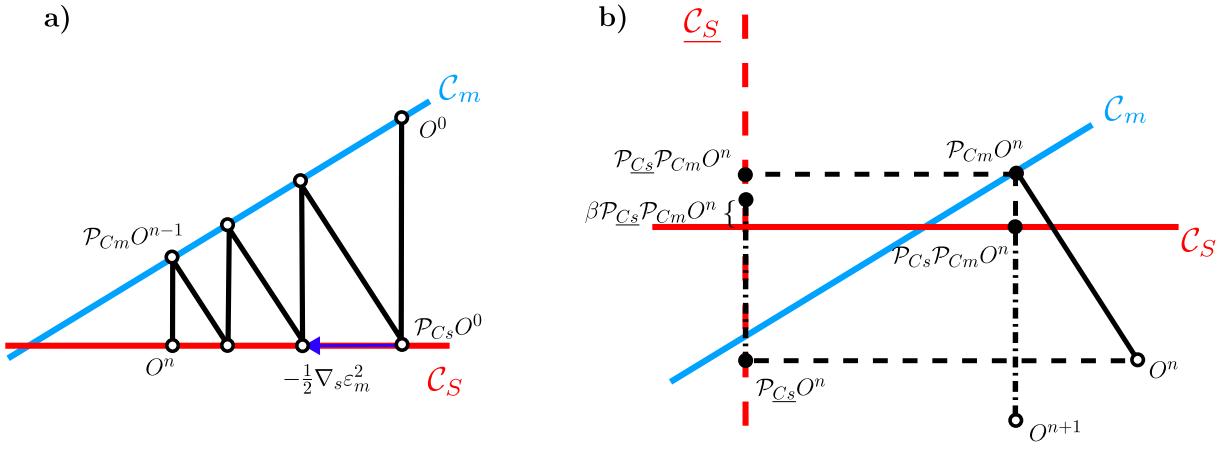
844 which makes explicit the equivalence between ER and steepest descent projected onto the  
 845 support constraint set, with a fixed step size of  $1/2$ .

### 846 3.3.2 Hybrid Input-Output (HIO)

847 The Hybrid-Input Output (HIO) algorithm introduces a nonlinear feedback that is essential to  
 848 escape local minima. Specifically, we can express the object at the  $k$ -th iteration as:

$$O^{k+1}(\mathbf{r}) = \begin{cases} \mathcal{P}_{Cm}(O^k(\mathbf{r})), & \mathbf{r} \in \mathcal{S} \\ (\mathcal{I} - \beta \mathcal{P}_{Cm})(O^k(\mathbf{r})), & \mathbf{r} \notin \mathcal{S} \end{cases} \quad (3.12)$$

849 where  $\beta$  is a positive hyperparameter with value typically around 0.9. Inside the support,  
 850 the estimate is replaced by its Fourier-modulus projection. Outside the support, instead of  
 851 being set to zero (as in ER), the estimate is updated with a feedback term proportional to the  
 852 modulus projection. This subtraction, which can in principle yield negative values as well,  
 853 prevents stagnation and allows the algorithm to explore solutions that are consistent with  
 854 both the support and modulus constraints. However, it is worth mentioning that due to the  
 855 nonlinear feedback outside the support, HIO does not converge to local minima but rather  
 856 towards *saddle* points of the error functional. For this reason HIO is used in combination with  
 857 ER: HIO drives the reconstruction away from traps by oscillating near saddle points, and ER  
 858 then provides stable convergence once the estimate is close to a true solution.



**Figure 3.2:** Geometrical representation of (a) the ER algorithm. Red and blue lines represent the support and Fourier modulus constraint sets respectively. The alternated projections on the two constraint sets drive eventually the estimate to the intersection. This is however only valid in a local subset of the solution space, where the - generally non-convex - modulus constraint set is convex and the estimate  $O^0$  is near the solution (b) geometrical representation of the HIO algorithm. Inspired by [84]

### 859 3.3.3 Relaxed Averaged Alternating Reflections (RAAR)

860 Another AP algorithm typically used in CDI is the Relaxed Averaged Alternating Reflections  
 861 (RAAR) developed by Luke in 2004 [3]. According to this algorithm the object at the  $k$ -th  
 862 iteration is:

$$O^{k+1}(\mathbf{r}) = \beta \frac{1}{2} (\mathcal{R}_{Cs} \mathcal{R}_{Cm} + \mathcal{I})(O^k(\mathbf{r})) + (1 - \beta) \mathcal{P}_{Cm}(O^k(\mathbf{r})) \quad (3.13)$$

863 where  $\beta \in [0, 1]$ . Eq.3.13 can be split into two terms. The first term, known as Average  
 864 Alternating Reflection (ARR) [77] acts like an average of the current estimate and its reflection  
 865 across both constraints sets. The alternated reflections search for the intersection of the two  
 866 sets, which is a fixed point of the problem, while exploring more broadly the solution space as  
 867 they do not project directly on the constraint sets. The second term is a relaxation term that  
 868 projects the current estimate on the modulus set, like in the HIO and ER.

869 In short, one could see the RAAR as a controlled (through  $\beta$ ) balance of exploration and  
 870 stability.

### 871 3.3.4 Support update

872 At this point of the discussion we should ask ourselves how do we know the *support* function  
 873  $\mathcal{S}$  required in all PR algorithms? We have said that in the BCDI case we typically do not  
 874 know the shape of the particle a priori, and should therefore come as product of the PR. It is  
 875 common practice in iterative phase retrieval to estimate the initial support from the object's  
 876 autocorrelation,

$$A(\mathbf{r}) = \mathcal{F}^{-1}\{I(\mathbf{q})\} \quad (3.14)$$

877     However, the autocorrelation extends over a region roughly twice the linear size of the true  
 878     object (see Fig. 3.4).

879     An important step forward was introduced by Marchesini in 2003 [94]: the *shrinkwrap*  
 880     algorithm, which adaptively refines the support during reconstruction. After a given number  
 881     of phase retrieval iterations, the modulus of the current object estimate is convolved with  
 882     a Gaussian kernel and subsequently thresholded (typically at  $\sim 20\%$  of the maximum value),  
 883     yielding a binary mask that defines the updated support.

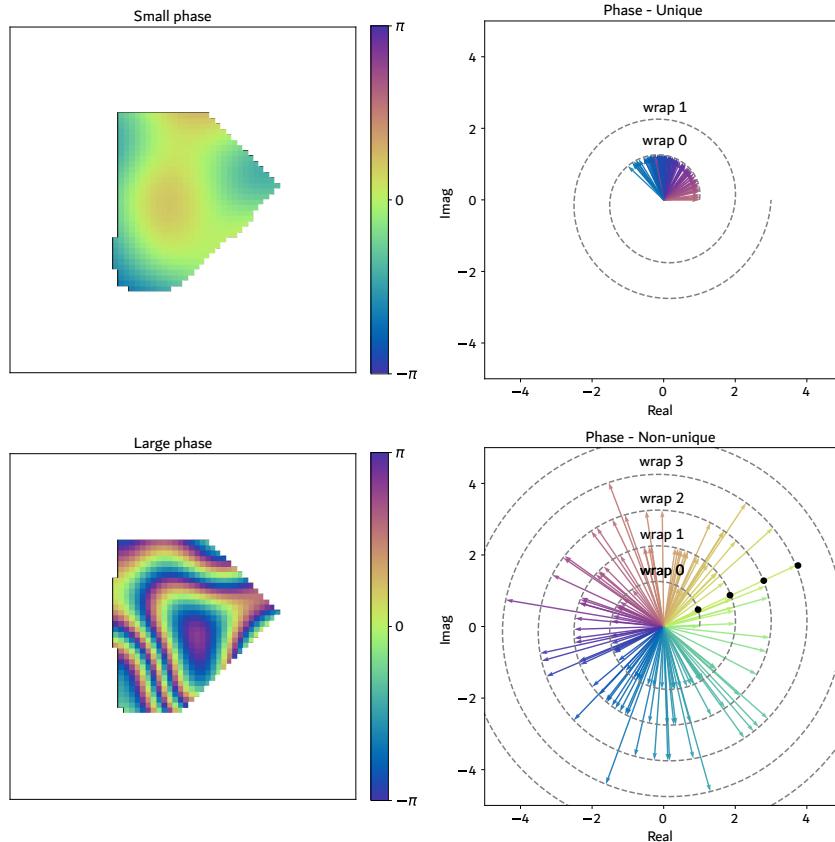
884     This adaptive support refinement was shown to significantly improve convergence and  
 885     reconstruction quality in many experimental cases (depending on the signal-to-noise ratio and  
 886     choice of threshold), and it is now a standard component of BCDI phase retrieval pipelines.  
 887

888     Later in the text we will refer to “standard iterative algorithms” or “conventional PR” as the  
 889     combination of AP algorithms and support updates like Shrinkwrap.

## 890     3.4 BCDI Phase Retrieval in presence of high-strain

891     At this point it is interesting to understand how and why the presence of high-strain complicates  
 892     the PR. and therefore postulate on the possible strategies that can be adopted to regularize the  
 893     problem and facilitate convergence.

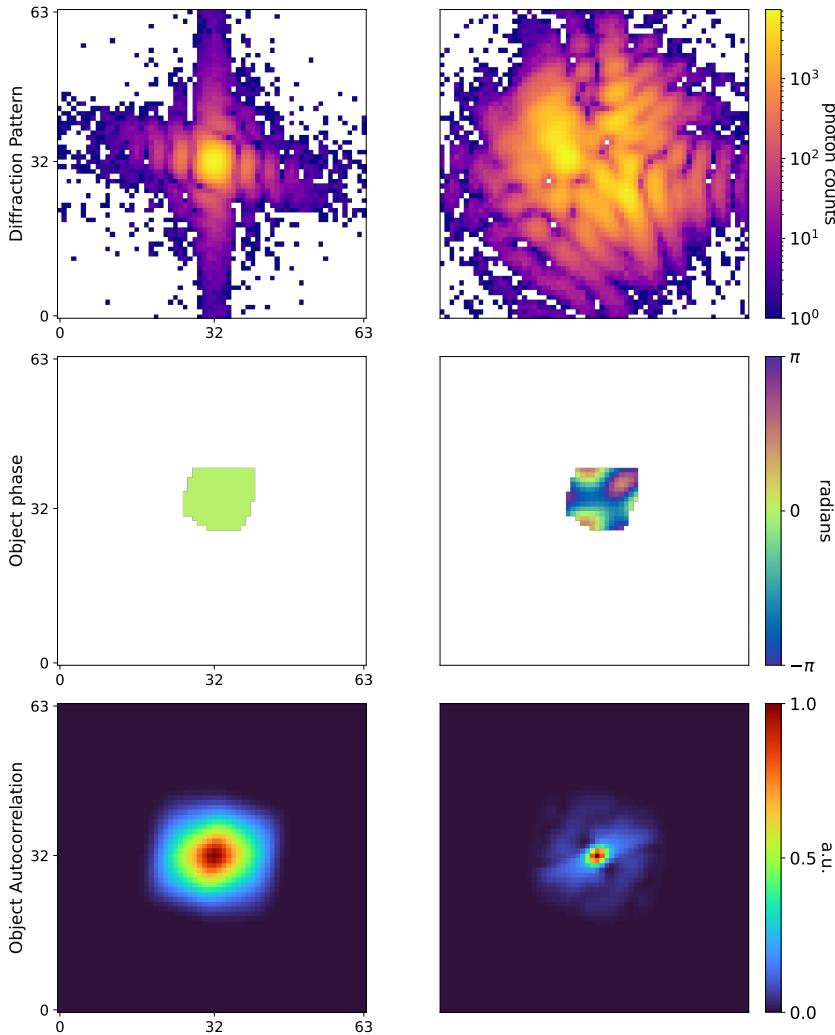
894     From a pictorial point of view it could help to see the problem as follows. If we consider  
 895     each voxel constituting the object as a complex vector made of modulus (normalized between 0  
 896     and 1) and phase, we can plot the full object as the ensemble of these complex vectors lying on  
 897     the Argand plane, where modulus and phase are also seen on the Cartesian real and imaginary  
 898     axes (see Fig.3.3). For small phases we would see the vectors spreading over a limited range of  
 899     the unitary circle. On the contrary large phases will cause the vectors to spread even more to  
 900     the full circle and eventually extend to multiple phase wraps. At this point, if the phases are  
 901     bounded within the  $2\pi$  range, each vector’s angle is uniquely determined by its coordinates  
 902     on real and imaginary axis. On the contrary, if the range exceeds  $2\pi$ , some of these vectors  
 903     with phase  $\phi$  may overlap with some other vectors whose phase is  $2\pi + \phi$ , thus making each  
 904     point on the unitary circle no longer uniquely assigned to a single phase. This degeneracy is  
 905     responsible for the presence of additional local minima that can hinder the PR.



**Figure 3.3: Pictorial visualization of phases on the Argand plane.** The figure provides an intuitive illustration of the origin of additional local minima that arise for large phase ranges. Each voxel is represented by an arrow vector directed according to its phase, unwrapped along a spiral line. This representation highlights that objects with small phase variations (within  $2\pi$ ) correspond to solutions of the (PR) problem that lie on the 0-th wrap, where each phase value is uniquely determined. In contrast, objects with large phase ranges ( $> 2\pi$ ) extend the solution space to higher-order wraps, creating degeneracies with lower-order wraps ( $\phi = \phi + 2\pi \times \text{wrap}$ ). This degeneracy in the solution space increases the ambiguity of the phase problem.

906 In the literature it is known that particles with phase range beyond  $2\pi$  are more challenging  
 907 to reconstruct and these occurrences are often referred to as “strong-phase” or similarly “high-  
 908 strain” cases [95–97]. Although strictly related, these two concepts are different, since the strain  
 909 is ultimately given by the *gradient* of the phase. From the PR perspective, it is the strong-phase  
 910 complicating the PR. Practically, given the finite number of voxels over which the phase can  
 911 extend, the strong-phase case coincides to the high-strain case. Later in the text the two terms  
 912 will be used interchangeably.

913 Another reason that can practically hamper the PR is difficult initialization of the algorithms.  
 914 In fact, it is interesting to notice how the autocorrelation changes for highly strained particles  
 915 and how this can affect the PR algorithms initialization and consequent convergence. Fig. 3.4  
 916 depicts two simulated diffraction patterns from a particle with the same support but different  
 917 applied strain fields. While in the first case (Fig. 3.4a) the low-strain does not distort the  
 918 diffraction pattern and therefore the autocorrelation function, beside the sharp central peak, is  
 919 uniform and its support well-defined. On the contrary, in Fig. 3.4b, the high-strain distorts the  
 920 diffraction pattern and the calculated autocorrelation presents a poorly homogeneous support.



**Figure 3.4: Object autocorrelation** **a)** In case of low strain the object autocorrelation function has uniform amplitude outside the central peak, with well-defined support. One can notice that it clearly extends over a domain that is double the size of the object. In this case the sharp autocorrelation support edges facilitate the shrink-wrap algorithm to extract the support. To the same particle a large phase was applied **b)** and the calculated autocorrelation shows a sharper central peak and a much less intense and inhomogeneous surrounding region. It follows that in this case a lower shrink-wrap threshold needs to be adopted in the initialization. However, this can include holes in the support which might be an effect of the large phase and not physical.

921 The result of the local minima and the hard estimate of the object support make such  
 922 that conventional PR algorithms have to be launched multiple times (in the order of tens or  
 923 hundreds normally), with different initializations. In this way the probability of landing near  
 924 the solution is increased, and the convergence is more likely. This is however computationally  
 925 expensive and time-consuming.

926 In the years, and during this PhD as well, different approaches aiming at improving the PR  
 927 of highly-strained crystals have been developed, as it still represents a major limitation of  
 928 the BCDI technique. Solutions that employ the use of structured 2D modulators placed before  
 929 or after the sample have been studied by respectively Calvo *et al.* [98] and Zhao *et al.* [99].  
 930 These methods allow to introduce an additional constraint, given by the modulator, during

the PR. From the algorithmic perspective one often seeks regularization of the problem with some prior knowledge. Typically, the assumption of homogeneous electron density inside the support is leveraged with uniformity constraints during PR [58, 100, 101] or object's modulus minimization in  $\mathcal{L}_1$  norm [96], or guiding the algorithm by promoting objects with uniform and sharp modulus [102]. Recent studies have also explored the combination of genetic algorithms with standard PR [103] for improved exploration of the solution space. In other cases the additional knowledge is provided by multiple and different measurements of the same sample. For instance multiple Bragg reflections of the same particle can be phased simultaneously [104] to constrain the solution to a unique reconstruction fulfilling each measurement. Alternatively, if the sample undergoes phase transitions inducing high-strain, the support retrieved at low-strain level can be fixed across the transition during the PR of high-strain states [105].

The regularization of the strongly ill-posed PR for highly strained crystals with prior knowledge is the key to successful reconstructions. Another type of strategy concerns the use of *learned* priors, i.e. that emerge from the properties of the learned statistical distributions of the particular class of inverse problems. It is in this framework that Convolutional Neural Networks models presented in Chapter 6 were conceived.

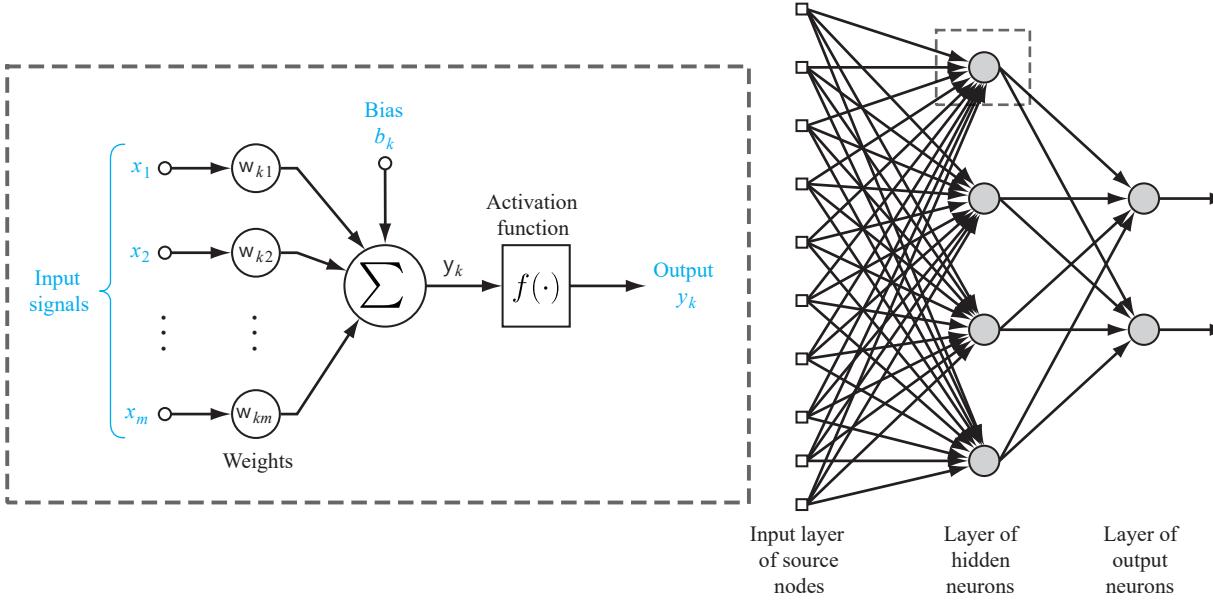
948 **KEYNOTES ON CONVOLUTIONAL NEURAL NET-  
949 WORKS**

950 In this chapter a short overview on the basic concepts of Convolutional Neural Networks  
951 (CNNs) will be given. The scope is to give the necessary background for the understanding of  
952 the structure and the motivations behind the Deep Learning (DL) models employed for the  
953 analysis of BCDI data, presented in the next chapters.

954 For more comprehensive and exhaustive dissertations about Machine Learning (ML) and  
955 Artificial Neural Networks (ANNs) the book of Goodfellow [106] and the more recent from  
956 Prince [107] are suggested to the reader.

957 **4.1 Artificial Neural Networks (ANNs)**

958 ANNs are a type of machine learning algorithm inspired by the biological neuron structure.  
959 ANNs are generally composed of interconnected nodes where the signal is processed through  
960 operations with tunable parameters named weights and biases for multiplications and addition  
961 respectively. An important feature of each node is the *activation function*, that introduces a non-  
962 linear operation and returns the node's output [108]. Several kinds of these activation functions  
963 exist and their use depends on the properties of each (bounds, derivatives, positivity, etc.) [109].  
964 In the following chapters the modified rectifying linear unit, known as LeakyReLU [110], and  
965 the sigmoid, also known as logistic, function will be used. Neurons are generally organized  
966 into *layers* and are connected to neurons of other layers. In *feed-forward* neural networks the  
967 information flows from the input layer to the output layer with forward connections only. In  
968 other neural networks, like *recurrent* ones, the connections are also designed backwards.



**Figure 4.1: Schematic of an artificial neuron and a neural network.** On the left, a series of input signals  $x_i$  are multiplied by the tunable weight  $w_k$  and summed together with the tunable bias  $b_k$  relative to the  $k$ -th neuron. The output is then passed through the activation function  $f$  which produce an output  $y_k$  which is a non-linear combination of the inputs. On the right, the feed-forward network composed of two *hidden* layers of artificial neurons processing the ten units long vector  $x$  and returning a binary output. Networks of this type are said to be *fully connected* as each input component, or node's output is processed by all neurons. Adapted from [111]

#### 969 4.1.1 Neural networks as universal approximators

970 The use of non-linear functions is found to be fundamental for the powerful analytical and  
 971 statistical properties of ANNs, which have been progressively established over the years. It was  
 972 shown [112–114] that ANNs with appropriate activation functions and sufficient parameters  
 973 can approximate *any* continuous function on a compact domain to arbitrary accuracy (*universal*  
 974 *approximation theorems*). Later the proof has been extended to deep convolutional neural  
 975 networks as well [115].

976 Formally, we can consider  $\mathcal{X}$  as the set of all events belonging to the same statistical  
 977 distribution and  $\mathcal{Y}$  similarly. We also assume that an unknown mapping  $\mathcal{M}$  between an input  
 978  $x \in \mathcal{X}$  and the output  $y = \mathcal{M}(x)$  exists. At this point, the goal of the ANNs is to be the  
 979 closest approximation of  $\mathcal{M}$ . Typically, this implies seeking the combination of parameters  
 980  $\theta_k = (w_k, b_k)$  of the neural network  $\mathcal{N}_\theta$  such that  $\mathcal{N}_\theta \approx \mathcal{M}$

981 **How is this mapping found?** The core idea is known as Empirical Risk Minimization  
 982 and states that when the  $\theta_k$  are adjusted to fit a sufficiently large dataset consisting of samples  
 983 drawn from an underlying distribution, the function being approximated reflects the statistical  
 984 relationships encoded in that distribution [116]. By the Law of Large Numbers, the empirical  
 985 distribution observed in the *training set* converges to the true distribution as the sample size  
 986 grows, and thus the empirical risk minimized during training approaches the true risk. This  
 987 statistical foundation explains why ANNs are able to generalize to new, unseen data drawn

988 from the same population.

989 It is therefore sufficient to have a large but finite number of samples representative of both  
 990 the sets  $\mathcal{X}$  and  $\mathcal{Y}$  to approximate  $\mathcal{M}$ . This powerful statistical property underlies the concept  
 991 of *supervised* training of ANNs.

992 More formally, one can consider having a limited set of  $N$  examples  $(x_1, y_1 = \mathcal{M}(x_1)), \dots, (x_N, y_N =$   
 993  $\mathcal{M}(x_N))$  which we call *training set*  $\mathcal{T}$ . Here, each  $x_i$  is an input instance and  $y_i$  the correspond-  
 994 ing ground truth transformation operated by  $\mathcal{M}$ . We can now introduce a non-negative scalar  
 995 real-valued *loss function*  $L(\hat{y}, y)$  which measures the difference between the ground truth  $y$   
 996 and the output of the neural network  $\hat{y} = \mathcal{N}_\theta(x)$  for each element of the training set.

997 It follows that the best approximation of the mapping  $\mathcal{N}_\theta^{\mathcal{T}} \approx \mathcal{M}^{\mathcal{T}}$  is obtained when the  
 998 score of the loss function averaged across the number of training samples is lowest. The  
 999 problem of approximating the mapping between the input and ground truth in the training set  
 1000 is then formulated as a minimization problem of the type:

$$\hat{\theta}_{\mathcal{T}} = \arg \min_{\theta} \left\{ \frac{1}{N} \sum_{i=1}^N L(y_i, \mathcal{N}_\theta^{\mathcal{T}}(x_i)) \right\} \quad (4.1)$$

1001 For a sufficiently large  $N$  the  $\hat{\theta}_{\mathcal{T}} \approx \hat{\theta}$  valid for the whole statistical distribution. The size of  
 1002  $N$  needed to approach the true mapping depends obviously on the complexity of the mapping  
 1003 and on the complexity of the minimization task. While the first is inherent to the problem, the  
 1004 second can be engineered with the choice of a metric for which the loss function minimization  
 1005 is easier. This will be clear in Chapter 6 where different loss functions are tested. In the  
 1006 same way, the network design can impact the facility with which the ANN converges to the  
 1007 approximation of the desired mapping. In this regard, the use of multiple intermediate layers  
 1008 of neurons (*hidden layers*) was shown to strongly improve the ability of the NN to fit more  
 1009 complex functions (see [107] - Chapter 4.5). For this reason these network called *deep* have  
 1010 taken over *shallow* ones. Another example, briefly discussed in the next section, is given by  
 1011 Convolutional Neural Networks (CNNs), a type of ANN suited for natural image processing.

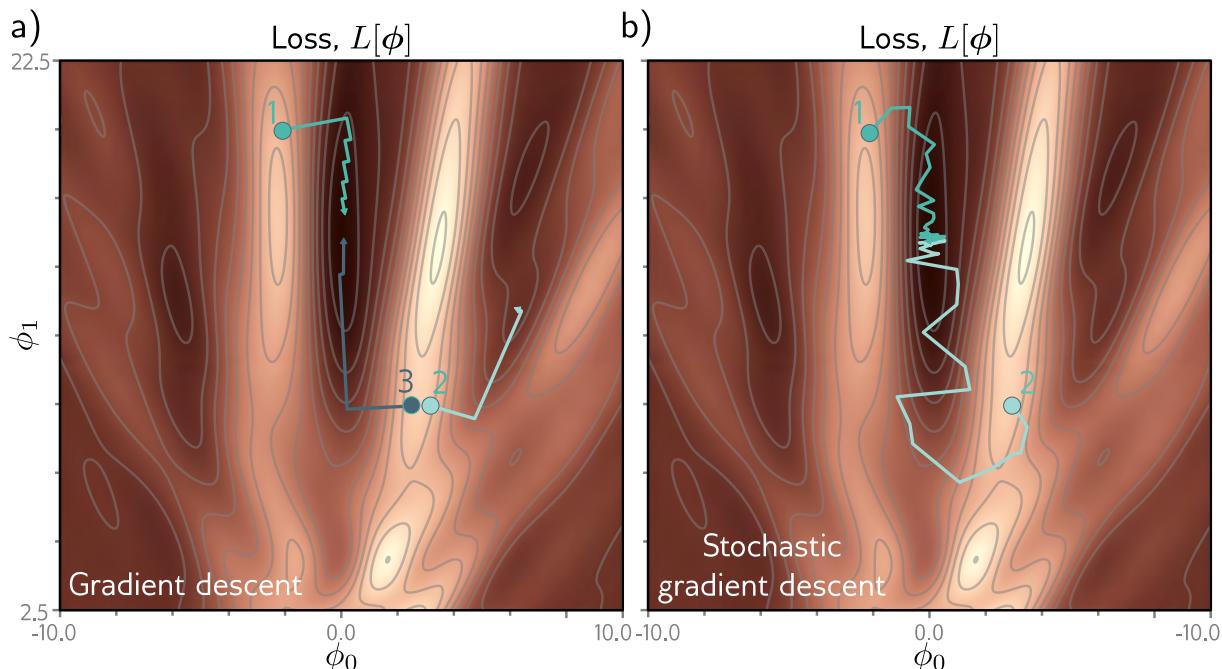
1012 Moreover, in cases in which one does not have a training set composed of input - ground  
 1013 truth pairs, but instead incorporates prior knowledge into the model architecture or the loss  
 1014 function, the training is said to be *unsupervised*. This approach has not been explored in the  
 1015 context of this PhD.

### 1016 4.1.2 Gradient Descent

1017 At this point another question may arise: **How do we solve the minimization problem?** The  
 1018 most straightforward manner to solve a minimization problem is with gradient descent. This  
 1019 implies an iterative process in which at each iteration the derivatives of the loss function with  
 1020 respect to each parameter  $\theta$  are calculated, and each parameter is updated correspondingly.

$$\theta^{t+1} = \theta^t - \eta \nabla_{\theta} \left[ \frac{1}{N} \sum_{i=1}^N L(y_i, \mathcal{N}_{\theta}(x_i)) \right] \quad (4.2)$$

1021 Where  $\eta$  is the *learning rate* that is given as an external parameter (*hyperparameter*) to the  
 1022 model. Eq.4.2 implements the steepest gradient descent. However, in most machine learning  
 1023 optimizers a variant of this algorithm is computed. Namely, the gradients are calculated for  
 1024 each sub-set, often called *mini-batch* of the whole training set. During the training, within each  
 1025 *epoch* as many mini-batches as are needed to make the full dataset, are minimized in series.  
 1026 This approach, called *stochastic* gradient descent (SGD), is less expensive in terms of memory  
 1027 and offers well established convergence properties that outperform classical steepest descent  
 1028 methods [117]. In fact, the “noise” affecting the updates induced by the minimization of a small  
 1029 sub-set can be beneficial for escaping saddle points which may trap the search. Over the years,  
 1030 different variants of SGD have been proposed. The “momentum” calculation [118], that keeps  
 1031 track of the magnitude and direction of the updates and determines the next update as a linear  
 1032 combination of the gradient and the previous update, was first applied to SGD [119]. Later,  
 1033 adaptive approaches have aimed at tuning the learning rate differently for each parameter  
 1034 (AdaGrad [120], ADAM [121]). Later in the text, the DL models employed have adopted the  
 1035 ADAM optimizer. It is worth mentioning that, though the most widely used, gradient descent  
 1036 approaches are not the only strategies that have been explored to minimize the loss function.  
 1037 Evolutionary algorithms inspired by natural selection mechanisms and tensor optimization  
 1038 techniques developed in the quantum-many body field have been employed as well [122, 123].



**Figure 4.2: Gradient descent and Stochastic gradient descent.** **a)** Depicts the trajectory of the error score in a 2D loss function in case of gradient descent with line search from three different initializations. The algorithm converges to the global minimum when it starts from points 1 and 3 while it fails when it starts from point 2, located outside the valley of the global minimum. **b)** The same experiment solved with stochastic gradient descent achieves good convergence also when initialized from point 2. Indeed, the “noise” introduced by the mini-batches drives the updates outside the wrong valley and allow the convergence to the global minimum. Adapted from [107]

### 1039 4.1.3 Backpropagation and automatic differentiation

1040 At this stage, a crucial question arises: **how can SGD be efficiently implemented in practice?**  
 1041 ANNs often involve millions or even billions of parameters  $\theta$ , thus computing exact derivatives  
 1042 with respect to such a large number of variables, for every batch and across multiple epochs,  
 1043 requires highly optimized algorithms and hardware acceleration. Indeed, the practical feasibility  
 1044 of training neural networks was significantly advanced by the seminal work of Rumelhart,  
 1045 Hinton, and Williams in 1986 [119], which introduced the efficient *back-propagation* technique  
 1046 that made large-scale training computationally tractable.

1047 Back-propagation bases its principles on the fundamental *chain rule* proposed by Leibniz  
 1048 in 1676 to calculate derivatives of function compositions. In fact, a ANNs can be seen as a  
 1049 composition of functions (the layers) in which at each stage the layer  $i$ -th processes the output  
 1050 of the layer  $(i - 1)$ -th. Formally, one can write:

$$\mathcal{N}_\theta = f_\theta^{(L)} \circ f_\theta^{(L-1)} \circ \dots \circ f_\theta^0 \quad (4.3)$$

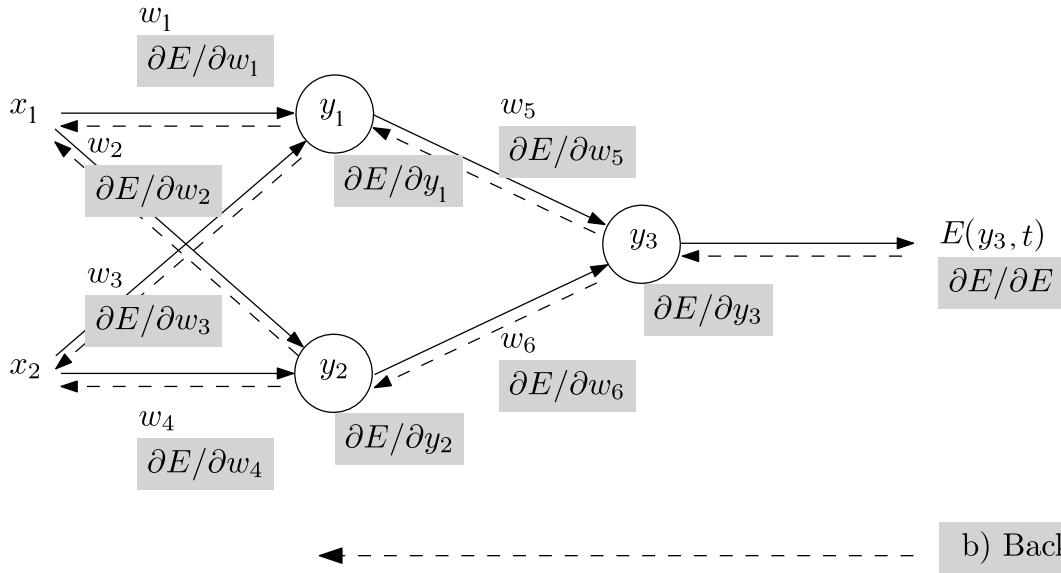
1051 Where  $f_\theta^{(L)}$  is the function representing the  $L$ -th layer of neurons of parameters  $\theta$ . The  
 1052 gradient of the loss function with respect to the parameters is therefore expressed, according  
 1053 to the chain rule as:

$$\nabla_{\theta^{(l)}} L = \frac{\partial(L)}{\partial f^{(L)}} \cdot \frac{\partial f^{(L)}}{\partial f^{(L-1)}} \cdot \dots \cdot \frac{\partial f^{(l)}}{\partial \theta^{(l)}} \quad (4.4)$$

1054 This method decomposes the calculation of the full gradient into a sequence of smaller,  
 1055 local derivatives, each associated with the intermediate states of the network. During the  
 1056 forward pass, the activations and local Jacobians are computed and stored in memory, a process  
 1057 sometimes referred to as *forward accumulation*. These stored quantities are then systematically  
 1058 reused during the backward pass to propagate gradients from the output layer back through  
 1059 the network parameters. This reuse of intermediate computations is what makes the algorithm  
 1060 highly efficient, and it is from this backward flow of gradients that the name *back-propagation*  
 1061 originates.

1062 In modern machine learning frameworks such as PyTorch [124] and TensorFlow [125],  
 1063 back-propagation is implemented through the construction of a *computational graph*. Each  
 1064 node in the graph represents an operation, while edges encode the dependencies between  
 1065 operations. Once the forward pass has been executed, this graph allows the backward pass  
 1066 to be automatically derived and executed, efficiently parallelized across GPUs or TPUs. In  
 1067 fact, back-propagation is a specific instance of a more general family of techniques known as  
 1068 *automatic differentiation* (or *algorithmic differentiation*) (AD) [126, 127], and more precisely  
 1069 corresponds to *reverse-mode automatic differentiation*. This connection highlights that back-  
 1070 propagation is not unique to neural networks, but rather belongs to the rigorous and general  
 1071 framework for computing exact derivatives of functions defined by computer programs.

a) Forward pass



b) Backward pass

**Figure 4.3: Schematic of Back-propagation on computational graph.** a) The training inputs  $x_i$  are processed in the forward pass by the network that produces intermediate states  $y_{1,2}$  using weights  $w_i$ , and the final output  $y_3$ . b) The error  $E$  computed between the output and the estimate  $t$  is propagated backward and the gradients with respect to the weights  $\nabla_{w_i} E = (\frac{\partial E}{\partial w_1}, \dots, \frac{\partial E}{\partial w_6})$  are obtained from the chain rule of derivatives. Adapted from [127]

## 1072 4.2 Convolutional Neural Networks (CNNs)

1073 It was anticipated earlier in the Chapter that the network design can strongly impact the  
 1074 learning of the mapping  $\mathcal{M}$ . In the specific case in which  $\mathcal{X}, \mathcal{Y}$  represent sets of natural images,  
 1075 neural networks with layers employing convolution operations have shown to outperform  
 1076 fully connected ones [128–130]. The reasons behind the success of convolutions are intuitively  
 1077 explained as follows:

- 1078 • Images are inherently high-dimensional, as they are defined on 2D or 3D grids. As a result,  
 1079 the number of parameters required to connect each pixel in a fully connected neural  
 1080 network grows rapidly and soon becomes intractable. Convolutional layers address this  
 1081 issue by employing a *kernel*: a small trainable filter that is convolved with the input  
 1082 image or intermediate feature maps. This mechanism enables weight reuse across spatial  
 1083 locations, drastically reducing the number of trainable parameters, a property commonly  
 1084 referred to as *parameter sharing*. Furthermore, the localized receptive field of the kernel  
 1085 induces a form of structured sparsity, effectively regularizing the model by constraining  
 1086 the mapping to be approximated using compact, spatially coherent information.
- 1087 • In images, spatially adjacent pixels are typically correlated, while uncorrelated variations  
 1088 are often attributable to noise. Fully connected layers process each pixel independently, re-  
 1089 quiring the network to learn spatial dependencies solely through training. Convolutional  
 1090 layers, in contrast, explicitly exploit local spatial correlations by aggregating information  
 1091 from neighboring pixels, thereby embedding structural priors into the architecture. (see  
 1092 Fig. 4.4)
- 1093 • The image of a tree and another image of the same tree, translated require a different  
 1094 tuning of the weights in a fully-connected layer, although the interpretation of the  
 1095 image has not physically changed. Learning a different parameter configuration for  
 1096 each different translation of the same object is highly inefficient. On the contrary, an  
 1097 additional benefit of the parameter sharing in convolutional layers is the equivariance  
 1098 to translation that inherently optimize the learning. This property allows the output to  
 1099 respond to the input change in the same way. Note that this equivariance is not valid for  
 1100 other geometrical transformations like rotation and magnification.

1101 We have seen in Chapter 2 that BCDI data is in the form of 3D images in which peculiar  
 1102 spatial structures made of peaks and fringes are clearly visible, hence the natural choice of  
 1103 convolutional neural networks. Moreover, the tasks addressed in this PhD thesis, namely the  
 1104 gap-inpainting and the phase retrieval, are classified as inverse problems where the goal is to  
 1105 reconstruct missing or unmeasured information from incomplete observations. As highlighted  
 1106 in recent surveys and related works [131, 132], CNNs are particularly well suited for these  
 1107 problems because they can learn powerful image priors from data, act as regularizers even  
 1108 without extensive training (as shown in the Deep Image Prior framework [133]), and can be  
 1109 combined with physics-based models to enforce data consistency.

1110 Let us present now the building blocks of a typical convolutional neural network.

**4.2.1 Convolutional layer**

It needs to be clarified here that to improve computational efficiency, machine learning libraries typically implement the *cross-correlation* operation rather than a convolution. The difference is that the convolution flips the kernel before computing the sum, whereas cross-correlation does not. This simplifies the implementation with no expressive power loss, since the weights are learned during training but in turn breaks the commutative property of the convolution. Given an input vector  $\mathbf{x}$  with  $N$  entries the output of a convolutional layer  $\mathbf{y}$  with  $M$  entries can therefore be written as a matrix - vector multiplication of the type:

$$\mathbf{y} = \mathcal{W}\mathbf{x} \quad (4.5)$$

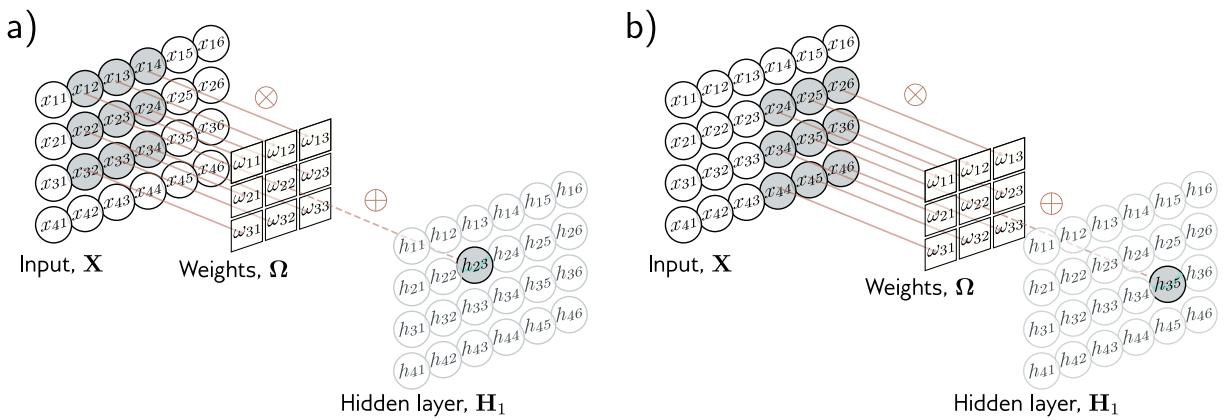
Where  $\mathcal{W} \in \mathbb{R}^{M \times N}$  is a Toeplitz matrix, where each descending diagonal contains a kernel entry, arranged such that the matrix performs the cross-correlation as a linear operation.

The initialization of a convolutional layer in typical machine learning libraries involve the definition of:

- An integer number of kernels. This number corresponds to number of filters through which the input is processed in a convolution operation. Each filter is normally associated to a different feature of the input image that is extracted by the corresponding kernel. The output of the convolutional layer will have an extra dimension (called *channel*) with size corresponding to the number of filters.
- The integer-valued size of the kernel. In 2D convolutions the kernel is a matrix and similarly extended to a 3D tensor for three-dimensional inputs. It controls the scale of the features that the convolutional layer can “see”. Small kernels highlight small features and vice-versa.
- The padding parameter which handles the convolution at the borders of the image, where the kernel would otherwise extend beyond the available data. It determines whether the output feature map maintains the same spatial dimensions as the input (“same” padding) or is reduced in size (“valid” padding).
- The stride parameter: This integer-valued parameter (matrix in 2D and tensor in 3D) indicates the lateral shift that is applied to the kernel across the image. If greater than 1 some pixels/voxels are skipped resulting in a sort of sub-sampling allowing to filter out some low-level features.
- The dilation rate, which is another integer-valued parameter (matrix in 2D and tensor in 3D) that allows to insert holes between consecutive elements of the kernel. In this way the actual size of the kernel is increased by the dilation value, but the overall number of weights is unvaried. This enables to cover a larger area of the input without increasing the number of trainable parameters.

Additional features of a typical convolutional layer include the addition of trainable biases and the initialization and regularization of the kernels and biases. Lastly, a non-linear activation is usually applied to the output of the convolutional layer.

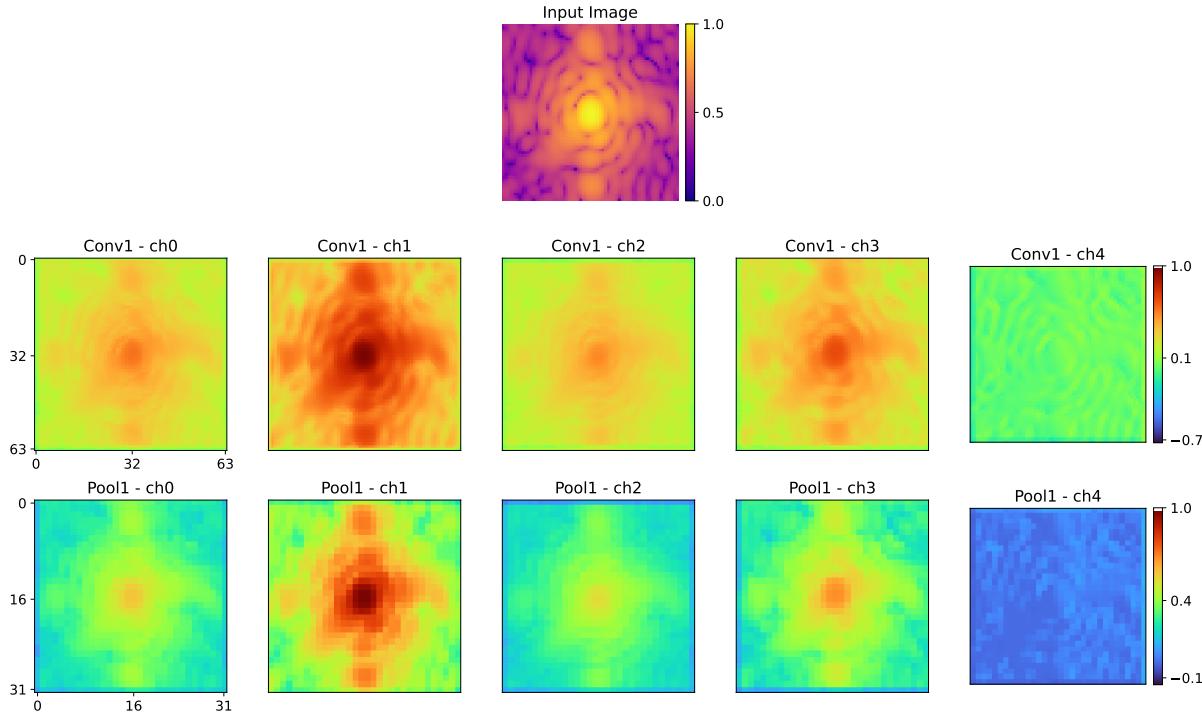
1148 We mention here the existence of so-called *transposed convolutions* which are sometimes  
 1149 used in CNNs. Here the conceptual difference is that Eq.4.5 is computed with the transposed  
 1150 Toeplitz matrix  $\mathcal{W}^T$ . Transposed convolutions naturally produce an output that is larger than  
 1151 the input, because they reverse the spatial reduction of a standard convolution. For this reason,  
 1152 they are often used to replace explicit *up-sampling* layers, which increase the spatial size  
 1153 through fixed interpolation (e.g. nearest- neighbor or bilinear). Unlike standard interpolation,  
 1154 transposed convolutions perform a *learned* up-sampling, where the network optimizes the  
 1155 kernel weights to reconstruct high-resolution features in a data-driven way.



**Figure 4.4: Schematic of 2D convolutional layer:** **a)** The convolution is operated over a region of the image with same size as the kernel ( $3 \times 3$ ). The result of the multiplication of each pixel value ( $x_{12}, \dots, x_{34}$ ) times the weight ( $w_{11}, \dots, w_{33}$ ) is then summed into a scalar  $h_{23}$ . **b)** The same operation with the same kernel is performed to a shifted window of the input image. Adapted from [107]

## 1156 4.2.2 Pooling Layer

1157 After the convolutional layer and the activation function, it is often the turn of a *pooling*  
 1158 layer. For the 2D case, the output of the layer at specific location  $i, j$  is the result of a filtering  
 1159 operation on the neighboring pixels of the input. For instance, the “Max Pooling” layer replaces  
 1160 a rectangular region of the input with its maximum value, thus retaining only the strongest  
 1161 activation within that area. In contrast, the “Average Pooling” layer assigns to the output pixel  
 1162 the mean value of all pixels in the corresponding region. The benefits deriving from the pooling  
 1163 layer are mostly twofold. First, the size of the output is reduced by a factor proportional to the  
 1164 area of the pooling window (or the volume in the 3D case), thus lowering the computational  
 1165 and memory cost of subsequent operations. Second, by condensing information into a smaller  
 1166 region, pooling encourages the network to learn representations that are more robust to small  
 1167 translations. For instance, with Max Pooling, a shift of the most activated pixel within the  
 1168 pooling window does not affect the output. This behavior can be interpreted as an implicit  
 1169 prior that biases the optimization towards translation-invariant approximations.



**Figure 4.5: Example of Convolutional and Max pooling layers.** A simulated 2D BCDI pattern is input image (first row) of the CNN model presented later in Chapter 6, after the training. The output of the first convolutional layer + LeakyReLU activation is displayed in the second row (first 5 channels) while the output of the subsequent Max Pooling layer is displayed in the third row. Notice the similarity between the two rows despite the halved size of Max Pooling output. This shows that it efficiently condenses the information into smaller sizes.

1170     The typical convolutional block is therefore composed of these three layers (convolutional,  
1171     activation, pooling). A sequential application of the convolutional blocks to a 2D or 3D input  
1172     image is often called *encoder* and can reduce the dimension of the feature map to as many 1D  
1173     vectors as the number of filters of the last convolutional layer. During the training, this reduced  
1174     representation of the input is more and more driven to capture the essential information in  
1175     a lower-dimensional space. At this point, for *classification* tasks the feature map is flattened  
1176     into a single 1D vector being the input of a fully connected layer. The output of this last layer  
1177     is then usually interpreted as the probability score for the input image to belong to a specific  
1178     class. Among the milestone CNN models for classification, it is worth highlighting LeNet-5,  
1179     introduced by LeCun in 1998 [134], and AlexNet, which revolutionized the field by leveraging  
1180     more efficient GPU-based training [135].

1181     In *image generation* tasks, the lower-dimensional space, also called *latent space* serves as  
1182     input to a sequence of *deconvolutional* blocks, where *transposed* convolutions combined with  
1183     activation functions progressively reconstruct the output image at the desired resolution. This  
1184     sequence of deconvolutional layers is referred to as the *decoder*, as it mirrors the encoder.  
1185     In some architectures, up-sampling layers followed by standard convolutions are employed  
1186     instead of transposed convolutions. Encoder-decoder architectures for image generation were  
1187     first introduced by Hinton in 2006 [136]. This class of models is employed in a variety of tasks  
1188     such as de-noising, image compression, inpainting, and segmentation. Although other types  
1189     of models are also used for image generation, including Generative Adversarial Networks

1190 (GANs) [137] and Vision Transformers (ViTs) [138], in this manuscript we restrict our focus to  
1191 encoder-decoder architectures.

### 1192 4.2.3 Skip connections

1193 Before concluding the chapter it is worth mentioning the concept of *residual connections*, or  
1194 more in general *skip connections* which has been introduced in deep convolutional neural  
1195 networks with the ResNet model [139]. In this type of architecture the output feature maps in  
1196 the encoder blocks is summed to the input of the decoder blocks with corresponding size. In  
1197 other cases, like the U-Net architecture [140] a concatenation of the two feature maps along  
1198 the channel dimension is performed instead of the sum. The goal of the skip connections is to  
1199 facilitate the training of deep generative networks by “reminding” the model about the features  
1200 extracted in the beginning of the forward process. In fact, the more difficult training of deep  
1201 networks as compared to shallow ones is attributable to the more unstable gradients [141].  
1202 Tiny changes in the input result in a completely different gradient. A study conducted by Li *et*  
1203 *al.* has shown the benefits of skip connections through a study of the loss function landscape,  
1204 which tends to have less saddle points and local minima [142].

1205

1206 This paragraph concludes the introductory background on neural networks. More broadly,  
1207 the chapter complements the presentation of the physics and conventional data analysis of the  
1208 BCDI technique from the first two chapters, by introducing the concepts, terminology, and  
1209 insights needed to approach the content of the following three chapters.



---

## 1211 DEEP LEARNING FOR DETECTOR GAPS INPAINT- 1212 ING

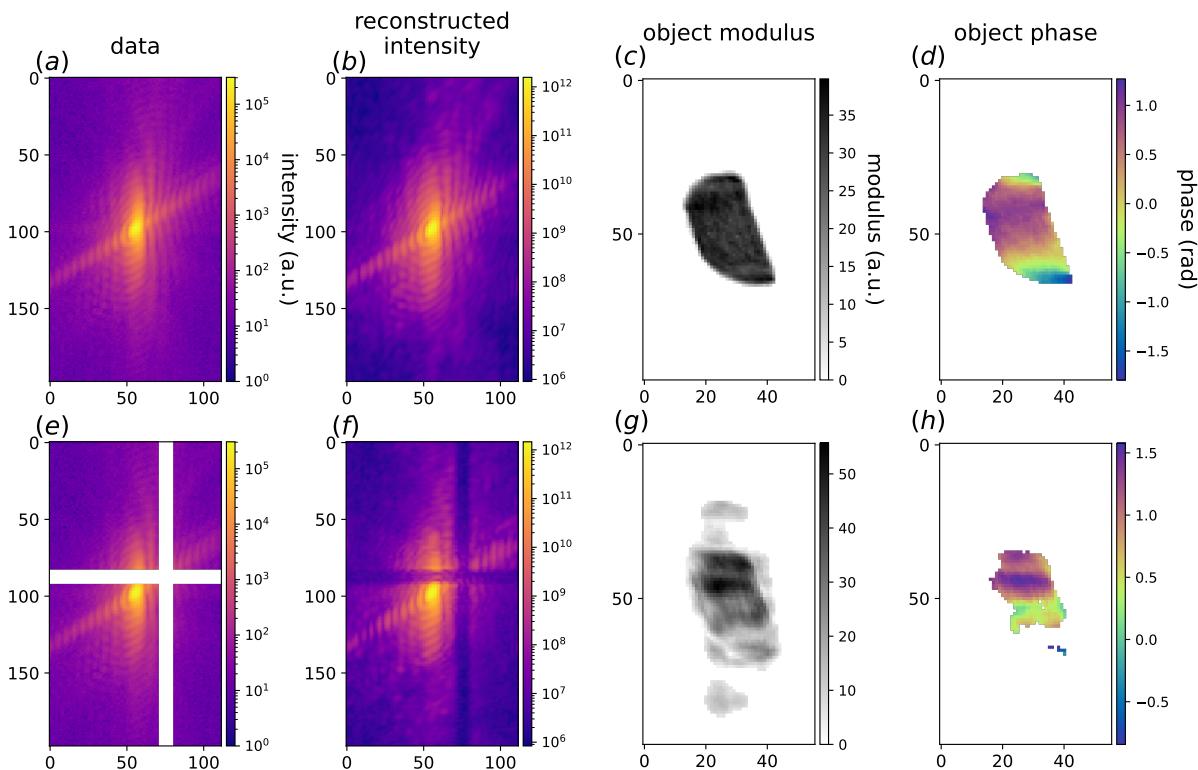
1213 In this chapter, the “detector - gaps problem” in Bragg Coherent Diffraction Imaging and  
 1214 the approach to solve it using Deep Learning (DL) are discussed. The main state-of-the-  
 1215 art methods are presented briefly and the topic of image inpainting with DL is introduced.  
 1216 The focus will then shift to the works that led eventually to the optimal “Patching-based”  
 1217 approach that can also be found in the published paper entitled “*Patching-based deep learning*  
 1218 *model for the Inpainting of Bragg Coherent Diffraction patterns affected by detectors’ gaps*”  
 1219 (<https://doi.org/10.1107/S1600576724004163>). The chapter is closed with some analyses of the  
 1220 performance of the DL models in a variety of simulated and experimental cases.

### 1221 5.1 The “Gap Problem”

1222 At time of writing, standard BCDI experiments employ pixelated photon counting detectors to  
 1223 record the diffraction patterns. These detectors can guarantee high spatial resolution, low-noise  
 1224 counting and fast read-out times. Two examples of these devices, currently used at the ID01  
 1225 beamline are the MAXIPIX and EIGER detectors [49, 143]. These detectors are often built by  
 1226 tiling together several sensing chips in order to cover a larger area, and are typically bonded to  
 1227 an Application-Specific Integrated Circuit (ASIC) using bump bonding. This implies always  
 1228 some physical connection which is not a pixel on one side of a chip. Additionally, intra-chip  
 1229 edges also constrain larger pixels which disrupt the oversampling condition, hence cannot  
 1230 always be used. These issues result in the presence, in the overall sensing region, of vertical  
 1231 and/or horizontal stripes that are not filled with useful signal. The width of these lines varies  
 1232 depending on the device but normally does not exceed the equivalent of some tens of pixels.  
 1233 Specifically, the MAXIPIX detector, with sensing area of  $516 \times 516$  pixels of  $55\mu m \times 55\mu m$ , is  
 1234 composed of four modules separated by  $220\mu m$  wide gaps (equivalent of 4 pixels) while the  
 1235 edge pixels measure  $165\mu m$  (i.e. 3 pixels), thus making a gap, when adjacent, of 6 pixels width.

1236 The EIGER detector instead has two types of larger gaps of 12 pixels and 38 pixels width.  
 1237 The detector gaps problem does not affect BCDI only, but it is shared among other X-ray  
 1238 techniques that deal with single photon-counting pixelated detectors and/or beamstops. We  
 1239 have seen in chapter 2 that during a BCDI scan the 2D images acquired by the detector are

stacked to form a 3D array. This leads these lines to become planes of missing signal in the dataset. The problems arise when reconstructing the data with these gaps. In fact, these regions of non-physical zero intensity deceive the Phase Retrieval algorithms inducing the presence of artifacts in the reconstructions [144] (see Fig.5.1).



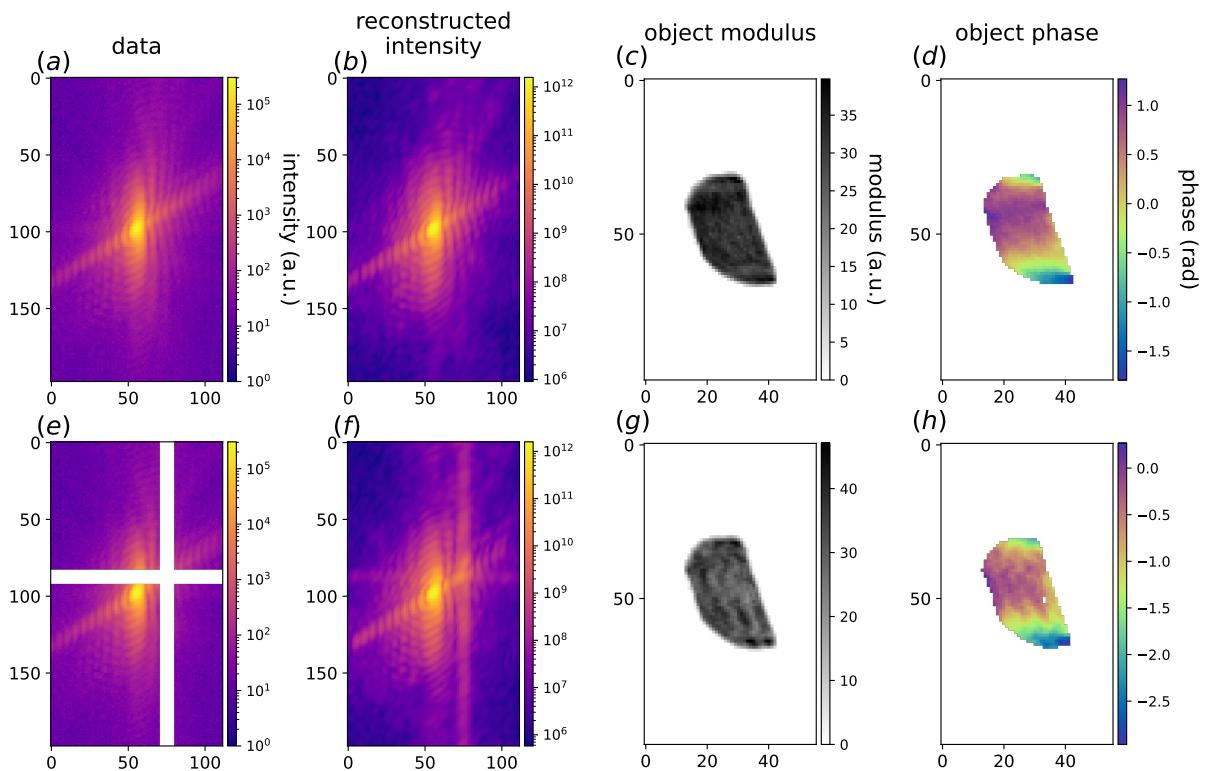
**Figure 5.1: Effect of detector gaps in BCDI reconstructions** (a) The central slice of an experimental diffraction pattern. (b) The same slice of the diffracted intensity calculated from the retrieved object. (c - d) slice of the modulus and phase respectively of the particle obtained from the phasing of the gap-less dataset. (e) Same slice as in (a) with an artificially added 9 pixel-wide, cross-shaped gaps to mimic the detector's ones. (f) The same slice of the diffracted intensity calculated from the retrieved object when not masking the gap regions. (h - g) slice of the modulus and phase respectively of the particle obtained from the phasing of the gap-affected dataset. The distortions caused by the gaps are evident. Note that for this figure an extreme, but explanatory case of large gaps near central peak of a low-oversampled data was illustrated.

It follows that the reliability of the reconstructions in this case is compromised as the strain distribution can be deeply affected by the artifacts. A good practice during standard BCDI experiments is to avoid the gaps by moving the detector if possible. However, this tends to be problematic for the case of high-resolution BCDI, i.e. when the diffraction pattern measurement extends to higher q-values, thus covering more than one sensing chip and necessarily crossing a gap region. Under these circumstances it becomes important to reduce the amount of artifacts deriving from the gaps.

## 1251 5.2 State of the art

1252 Here we will discuss the current strategies employed to treat the detector gaps. As someone  
 1253 could argue the simplest, yet not practical, solution would be to slightly move the detector  
 1254 sideways and acquire a second full scan with the gap hiding a different region of the same Bragg  
 1255 peak, and then merge the two measurements into a single gap-less one. This would at least  
 1256 double the acquisition time making it, de facto, not an option during standard experiments.

1257 The PyNX software, routinely used for the BCDI phase retrieval at ID01, allows the user to  
 1258 define a mask of the gap regions and ignore those pixels during the execution. This improves  
 1259 the quality of the reconstruction, but one can still notice the presence of oscillations appearing  
 1260 in both the object's modulus and phase. The origin of these artifacts can be found in the  
 1261 diffracted intensity calculated from the reconstructed particle as one can clearly see that the  
 1262 gap-regions is filled with non-physical high intensity (see Fig. 5.2(f))



**Figure 5.2: Masking the gap region during phasing** (a) The central xz slice of an experimental diffraction pattern. (b) The same slice of the diffracted intensity calculated from the retrieved object. Comparing this figure with 5.1(b) one can see that when excluding the gap region from the phasing with a mask, the calculated intensity shows bright non-physical streaks instead of the gaps. (c - d) xz slice of the modulus and phase respectively of the particle obtained from the phasing of the gap affected data with a mask of the gap regions. Despite the much higher quality of the reconstruction, one can notice some oscillatory artifacts appearing in both the modulus and the phase of the retrieved object.

1263 Methods relying on compressive sensing and TV regularization were proposed by He in  
 1264 [145] and Malm [146] respectively. Another, more invasive, option is to *fill* these gaps with an  
 1265 estimate of the intensity distribution that would be there, before the phase retrieval. The task  
 1266 of filling gap in images is usually referred to as “inpainting”.

### 5.2.1 Background on Image Inpainting Research

Computational image inpainting has been widely studied in the field of photography and imaging for many years [147, 148]. The inpainting problem can be defined as the task of using known information extractable from the image, to repair the parts where this information is missing, where for known information the colors, the textures and the semantic features are intended. In the history of image inpainting a clear-cut can be observed when deep learning methods have started to be employed. For traditional inpainting, different techniques have been explored, from the texture synthesis methods pioneered by Efros and Leung [149] to the use of PDEs as Navier-Stokes equations proposed by Bertalmio *et al.* [150] and then again from sparse representations [151] to hybrid methods combining variational and statistical methods [152].

More recently, Deep Learning models, headed by Convolutional Neural Networks (CNN), have taken the place of more traditional methods as they can attain higher accuracy for more complex inpainting tasks. By undergoing a training process, CNNs can “learn” to recognize and reproduce the semantic features of the training dataset, and thus leverage them during inference as additional information beside the colors and textures of the specific image to restore. As we have seen in Chapter 4, the typical CNN architecture for image generation consists of an encoder, which retains the features of the input image and compresses them into a lower dimensional latent space, and a decoder, which is responsible for the generation of the output image starting from the latent space. The model is then trained according to a loss function that pushes the model’s predictions to be close to a given ground truth reference. In some cases, the loss function can be replaced by another CNN that is trained to discriminate true images from the ones predicted by the model. These complementary networks are known as Generative Adversarial Networks (GAN), firstly proposed by Goodfellow *et al.* [137], and have also been used for image inpainting (e.g. [153]).

Since reviewing the vast amount of works about CNN for image inpainting is beyond the scope of this thesis and for more information, we redirect the reader to the reviews written by Elharrouss *et al.*, Xiang et al. and Xu *et al.* [148, 154, 155]. For what concerns the application of DL based inpainting for scientific imaging, early works date back to 2018 as in the case of Sogancioglu *et al.* for X-ray human chest 2D radiographic images [156] and to 2020 for 2D microscopic images [157]. A couple of years later Tanny Chavez and coauthors published a paper comparing the performance of different CNN models for the inpainting of 2D X-ray diffraction images [158]. The work is precisely addressing the gap problem for X-ray detectors used for powder diffraction measurements and is awarding U-Net and Mixed Scale Dense (MSD) models for the best performance on experimental data. The DL models outperform interpolations obtained with biharmonic functions across 7 and 17 pixel-wide gaps. This work has been of inspiration for the design of our DL model for BCDI gaps inpainting. In the same year, another work on DL based inpainting for X-ray detector gaps was published by Alfredo Bellisario and coauthors [159]. The authors tested a U-Net-like model on the inpainting of 2D simulated, noiseless coherent diffraction patterns against gaps of different sizes (2 to 20 pixels) along the central row. The gaps were placed such that the center of the peak was covered, a choice that, as we will see later, yields better results than predictions on peripheral areas. To our knowledge, at the time of writing, no other works about deep learning based inpainting for X-ray detector gaps are present in the literature.

## 1312 5.3 Model design: 2D case

1313 For simplicity, I have begun with the 2D case, using simulated diffraction patterns and inpainting  
 1314 randomly placed vertical gaps of different width. First, a training set of simulated data was  
 1315 created, composed of pairs of gap-affected images and corresponding gap-free ground truths,  
 1316 then a U-Net-like model was built and trained in a supervised fashion.

### 1317 5.3.1 Dataset creation

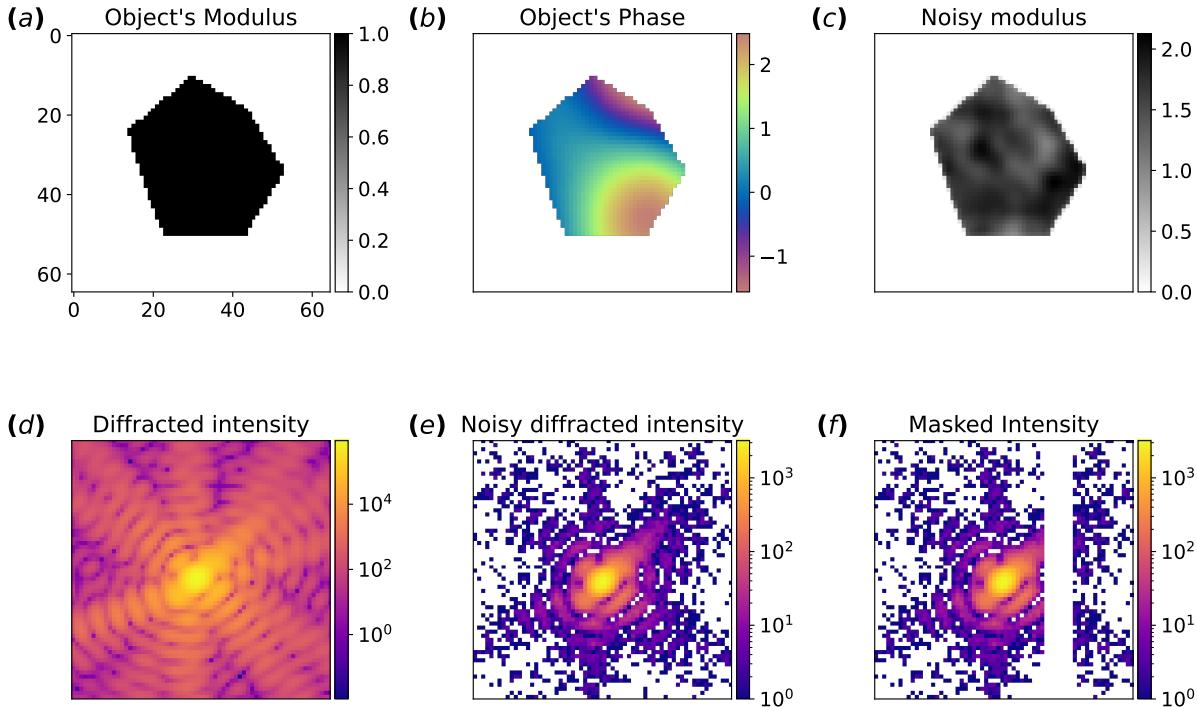
1318 The creation of training datasets of simulated 2D BCDI patterns for both the gap-inpainting  
 1319 and phase retrieval tasks has followed the procedure described in this paragraph.  
 1320 First, once the size of the array was chosen, a randomly shaped polygon was created in the  
 1321 center using `scipy.spatial.ConvexHull` function. This guarantees the object to have  
 1322 a compact support with homogeneous electron density as assumed for BCDI. Subsequently,  
 1323 a random phase field of the same size with variable phase range and correlation length was  
 1324 generated thus the complete complex object was formed.

1325 In order to make the object more realistic a Gaussian filter and Gaussian - correlated noise  
 1326 [160] have been applied to the object's modulus, to smoothen the edges and simulate real cases  
 1327 respectively. At this point the object was resized to the shape required to match the chosen  
 1328 oversampling ratio and the 2D Discrete Fourier Transform was computed. Finally, Poisson  
 1329 noise was added to the diffraction patterns with different magnitudes to simulate various X-ray  
 1330 flux conditions (see Fig.5.3).

1331 Datasets contain a number of diffraction patterns in the order of thousands and for each of  
 1332 them the random variables have been varied as well as the oversampling ratios. In the datasets  
 1333 for the training of phase retrieval models (presented later in Chapter 6), the reciprocal space  
 1334 phase corresponding to each diffraction pattern was saved as well and used as ground truth  
 1335 label. For inpainting tasks a randomly located vertical gap mask was created and applied  
 1336 to the intensity data. In some cases cross-shaped gaps were added instead to simulate the  
 1337 experimental condition of the Bragg peak in the vicinity of the corner of the sensing area. The  
 1338 size of the gaps was chosen to be consistent across the dataset and four different cases were  
 1339 studied (3px, 6px, 9px, 12px). Fig.5.3.

### 1340 5.3.2 2D Model design

1341 The 2D model that I have implemented is a U-Net that takes in input batches of 32 simulated  
 1342 BCDI patterns affected by both vertical and cross-shaped gaps. Each diffraction pattern was  
 1343 transformed into logarithmic scale to enhance the spatial features and then normalized between  
 1344 0 and 1. This normalization was proven to be convenient to any DL model [161]. Regarding  
 1345 the logarithmic transformation, it is important to notice that in order to avoid problems for  
 1346 zero intensity values, the  $\log(I + 1)$  was taken. The shape of each image was chosen to be  
 1347 of  $128 \times 128$  pixels. The encoder of part of the models was built as follows. The inputs go  
 1348 through five convolutional blocks inside each of which a convolutional layer, a Leaky ReLU  
 1349 activation function and a MaxPooling operation are applied. The feature map's size is therefore  
 1350 reduced down to  $2 \times 2$  while the channel dimension is brought up to 768 filters, and the kernel



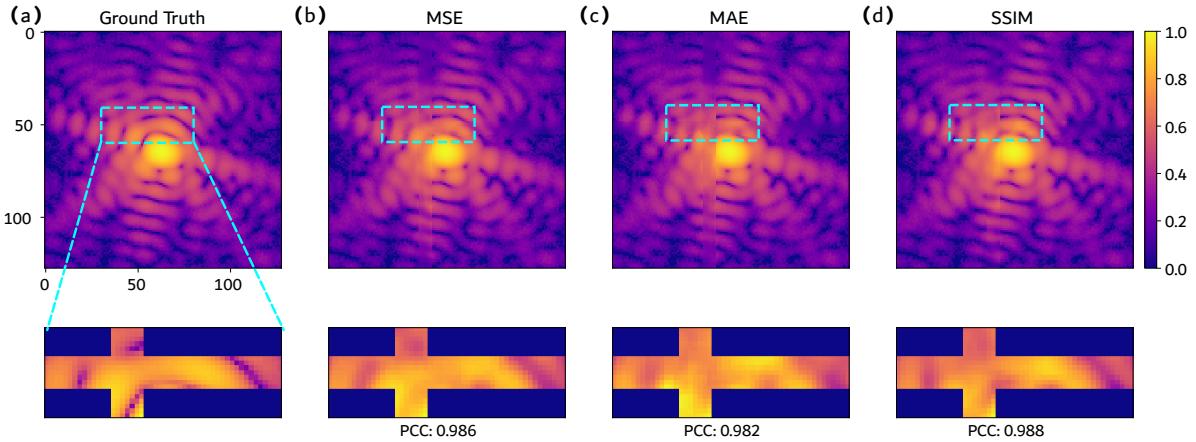
**Figure 5.3: Steps for the simulation of a single 2D diffraction pattern** (a) Simulated modulus of a 2D object with random shape and compact support. The object is padded with zeros to match the chosen oversampling ratio. (b) Simulated object's phase (c) Object's modulus after smoothening the edges and adding random Gaussian noise. (d) Squared modulus of the Fourier Transform of the complex object (in log scale). (e) Poisson noise is added to the simulated diffracted intensity. (f) A 6 pixel-wide vertical gap is added to the diffracted intensity at a random position.

1351 size is kept at  $3 \times 3$ . At the end of the encoder the feature map passed to the decoder is a  
 1352 (32, 2, 2, 768) tensor. The decoder, mirroring the encoder, is composed of five blocks inside  
 1353 each of which there is a transposed convolutional layer that upsamples the feature maps (stride  
 1354 = 2) and a Leaky ReLU activation function. Skip connections connecting each encoder block to  
 1355 its corresponding shape-like decoder block have been implemented as well. This measure has  
 1356 proven to be beneficial for the information flow between encoder-decoder [142]. The last activa-  
 1357 tion layer of the model is a sigmoid function that guarantees an output bounded between 0 and 1.  
 1358

1359 A simple Mean Squared Error (MSE) was used as loss function inside the gap region only,  
 1360 training the model on 12'000 diffraction patterns over 10 epochs, with ADAM optimizer and  
 1361 a learning rate of  $10^{-4}$ . The Mean Absolute Error (MAE) and the Structural Similarity Index  
 1362 Measure (SSIM) [162] loss functions were tested as well and the corresponding results obtained  
 1363 after the same training were compared. Here in Fig. 5.4 we report the comparisons for the 9  
 1364 pixel-wide gap on a test simulated diffraction pattern. The accuracy scores were calculated  
 1365 using the Pearson Correlation Coefficient (PCC).

$$PCC = \frac{\sum_{i \in \text{gap}} (\mathbf{I}_i^{\text{true}} - \langle \mathbf{I}^{\text{true}} \rangle)(\mathbf{I}_i^{\text{pred}} - \langle \mathbf{I}^{\text{pred}} \rangle)}{\sqrt{\sum_{i \in \text{gap}} (\mathbf{I}_i^{\text{true}} - \langle \mathbf{I}^{\text{true}} \rangle)^2} \sqrt{\sum_{i \in \text{gap}} (\mathbf{I}_i^{\text{pred}} - \langle \mathbf{I}^{\text{pred}} \rangle)^2}}, \quad (5.1)$$

1366 Where  $\mathbf{I}$  is the intensity inside the gap. The PCC measures the linear correlation between  
 1367 two sets of data, and it is naturally normalized between -1 and 1 for which the two datasets are  
 1368 said to be linearly correlated. On the contrary the 0 value implies uncorrelated sets.



**Figure 5.4: Comparison of different losses** Results on a test simulated diffraction pattern for the inpainting of a 9 pixel-wide cross-shaped gap produced by the same U-Net model trained for 10 epochs with different loss functions. (a) Shows the ground truth. (b) The prediction of the model trained with the MSE, (c) with the MAE, (d) with the SSIM. Corresponding accuracy scores calculated with the Pearson Correlation Coefficient (PCC) are shown as well. While MAE fails to recover the oscillations, SSIM yields better results.

1369 In the light of these results, the MAE metric was discarded and the sum of MSE and SSIM  
 1370 adopted instead. Finally, another term that computes the MSE between the *gradients* of the  
 1371 ground truth and predicted intensity inside the gap region was added in the definitive loss  
 1372 function.

1373

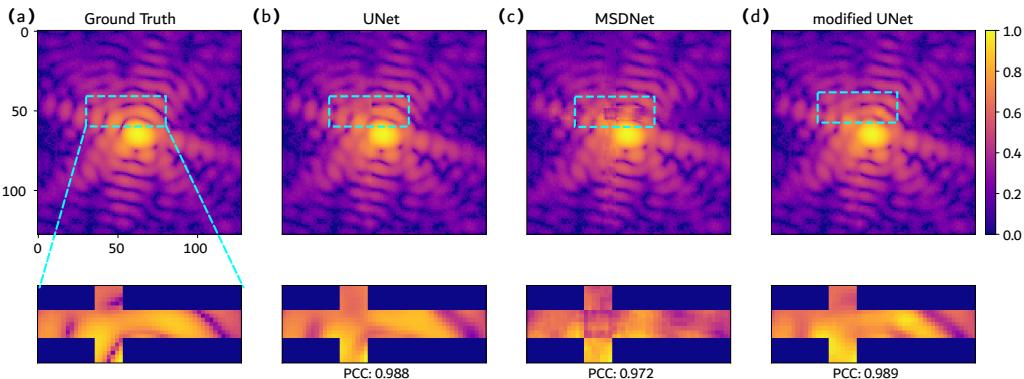
1374 Once established what was considered the best loss function, different models have been  
 1375 explored . Following the work of Chavez *et al.* mentioned above ([158]), a Mixed-Scale Dense  
 1376 Network (MSD-Net) was used [163]. The advantage of this type of network is the significant  
 1377 reduction of trainable parameters, and the use of *dilated* convolutions with respect to U-Net  
 1378 ones. While the former property guarantees faster training and lower chances of overfitting, the  
 1379 latter enhances the capture of long-range correlations. Moreover, in an MSD-Net, the image's  
 1380 spatial dimensions are kept constant throughout the whole network as no downsampling nor  
 1381 upsampling is made. The MSD-Net consists of sequential blocks in each of which the input is  
 1382 transformed by two different convolutional layers with growing dilation rates. Each output of  
 1383 the convolutional layers is concatenated to the input feature map and the result is passed to  
 1384 the following block. While the kernel size is kept constant to  $3 \times 3 \times 3$  pixels the dilation rate  
 1385 increases linearly from 1 to 30. The last layer is a sigmoid function as well as for the U-Net.  
 1386 The total number of trainable parameters is of the order of 320'000, two orders of magnitude  
 1387 lower than the U-Net.

1388 In order to combine the hierarchical dimensionality reduction of the U-Net with the fine-  
 1389 features capturing of the MSD-Net we have implemented a modified U-Net that adopts dilated  
 1390 convolutions inside the first three encoder blocks. In particular, they return the input tensor  
 1391 concatenated with the outputs of four dilated convolutional layers computed from the input.  
 1392 Dilation rates of (16,8,4,2), (10,5,3,1) and (5,3,2,1) were chosen respectively. As the MaxPooling

1393 operation down-samples the feature maps into smaller sizes, we limited the dilated convolutions  
 1394 to the first three blocks. Moreover, they were used in the encoder layers only as they are mostly  
 1395 used for feature extraction [164]. The characteristics of each model are summarized in form of  
 1396 pseudo - code in Table 5.1.

1397

1398 The three different models have been trained with a combined loss function (MSE + SSIM +  
 1399 MSE on the gradients) on the same training dataset for 10 epochs each. The results showed  
 1400 poor performances of the MSD-Net with respect to the two U-Nets. Slightly higher accuracy  
 1401 was achieved by the modified U-Net (see Fig. 5.5).



**Figure 5.5: Comparison of different models** Results on a test simulated diffraction pattern for the inpainting of a 9 pixel-wide cross-shaped gap using three different models trained with the same loss function. (a) Shows the ground truth. (b) The prediction of the U-Net, (c) of the MSD-Net, (d) of the modified U-Net. Corresponding accuracy scores calculated with the Pearson Correlation Coefficient (PCC) are shown.

1402 We conclude here the introductory studies on 2D simulated data. These preliminary tests  
 1403 allowed to explore different DL architectures and loss functions and select the optimal choices  
 1404 for the inpainting of BCDI detector gaps.

### 1405 5.3.3 Accuracy VS Gap position

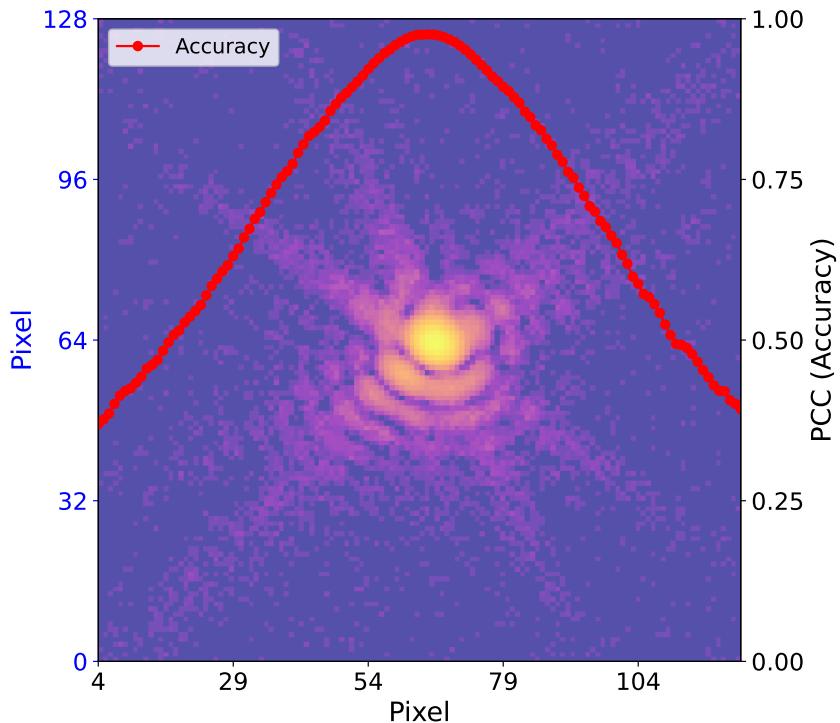
1406 Before moving to the 3D case, it is worth spending a few words on the assessment of the DL  
 1407 model upon different conditions. Namely, the accuracy as function of the position of the gap  
 1408 inside the diffraction pattern and as a function of the oversampling ratio. For the first test  
 1409 150 2D diffraction patterns were simulated starting from random particles shapes, random  
 1410 oversampling ratios and Poisson noise intensity. For each diffraction pattern a vertical 9 pixel-  
 1411 wide gap was placed at all positions from left to right, then the DL prediction and corresponding  
 1412 accuracy score when compared to the ground truth were calculated. The accuracy was again  
 1413 calculated with the Pearson Correlation Coefficient. This score was then averaged for each gap

	U-Net	MSD-Net	U-Net_mod
<b>block1</b>	<pre> def encoder_block(x_input,                   num_filters, ker):     s = Conv2D(num_filters, ker,               'leaky_relu')(x_input)     x = MaxPool2D(2)(s)     return x, s </pre>	<pre> def MSD_block(x, in_channels,               dilations,kernel_size=3):     if isinstance(dilations, int):         :         dilations = [(j % 10) + 1                       for j in range(                           dilations)]      out_channels = in_channels +         len(dilations)     for d in dilations:         x1 = Conv2D(out_channels                     //2,kernel_size,1,                     dilation_rate=                     dilation, 'same', ,                     'leaky_relu')(x)         x = tf.concat([x1,x] ,axis                       = -1)     return x, out_channels </pre>	<pre> def encoder_block_mod(x_input,                       ker, num_filters, rate):     f = num_filters // 4     s = tf.concat([x_input] + [         Conv2D(f, ker,                dilation_rate=r, ,                'leaky_relu')(x_input)         for r in rate], axis=-1)     return MaxPool2D(2)(s), s </pre>
<b>block2</b>	<pre> def decoder_block(x_input,                   num_filters, ker,                   skip_input = None):      if skip_input is not None:         x_input = Concatenate()([             x_input, skip_input])      x = Conv2DTranspose(         num_filters, ker,         strides=2, 'leaky_relu')         (x_input)     return x </pre>		<pre> def decoder_block(x_input,                   num_filters, ker,                   skip_input = None):      if skip_input is not None:         x_input = Concatenate()([             x_input, skip_input])      x = Conv2DTranspose(         num_filters, ker,         strides=2, 'leaky_relu')         (x_input)     return x </pre>
<b>body</b>	<pre> x, s1 = encoder_block(inputs,                       48,3) x, s2 = encoder_block(x,                       96,3) x, s3 = encoder_block(x,                       192,3) x, s4 = encoder_block(x,                       384,3) x, s5 = encoder_block(x,                       768,3)  x = Conv2D(1536,3, 'leaky_relu')(x)  x = decoder_block(x,768,3) x = decoder_block(x,384,3, s5 ) x = decoder_block(x, 192,3,s4 ) x = decoder_block(x, 96,3,s3) x = decoder_block(x, 48,3,s2)  x = Conv2D(24,5,'leaky_relu') (x) x = Conv2D(12,5,'leaky_relu') (x) x = Conv2D(6,5,'leaky_relu')(x)  out = Conv2D(1,5,'sigmoid')(x) </pre>	<pre> x,out_ch = MSD_block(inputs                       ,1,[1,2]) x,out_ch = MSD_block(x,out_ch                       ,[3,4]) ... x,out_ch = MSD_block(x,out_ch                       ,[31,32]) out = Conv2D(1,3,'sigmoid')(x ) </pre>	<pre> x, s1 = encoder_block_mod(     inputs,3,48,[16,8,4,2]) x, s2 = encoder_block_mod(x     ,3, 96,[10,5,3,1]) x, s3 = encoder_block_mod(x     ,3, 192,[5,3,2,1]) x, s4 = encoder_block(x, 384     ,3) x, s5 = encoder_block(x, 768,     3)  x = Conv2D(1536,3,'leaky_relu     )(x)  x = decoder_block(x,768,3) x = decoder_block(x,384,3,s5) x = decoder_block(x,192,3,s4) x = decoder_block(x,96,3,s3) x = decoder_block(x,48,4,s2)  x = Concatenate()([x, s1]) x = Conv2D(24,5,'leaky_relu') (x) x = Conv2D(12,5,'leaky_relu') (x) x = Conv2D(6,5,'leaky_relu')(x )  out = Conv2D(1,3,'sigmoid')(x ) </pre>
<b>parameters</b>	31,827,673	322,458	32,652,337

**Table 5.1:** Comparison of U-Net, MSDNet, and U-Net\_mod components.

position, over the 150 diffraction patterns and the result was plotted as a function of the gap position. Fig. 5.6 shows the resulting curve that clearly highlights that the model performs better in regions with high intensity. This can be qualitatively explained: (i) central pixels contain larger spatial features both because of the nature of the Bragg peak, and because of

the lower noise level. This makes it easier for the model as it reduces the complexity of the prediction. (ii) As we move away from the center of the Bragg peak, the Signal to Noise Ratio (SNR) decreases, along with the *density of signal*. High accuracy scores in these regions would require the model to be able to predict noise correctly which is by definition impossible as it is an uncorrelated random process. One could argue that the accuracy curve would follow the statistical distribution of the gap positions inside the DL model training dataset. However, each mask has been applied at a position drawn from a discrete uniform probability function spanning the full data size, so can be excluded.

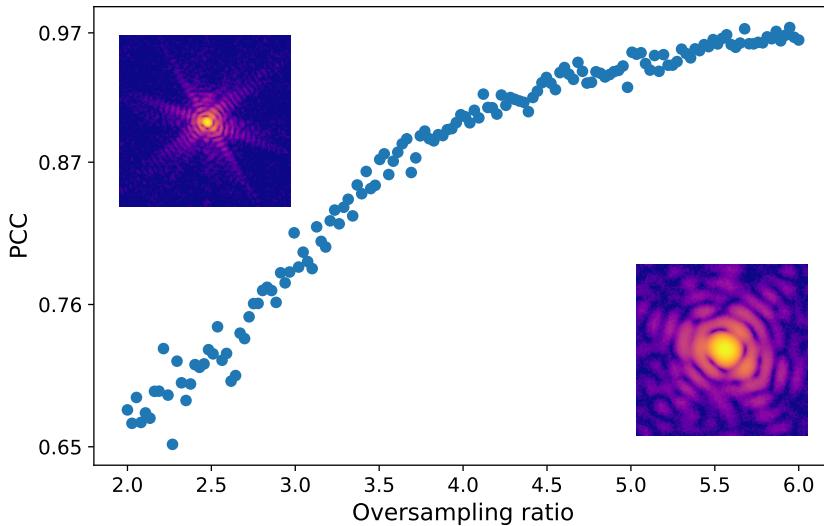


**Figure 5.6: (Accuracy VS Gap position)** Average Pearson Correlation Coefficient calculated over 150 9 pixel-wide vertical predicted gaps for each position of the gap inside the diffraction patterns. The model shows higher accuracies when the gap masks high intensity regions.

To conduct the second test, 150 diffraction patterns were simulated for the same particle varying gradually the oversampling ratio (see Sec.3.1) between 2 and 6. For each image a 9 pixel-wide vertical gap was then applied at all  $X$  positions and the DL prediction was computed. The accuracy scores have been averaged for each prediction in the same image and plotted against the oversampling (Fig. 5.7). As expected from the previous discussions, the model performs better for larger oversampling ratios, because of the bigger size of the features with respect to the gap width and because of the more uniform *density of signal*. It is worth clarifying that, for a given particle, the total amount of intensity in the diffraction patterns is in principle constant regardless of the oversampling ratio as it is fixed by Parseval's theorem<sup>1</sup>. However, if the size of the dataset is kept fixed for different oversamplings, the effect is the same of a zoom lens that increases or reduces the field of view. Therefore, while for low oversampling ratios the full peak is recorded, for higher ones the peak is cropped, and less intensity is present in the image.

<sup>1</sup>Interpreted also as an energy conservation law, Parseval's theorem states that  $\int |f(x)|^2 dx = \int |\mathcal{F}\{f(x)\}|^2 dk$ . In other words, the overall amount of signal is constant in direct space as in Fourier space.

1439 This effect, coupled with the typical radial intensity decay of Bragg peaks and of Poisson  
 1440 noise, reduces the *density of signal* in largely oversampled BCDI patterns. Here the *density*  
 1441 *of signal* refers to the amount of information per pixel, which becomes smaller and more  
 1442 uniform. On the contrary, for low oversampling ratio the *density of signal* is less uniform as it  
 1443 is high inside bright regions (a lot of information concentrated in few pixels) and low in noisy  
 1444 regions far from the peak. It follows that in order to properly assess the accuracy against the  
 1445 oversampling ratio one should consider diffraction patterns over the same extent in  $q$ -space,  
 1446 thus changing the size of the images. This more accurate evaluation was carried out for the 3D  
 1447 case and can be found in the next section.



**Figure 5.7: (Accuracy VS Oversampling ratio)** Average Pearson Correlation Coefficient calculated over 280 9 pixel-wide vertical predicted gaps for each position of the gap inside the diffraction patterns. The model shows higher accuracies for high oversampling ratios.

## 1448 5.4 3D case - Patching approach

1449 When considering the 3D case, and especially the experimental conditions, there are a few practical  
 1450 issues that need to be overcome. In fact, experimental BCDI datasets that are more often  
 1451 affected by a detector's gaps are necessarily large datasets (e.g.  $512\text{px} \times 512\text{px} \times N_{\text{rocking\_steps}}$   
 1452 where  $N_{\text{rocking\_steps}}$  is of the order of hundreds of pixels in typical BCDI experiments). Training a  
 1453 U-Net like model for 3D images of that size is overly expensive in terms of computing memory  
 1454 and time. Moreover, a common problem with this type of architectures is that the size of the  
 1455 images they can process is fixed by the first initialization. This means that one would need to  
 1456 resize, via binning or interpolation, the experimental datasets to the shape accepted by the DL  
 1457 model, and back to the original shape after the inpainting. Besides the impracticality, these  
 1458 operations are not recommended as they induce further modification and information loss to  
 1459 the original data. For these reasons a patching approach that loosens these constraints while  
 1460 preserving sufficiently high accuracies was considered.

1461

1462 The patching method exploits the regularity of the oscillations within BCDI datasets. The  
 1463 periodicity of the fringes in reciprocal space, peculiar property of this coherent diffraction

technique, follows from the oversampling condition as it is given a product of the Fourier transform of an object less than half of the Fourier window containing the diffraction pattern. It can be observed by eye and in many cases makes the prediction inside a gap region intuitively possible starting from just a few neighboring pixels. In our case we have decided to work with 32 pixel-sided cubic sub-volumes (patches from now on) cropped out of entire diffraction patterns. Among the “GPU-friendly” tensor sizes [165] we opted for 32 as a good trade-off between amount of contained information and computing power required for training and inference. Moreover, this size is also greater than twice the size of the usual gap sizes and enough to catch several fringes in typical experimental oversampling conditions (< 10 usually).

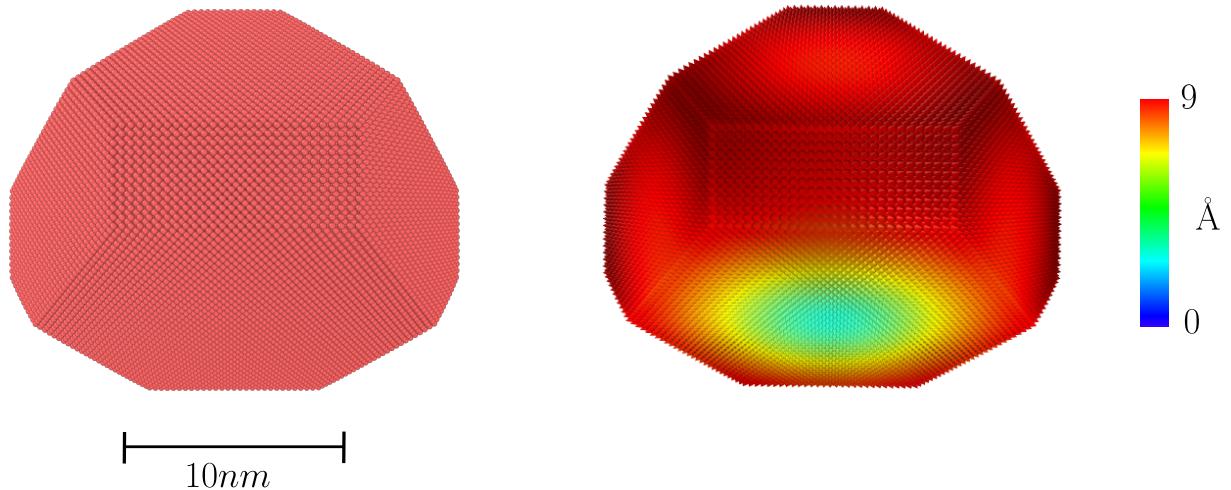
### 5.4.1 Dataset creation

The training dataset consists of 50% patches from experimental data and 50% from simulated data. The experimental measurements were acquired at the ID01 beamline of the ESRF during different beamtimes on different particles. Namely, (i) Pt particles de-wetted on sapphire and YSZ (yttria–stabilized zirconia) with Winterbottom shape, measured under various temperatures and gas conditions, (ii) Pd and PdCe particles on glassy carbon, with Wulff shape, measured in an electrochemical environment following hydrogen loading. (iii) Ni particles on sapphire undergoing changes during  $CO_2$  adsorption and (iv) “cubic  $CaCO_3$ ” particles on glassy carbon. The synthetic diffraction patterns were instead simulated in three steps. The first step consisted in the creation of simulated 3D particles of different shapes (Winterbottom, Wulff, Cubic, Octahedral and random) using pre-existing scripts developed by Dr. Dupraz and Dr. Bellec [166]. These codes allow the user to construct a cubic FCC crystal of a given element, taking into account the inter-atomic potential, the atomic mass and the lattice parameter.

The final particle was finally obtained by "cutting" off atomic planes along given (or random) directions, depending on the chosen shape. Only Gold nano-particles were simulated, and this is, in first approximation, equivalent to any generic element as a different lattice parameter would just shift the Bragg peak to a different position in reciprocal space, with no significant alterations of the diffraction pattern itself. Each particle's configuration was then automatically saved in a LAMMPS-readable file. In a second stage, energy relaxation using LAMMPS software for Molecular Dynamics was performed [167]. This step induces small displacements to the perfect lattice, especially near the surface and the substrate. In the last stage, the 3D diffraction pattern of a selected Bragg reflection was computed using PyNX scattering package [55]. This software, optimized for GPU acceleration, produces a 3D representation of a selected Bragg peak using the kinematic sum. It was then possible to adjust the parameters that control the oversampling ratio, the size of the array in which the Bragg peak was centered and the rotation of the  $q$ -space. In this case, 128 pixel-size cubic diffraction patterns were simulated and, in order to augment the training dataset, various oversampling ratios (from 2 to 5) and different rotations for each particle were selected.

As we have seen in Chapter 2, in the kinematic scattering approximation the energy of the incident X-ray does not alter the diffraction pattern if not as a “zooming” factor. It follows that simulating diffraction patterns for different X-ray energies for a fixed pixel size and distance results in different oversampling ratios.

Before taking portions of these simulated BCDI patterns, Poisson noise was applied, randomly scaling the  $\lambda$  parameter for each image.



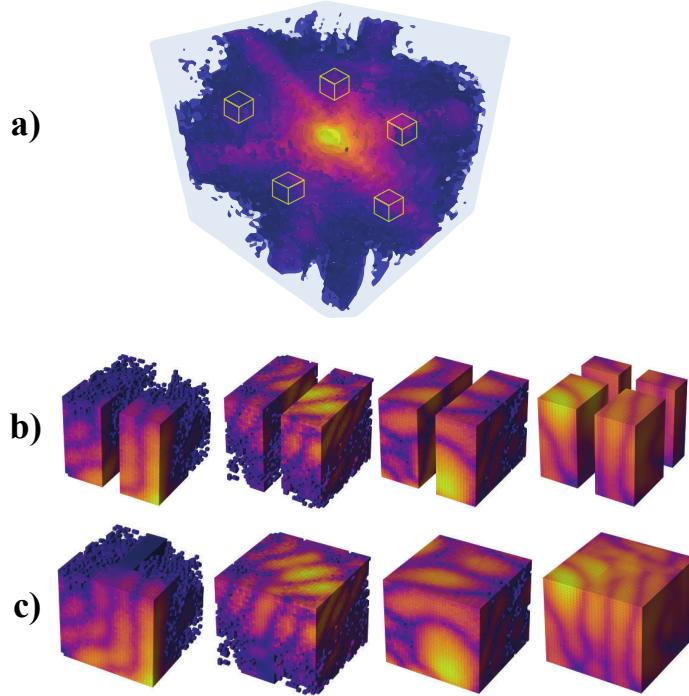
**Figure 5.8:** **Left:** Simulated Au particle with Winterbottom shape (134114 atoms). **Right:** Atomic displacement field of the same particle after LAMMPS energy relaxation showing the typical distribution at the interface with the substrate.

At this point, I proceeded with the extraction of patches taken at *pseudo*-random locations inside each 3D pattern. The selection in fact was not totally random as the extraction of patches from the outer regions, far from the center of the peak was favored. There are mainly two reasons for this choice, namely (i) compensate the inherent uneven accuracy score against the position of the gap (see Fig.5.6) by increasing the training data far from the center and (ii) emulate as much as possible the experimental conditions, in which unavoidable gaps are typically far from the center of the peak. For each sub-volume a 3D mask of the gap was created for different gap sizes (3,6,9,12 pixel-wide). The gap was placed vertically, in the center, along the third dimension, resulting in a “empty slab”. Notice that, while in practice a gap may obscure the diffraction pattern at any position, it is unnecessary to consider other placements within a patch, since one can always select a patch symmetrically around the gap. Cross-shaped gaps were also included in the training dataset, with a population ratio of 1:5 compared to vertical gaps. They were created by adding a horizontal gap at a random height to an existing vertical gap (Fig. 5.9). The final training dataset consisted of 30'000  $32 \times 32 \times 32$  pairs of patches, with gap and ground truth, created as described above.

## 5.5 3D model architecture

The DL architecture used for the 3D patching inpainting is illustrated in Fig. 5.10. Given the reduced size of the inputs, the encoder was composed of four blocks only, in each of which there were convolutional layers and max pooling layers. The feature map was thus reduced to a  $2 \times 2 \times 2 \times 478$  tensor before being passed to the decoder. Notice that, as introduced above in section Sec.5.3, dilated convolutions were employed in the first two blocks to enhance the extraction long-range correlated features. After four decoder blocks three simple convolutional layers with 24,12 and 6 channels were used respectively in order to restore the possible smoothing effect of the decoder. As in the 2D model, the last activation function is a sigmoid that ensures the output to be in the range (0,1). The model contains 2'770'000 trainable parameters, significantly less than the 2D models working on full size patterns.

The training was performed loading batches of 32 images at the time over 100 epochs using



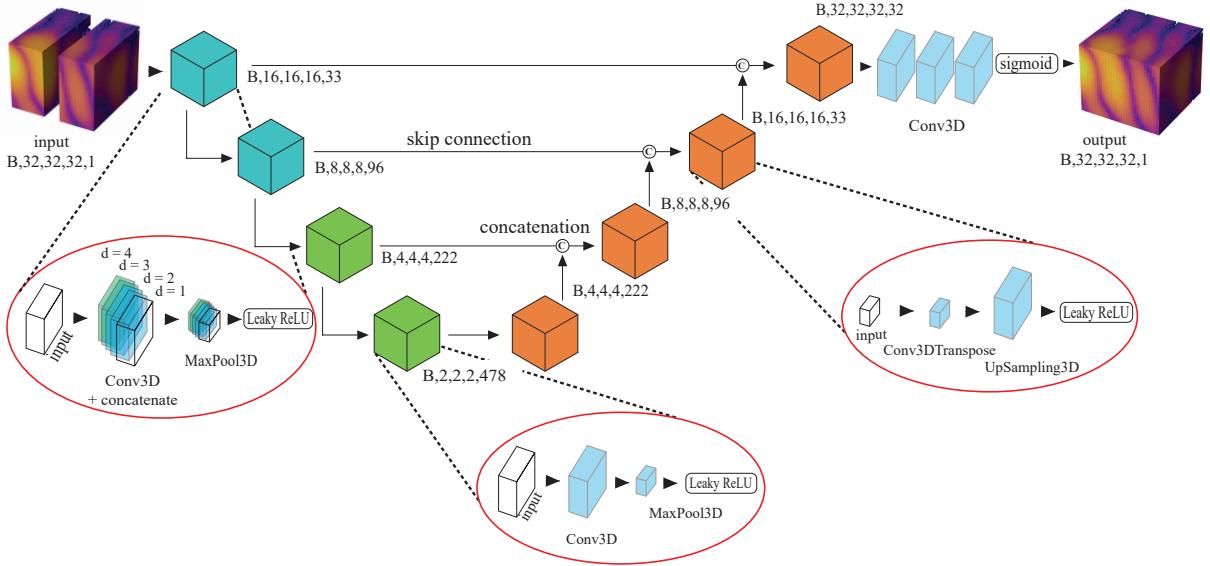
**Figure 5.9: Schematic of the patches extraction.** a) The 3D BCDI diffraction pattern and the patches. b)  $32 \times 32 \times 32$  pixel-size patches with 9 pixel-wide vertical and cross-shaped gaps. c) Same patches with the DL inpainted gaps.

1534 ADAM optimizer [121]. The optimizer was initialized with a learning rate of  $10^{-3}$  and decreased  
1535 it progressively with the ReduceLROn-Plateau callback feature available in Tensorflow.  
1536 In order to exploit the maximum of the training dataset only 4% and 2.5% of the whole dataset  
1537 was left for validation and testing respectively.

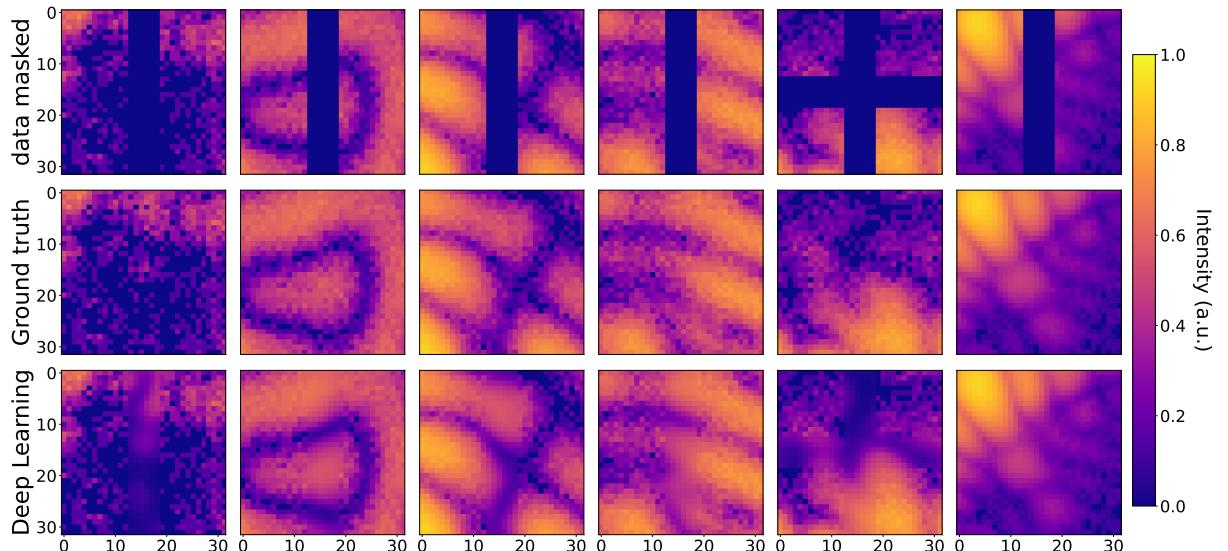
## 1538 5.6 Results in detector space

1539 In this section the results of the DL model on both simulated and experimental diffraction  
1540 patterns are presented. The following section will focus on the results in real space and the  
1541 reduction of the artifacts in the reconstructed objects.

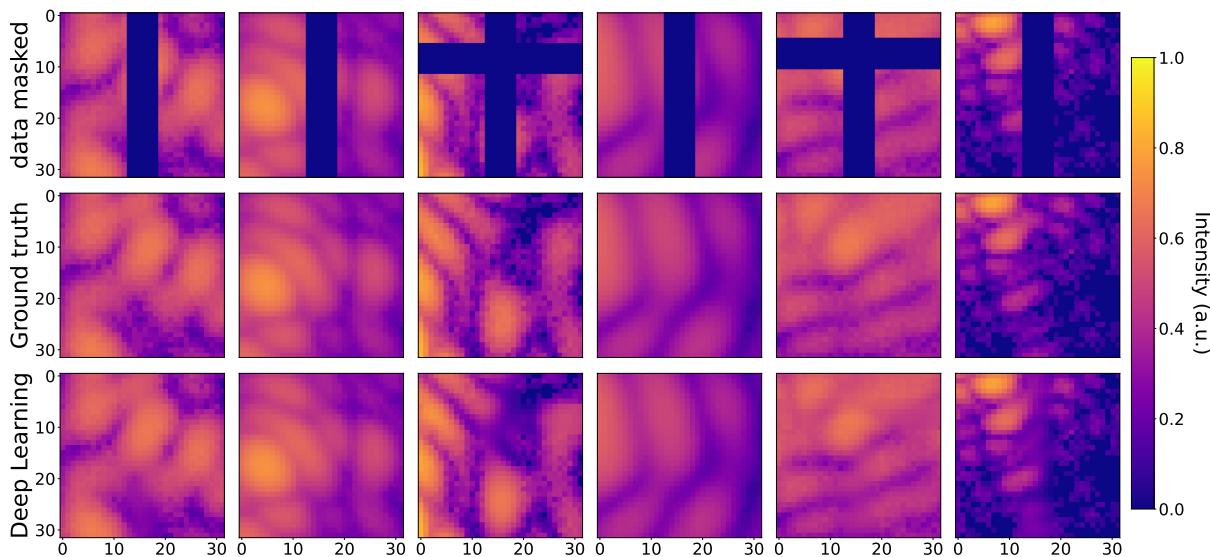
1542  
1543 Once completed the training, the model was first tested on portions taken from the test  
1544 dataset. It is possible to qualitatively observe that the model works equally well for both  
1545 simulated and experimental data (see Figs. 5.11 - 5.12). From a first visual assessment one  
1546 can also confirm that low noise regions with larger features are better restored than others as  
1547 previously stated in Sec. 5.7. Another curious effect that can be observed, is the “smoothing” of  
1548 features around noisy areas (see first column in Fig. 5.11 and last column in Fig. 5.12). In fact,  
1549 the “grainy” aspect of these regions is caused by Poisson noise which cannot be predicted by  
1550 the DL model as it is uncorrelated. In those regions the DL performs therefore an averaging  
1551 that “smoothens” the features and acts like a denoiser. This effect has been already studied in  
1552 the literature and exploited for denoising applications like the Noise2Void model [168].



**Figure 5.10: Schematic of the 3D model architecture** The model uses a modified U-Net structure. In the first two encoder blocks (highlighted by the left red circle), dilated convolutions are applied where the original input is concatenated with its convolutions at various dilation rates ( $d = 4, 3, 2, 1$ ) prior to the MaxPooling operation. The input consists of small gap-affected portions, grouped into batches of 32 ( $B$ ). These portions (top left) are progressively processed by the encoder until they are reduced to a  $2 \times 2 \times 2$  pixel-size feature map. In the decoder, each building block (represented as orange cubes) receives as input the concatenation of the output from the previous block and the matching output from the encoder block of the same size. The final result (top right) is a batch of inpainted versions of the input portions.



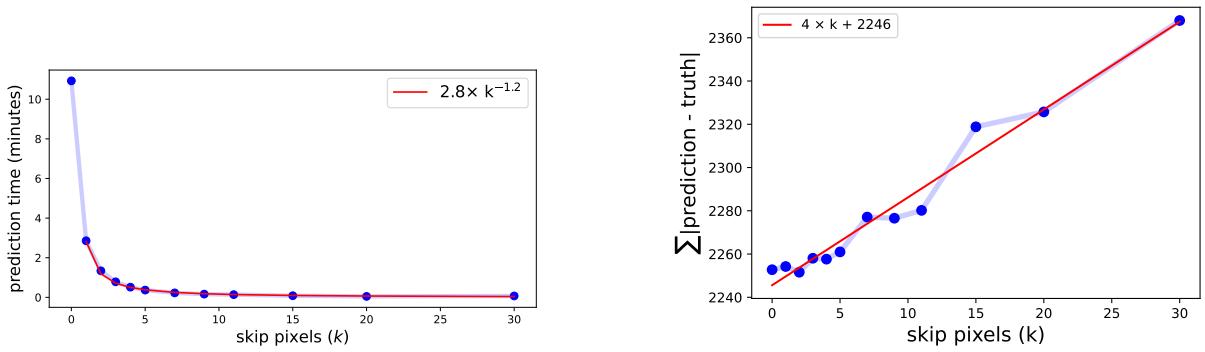
**Figure 5.11: Results on portions of test simulated data.** Central slices of portions taken from the simulated test dataset. Masked input with 6 pixel-wide gap in the first row, corresponding ground truth and DL inpainted in second and third row respectively.



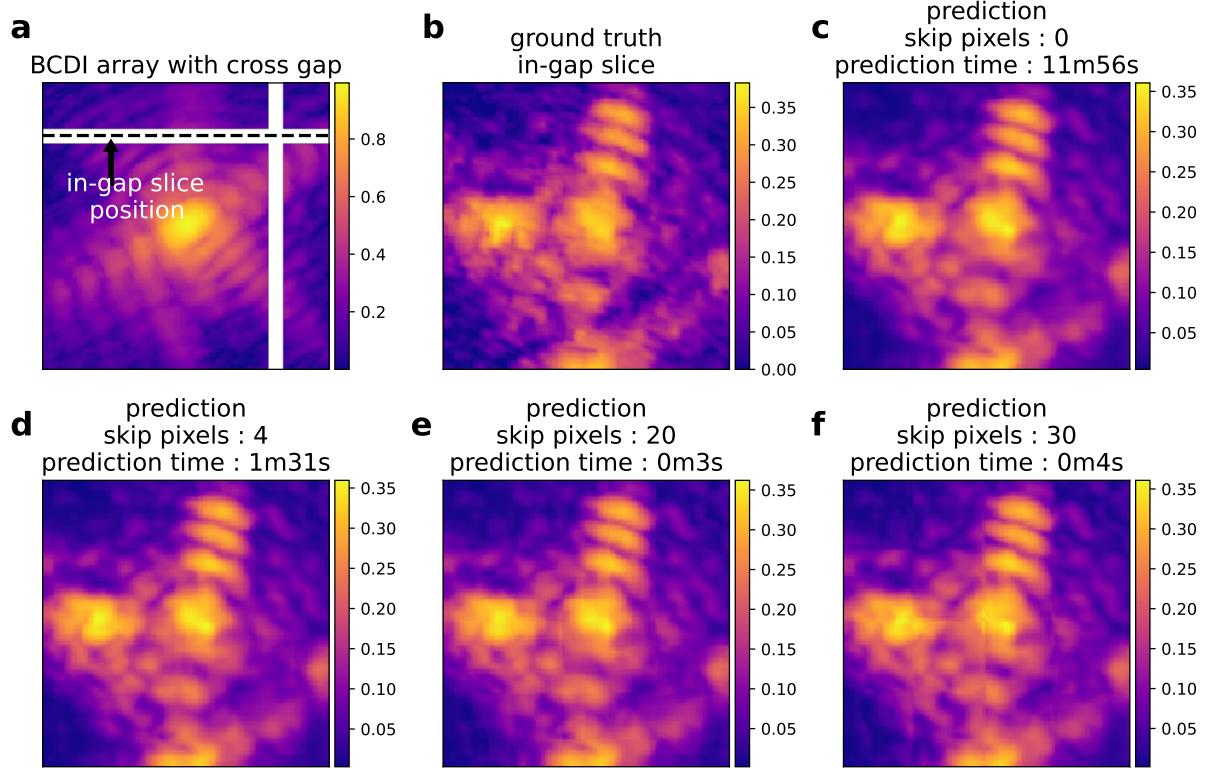
**Figure 5.12: Results on portions of test experimental data.** Central slices of portions taken from the experimental test dataset. Masked input with 6 pixel-wide gap in the first row, corresponding ground truth and DL inpainted in second and third row respectively.

1553    **5.6.1 Full gap inpainting**

1554    For the inpainting of a gap inside a full 3D BCDI pattern it is sufficient to apply repeatedly  
1555    the DL model on sub-volumes cropped such that the gap plane lies vertical in the center of  
1556    the array perpendicularly to the third dimension. Each sub- volume needs to be preprocessed  
1557    exactly in the same way described above, i.e. transformed into logarithmic scale and normalized  
1558    between 0 and 1. Moreover, it is advised to apply a mask on the gap, to match exactly the  
1559    gap width the model has been trained with. One can then proceed along the gap moving  
1560    forward one pixel at the time, compute the inpainted gap and average the prediction over the  
1561    overlapping pixels with the previous predictions. By doing this, potential errors are averaged  
1562    out and the accuracy of the prediction is maximized. However, for large datasets this can  
1563    be time-consuming. For example, for a  $128 \times 128 \times 128$  pixel-size diffraction pattern with a  
1564    cross-shaped gap the time needed to compute the full inpainting amounts to 11 minutes (using  
1565    a NVIDIA Tesla V100-SXM2 GPU with 32GB RAM). However, it is possible to increase the step  
1566    size to significantly reduce the computing time without affecting excessively the accuracy (see  
1567    Fig.5.14). It was proven that the amount of time for the full inpainting follows a power law (see  
1568    Fig. 5.13a) and the accuracy starts dropping significantly for more than 5 pixels skipped at the  
1569    time (see Fig. 5.13b).



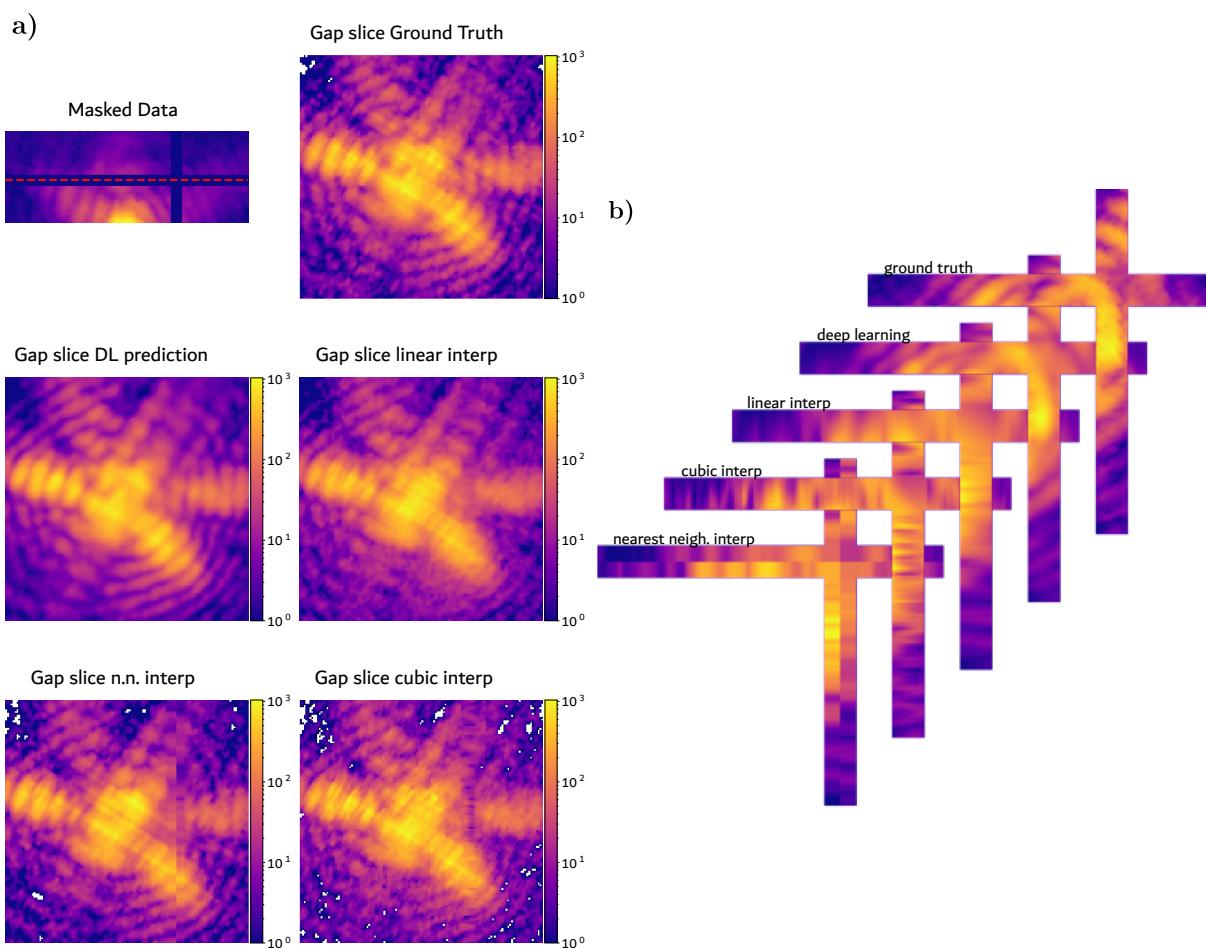
**Figure 5.13:** **Left** Full inpainting time for a 6 pixel-wide cross-shaped gap on a  $128 \times 128 \times 128$  pixel-size diffraction pattern as function of the step size, i.e. the amount of pixels skipped between patch DL predictions along the gap. **Right** Sum of the absolute errors as function of the skipped pixels.



**Figure 5.14:** Full inpainting of an experimental BCDI pattern for different amounts of skipped pixels. **a** a slice of the diffraction pattern perpendicular to the gap plane. **b** Ground truth intensity inside the gap. **c-d-e-f** In-gap prediction with 0, 4, 20 and 30 skipped pixels respectively, with corresponding execution time. Skipping 4 pixels is a good trade off between time and accuracy.

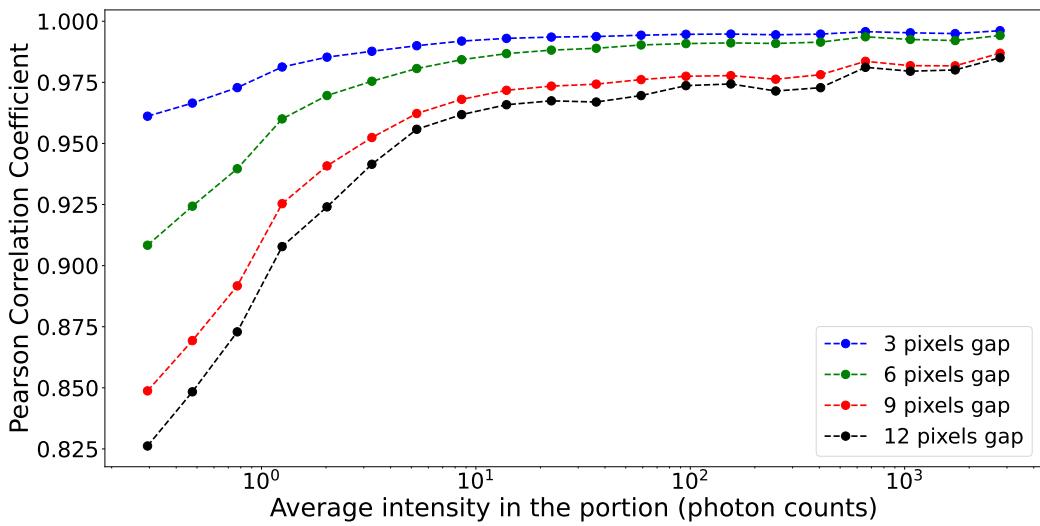
## 5.7 Performance assessment

In order to assess the DL model's performance with respect to other inpainting methods, it was tested against conventional interpolation methods. Specifically, an experimental BCIDI pattern with a 6 pixel-wide cross-shaped gap was considered and the inpainting results of the DL model with (i) linear interpolation (ii) cubic interpolation (iii) nearest-neighbor interpolation were compared. These techniques allow for a quick estimation of the intensity distribution inside the gaps but fail to recover fine features (see Figs. 5.15). In particular, it can be noticed in the *in-gap slice* (Fig. 5.15a) that linear interpolation for instance doesn't retrieve correctly the space curvature of the fringes while nearest neighbor and cubic interpolations show artifacts in correspondence of the perpendicular gap. When considering the central slice perpendicular to the gap planes (along the rocking curve dimension) one can notice even more how the DL model outperforms conventional interpolations (Fig. 5.15b).



**Figure 5.15: Comparison of DL inpainting and standard interpolations on experimental data: a)** In-slice cut of the ground truth data and the predictions for each inpainting method. The DL outperforms standard interpolations, correctly recovering the fringes. The “blur” effect on the DL predicted slice is the result of the averaging of features in presence of noise, typical of CNNs [168]. **b)** Cross-shaped gaps for the same dataset (perpendicular view with respect to a)). The DL model outperforms other methods from this perspective as well.

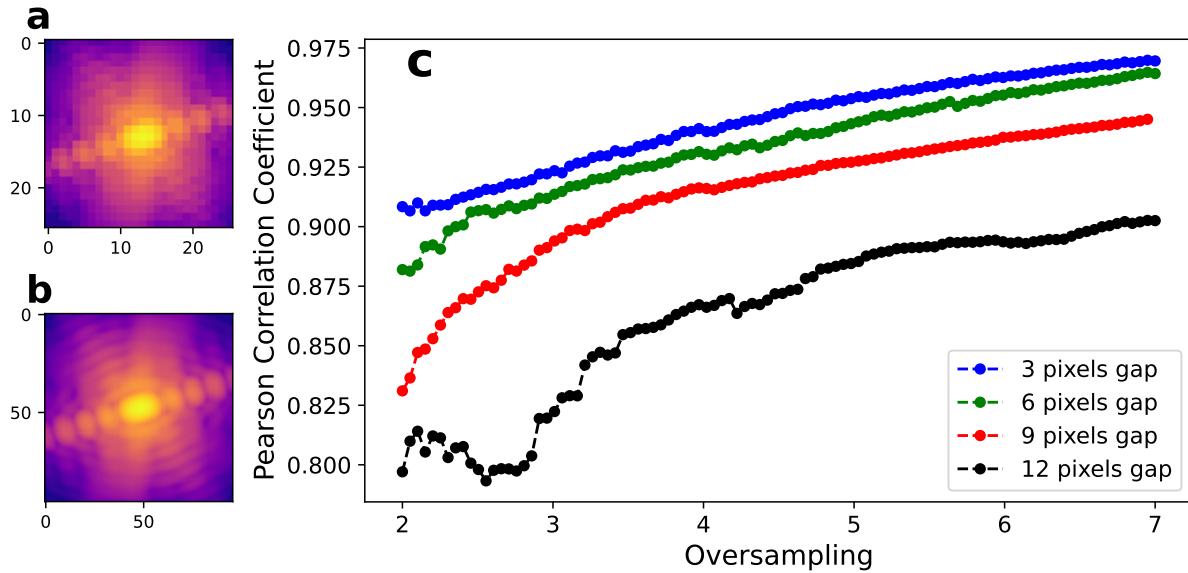
Similarly to the 2D case above, the performance of the model has been evaluated against the amount of intensity inside the sub-volume and against the oversampling ratio. The test was repeated for different gap widths, namely 3,6,9,12 pixel-wide, using vertical gaps placed in the center of each patch. For the first performance assessment a full simulated  $128 \times 128 \times 128$  pixel-size BCDI pattern was considered, and 1000 patches were randomly cropped out of it. A vertical gap was then applied in the center of each patch for different gap sizes and then computed the prediction with the corresponding DL model. The intensity (in photon counts) inside each patch was then summed and the obtained values for the 1000 samples were binned into 20 classes for better visualization. The accuracy scores, calculated with the PCC, were then averaged inside each bin class. The results are displayed in Fig. 5.16. As expected from the 2D case, better accuracy scores are obtained for portions containing larger amount of signals, where noise levels are lower and the features of the diffraction pattern are more visible. Moreover, the plot logically shows that smaller gaps are generally better recovered, but it is worth noticing that the accuracy spread across different gap sizes widens for noisy portions and narrows down as the amount of signal increases. These trends suggest that DL models are overall robust to different gap sizes especially for high intensity regions, which are the most important ones as they contribute the most during the reconstruction.



**Figure 5.16:** Accuracy scores (PCC) of the DL patching model versus the amount of signal of the patch for the four different gap sizes. As expected, the model perform better on smaller gaps and high SNR. Surprisingly, for high-signal regions the accuracy scores for large gaps approach the ones for smaller gaps.

The last test concerns the study of the accuracy for different oversampling ratios. As anticipated above for the 2D case, to carry out properly this evaluation, one should consider the same diffraction pattern extending to the same equivalent  $q$ -space value for each oversampling ratio. This in practice is done increasing the  $dq$  per pixel as decreasing the oversampling ratio, results in a smaller size of the overall BCDI pattern. In this case the same BCDI pattern was simulated for oversampling ratios spanning from 2 to 7. For each oversampling ratio, a vertical gap mask was applied to the whole BCDI array and the DL prediction was calculated with no-skip pixel (see Sec. 5.6.1). The gap was then shifted laterally, and this procedure was

repeated until the whole BCDI array was predicted using our model, thus leading to a full BCDI predicted image. The PCC was then calculated using the whole BCDI array for different oversampling ratios and model gap sizes. The results are displayed in Fig. 5.17. As expected, the predictions are more accurate for large oversampling ratios and small gap sizes (i.e., large oscillation periods relative to the gap width).



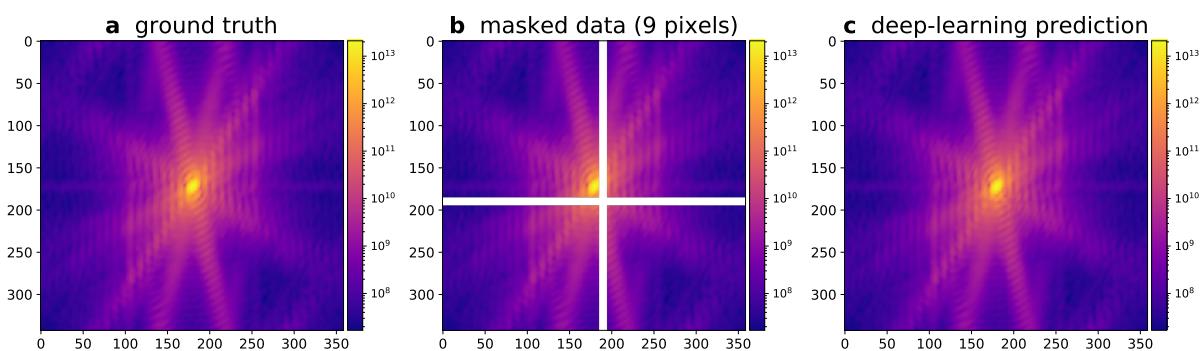
**Figure 5.17: Accuracy scores (PCC) of the DL patching model against the oversampling ratio.**  
**a)** Simulated diffraction pattern with low oversampling ratio. **b)** The same pattern simulated on a larger grid for an increased oversampling ratio. **c)** Pearson Correlation Coefficient (PCC) of the full inpainted diffraction pattern for the four different gap sizes considered. The trend overall confirms the expectations.

## 5.8 Results in real space

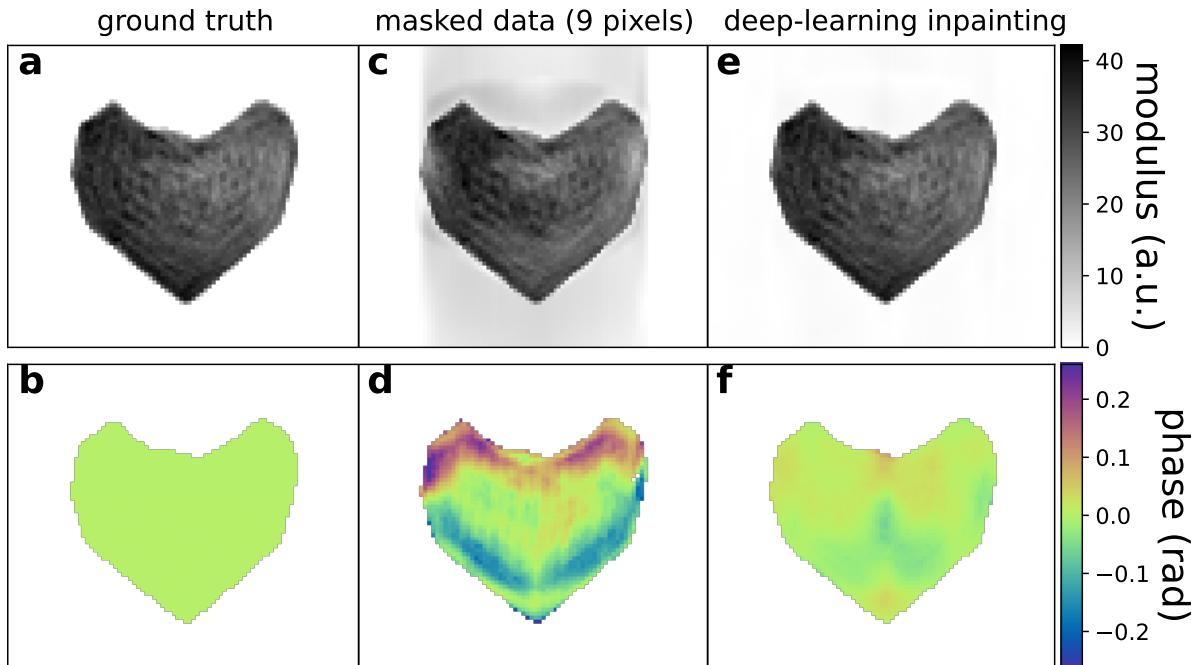
In this section the effects of DL inpainting on the reconstructed objects will be discussed for both simulated and experimental data. In particular, the assessment, both qualitatively and quantitatively, of the gap induced artifacts in the modulus, phase and strain fields of the reconstructions and their reduction thanks to the DL inpainting are presented. To carry out these analyses an experimental BCDI dataset acquired at ID01 was considered. The same dataset had already been exploited by Carnis *et al.* in 2019 for similar studies on gap-induced artifacts [144]. The dataset corresponds to the BCDI pattern around the  $(111)$  peak of a Pt particle (400 nm in size). Similarly to what the authors did, I have kept the modulus of the reconstructed object and set the real space phase to zero, making it our reference ground-truth object  $\mathbf{O}$ . This measure helps to highlight the gap induced artifacts of the phase and strain fields as there is zero-phase reference to compare the results with. The corresponding diffraction pattern was then calculated with the Discrete Fourier transform (DFT) obtaining a complex diffracted amplitude  $\mathbf{A} = \mathcal{F}(\mathbf{O})$ . A cross-shaped gap was subsequently applied to  $\mathbf{A}$  and the corresponding object  $\mathbf{O}_{gap}$  was calculated with the inverse DFT. From the same gapped  $\mathbf{A}$ , the intensity  $\mathbf{I} = |\mathbf{A}|^2$  was also “inpainted” using our DL model and corresponding object  $\mathbf{O}_{DL}$  was calculated with the inverse DFT as well, using the ground truth reciprocal space phase.

1629 The procedure was repeated for four different gap sizes (3, 6, 9, 12 px-wide) matching exactly  
 1630 the cases mentioned in the work of Carnis and coauthors. Figure 5.18 shows the projection  
 1631 along the rocking curve axis (XY slice in this case) of the ground truth, the gapped and the DL  
 1632 inpainted diffracted intensity for the 9 pixel-wide gap case.

1633 Notice that in all cases of Fig 5.18, there are vertical and horizontal lines of higher intensity  
 1634 not attributable to the typical streaks from the truncation rods. These arise from the detector  
 1635 gaps in the original dataset, which are in fact responsible for the oscillatory stripe artifacts  
 1636 visible on the object's modulus for all three cases in Fig. 5.19, ground truth included. The DL  
 1637 inpainting of these original gaps is presented later in Sec. 5.8.1.

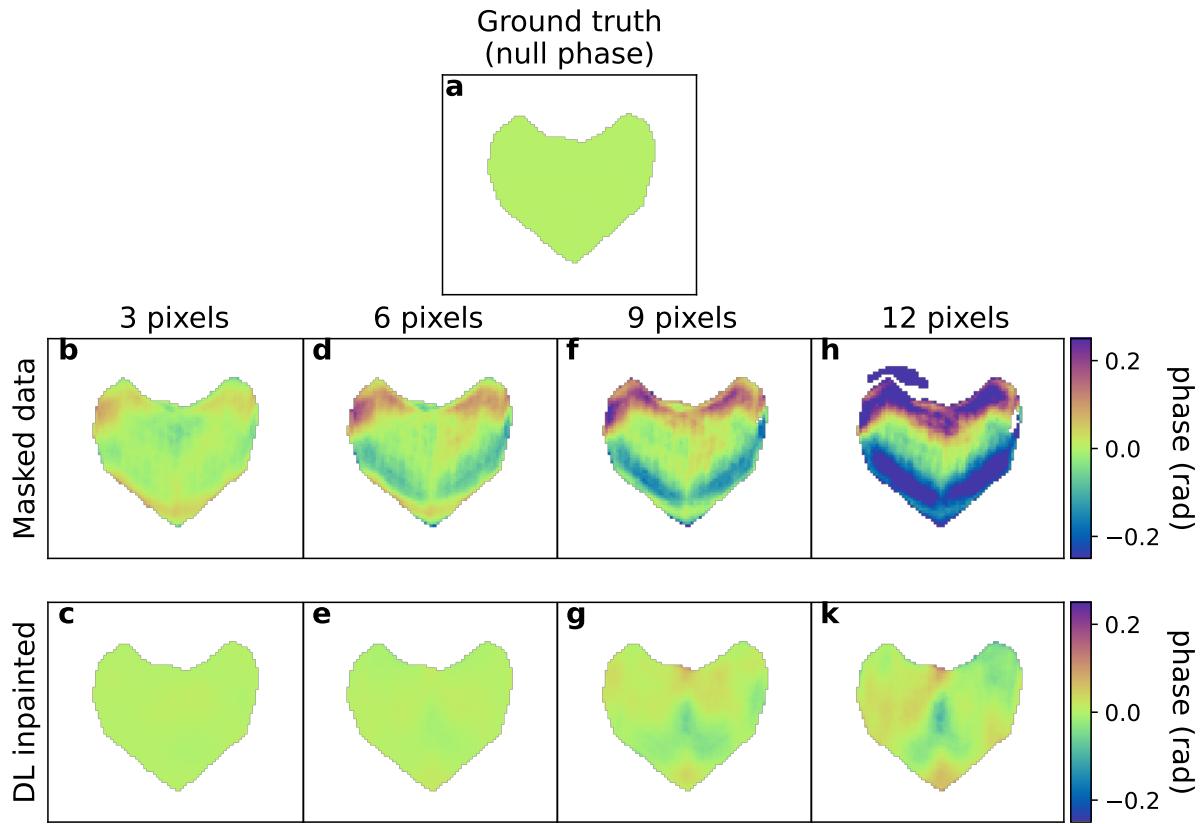


**Figure 5.18: Projections along the rocking curve axis of the studied diffraction pattern in log scale.** **a** Ground truth pattern obtained from the  $|\mathcal{F}(\mathbf{O})|^2$ . **b** Pattern with a 9 pixel-wide cross shaped gap. The position close to the center of the peak is experimentally unlikely but here it allows us to enhance the artifacts in the reconstructions. **c** Corresponding DL inpainted BCDI pattern. The presence of aliasing can be seen due to the DFT calculation as compared to the correct kinematic sum. This effect is however not relevant for the scope of these analyses.

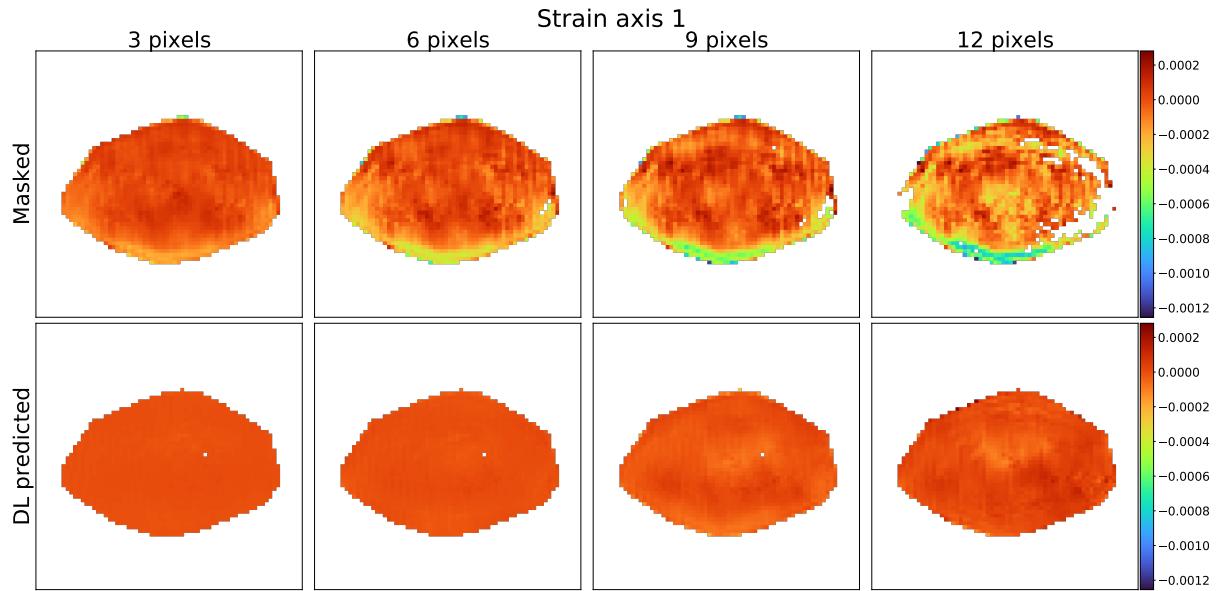


**Figure 5.19: Reconstructed objects (central slice).** a-b XY central slice of ground truth modulus and phase obtained from the reconstruction of an experimental gap-less dataset. The object's phase has been set to zero artificially after the reconstruction for better assessment of the gap-induced artifacts. c-d Modulus and phase of  $\mathbf{O}_{gap}$ , the object obtained from the inverse DFT of the diffraction pattern (with reconstructed reciprocal space phase) with a 9 px-wide cross-shaped gap placed as shown in Fig.5.18. e-f Modulus and phase of  $\mathbf{O}_{DL}$ , the object reconstructed from the DL-inpainted diffraction pattern. The phase of this last object shows reduced artifacts with respect to d.

1638     Figure 5.19 illustrates instead the central slice (XY) of the reconstructed objects for the  
 1639 three cases. It is evident that while  $\mathbf{O}_{gap}$  shows significant abnormalities in both modulus and  
 1640 phase,  $\mathbf{O}_{DL}$  is much closer to the ground truth. In particular one can notice that the gap plane,  
 1641 horizontal in the XY plane, induces artifacts along its perpendicular direction. The result is  
 1642 indeed a stripe of non-zero modulus outside the support and, most importantly, an overall  
 1643 phase variation of 0.4 radians along the vertical direction. This phase variation results in an  
 1644 error of  $\pm 7$  pm in the lattice displacement field for the 111 Pt reflection, with more intensity  
 1645 around the surface. These artifacts are particularly problematic in the cases of (electro-)catalytic  
 1646 experiments [22] or in situ gas experiments [169–173], where the particle's surface is primarily  
 1647 involved in the reaction and one could follow the process by monitoring the evolution of the  
 1648 strain in that region. As one could expect the artifacts become more severe as the gap size  
 1649 increases. Fig.5.20 depicts the phases of  $\mathbf{O}_{gap}$  and  $\mathbf{O}_{DL}$  for the different gap sizes considered in  
 1650 this study while Fig.5.21 the strain distribution in the XY plane.

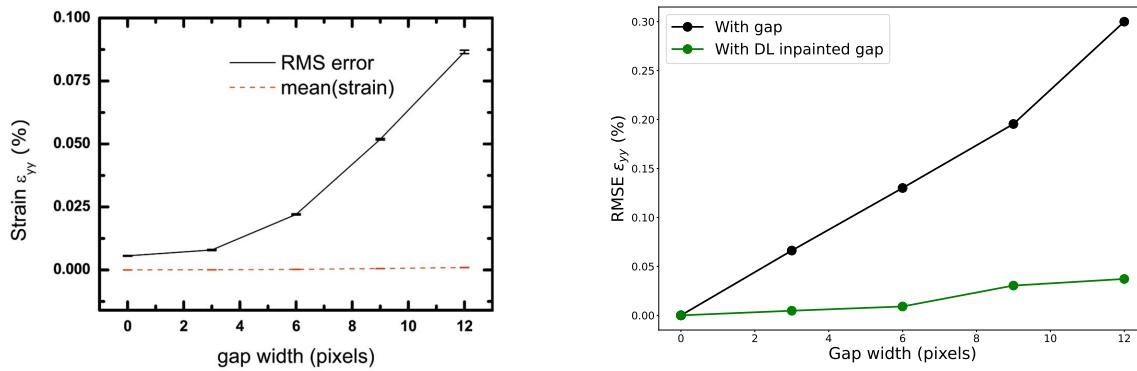


**Figure 5.20: Comparison of the phase artifacts vs gap size.** XY central slice of the phase of  $\mathbf{O}_{gap}$  for different gap sizes (b-d-f-h), and phase of the corresponding  $\mathbf{O}_{DL}$  (c-e-g-k). The comparison with the ground truth null phase (a) shows a significant increment of the artifacts in  $\mathbf{O}_{gap}$  with the gap size. On the contrary, in  $\mathbf{O}_{DL}$  artifacts are severely reduced for all gap sizes.



**Figure 5.21: Comparison of the strain vs gap size.** XZ central slice of the strain distribution for of  $\mathbf{O}_{gap}$  (first row) and  $\mathbf{O}_{DL}$  (second row) for different gap sizes.

1651 The deviation from the ground truth zero value of the retrieved strain for both  $\mathbf{O}_{gap}$  and  
 1652  $\mathbf{O}_{DL}$  can be measured with the root mean squared error (RMSE) across all the different gap sizes.  
 1653 This calculation was already proposed in the aforementioned work of Carnis and coauthors in  
 1654 which the results were plotted in Fig.4. We have reproduced a similar figure adding the results  
 1655 of our DL model (Fig.5.22). The trend of the strain RMSE resembles the curve shown in [144],  
 1656 increasing significantly with the gap size while the DL equivalent curve lies below, dampening  
 1657 the error by approximately a factor 5.



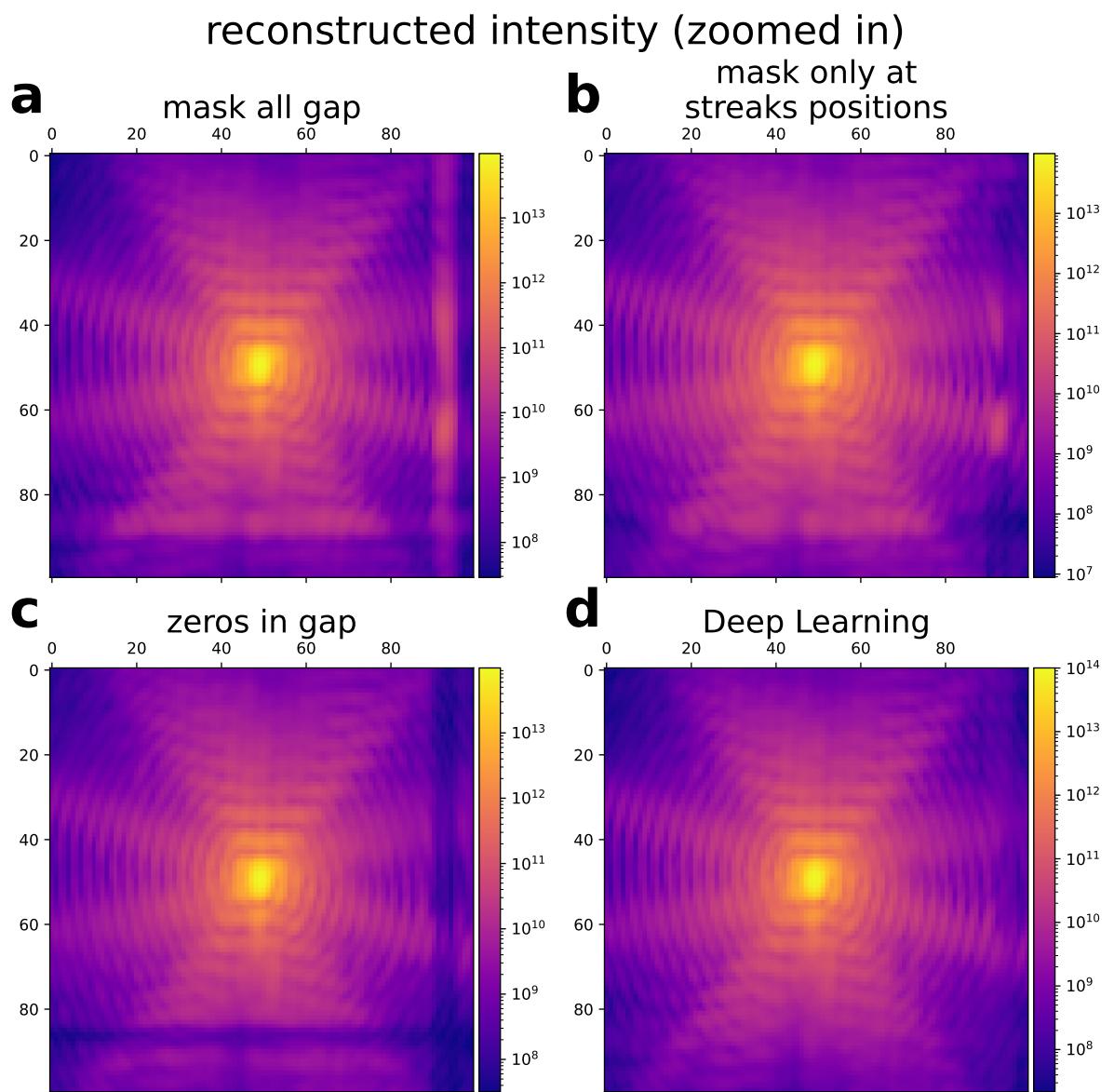
**Figure 5.22: RMSE of the strain field versus gap size** (**Left**) The plot realized by Carnis *et al.* in [144] depicts the RMSE of the  $\varepsilon_{yy}$  strain component for the objects reconstructed from gapped diffraction patterns with different gap-size. The mean strain around zero corresponds to the null object's phase set artificially. (**Right**) Plot of the RMSE of the  $\varepsilon_{yy}$  strain component versus same gap-size as (**Left**) for gapped (blue line) and DL inpainted (orange line) diffraction patterns.

### 1658 5.8.1 DL inpainting for high resolution BCDI

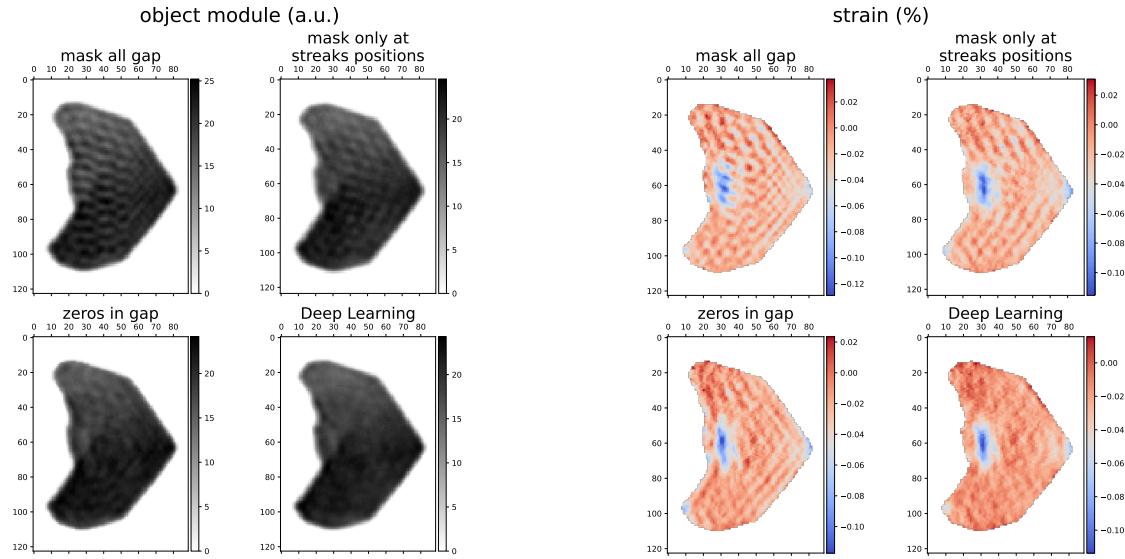
1659 As anticipated in the introduction to the chapter, the most prominent case in which a BCDI  
 1660 datasets is affected by a detector gap is a “high-resolution” dataset. The acquired data here  
 1661 extend for large  $q$  ranges in all directions resulting in parts of the diffracted signal that cross a  
 1662 region on the detector with a vertical or horizontal gap, hence the need for gap-inpainting. All  
 1663 the more so, it is convenient to use a patching approach since treating the full volume would  
 1664 be computationally too expensive. Moreover, any binning or interpolation to smaller sizes will  
 1665 induce information loss, and is not advised.

1666 An example of high-resolution BCDI dataset of this type is the one we have used so far from  
 1667 the work of Carnis and coauthors. The original dataset is indeed a large  $(256 \times 300 \times 300$   
 1668 pixels) array that contains a cross-shaped, 6 pixel-wide gap. Here we show how the artifacts  
 1669 can change depending on the type of masking of the gaps is chosen during the phasing and  
 1670 how our DL model can outperform these methods. A common approach, when using PyNX  
 1671 software, is to mask the gap such that those pixels don't contribute during the phasing and  
 1672 are left free to evolve (**a**). Moreover, one could mask only near the intensity streaks affected  
 1673 by the gap (**b**) or simply leave the gap with zeros and remove the contribution of the gap  
 1674 voxels during the whole phasing (**c**). These strategies have been used during the phasing of the  
 1675 cited Pt diffraction pattern and the results in object space compared with that obtained from  
 1676 the DL inpainted pattern. The results, illustrated in Figs. 5.23 - 5.24, show that the amount

1677 of oscillatory artifacts progressively decreases as we go from method **a** to **c**, proving the DL  
 1678 inpainting to be the optimal method among them.



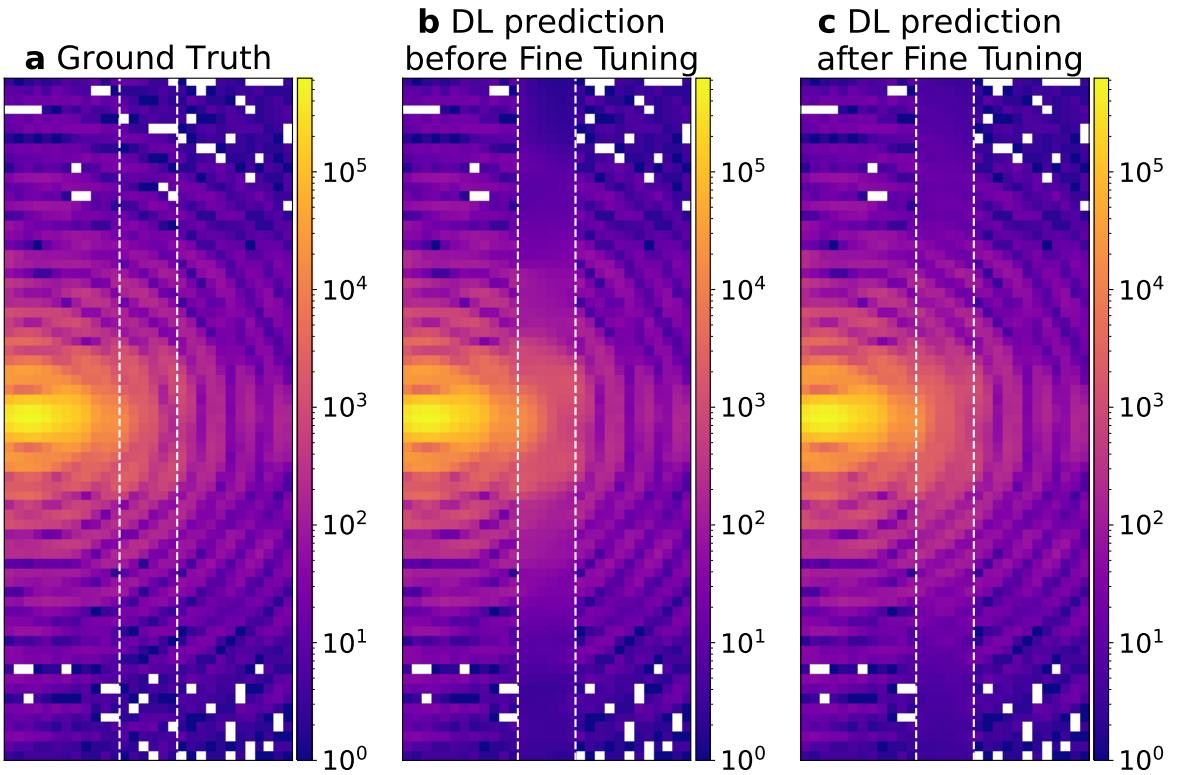
**Figure 5.23:** Zoom on the projection along the rocking curve axis of the Pt BCDI pattern in Fig.5.18 calculated from the reconstructed object obtained with PyNX software using **(a)** a mask on the gaps, **(b)** a mask on the streaks only, leaving zeros inside the gaps **(c)** and inpainting the gaps with the DL model **(d)**. Inside the regions where the mask was applied the intensity is free to vary throughout the PR iterations.



**Figure 5.24:** Modulus (Left) and strain (Right) of the object's reconstructions for each case mentioned above in Fig. 5.23. The oscillatory artifacts are smallest for the object obtained after DL inpainting (bottom right corner on both left and right images).

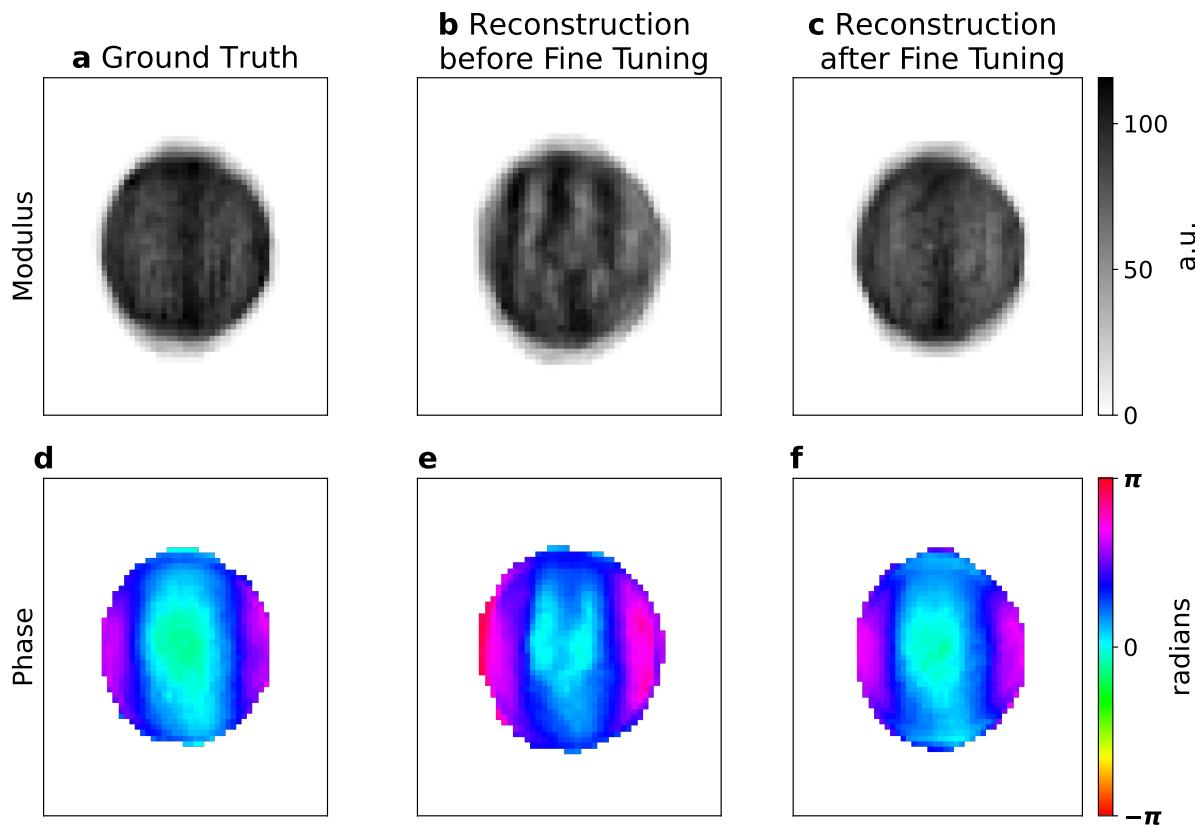
## 1679 5.9 Fine-tuning

1680 For those cases in which the DL model does not yield satisfactory results when inpainting a new  
 1681 experimental BCDI pattern a fine-tuning of the model was conceived in order to improve the  
 1682 accuracy of the prediction. This fine-tuning is enabled by the patching approach as it consists  
 1683 of a secondary short training of the general model on a small dataset made of portions extracted  
 1684 from the new BCDI pattern to be inpainted. In particular, after loading the gap affected BCDI  
 1685 pattern, 2000 portions were randomly cropped out of it, paying attention not to include the  
 1686 gap region. The model has been trained for the corresponding gap width for 2 epochs with  
 1687 a learning rate of  $10^{-5}$  and batches of 32 images ( $\sim 2$  mins on a NVIDIA Tesla V100-SXM2  
 1688 GPU with 32GB RAM). Biasing the model to fit the features of that specific diffraction pattern  
 1689 (oversampling ratio, particle shape, noise level, fringes shape) better results can be obtained  
 1690 on the real gap. An example is shown in Fig.5.9. There, the general DL model was not able to  
 1691 predict the fringes with the correct periodicity inside the gap. After the fine-tuning instead,  
 1692 the model properly recovers the fringes improving the accuracy. This fine-tuning technique is  
 1693 a further example of the advantage of using a patching approach and its usage depends on the  
 1694 user judgement on the quality of the general model inpainting.



**Figure 5.25: Example of improved accuracy after fine-tuning of the DL model on experimental data.** a) The ground truth portion of the central slice of the BCDI experimental diffraction pattern. The dashed line shows the position of the artificial 9 px-wide vertical gap applied for this test. b) Same slice for the DL inpainted pattern *before* fine-tuning. The fringes' periodicity is not recovered correctly. c) Results *after* fine-tuning. The fringes are now better recovered after 2 epochs of fine-tuning on a dataset of 6400 patches cropped out of the same diffraction pattern (except the masked region). The fine-tuning training took 110 seconds.

1695        The effect of the fine-tuning can also be assessed on the reconstructed object. Here, the  
 1696        diffraction patterns showed in Fig.5.25 has been phased with PyNX and the results are compared  
 1697        in Fig.5.26. For the phase retrieval, in all the three cases a single run of 400 HI0 + 1000 RAAR +  
 1698        300 ER iterations has been performed starting from the autocorrelation of the object and with  
 1699        the shrinkwrap threshold of 0.2. As expected, already from visual inspection, the modulus and  
 1700        phase of the object reconstructed from BCDI pattern inpainted after the fine-tuning are more  
 1701        similar to the ground truth ones. On the contrary, the one obtained before the fine-tuning  
 1702        shows some oscillatory artifacts caused by the not correctly restored fringes.



**Figure 5.26: Reconstructed object before and after fine-tuning of the DL model** From the diffraction pattern in Fig. 5.25, **a - d** Modulus and phase of the ground truth object. **b - e** and **c - f** modulus and phase of the object reconstructed from the DL inpainted diffraction pattern *before* and *after* fine-tuning respectively. The not well recovered fringes of Fig. 5.25**b** produce artifacts in the reconstructed object that are eliminated with the inpainting of the fine-tuned DL model.

## 1703 5.10 Conclusions

1704 A novel DL model for the inpainting of BCDI patterns with detectors' gaps has been presented  
 1705 and tested for different experimental circumstances. The "patching approach" has proven to be  
 1706 adequate for BCDI for different reasons here summarized.

- 1707 • Due to the oversampling condition, the BCDI patterns exhibit spatially regular and well-  
 1708 separated oscillations. This local "redundancy" in detector space can be exploited by  
 1709 isolating smaller sub-volumes of the diffraction pattern (*patches*). In the presence of gaps,  
 1710 the missing signal within each patch can then be reconstructed from neighboring voxels,  
 1711 with the reconstruction accuracy depending on the relative size of the gap compared to  
 1712 the patch.
- 1713 • The patching approach naturally allows for larger training datasets, faster training  
 1714 compared to full-image direct inpainting approaches. Moreover, no lossy resizing is  
 1715 needed to adapt the whole BCDI pattern to the DL model size, as in full-image direct  
 1716 inpainting.

- 
- 1717 • DL inpainted experimental BCDI patterns have shown reduced artifacts in the recon-  
1718 structed objects for different gap-sizes, thus confirming improved reliability of the recon-  
1719 structions.
- 1720 • Unlike full-image direct inpainting models the patching approach enables a model fine-  
1721 tuning on each new specific gap-affected dataset to be restored. This additional training  
1722 can significantly improve the inpainting accuracy, at the cost of only a few extra minutes  
1723 of computation.

1724 The presented DL model is planned to be implemented in the PyNX package in the near  
1725 future, such that it can be employed by a broader BCDI user community in standard data  
1726 analysis pipelines. In these regards, it is worth mentioning that for an unbiased reconstruction,  
1727 during the last iterations PR iterations, the inpainted pixels would need to be “freed” with a  
1728 proper mask. In such a way, when the PR algorithm is near convergence, the data-fidelity is  
1729 preserved.

1730 Future perspective can envision the extension of this patching approach to other CDI  
1731 techniques. In the X-ray imaging community the detector gaps are in fact a common problem,  
1732 as well as other limiting elements (beam-stops, defective pixels, overexposure, etc.). Given  
1733 high-intensity signal covered by beam-stops and the higher accuracy of the DL model on  
1734 bright patches, one could infer that CDI techniques like small-angle CDI and others in forward  
1735 geometry could benefit from the presented DL inpainting approach.



---

## 1737 DEEP LEARNING FOR PHASE RETRIEVAL

1738 We enter now the core topic of the thesis. Most of the efforts during this PhD have been  
 1739 dedicated to the study of the Phase Problem for Bragg Coherent Diffraction Imaging using  
 1740 DL based approaches. Here I will discuss the main steps of this journey, starting off from the  
 1741 analysis of the most relevant works in literature and concluding with some comments on the  
 1742 final version of a DL model for highly strained particles. The latter has become the subject  
 1743 of an article, currently submitted, entitled “*Phase Retrieval of Highly Strained Bragg Coherent*  
*1744 Diffraction Patterns with Supervised Convolutional Neural Network*” [174]. The process that led to  
 1745 the final version of the model will be unraveled, and particular attention will be given to  
 1746 elucidating the key steps and the critical issues encountered along the way. The structure of  
 1747 the chapter can be summarized as follows:

- 1748 • **State of the art and Reciprocal space phase phasing:** introduces the works on DL for  
 1749 phase retrieval in BCDI and discusses the novel concept of the prediction, using DL, of  
 1750 the reciprocal space phase (RSP) lost during the measurement.
- 1751 • **2D case and Weighted Coherent Average loss function:** presents the preliminary  
 1752 studies on the 2D case, for low-strain and high-strain simulated particles. Introduces the  
 1753 novel “Weighted Coherent Average” loss function, designed specifically for the prediction  
 1754 of complex phases.
- 1755 • **3D case - Patching approach:** discusses a patching-method for the low-strain and  
 1756 high-strain cases in 3D.
- 1757 • **Final 3D model:** presents the final DL model for the prediction of the RSP of highly  
 1758 strained 3D BCDI patterns. Results on simulated and experimental data are shown as  
 1759 well as the combination of DL prediction and iterative algorithms for refinement.
- 1760 • **Performance assessment:** the final DL model and standard algorithms are tested on  
 1761 simulated data with different strain distributions to assess the benefits of the DL approach.

## 1762 6.1 State of the art

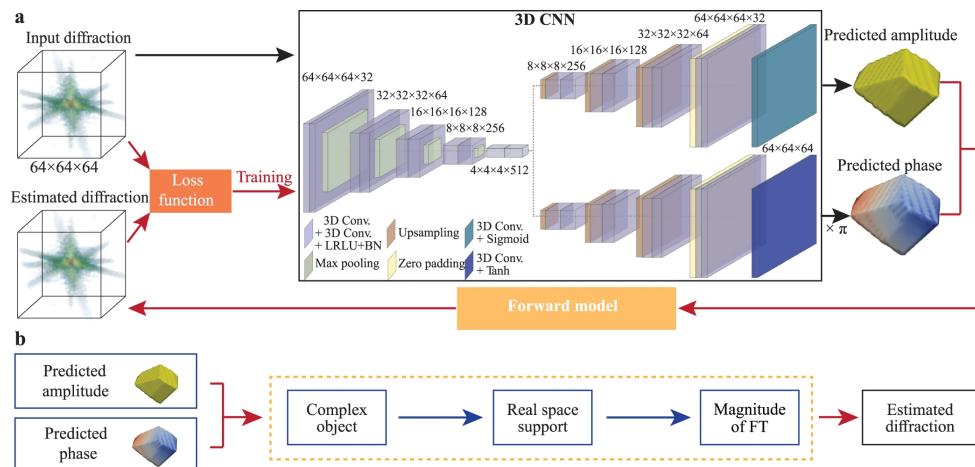
1763 In this paragraph I will focus on the state of the art for what concerns the Phase Retrieval of  
1764 BCDI diffraction patterns with deep-learning, tensor-computation and automatic differentiation  
1765 methods. Conventional phase retrieval iterative algorithms are discussed in the introduction  
1766 chapter as well as other approaches.

1767  
1768 Given the relatively new development of neural networks and more specifically even more  
1769 recent for BCDI phase retrieval, I will try to give a chronological overview of the main works  
1770 in the literature pointing out strengths and weaknesses. The use of deep neural networks for  
1771 inverse problems in imaging was proven to be successful in lensless computational imaging by  
1772 Sinha in 2017 [175] and holography by Rivenson [176]. However, the first work pioneering  
1773 the field for BCDI is “Real-time coherent diffraction inversion using deep generative networks”  
1774 published by Cherukara *et. al* in 2018 [177]. The paper presents two CNNs for the phase  
1775 retrieval of small ( $32 \times 32$  pixels) 2D simulated BCDI patterns, one predicting the support  
1776 and the other the phase. A U-Net like architecture with encoder-decoder was implemented,  
1777 and the model was trained for 10 epochs in a supervised fashion with a cross-entropy loss  
1778 function (see Appendix). The results showed an excellent agreement between prediction and  
1779 ground truth also in presence of relatively strong phases. The potential of this new approach  
1780 for phase retrieval becomes immediately clear when considering the drastic reduction of  
1781 computational time and resources needed for the model inference. Once the model is trained,  
1782 the reconstruction can be obtained within a few milliseconds on a desktop machine.

1783 In 2020 Scheinker and Pokharel proposed another approach [178] that employs a CNN model  
1784 for 3D diffraction patterns. The fundamental difference is that the object’s support was defined  
1785 by its surface only, as it is assumed to be *compact* and *homogeneous* inside. Moreover, the surface  
1786 was parametrized by spherical harmonics and the DL model was trained to predict 28 of the first  
1787 even coefficients of the spherical harmonics. The model architecture was therefore essentially  
1788 different since, while the encoder is just transposed to a 3D one, the decoder is replaced by  
1789 a flattening and dense layer with 28 different classes as output. The model showed good  
1790 performance on both simulated and experimental data, marking the first DL-based approach  
1791 capable of real 3D BCDI phase retrieval. In the same year, Wu and coauthors, [179], opted  
1792 for an architecture made of a single encoder and two identical decoders for the prediction  
1793 of amplitude and phase of single crystals from the central slice of the BCDI pattern. They  
1794 conducted the study on simulated data and tested it on one experimental case as well. What is  
1795 evident from their work is the winning combination of DL prediction and iterative refinement.  
1796 The speed and generalization capabilities of the CNN allows for fast and good estimations of the  
1797 object’s support and phase. In addition, the precise and well established iterative methods can  
1798 bring this initial guess to a more polished and accurate solution in fewer cycles than without  
1799 DL prediction. This successful combined approach has been later adopted in other works, ours  
1800 included.

1801 In 2021 two important works were published. First, Chan *et al.* in [180] extended the  
1802 encoder/2-decoders architecture to the 3D case. In their work they first created a “physics-  
1803 informed” training set obtained building particles by clipping planes from a cubic FCC structure  
1804 of atomic positions, relaxing them with LAMMPS software for molecular dynamics and com-  
1805 puting the BCDI pattern around the (111) Bragg peak. The procedure is very similar to the  
1806 one adopted by Lim *et al.* in [166] and described above in Section 5.4.1. Training the CNN

on a restricted set, of such created BCDI patterns, biases the predictions towards physically meaningful particles. Moreover, it is interesting to notice that the training of the model was conducted in a sort of unsupervised fashion as the loss function calculates the differences between the target diffracted intensity and the intensity obtained by the squared modulus of the discrete Fourier transform of the predicted complex object. Although the authors managed to successfully test their model on an experimental BCDI pattern, the small size ( $32 \times 32 \times 32$  pixels) of the images accepted by the CNN was not yet enough for proper experimental use. The work of Wu *et al.* [181] published in the same year, lifted the size to 64 pixel-sided cubes, enabling their model to be tested on several experimental cases. Their CNN model maintained the encoder/2-decoders architecture for a simultaneous prediction of the object's amplitude and phase and explores for the first time the unsupervised training for refinement as well. The authors claimed that this approach is able to achieve better reconstruction quality with respect to current state-of-the-art iterative algorithms in use. The year after, Yao and coauthors published AutoPhaseNN [182], again an encoder/2-decoders architecture that completely trained in an unsupervised manner. This approach is beneficial as it doesn't require datasets labeled with a ground truth, which means that experimental data can be directly used in the training set. Another advantage is that it relaxes the limitation of simulating a sufficiently diverse population of samples, capable of constituting a comprehensive distribution of real cases. AutoPhaseNN was trained to predict an object the diffracted intensity of which matches the observed one according to a normalized Mean Absolute Error metric. The model showed to work on simulated data as well as on experimental data and once more the winning method lies in the combination of DL prediction and iterative refinement. AutoPhaseNN has marked a milestone in the BCDI data analysis, attaining 10X to 100X phase retrieval speed up with reduced efforts for the model training.



**Figure 6.1: Schematic of the AutoPhaseNN model for unsupervised BCDI Phase Retrieval.** **a** The convolutional neural network is divided into an encoder and two parallel decoders for object's modulus and phase respectively. **b** The loss function is calculated in reciprocal space, between the observed and calculated intensities, therefore enabling unsupervised training. Adapted from [182]

Although of different nature, it is worth mentioning the work of Zhuang and coauthors [183] in which two CNNs are used in the “deep image prior” (DIP) framework. DIP [133] typically implies the use of a CNN for an enhanced representation of an image, often to solve inverse problems like super-resolution, de-noising and inpainting. However, it differs from classical deep learning as there is no training dataset but a fit of the target problem exploiting

the parameters of the convolutional layers and the efficient gradient descent provided by the automatic differentiation. In their work, Zhuang *et al.* formulated the more general far-field phase retrieval problem as an optimization problem and considered the phase symmetries that affect this class of solutions [184] (see Introduction chapter). Their work employs two DIPs, one for the modulus and one for the phase, and successfully manages to reconstruct simulated objects even in the presence of strong phases. The last interesting contribution is the work of Yu and *et al.* [185]. In this paper the authors proposed a DL model that computes complex convolutions, handling real and imaginary parts of the complex tensor in a single passage through the convolutional block. Complex convolutional layers are claimed to be better at preserving the physical connection between real and imaginary parts inside the complex object. Moreover, the authors made use of *skip connections* between encoder and decoder to enhance the training. This is rather peculiar as this kind of residual links are typically used, in convolutional encoder-decoder networks, for tasks in which the input and output images are visually similar (i.e. segmentation, denoising, inpainting), thus, where it is more evident the information flow from the two blocks of the network. The model was used for the phase retrieval of experimental 2D diffraction patterns, for which an unsupervised refinement was used as well.

Before proceeding with our study, Table 6.1 summarizes the key features of the works from the two leading BCDI research groups at Brookhaven and Argonne National Laboratories, highlighting similarities and differences to guide the development of our model.

	Architecture	Last Activation Layer	Loss Function	Refinement
Cherukara - 2018 [177]	Two different UNets	Sigmoids	Cross Entropy	-
Wu - 2020 [179]	Encoder / 2 Decoders	ReLU	MSE on mod and phase + PCC on magnitudes	Iterative
Chan - 2021 [180]	Encoder / 2 Decoders	ReLU	MAE on normalized magnitudes	Automatic Differentiation
Wu - 2021 [181]	Encoder / 2 Decoders	LeakyReLU	MSE on mod and phase + PCC on magnitudes	Transfer learning + unsupervised training
Yao - 2022 [182]	Encoder / 2 Decoders	Sigmoid and Tanh	MAE on normalized magnitudes	Iterative (50 ER)
Yu - 2024 [185]	Complex encoder-decoder + skip connections	ReLU	MAE on real + MAE on imaginary	Transfer learning + unsupervised training

**Table 6.1:** Comparison of deep learning-based phase retrieval approaches.

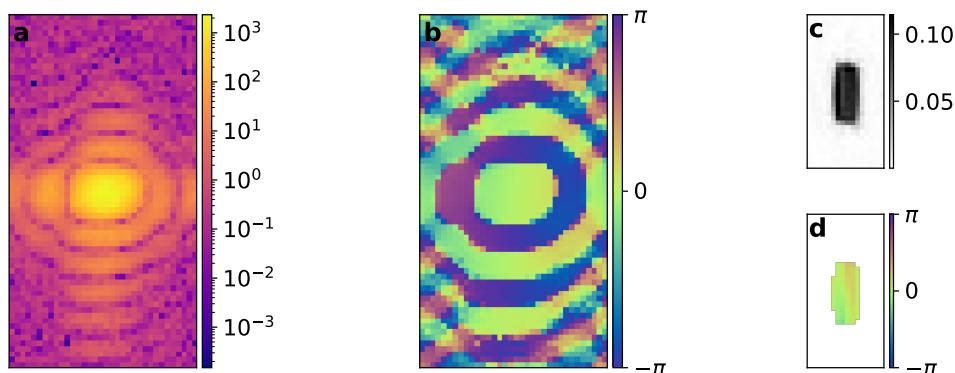
First, it is interesting to notice that the architecture's choice, from treating the object's modulus and phase separately with two different detached networks, moved over the years to a single "standard" U-Net that accounts for the complex nature of the data. Second, I noticed that the choice of the last activation layers, which are the ones producing the modulus and phase outputs, in their final value range, is not uniform throughout the articles. ReLU and sigmoid activations produce strictly non-negative outputs, making them naturally suited for quantities that are inherently positive, such as the modulus. In contrast, LeakyReLU and Tanh can generate negative values, which makes them valid choices for representing complex phases. However, the practical impact of this choice appears limited: in some cases, models can predict correct moduli using LeakyReLU and correct phases using ReLU or sigmoid. This can be explained by the fact that a global offset in the phase, shifting the entire range to positive values, does not change the physical solution. Consequently, a ReLU can produce a valid phase array, simply offset by a constant, and similarly for the sigmoid, provided the phase range fits within the activation's output limits.

1870

1871 One of the most important components of the model is the loss function, as its gradients  
 1872 regulate the model's parameters, hence the prediction. The first work that employs a cross  
 1873 entropy loss, normally used for classification tasks, whereas other works opt for Mean Absolute  
 1874 Error (MAE) and Mean Squared Error (MSE), which are standard for regression tasks and  
 1875 Pearson Correlation Coefficient (PCC). Typically, when the loss is calculated between intensities  
 1876 the MAE and the PCC are used as they are more suitable for the high dynamic range of  
 1877 the diffraction patterns. MSE in fact, “would overly de-emphasize errors in mid-intensity  
 1878 regions of the images” [180]. Lastly, I have listed the different ways used to refine the DL  
 1879 predictions. Here we can notice that GPU accelerated gradient descent methods have been  
 1880 used to replace of conventional iterative algorithms. The unsupervised training allows to  
 1881 easily switch from inference to refinement using the same model in the same GPU optimized  
 1882 computing environment guaranteed by machine learning libraries like PyTorch and Tensorflow.

## 1883 6.2 Reciprocal space phasing

1884 Taking inspiration from these works, but significantly changing the perspective, it was de-  
 1885 cided to predict the “reciprocal space” phase (RSP) that is lost during the measurement of  
 1886 the BCDI pattern rather than the complex object in real space. The main, intuitive, reason  
 1887 behind this choice is the visual similarity between the morphology of the diffraction pattern  
 1888 and its corresponding RSP. Furthermore, it is common that many samples studied with BCDI  
 1889 have facets that happen to be, to some degree, parallel with each other, thus interfering like a  
 1890 single slit, with the typical fringes of intensity that correspond to constructive interference,  
 1891 interspersed with dark regions arising from destructive interference. In these specific cases,  
 1892 the RSP shows a regular pattern in which there is always a  $\pi$  shift between two crests of the  
 1893 fringes as shown in Fig.6.2. Once the RSP is retrieved, one can then recompose the full complex  
 1894 diffracted wave-function and obtain the complex object via inverse Fourier transform.  
 1895



**Figure 6.2:** Central slice of a typical BCDI pattern (a) with the corresponding RSP (b) obtained after a successful reconstruction of the object (modulus and phase in c - d respectively). It highlights the structural similarity between the diffracted intensity in logarithmic scale and the RSP. Moreover, one can notice that in this case of low strain faceted particle, the RSP varies regularly between 0 and  $\pi$  (or  $-\pi$ ) in correspondence of the intensity fringes.

Given this “simple” law of constructive-destructive interferences, we hypothesized the possibility to predict patches of this RSP given a portion of diffraction pattern and then, similarly to the inpainting case, stitch them together and obtain the full RSP. This entails a number of complications related to the so-called phase symmetries (see Section 3.2) that I have encountered during the development of the algorithms and that will be discussed in the next sections.

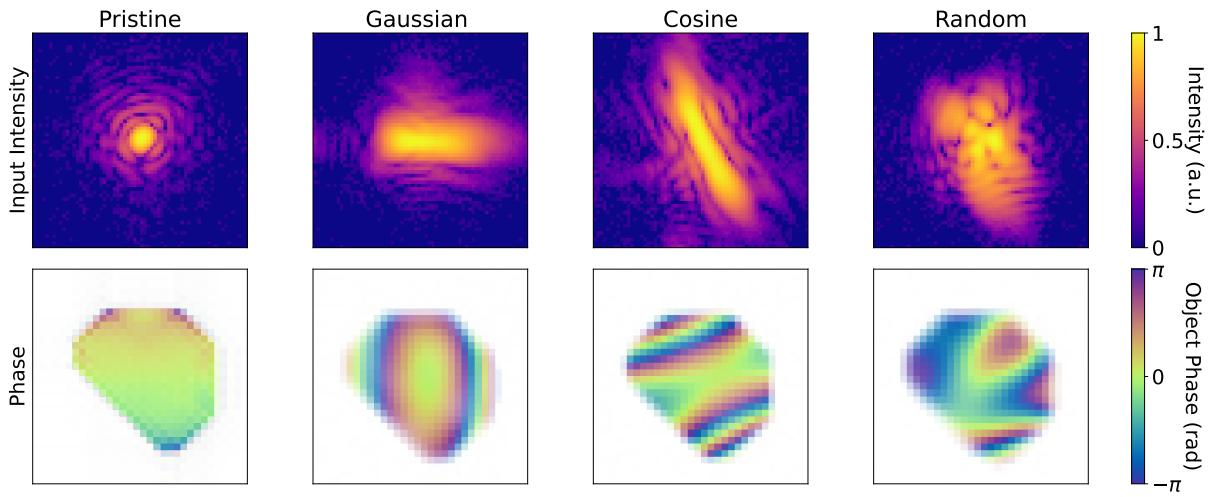
Ultimately, the goal of this DL model for phasing is to facilitate the reconstruction of highly strained particles. While other works in literature have mostly leveraged the gain in computing time, here the model aims at tackling those reconstructions for which conventional algorithms struggle to find convergence because of the high strain in the particle. However, in this case, the aforementioned RSP  $\pi$ -shifts in between two fringes is much more complicated since the strong and extended displacement fields inside the crystal alter the Bragg peak, merging and spreading the fringes into an irregularly distributed intensity pattern (see Fig 2.6).

## 6.3 Dataset creation

I have trained the model in a supervised manner, meaning, in this case, that the training was always conducted on simulated data only, as the RSP is never experimentally detectable. For this reason, I have simulated the training dataset following the same procedure described in Sections 5.3.1 and 5.4.1 for the 2D and 3D cases, respectively. However, in this case the simulated RSP was used as the ground truth label for training.

I will anticipate here that for the high strain case I created a dedicated training set simulating the strain by applying an artificial “strong” phase to the particles. In order to have a diverse population of strain distributions I have simulated each object’s phase using different functions and parameters, namely: with the sum of two Gaussian functions, with the sum of two cosine functions and using a correlated Gaussian random field [160]. In each case, centers, amplitudes, variances, frequencies, and correlation lengths were randomly chosen to ensure a phase variation within the particle ranging between  $\pi$  and  $5\pi$ . By doing this, strongly distorted BCDI patterns, similar to experimental high-strain ones, were obtained. In particular, the two Gaussian and cosine functions phase can closely emulate the effect of the substrate induced strain inside Winterbottom particles. (see Fig.6.3)

Similarly to the inpainting case, the BCDI patterns have been transformed into logarithmic scale and normalized between 0 and 1. Batches of 32 images at the time were used.



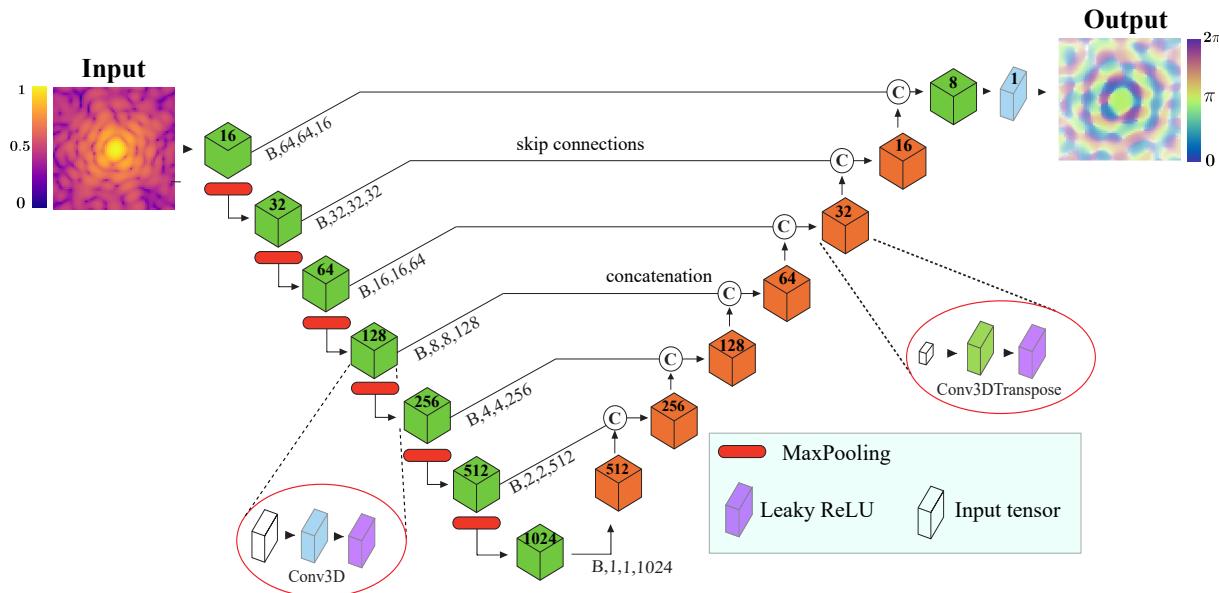
**Figure 6.3: Different simulated strain distributions.** Examples of each of the three adopted methods to simulate the high-strain. From the left, non-strained diffraction pattern and the corresponding pristine particle (central slice); the diffraction pattern obtained from the same particle with a phase built with two Gaussian functions, and relative object’s phase slice; the diffraction pattern when the phase is simulated with two cosine functions and corresponding object’s phase slice; the diffraction pattern obtained from the same particle with an applied gaussian-correlated random phase field. It is interesting to notice that for the first two objects, the variation of the phase are more pronounced along a specific direction, the same for which the diffraction pattern show an elongation of the Bragg peak, causing a visual “fusion” of the fringes. This phenomenon is often observed experimentally for particles grown on crystalline substrates (see Fig.6.40) [95, 100, 186].

## 1928 6.4 2D case low strain

1929 Similar to the approach for the inpainting case, I have first conducted some preliminary studies  
 1930 in 2D, on noise-less low strain data. Here I will briefly show the model’s architecture, the loss  
 1931 function and the results.

### 1932 6.4.1 Model structure

1933 The architecture that I used has a U-Net like structure with an encoder and a decoder. The  
 1934 encoder is composed of six convolutional blocks through which the input diffracted intensity is  
 1935 progressively reduced from the 64 pixel-side squares to a 1D flattened vector. Each convolutional  
 1936 block is composed of a convolutional layer, a LeakyReLU activation function and a MaxPooling  
 1937 layer that halves the feature’s map dimensions (see Fig. 6.4 for details). At the end of the encoder  
 1938 the so-called bottleneck, composed of a convolutional layer followed by a LeakyReLU activation,  
 1939 processes the feature map before passing it to the decoder which by means of transposed  
 1940 convolutions (with double stride to upsample each image) and LeakyReLU activations, brings  
 1941 back the feature map to the input’s size. Skip connections between encoder and decoder blocks  
 1942 are employed as well. The output tensor is the result of a last single-channeled convolutional  
 1943 layer with no activation function. In this way we let the model predict unbounded tensors to  
 1944 account for the phase symmetries.



**Figure 6.4: U-Net like architecture of the 2D model for phase retrieval.** The log-scaled, normalized 2D diffraction pattern is first passed through the encoder, composed of six convolutional blocks (green cubes) in each of which the feature map is filtered by a single stride convolutional layer with a  $3 \times 3$  pixels kernel. The number of filters, shown in bold inside each block, scales from 16 to 512. A LeakyReLU (alpha = 0.2) activates the output of each convolutional layer. A MaxPooling layer halves the size of the feature map down to 1D. The bottleneck, built with another convolutional block with 1024 filters, passes the feature map to the decoder. This last, mirroring the encoder, is composed of an equivalent number of de-convolutional blocks (orange cubes) in each of which the feature map is filtered and up-sampled by a two-strides de-convolutional layer with a  $3 \times 3$  pixels kernel and same LeakyReLU activation. Skip connections are employed as well between encoder and decoder. Copies of the feature map after each convolutional block are concatenated with same size feature maps in the decoder before transposed convolutions. A last convolutional layer with no activation function produces the RSP prediction. The total number of trainable parameters is of the order of 6.8 million.

#### 1945 6.4.2 Loss function

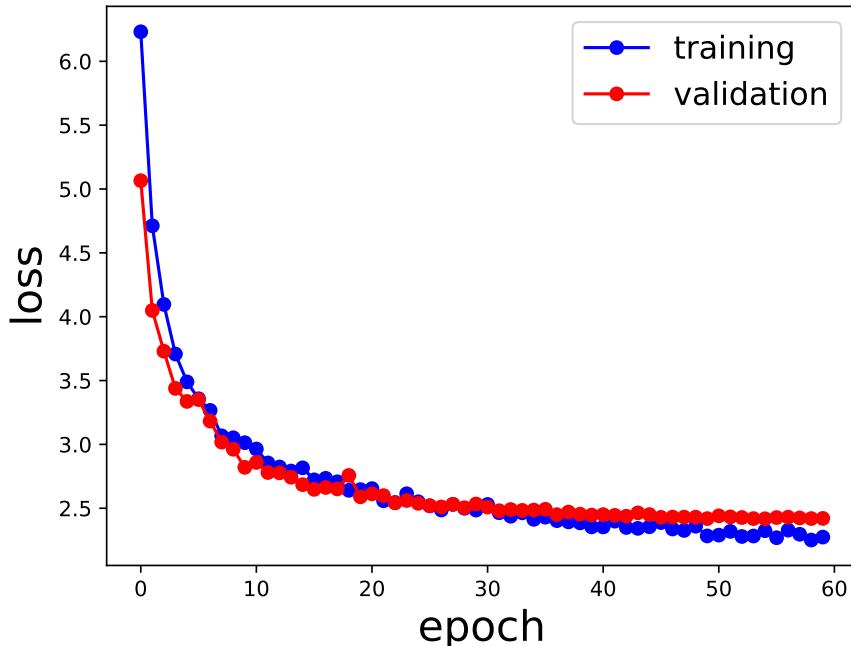
1946 The choice of the loss function was first based on what was used in literature. A sum of the MSE  
 1947 computed on the objects' amplitudes and one on the phases has thus been used (Eq. 6.2). The  
 1948 ground truth objects were indeed available from the simulated data while the predicted objects  
 1949 have been first calculated with a 2D inverse Fourier transform from the diffracted amplitude  
 1950 and the predicted RSP (Eq. 6.1).

$$\hat{O}(\mathbf{r}) = \mathcal{F}^{-1}\{\sqrt{I(\mathbf{q})} e^{i\varphi_{\text{pred}}(\mathbf{q})}\}(\mathbf{r}) , \quad (6.1)$$

$$\mathcal{L} = \frac{1}{N} \sum_{\mathbf{r}} \left( |\hat{O}(\mathbf{r})| - |O(\mathbf{r})| \right)^2 + \frac{1}{N} \sum_{\mathbf{r}} \left( \phi(\mathbf{r}) - \phi_{\text{gt}}(\mathbf{r}) \right)^2 , \quad (6.2)$$

### 1951 6.4.3 Results

1952 The training of the model was conducted on 8500 simulated BCDI patterns over 30 epochs with  
 1953 a learning rate of 0.0003 and monitored both training and validation loss. Here, Fig.6.5 shows  
 1954 the model's loss during the 30 epoch long training.



**Figure 6.5:** Training and validation loss over 30 epochs. The curve suggests a proper learning with no overfitting as both losses are decreasing reaching a plateau and the validation loss follows the same trend of the training loss.

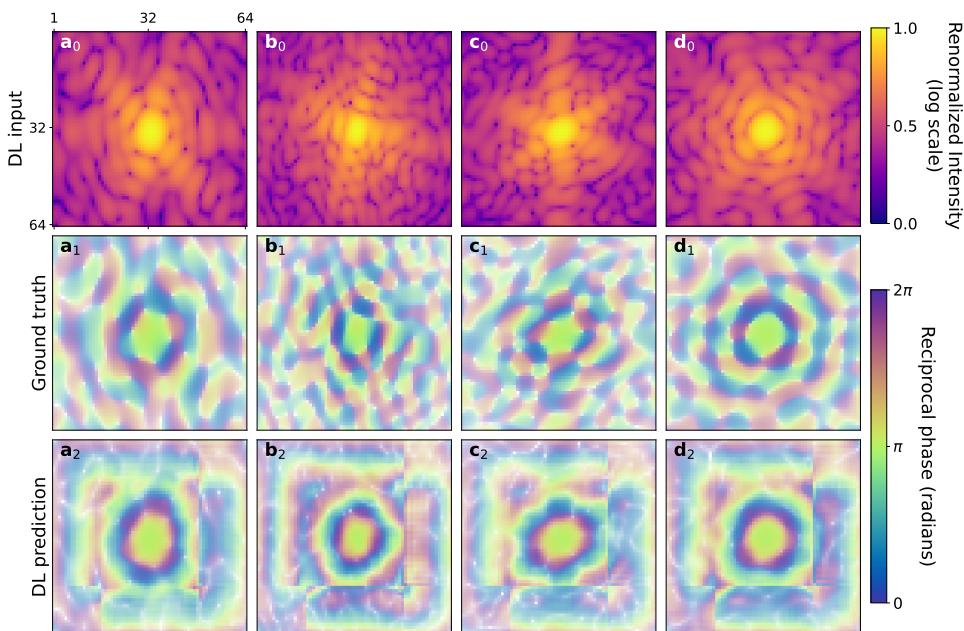
1955 Fig.6.6 illustrates the results of the predicted RSP of some test simulated BCDI patterns. Note  
 1956 that the displayed predicted RSP has been wrapped between 0 and  $2\pi$  for better comparison  
 1957 with the ground truth but the raw output of the model is in fact an “unwrapped” array. This is  
 1958 expected since no activation layer was applied to the last convolutional layer, meaning that  
 1959 the last operation is the multiplication of the last feature map with the real values inside the  
 1960 convolutional kernel, hence linear.

1961 When comparing the reconstructed objects obtained from the predicted RSP with the ground  
 1962 truth ones (Fig. 6.7) one can draw some interesting conclusions about the model’s learning  
 1963 performance. First it can be observed that the model learns the approximate shape and size of  
 1964 the particle, it produces images that resemble reasonable particles, sometimes similar to the  
 1965 ground truth. The amplitude is concentrated inside the support with little noise outside and  
 1966 the phase is overall correct around zero.

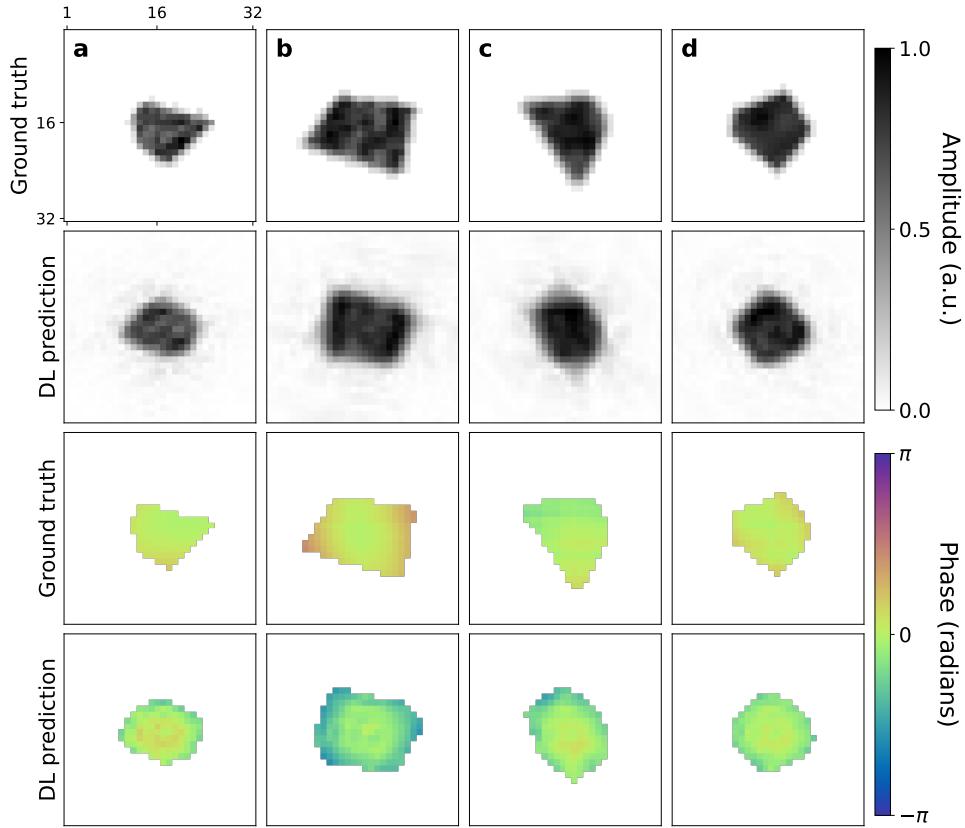
1967

1968 However, when looking more carefully, it is clear that the shape is not quite correct,  
 1969 especially for highly non-centrosymmetric objects. For instance, if we consider the object in  
 1970 Fig. 6.7 c, we see that the predicted shape seems to be derived from the incorrect superposition  
 1971 of the correct shape and its twin, also valid. In general it seems that the model tends to predict

1972 centrosymmetric objects. According to Sicairos *et al.* [90], if we name  $\varphi(\vec{q})$  the correct RSP,  
 1973 this phenomenon arises from a predicted RSP phase  $\phi$  composed of  $\varphi(\vec{q})$  in some regions of  
 1974 the  $q$ -space and  $-\varphi(\vec{q})$  elsewhere. In other words, the model is not fully able to break the sign  
 1975 symmetry. This subject was recently studied by Tayal *et al.* [184] and Zhang and coauthors  
 1976 in [187]. In this last study, the authors show that if not broken in the dataset, meaning that  
 1977 during the training the model is exposed to both cases ( $\varphi(\vec{q})$  and  $-\varphi(\vec{q})$ ) indistinctly, the model  
 1978 is deceived to produce a mix of the two, since the sign information cannot be recovered from  
 1979 the input intensity. The authors conclude that in order to prevent this detrimental effect, one  
 1980 should break the symmetry in the dataset to bias the model towards one preferred sign.

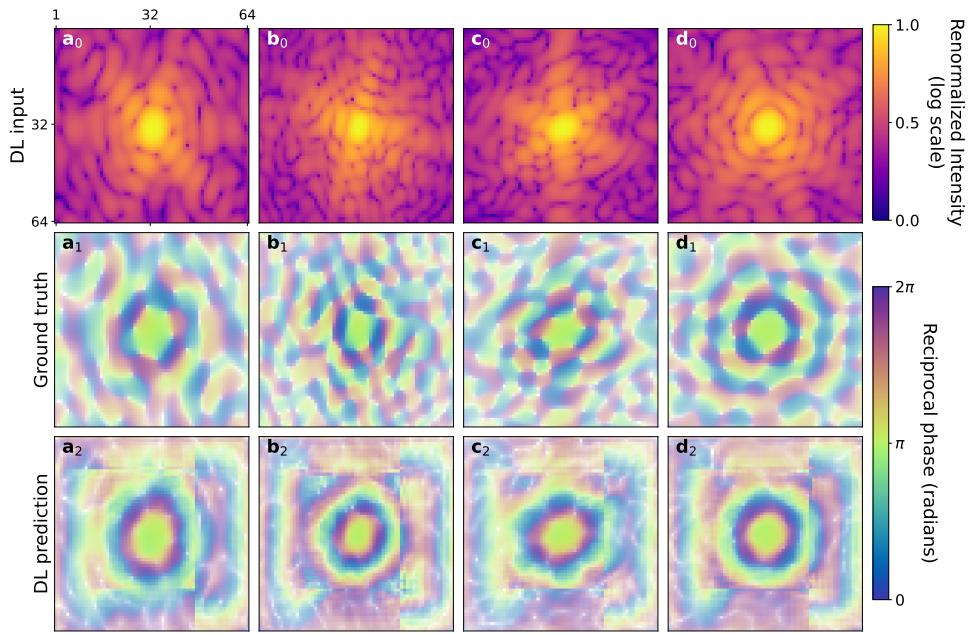


**Figure 6.6: Model testing on new 2D data using MSE loss function.** First row shows four simulated BCIDI patterns, second row the ground truth RSP corresponding to the pattern and last row the DL prediction. The superposition of the two twin solutions resulting in the object space can be inferred here by observing how iso-phase lines in the predicted RSP tend to be more “circular”, as it would be for round direct-space objects. Moreover, some squared artifacts are visible in the predicted RSP. Being the side of the square half the size of the image, one can hypothesize that is originated by an incorrect up-sampling in the last convolutional layer.

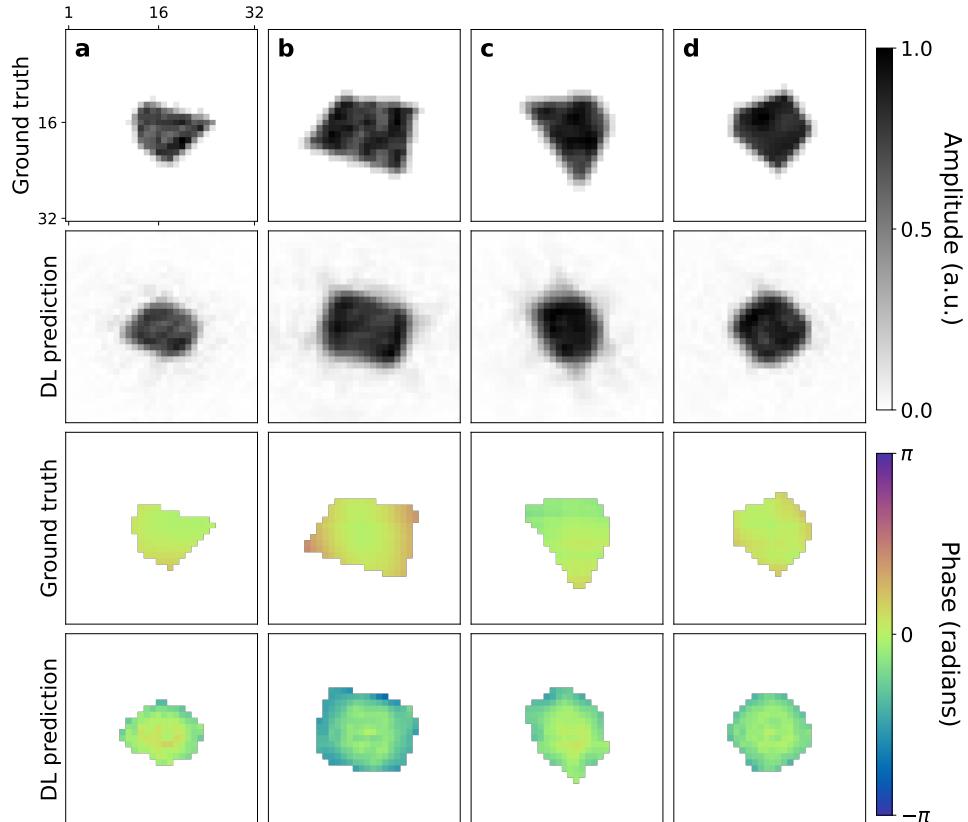


**Figure 6.7: Corresponding reconstructed objects.** Ground truth and predicted objects' amplitudes (first two rows respectively) and ground truth and predicted objects' phases (first two rows respectively). Note that the predicted objects resemble the superposition of the two twin true solutions.

The procedure presented in the article for the removal of the phase symmetries consists in:  
 (i) the centering of all the objects in real space (phase ramp removal), (ii) the shift of the RSP such that the  $k$ -th pixel for which  $\varphi_k = 0$  is the same across the dataset (phase offset removal), and lastly, (iii) the flip of the sign of the whole RSP when its value corresponding to a fixed position across the dataset is negative (if  $\varphi_{k+1} < 0$  then  $\varphi = -\varphi$ ). In our case the phase ramp symmetry was already broken by simulating particles with the center of mass in the center of the array. In this way the model is already biased towards the prediction of RSPs that yield centered objects. For the offset and the sign, the method proposed by Zhang *et al.* has been implemented in the model and the results are shown in Fig. 6.8 for the RSP and Fig. 6.9 for the reconstructed objects.



**Figure 6.8: Model testing using MSE loss function and biased dataset.** First row shows four simulated BCDI patterns, second row the ground truth RSP corresponding to the pattern and last row the DL prediction . The results do not seem to improve since the predictions look identical to the ones in Fig. 6.6.



**Figure 6.9: Corresponding reconstructed objects.** Ground truth and predicted objects' amplitudes (first two rows respectively) and ground truth and predicted objects' phases (first two rows respectively). No significant improvement can be observed after the adopted sign-symmetry breaking procedure.

Unfortunately, the proposed method did not seem to solve the sign ambiguity of the RSP. The model is still unable to discriminate between the plus/minus sign of the RSP and the result is the incorrect overlap of the object with its twin obtained by the inversion symmetry. The phase, though small for this case, is also showing a kind of centro-symmetry as its variations tend to spread radially from the center of the array.

#### 6.4.4 The Weighted Coherent Average loss function

At this point in the study, and in anticipation of applying the model to portions of the RSP, it became necessary to consider a loss function that would operate directly on the phase, without requiring transformations into direct-space. However, the main challenges were posed by the symmetries inherent to the phase. Upon further reflection, it was concluded that a Mean Squared Error (MSE) is not an appropriate metric for comparing the phases of complex functions. Indeed, MSE fails to account for the  $2\pi$  periodicity and the possibility of a global phase offset. One could argue that  $2\pi$  wraps can be fixed with a modulo  $2\pi$  operation and the offset can be removed by shifting the tensor by a constant. However, the modulo wrapping function jumps abruptly by  $2\pi$  every time the phase crosses an integer multiple of  $2\pi$ , meaning that the gradients are infinite thus not advised for gradient-based optimizations. Moreover, the MSE (or MAE and other *divergent* metrics) will have problems at the  $0$ - $2\pi$  boundary. In fact, when considering the phase mapped in the  $0$ - $2\pi$  range, if we suppose a  $\varphi_{pred}^0 = -0.1$  where  $\varphi_{G.T.}^0 = 0$ , the wrap will move the  $\varphi_{pred}^0$  to the value  $2\pi - 0.1 = 6.183$  amplifying the error ( $\Delta\varphi$ ) from  $0.1^2$  to  $6.183^2$  improperly.

In order to bypass these shortcomings a new loss function was designed. Here it follows the reasoning process that leads to the mathematical expression of the loss. The best way to account for the periodicity and the wrap, without discontinuities and error unbalances, is to evaluate the ground truth - predicted phase differences ( $\Delta\varphi_k$ ) on the unit circle. To do such, it is necessary to express  $\Delta\varphi_k$  as angles of a complex exponential. This means that if  $\varphi_{pred}$  is an array of random values, each complex number  $z_k = e^{i\Delta\varphi_k}$ , when represented on the Argand plane, can be seen as a vector pointing at a random coordinate on the unit circle (Fig. 6.10c). Now, the goal of the optimization is not to minimize  $\Delta\varphi_k$  for all  $k$  but to have the same  $\Delta\varphi_k$  throughout  $k$ . In fact, for  $\varphi_{pred} \Leftrightarrow \varphi_{G.T.}$  each vector  $z_k$  points in the same direction, but it does not necessarily lie on the x-axis ( $\Delta\varphi_k = 0$  condition). Therefore, the loss function should ultimately drive all the  $z_k$  from randomly distributed to coherently aligned along a common direction.

A helpful quantity in this case can be the complex average vector  $\langle z \rangle = \frac{1}{N} \sum_{k=1}^N z_k = \frac{1}{N} \sum_{k=1}^N e^{i\Delta\varphi_k}$  where  $k$  runs over all the  $N$  pixels. In particular the length of  $\langle z \rangle$ , represented by the modulus  $|\langle z \rangle|$ , is an efficient metric for the measurement of the *degree of alignment* of all the  $z_k$  between each other. In other words it measures the “coherence” between  $\varphi_{GT}$  and  $\varphi_{pred}$ .

In fact,  $|\langle z \rangle|$  scores 0 for randomly oriented  $z_k$ , as opposite contributions cancel out each other because incoherent, while it scores 1 for perfectly aligned ones. It follows that one wants to maximize  $|\langle z \rangle|$  during the optimization. Moreover, given the natural normalization between 0 and 1 of this metric, it follows naturally that the loss function can be expressed as  $L = 1 - |\langle z \rangle|$ .

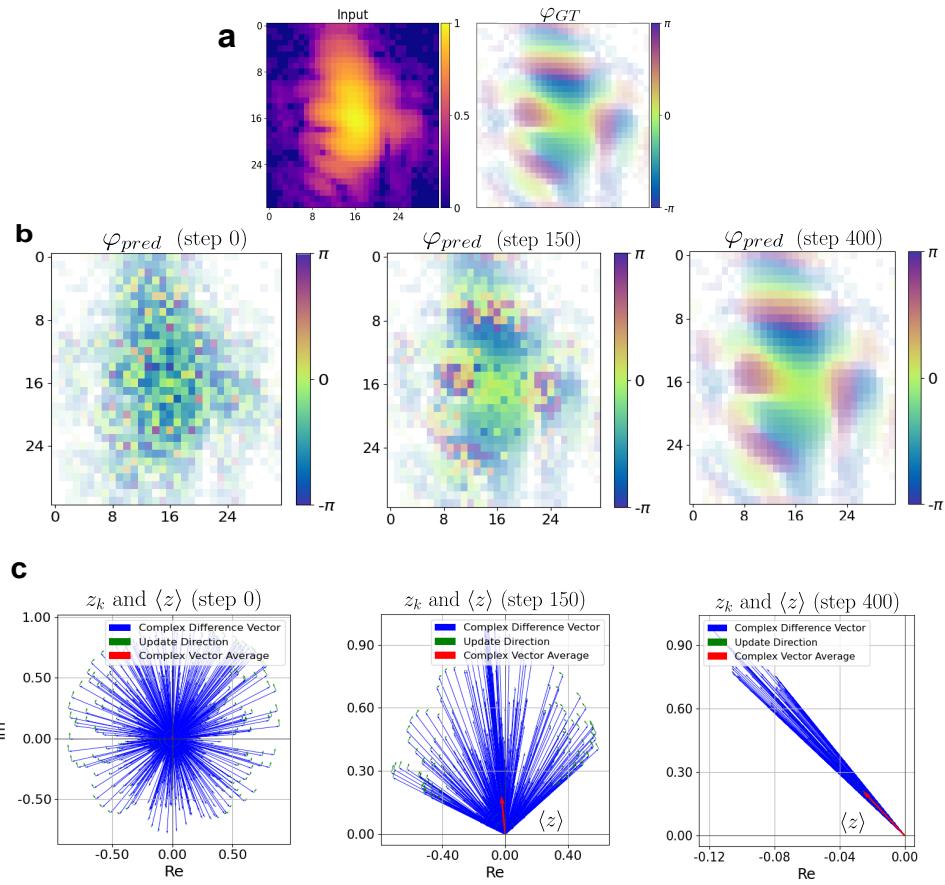
2033     Additionally, an importance mask can be applied during the averaging process. In particular,  
2034     we know that the brightest pixels of the BCDI pattern are the ones contributing the most to the  
2035     object's reconstruction. For this reason one could weight the complex average by multiplying  
2036     by the input magnitudes. The effect of this operation is to "give a direction" to the optimization,  
2037     meaning that the  $\langle \Delta\varphi \rangle$  the model will tend to converge to, will be mostly steered close to the  
2038      $\Delta\varphi_k$  of the brightest  $k$  pixels. The loss can now be expressed as:

$$L = 1 - \left| \frac{1}{N} \sum_{k=1}^N \sqrt{I_k} \exp(i(\varphi_{\text{GT},k} - \varphi_{\text{pred},k})) \right| \quad (6.3)$$

2039     Where  $N$  is the total number of pixels in each RSP array and  $k$  is the pixel index.  $\sqrt{I}$  is the  
2040     magnitude of the BCDI pattern normalized between 0 and 1 with respect to the sum, and  $\varphi_{\text{GT}}$   
2041     and  $\varphi_{\text{pred}}$  the ground truth and predicted RSP.  
2042     The last missing piece is the removal of sign symmetry. Rather than biasing the dataset  
2043     preferring one sign over the opposite, the function  $L$  is computed for both  $\varphi_{\text{GT}}$  and  $-\varphi_{\text{GT}}$   
2044     and in a second passage, the minimum of the two along the batch dimension is kept for  
2045     backpropagation. The final form of the Weighted Coherent Average (WCA) loss is then given  
2046     by:

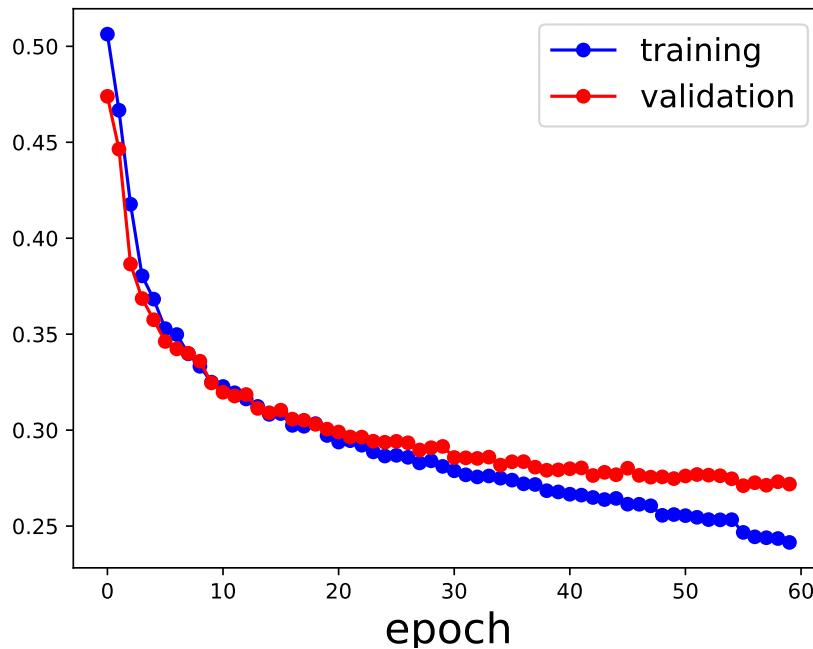
$$L_{\text{WCA}} = \min(L_+, L_-) \quad (6.4)$$

2047     To better visualize the functioning of the WCA loss function, a simple model has been  
2048     trained to fit the ground truth phase of a single 2D BCDI pattern using the WCA. The complex  
2049     phase differences vectors were extracted at each step of the optimization together with the  
2050     updates obtained from the gradients of the WCA with respect to the trainable parameters. Fig.  
2051     6.10 shows the evolution of the predicted RSP as well as the progressive alignment of the  $z_k$ .



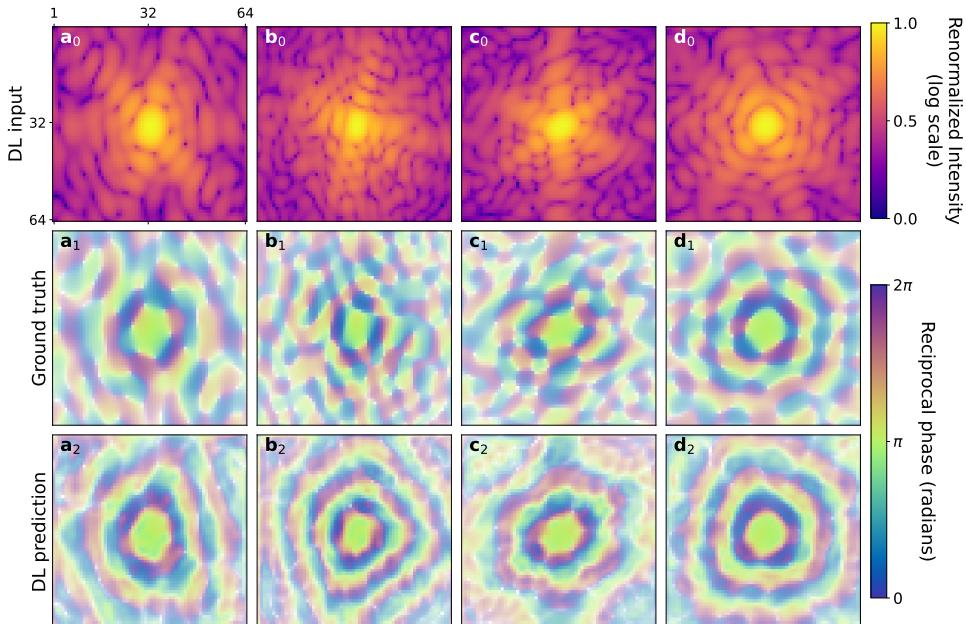
**Figure 6.10: Illustration of the WCA loss function.** **a** Input intensity (log-scale normalized) and ground truth RSP. **b** Predicted RSP in steps 0 - 150 - 400 of the optimization. **c** Corresponding complex phase-differences vectors  $z_k$  on the Argand plane (blue arrows), together with the updates (green arrows) obtained from the gradients of the WCA, and the resultant complex average  $\langle z \rangle$  (red arrow). It is visible that during the fit, as the  $z_k$  align around a common one, the amplitude of  $\langle z \rangle$  grows bigger and the predicted RSP converges to the ground truth one. Notice in **c** the vectors align around a  $\Delta\varphi \neq 0$  resulting in a  $\varphi_{pred}$  shifted by a constant offset. The offset has been removed before plotting in **b** for better comparison with the ground truth.

2052        The same model has been trained using the WCA for the same number of epochs on the  
 2053        same dataset and here the results are shown. First, it can be noted in Fig.6.11 that the training  
 2054        and validation loss values throughout the training are following different trends with respect  
 2055        to the model trained with the MSE loss (Fig. 6.5)

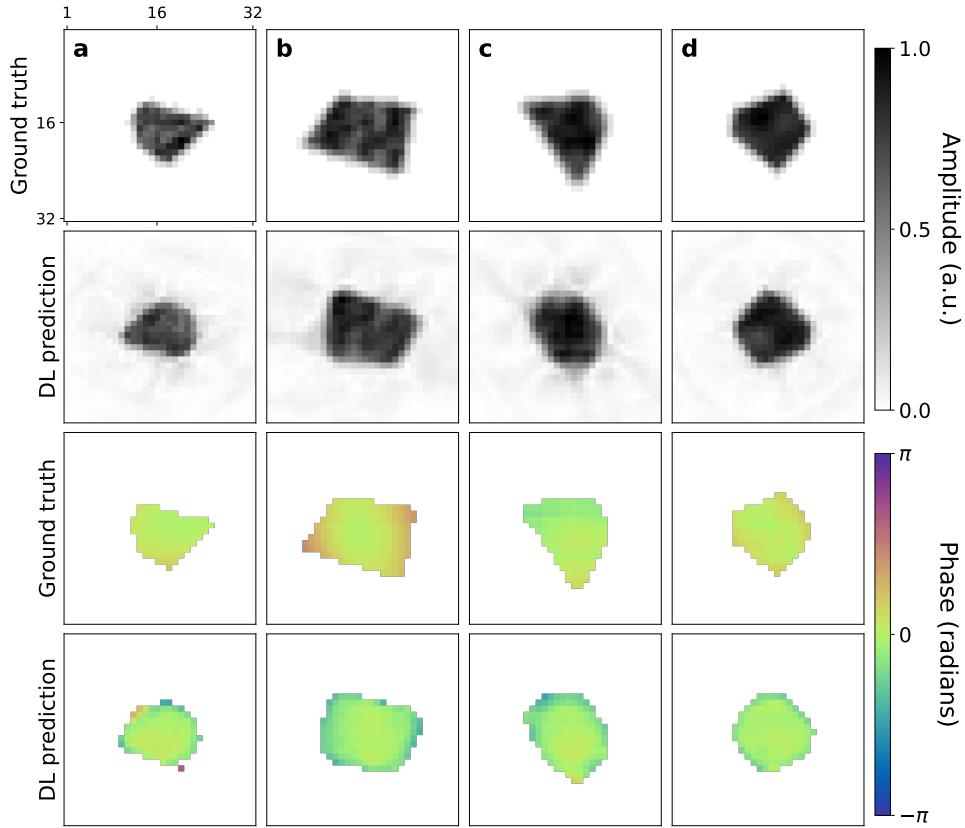


**Figure 6.11:** Training and validation loss curves over 60 epochs using the WCA loss function.

2056 In this case the correct learning curve does not reach a plateau within the first 25 epochs  
2057 but maintains a negative slope for longer, indicating a better learning. This suggests indeed  
2058 better results when used on test data. In particular, for the same input diffraction patterns  
2059 tested above in Figs.6.8 - 6.9 the model trained with the WCA yields the prediction shown in  
2060 Fig.6.12 for the RSP and Fig.6.13 for the corresponding reconstructed objects.



**Figure 6.12: Model testing using WCA loss function.** First row shows four simulated BCDI patterns, second row the ground truth RSP corresponding to the pattern and last row the DL prediction. Here the squared artifacts disappeared, suggesting a better learning.

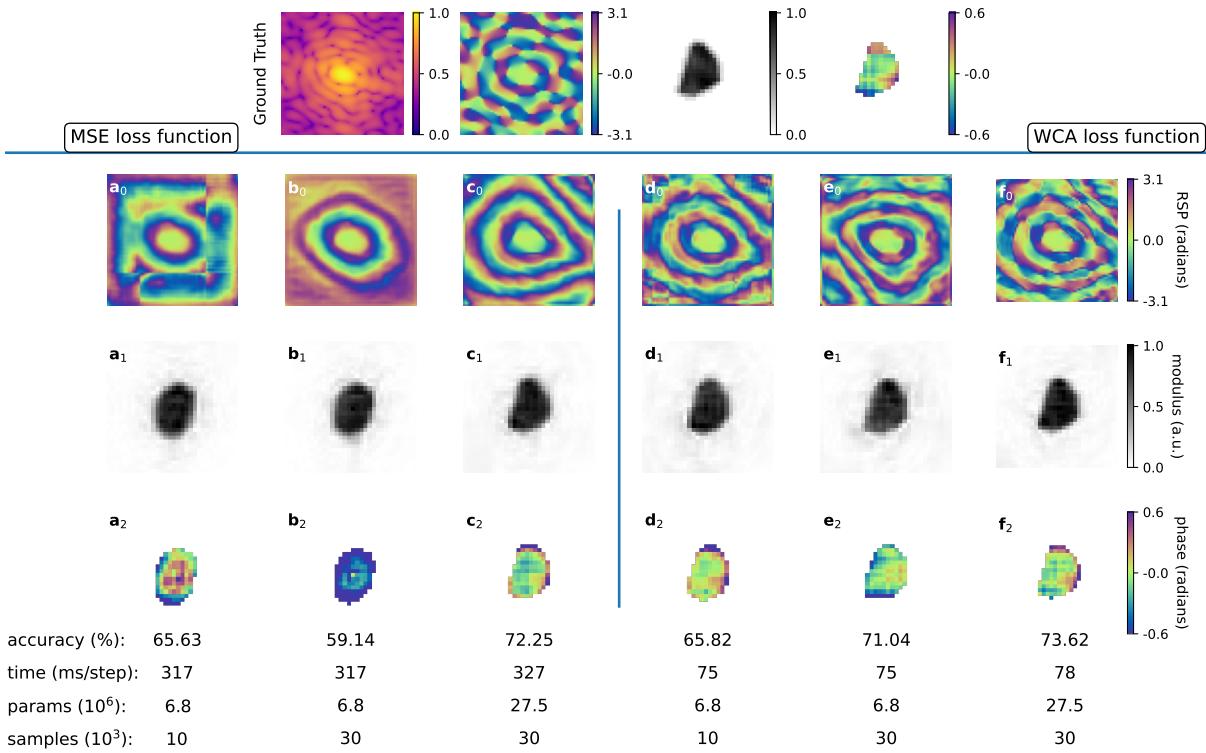


**Figure 6.13: Corresponding reconstructed objects.** Ground truth and predicted objects' amplitudes (first two rows respectively) and ground truth and predicted objects' phases (first two rows respectively). Here the twin problem mentioned above and visible in Figs.6.7 - 6.9 is less pronounced, the model seems to manage the two solutions, often preferring the twin image to the ground truth.

The results obtained from the model trained with the WCA loss function are visually better than the MSE ones. Although not completely removed, the sign symmetry that gives rise to the superposition of the object with its twin, is less pronounced. For example, particles in Fig. 6.13(a-b-d) have a clear orientation and a shape that matches the ground truth. In all those cases though, the model has opted for the conjugate solution as the predicted object are flipped with respect to the ground truth ones. In Fig. 6.13(c) instead the symmetry is not broken and the result is still a superposition of the particle with its twin. This suggests that the symmetry breaking method implemented in the WCA, and the one proposed by Zhang and coauthors, is only partially playing a role in the actual model learning. It is interesting to notice indeed that when the training dataset or the model trainable parameters are increased, the sign symmetry is completely removed in the most difficult cases as well. Fig. 6.14 shows the effect of the dataset and models sizes for both MSE and WCA loss functions on the same simulated test data. The first important piece of information this figure shows is that the model trained with the WCA reaches higher accuracy. Moreover, it is much faster to compute since no FFT or IFFT is involved (see Fig. 6.14), thus the training time is drastically reduced. For what concerns the accuracy metric, in order to properly account for both modulus and phase, it has been calculated using

$$\left( \frac{PCC(\rho_{pred}, \rho_{GT}) + WCA(\rho_{pred}, \phi_{pred}, \phi_{GT})}{2} \right) \times 100 \quad (6.5)$$

where  $PCC(\rho_{pred}, \rho_{GT})$  is the Pearson Correlation Coefficient on the object's moduli and  $WCA(\rho_{pred}, \phi_{pred}, \phi_{GT})$  is the WCA function applied to the object's phase weighted by the normalized predicted modulus (Eq.6.3-6.4). For what concerns the sign symmetry problem it is evident that while for the MSE trained model it is resolved only for a larger number of trainable parameters, for the WCA trained one it is already sufficiently overcome. Finally, it is interesting to notice that when the model size is kept fixed and the training dataset augmented, the WCA improves the performances while for the MSE it is not the case.



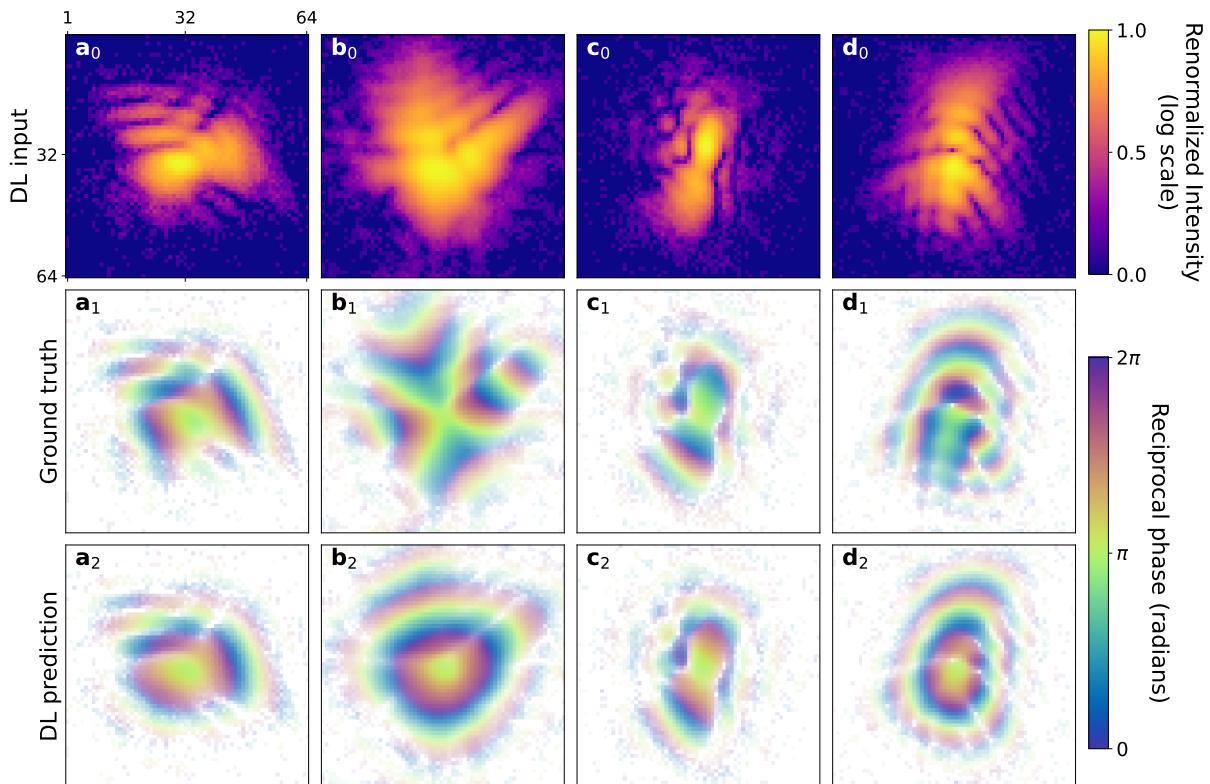
**Figure 6.14: Comparison of MSE and WCA loss function for different model and training dataset sizes** In the first row from left to right the input intensity, the ground truth RSP and the corresponding object (modulus and phase) are represented.  $a_0, b_0, c_0$

are the results of the predicted RSP obtained from the model trained with the MSE loss function with the initial number of parameters and training set (a), with the augmented dataset (b) and with both model and dataset size increased (c). In third and fourth rows the corresponding reconstructed objects are displayed.  $d, e, f$  columns symmetrically shows the results obtained with the model trained using the WCA loss. The WCA yields better results and requires shorter trainings.

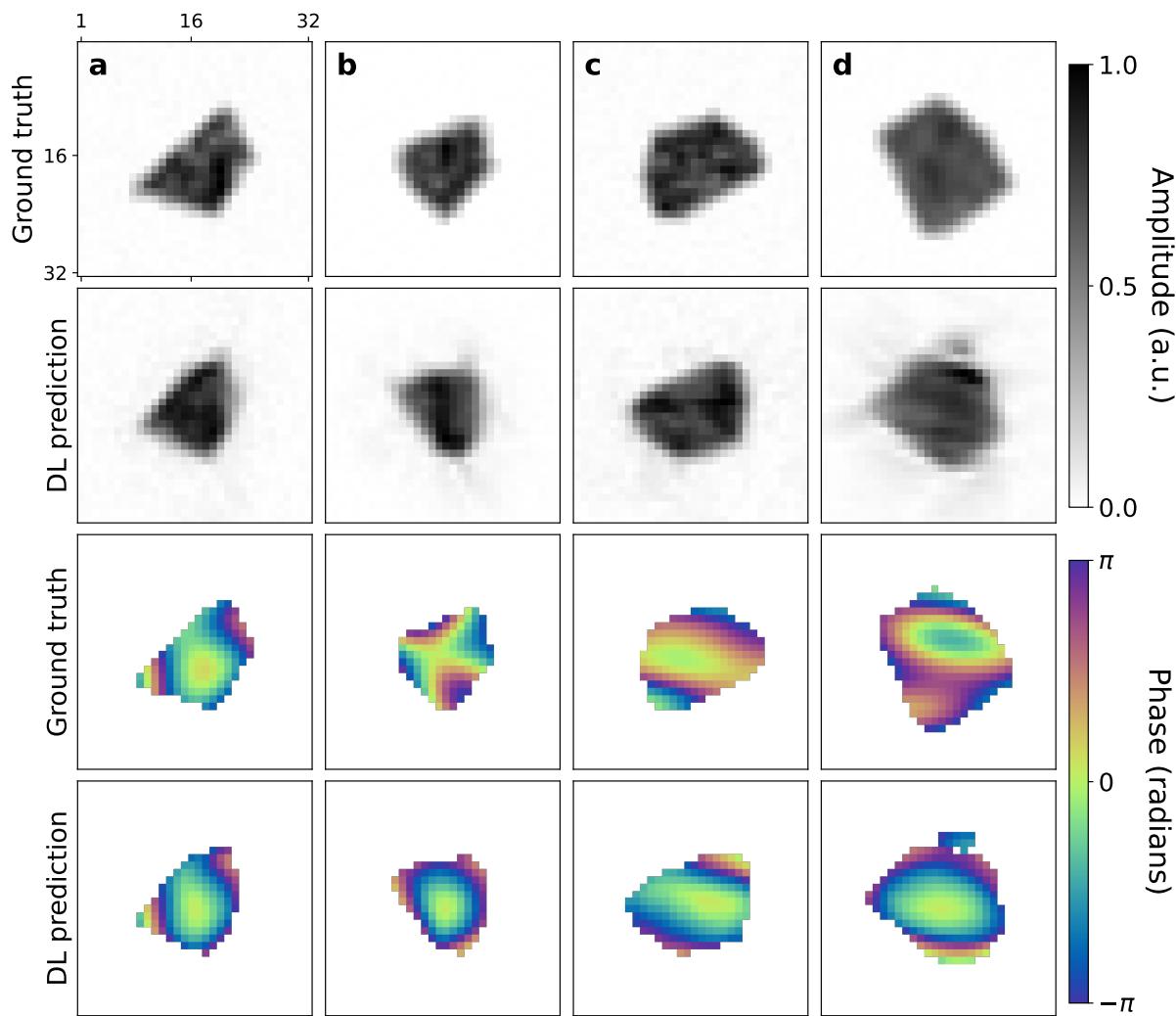
## 2085 6.5 2D high strain case

In this paragraph the model training was performed on a dataset of highly strained 2D BCDI pattern simulated as described above in section 5.3.1. In this case Poisson statistics was applied to each dataset to better simulate the experimental condition. A set containing 30'000 samples was created and the “bigger” model of 27.5M parameters mentioned in Fig 6.14 was trained over 50 epochs with a learning rate of 0.001. Similarly to the low-strain case described above the same model has been trained with the MSE and WCA losses separately for comparison.

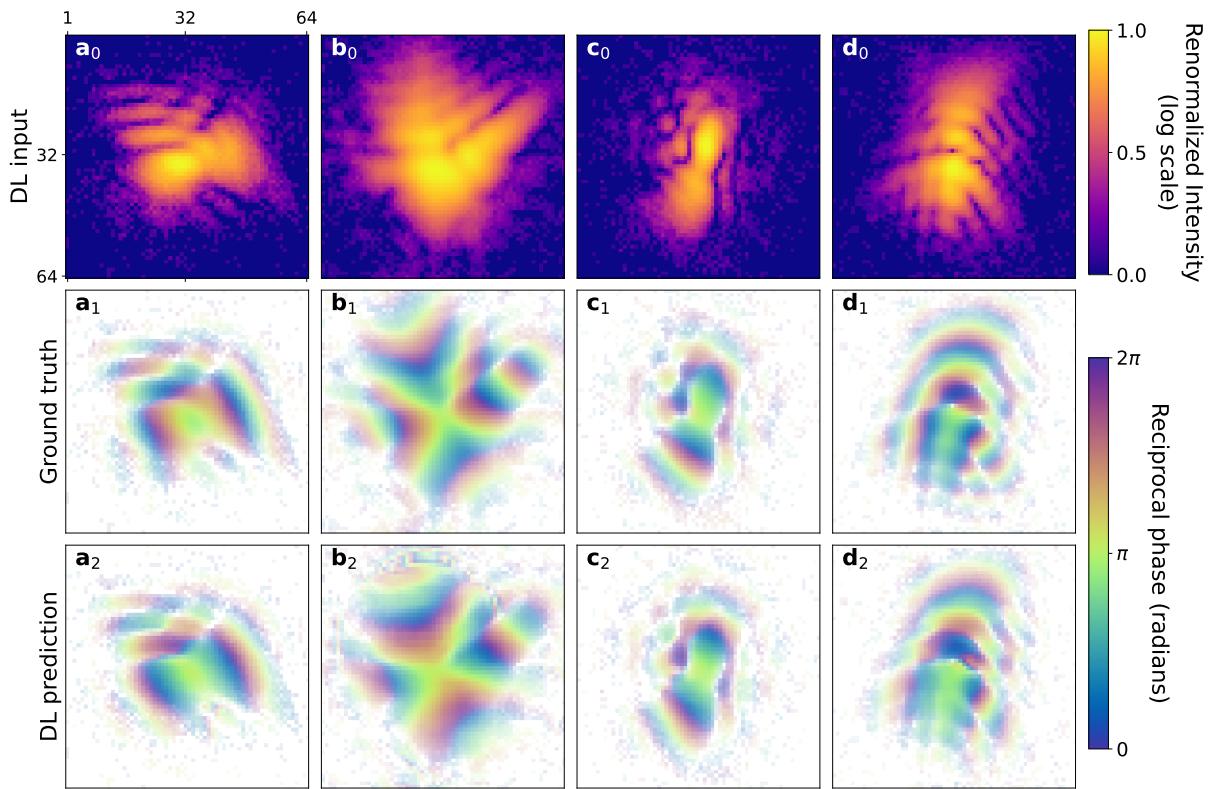
2092 The goal of this study is in fact to test the relevance of the loss function compared to the size  
 2093 of the model when the complexity of the task increases. The results on 4 test BCDI pattern are  
 2094 shown in Figs. 6.15-6.16 for the MSE one and 6.18-6.19 for the WCA.



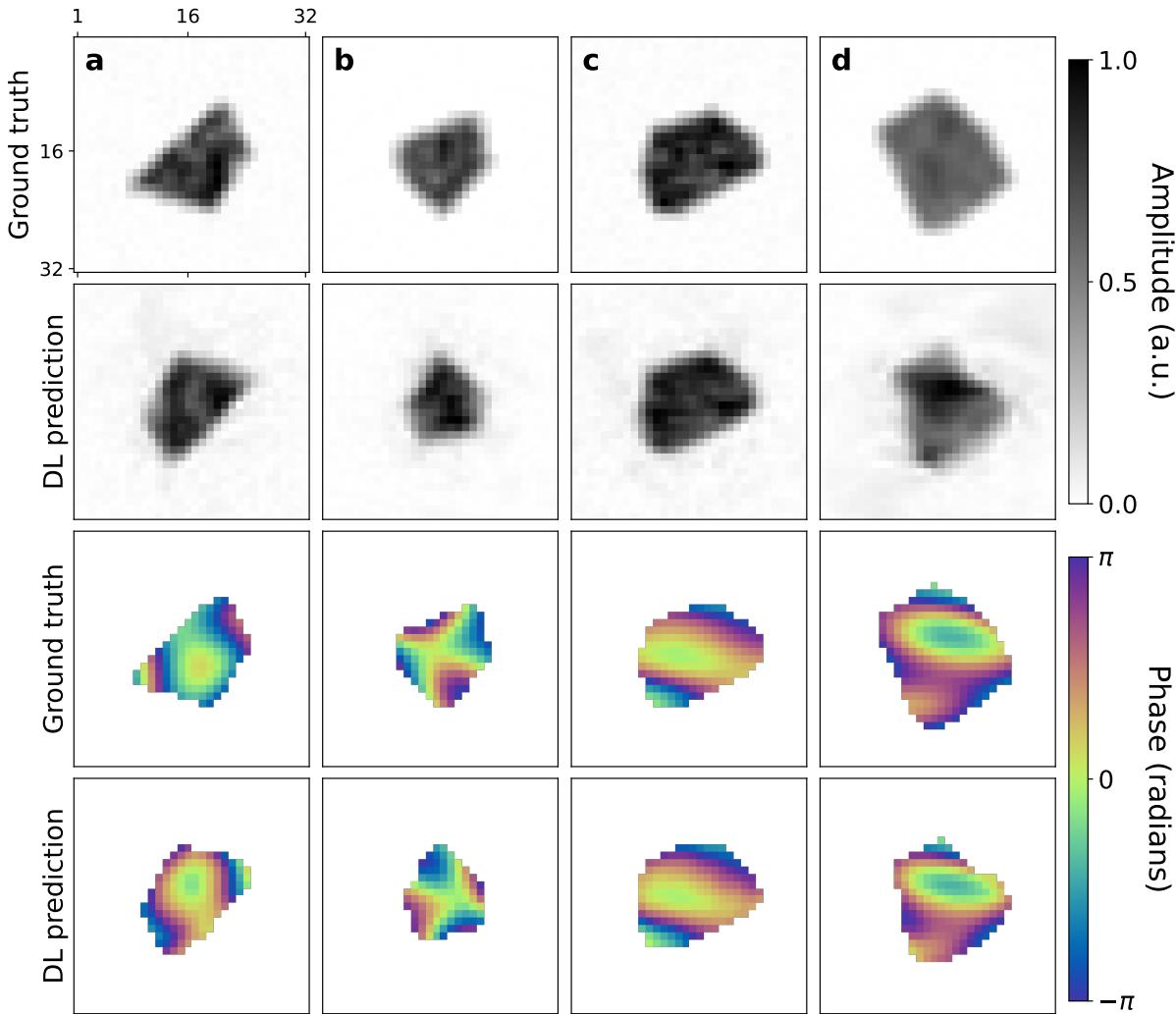
**Figure 6.15:** RSP predicted by the model trained with the MSE loss function calculated on both modulus and phase of the reconstructed object. First row: simulated 2D strained BCDI patterns (test dataset). Second row: corresponding ground truth RSP. Third row: predicted RSP. Once can notice that the model cannot predict correctly the RSP where the “iso-phases” do not have a circular symmetry (see **b-d**).



**Figure 6.16:** Reconstructed objects corresponding to Fig 6.15. First and third row: ground truth modulus and phase. Second and fourth row: model’s results of objects’ modulus and phase. Although the shape might at first sight look like the ground truth one (or the twin) the phase is often incorrect. It is curious to notice that better results are obtained when the object’s phase possesses a certain degree of symmetry with respect to the center, analogously to the corresponding RSP.



**Figure 6.17:** RSP predicted by the model trained with the WCA loss function. Here the model correctly retrieves the RSP for non-circularly symmetric structures as well (**b-d**)



**Figure 6.18:** Reconstructed objects corresponding to Fig 6.17. Compared to the objects obtained from the DL model trained with the MSE loss function (Fig.6.16) here the objects' phases are correctly recovered, also when the symmetry is not circular (**b**).

2095 The preliminary studies on the 2D case for low-strain particles have demonstrated the  
 2096 possibility to recover the RSP from the diffracted intensity pattern with a U-Net like architecture  
 2097 without ever calculating the object in real space. Moreover, the studies on the high strain  
 2098 particles has shown that this model configuration is well suited for this case as well. From  
 2099 these promising results, it was decided to investigate the mapping intensity-RSP for patches of  
 2100 the reciprocal space.

## 2101 6.6 Phasing patches: 3D case low strain

2102 In this section of the manuscript the DL prediction of “patches” of the RSP will be explored and  
 2103 discussed. Three-dimensional BCDI patterns of low strained particles were used to conduct  
 2104 this study. Although this patching approach has not given satisfactory results for the PR, it is  
 2105 nevertheless reported in the manuscript as study on the *local* rather than *global* relationship  
 2106 between the diffracted intensity and the RSP. It is indeed known that a unique mapping exists,

2107 barring some trivial RSP symmetries, between the diffracted intensity and the RSP in 3D [188].  
2108 What is interesting to investigate is whether this relationship exists also for subsets of the  
2109 reciprocal space, and in particular if it can be retrieved by a DL model. (From now on the term  
2110 “patches” will be used to refer to cubic subsets of the reciprocal space).

2111

2112 When deciding to work with patches, there is a number of questions that arise and the  
2113 answer to which is not straightforward nor unique in many cases. Namely:

2114

- What size is best?
- Can the patches be extracted at random positions or should there be an order?
- What about the normalization of the intensity range inside the patch?
- How are the patches stitched together into the full RSP eventually?
- How are the phase symmetries taken into account during the stitching?

2115 Here I will present the approach that allowed me to address these questions.

2116

### 6.6.1 The choice of the size

2117 Similarly to the inpainting case, 32 pixel-side cubic patches, cropped out of 128 pixel-side  
2118 simulated BCDI patterns were considered. The choice was supported by the following reasons:

2119

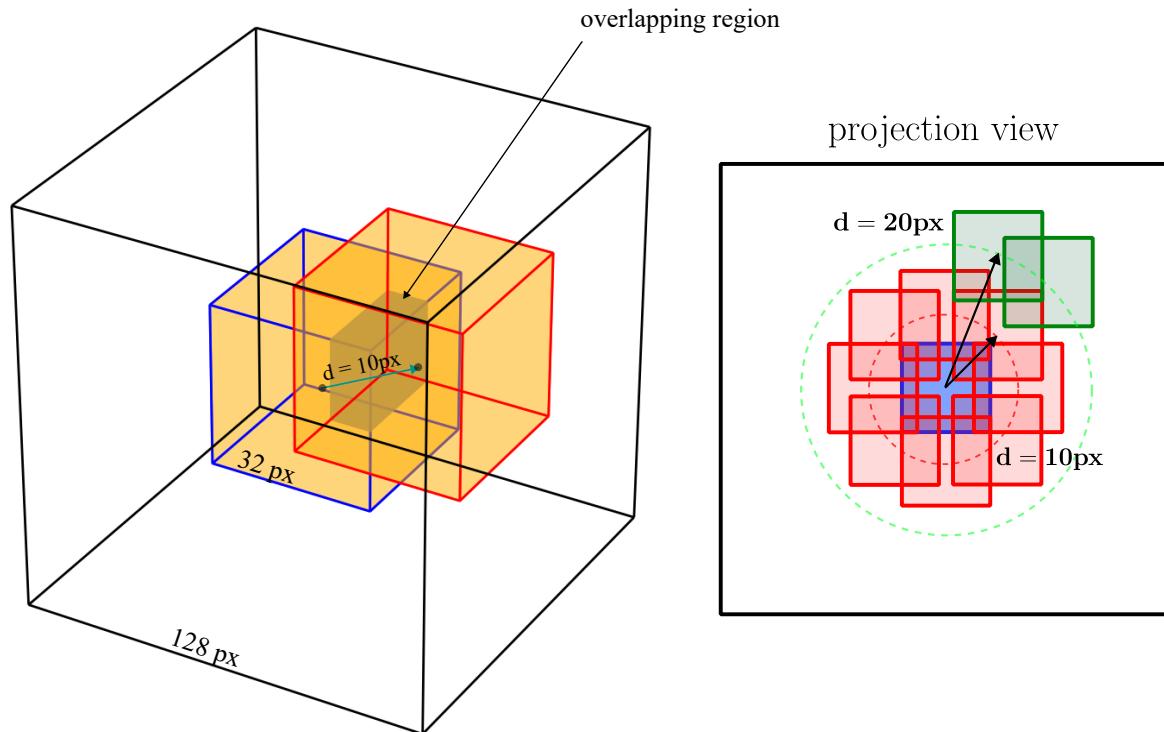
- The good results obtained for the inpainting case suggested that the amount of information contained inside a 32 pixel-side patch of reciprocal space is enough for the model to grasp spatial correlations.
- The average oversampling ratio of BCDI experimental data is such that in a 32 pixel-side volume a sufficient amount of fringes is contained, meaning intuitively that the model can predict the corresponding RSP.
- An even number multiple of 2 is usually considered GPU-friendly since it facilitates shared calculations across different threads.

2120

### 6.6.2 Patches division and stitching

2121 At first, the patches were extracted randomly from the full BCDI pattern as for the inpainting  
2122 case. However, by doing so the RSP of each patch would in principle have different offsets and  
2123 different wraps than their neighbors and this would complicate the stitching of the patches  
2124 back into the full RSP. For this reason, and considering the approximate spherical symmetry  
2125 of the average BCDI pattern it was decided to crop patches radially, starting from the region  
2126 around the center of the Bragg peak and the progressively moving outwards to higher q-values.  
2127 In this configuration, an integer step (10 pixels in our case) was chosen beforehand and the first  
2128 patch around the center of the Bragg peak was selected together with all the patches centered

in distances of integer multiples of the chosen step. Fig. 6.19 shows a simplified schematic of the patches' extraction. For the DL model training the patches of the intensity pattern need to be selected as well as the for the corresponding RSP for ground truth comparison.

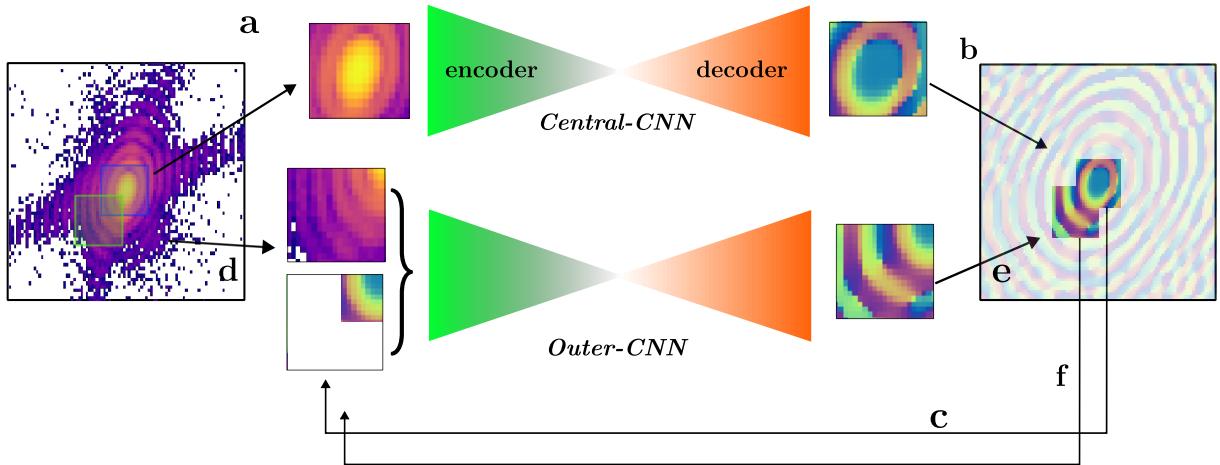


**Figure 6.19: Schematic of the cropping of patches (not in scale).** **(Left)** From the full BCDI pattern (white 128 px-side cube) the first patch is cropped out in the center and (orange 32 px-side cube with blue outline). Other patches are extracted radially from concentric shells separated by a 10 pixels distance. The gray shaded area highlights the overlapping volume between the two patches. **Right** The projection view shows the full first shell and two patches from the second shell for simplicity.

Being the step size smaller than the semi-diagonal of the 32 pixel-sided patches, it follows that the patches of adjacent cells have overlapping volumes. These common regions can have a twofold purpose. Firstly, they reduce the complexity of the stitching procedure since when this is executed progressively starting from the central patch, the sign and the offsets of the RSP are unambiguously fixed for all the following ones. Secondly, during the DL model training, for patches belonging to the outer shells, the overlapping volume of RSP belonging to the innermost adjacent shell can be provided as initial guess along with the input intensity patches. This of course cannot be exploited for the central patch that necessarily has to be predicted without initial RSP guesses.

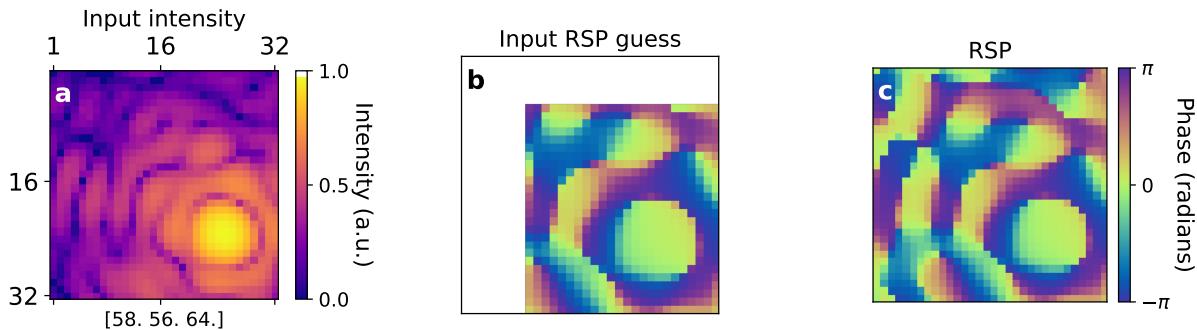
The last question to be answered concerns the normalization. Since each patch is processed independently of the others by the DL model, it was decided to normalize each patch between 0 and 1 (always in log scale).

To summarize, the final design implied the use of two distinct training datasets and two different CNNs as depicted in Fig.6.20.



**Figure 6.20: Schematic of the patching model for phase retrieval.** The central patch of the BCDI pattern is cropped and sent through the *Central-CNN* (a) for the prediction of the corresponding RSP. During training the corresponding ground truth RSP is used to compute the WCA loss function and update the model's parameters. During inference predicted central RSP is placed in the center of an empty 3D array (b). The outer patches on the first shell at 10px distance from the center are extracted from the BCDI pattern and concatenated with the RSP patch which is non-zero only in those voxels overlapping with the innermost shell. During training this initial RSP guess is taken from the ground truth simulated RSP while during inference this is taken from the previous predictions (c - d). The *Outer-CNN* takes as input the outer intensity patch and the initial guess RSP and completes the RSP patch with the prediction inside the empty voxels. During inference, the output is stitched with the previous predictions (e) to form the full RSP. The process is repeated until the outermost shell (f).

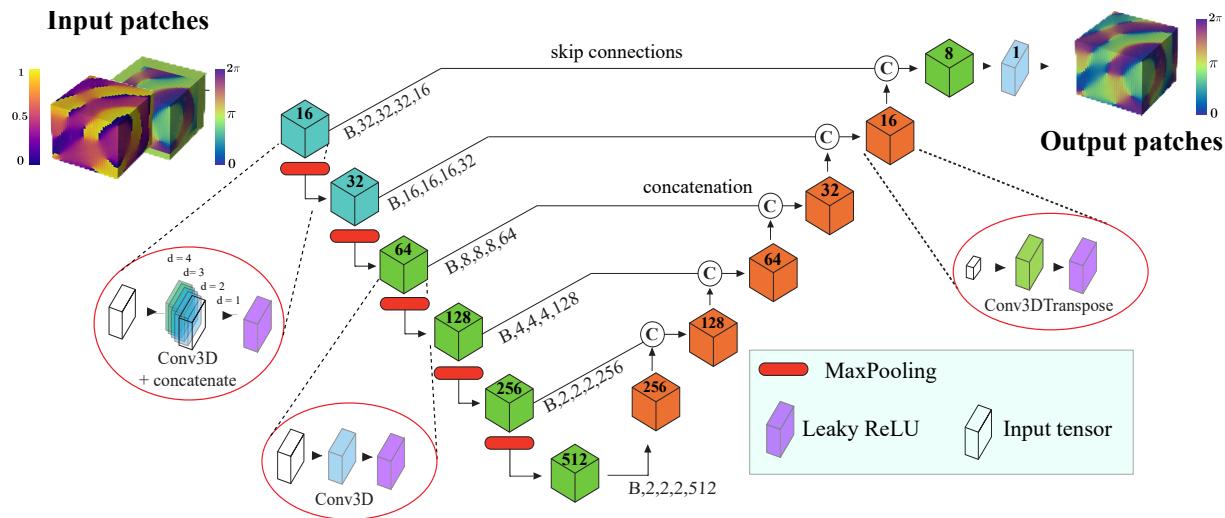
2157        The first dataset was dedicated to the central portion, therefore the first CNN (*Central-CNN*)  
 2158        was provided with 3D intensity patches in input (normalized log scale) and corresponding  
 2159        RSP patches as ground truth labels. A second training dataset containing patches from outer  
 2160        shells (5 concentric ones for a 128 pixel-sided full BCDI pattern) was created. Here each file  
 2161        was made of the pair intensity-RSP initial guess, from the closest neighbor patch belonging to  
 2162        the innermost shell, as input and the full RSP ground truth patch corresponding to the input  
 2163        intensity. This second dataset was used to train a second CNN (*Outer-CNN*) identical to the  
 2164        first one. One observation regarding the datasets is that there is an intrinsic imbalance between  
 2165        the number of central patches and the outer ones. In fact, for a single full BCDI pattern, the  
 2166        number of patches in the first shell is 1, while the number of outer portions can go up to several  
 2167        hundreds. Moreover, the central patch is the most important one as it contains a low resolution  
 2168        representation of the particle in real space. In order to balance the training, the first dataset  
 2169        was augmented with more simulated data.



**Figure 6.21:** Example of input-ground truth pairs for the outer patches model. **a** Central slice of the input intensity for the patch cropped from the first shell at position [58,56,64] (the central patch is at [64,64,64]). **b** Initial RSP guess deriving from the overlapping region of the intensity patch with the central one. The blank area represents the part that needs to be predicted. **c** Ground truth RSP corresponding to the intensity patch in **a**.

### 2170 6.6.3 Model architecture

2171 The model architecture is similar to the one used for the inpainting case, with a U-Net like  
 2172 structure (see Fig. 6.22). 5 encoder blocks reduce the feature map to 512 one-dimensional  
 2173 vectors in the bottleneck and 5 decoder blocks upsample the feature map back to the original  
 2174 size. Skip connections are used as well and similarly to the 2D case for Phase Retrieval, the last  
 2175 layer has been left with no activation function.



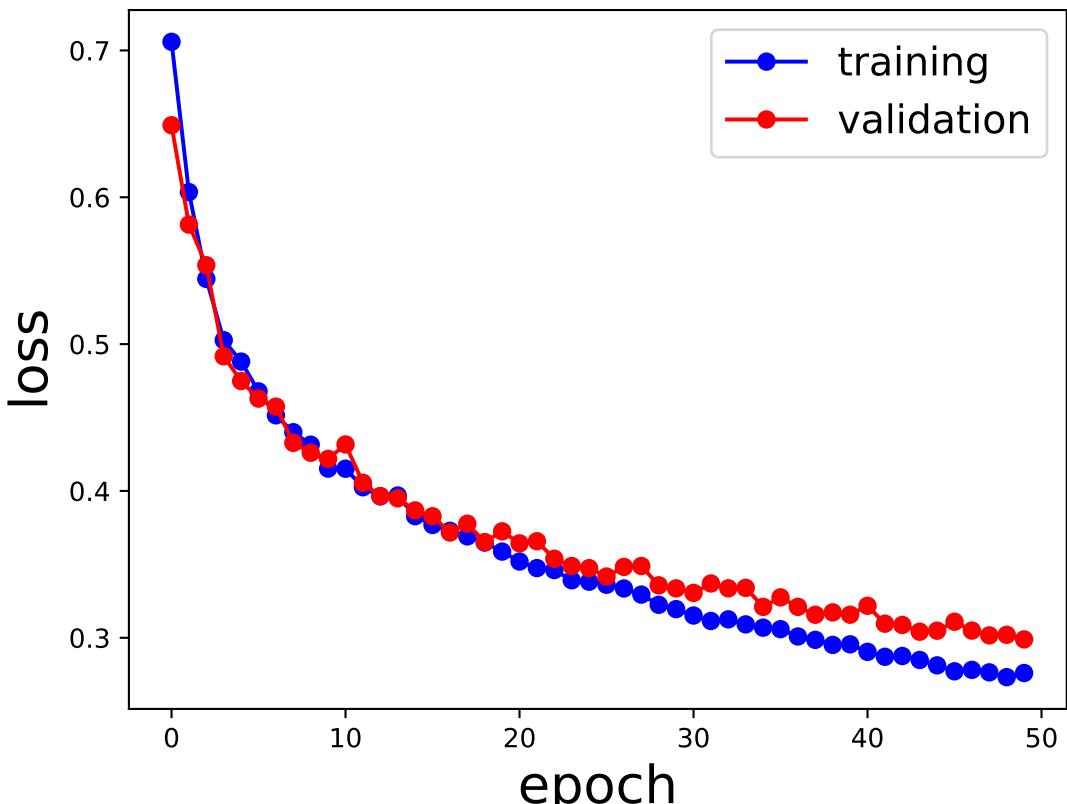
**Figure 6.22: Patching-model architecture.** Both *Central-CNN* and *Outer-CNN* share the typical U-Net like encoder-decoder structure for 3D images. Here only *Outer-CNN* is depicted. The description of each block can be found in the captions of Figs.5.10 - 6.4.

2176 The total number of trainable parameters is of the order of 3.5 million.  
 2177 For the second CNN the layers and features are identical to the first one but three modifications  
 2178 were made. Namely:

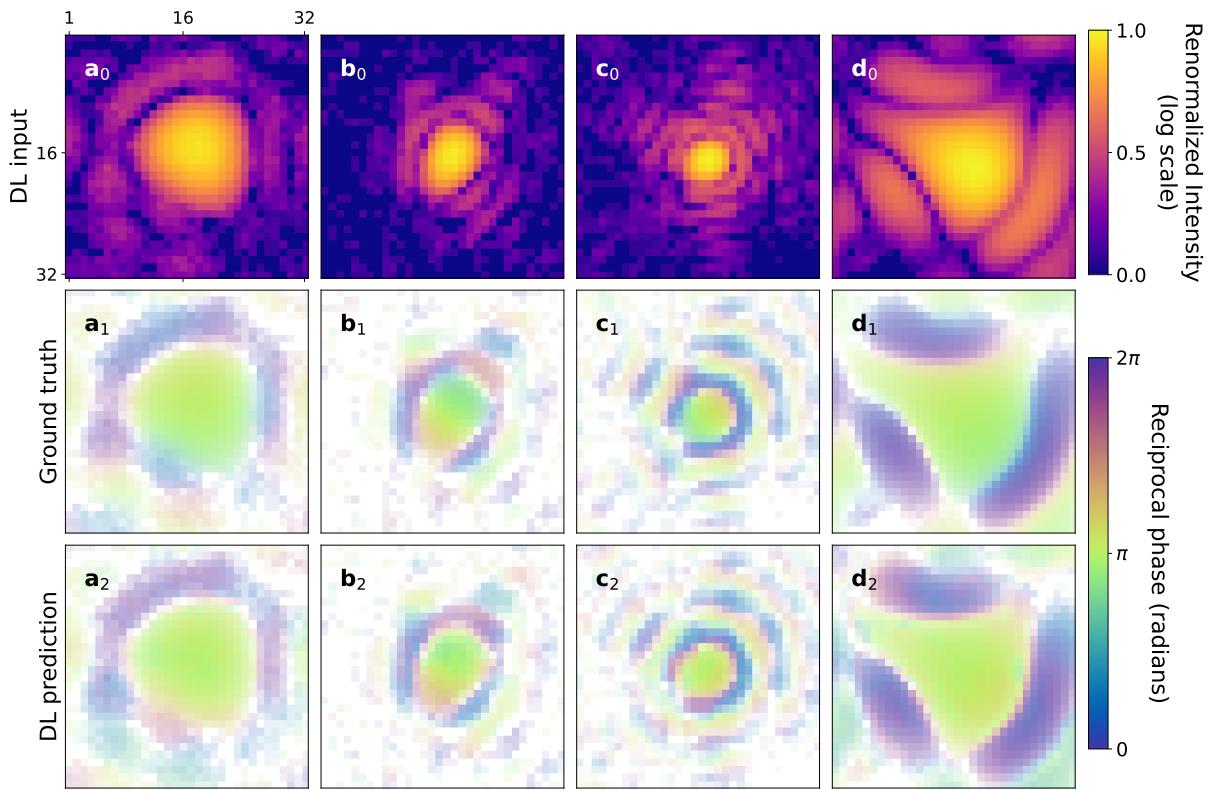
- 2179 • The input tensor was composed of the intensity patch concatenated with the initial RSP  
 2180 guess and a binary mask marking the RSP guess voxels from the others.
- 2181 • The mask was used at the exit of the decoder as well to select the new predicted voxels  
 2182 only for the back-propagation.
- 2183 • The WCA loss function has been restricted only to the positive sign of the RSP since no  
 2184 sign ambiguity is left when fixing an initial guess

2185 **6.6.4 Results on central patches**

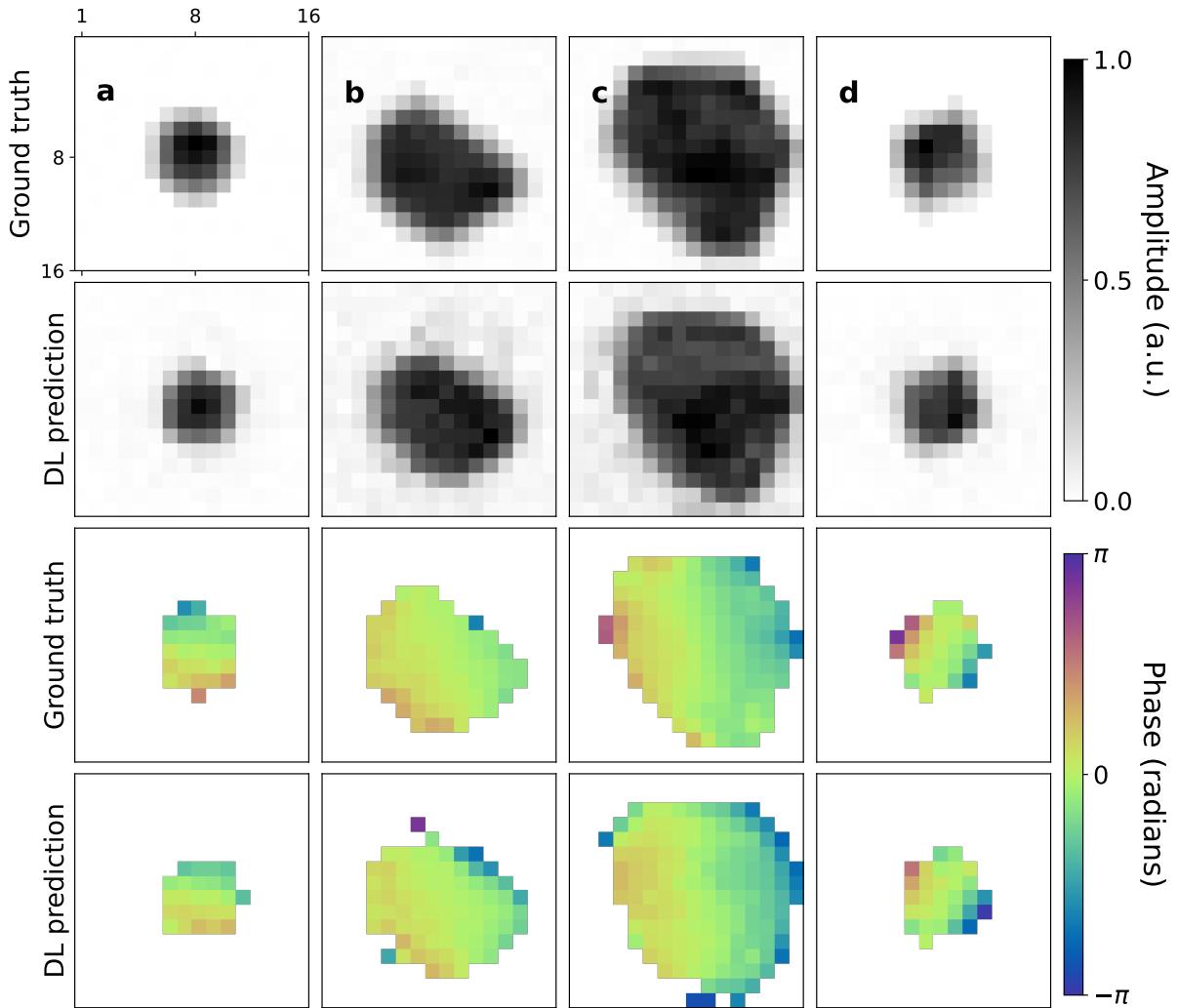
2186 *Central-CNN* has been trained over 50 epochs on 10000 samples of central  $32 \times 32 \times 32$  pixel-size  
 2187 patches cropped out of  $128 \times 128 \times 128$  pixel-size full BCDI simulated pattern using the WCA  
 2188 loss function. Fig.6.23 shows a good learning curve which is supported by the results on test  
 2189 data, illustrated in Figs. 6.24 - 6.25



**Figure 6.23:** Training and validation loss curves over the 50 epochs long training. The plots indicate a good learning of the model because both training and validation losses are decreasing monotonically with the same pace.



**Figure 6.24:** Slices of the central patches. First row shows four different examples of BCDI patterns cropped around the center of the peak. The small oversampling ratio of  $\mathbf{b}_0$  and  $\mathbf{b}_1$  makes such that the central portion already contains the full useful signal and shows the diversity of the dataset. Row from  $\mathbf{a}_1$  to  $\mathbf{d}_1$  shows the corresponding ground truth RSP while the last row shows the RSP predicted by the DL model.



**Figure 6.25:** Corresponding objects from the central RSP patches in Fig.6.24. First and third rows show the modulus and the phase of the ground truth objects while second and fourth rows show the predicted ones. Although the low resolution due to the limited reciprocal space window, the model proves to correctly find the shape and phase distribution of the particle.

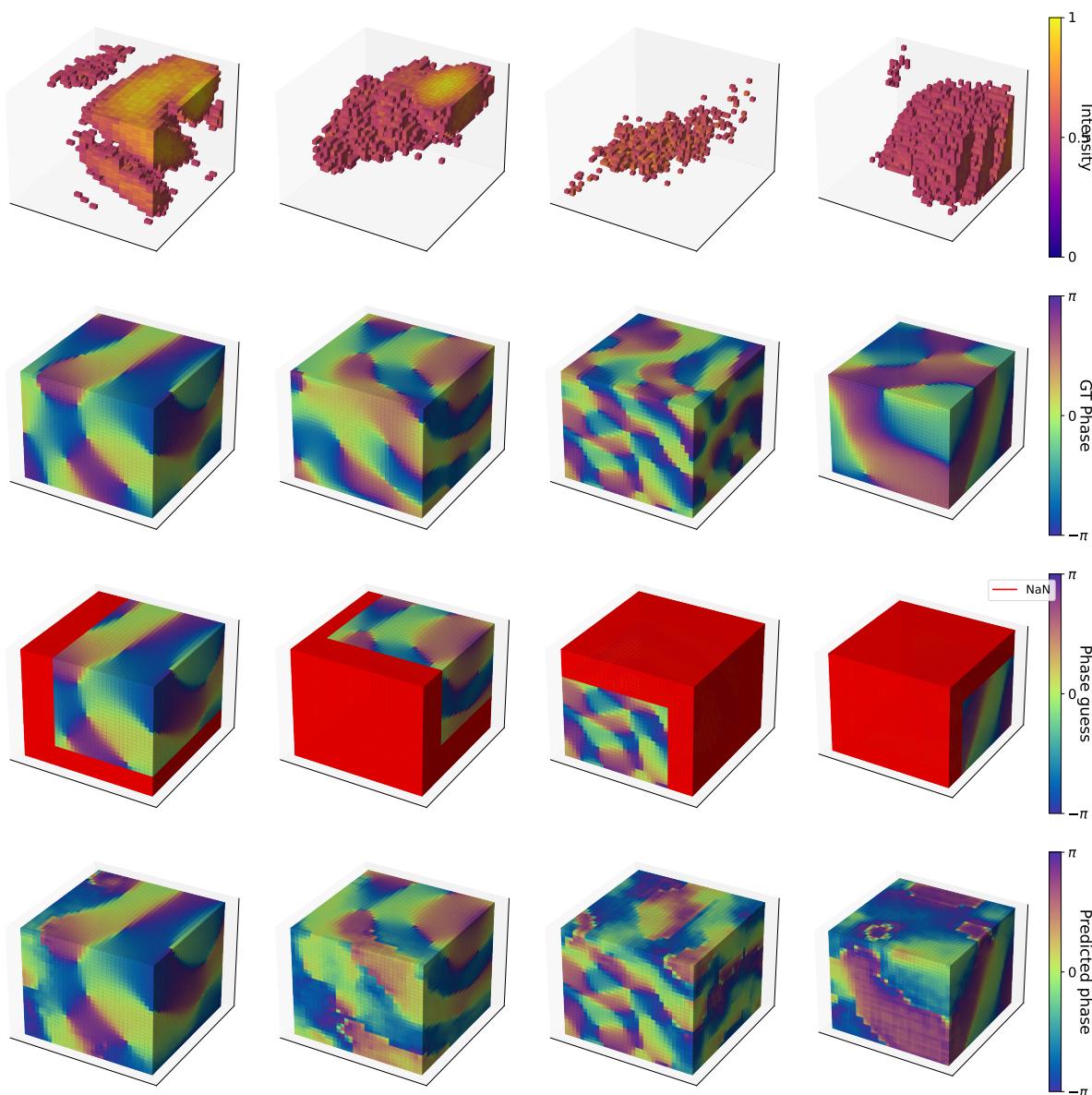
From the results obtained after the training of the model dedicated to the central portion of the simulated BCDI patterns, one can conclude that the model is capable of retrieving the correct RSP for the low strain case, meaning that the leap from the 2D case to the 3D case does not imply unforeseen complications. Moreover, given the diversity of the training dataset the model manages successfully for full peaks contained in a small patch.

2195

### 2196 6.6.5 Results on outer patches

2197 *Outer-CNN* was trained on patches extracted from outer shells over 50 epochs on a dataset  
 2198 containing 50000 samples. Fig. 6.26 illustrates some relevant results. In particular one can  
 2199 observe that the model can predict the RSP in the missing regions providing a relatively smooth  
 2200 transition between the “known” and “unknown” parts. This result is particularly interesting as

it proves that a CNN trained with the WCA loss function learns the map that links a portion of diffracted signal with the corresponding RSP with no information on the particle in real space nor the position of this portion with respect to the center of the diffraction peak. It also shows that the size of the patches contains a sufficient amount of information for this map to be learned. However, one can also notice that some “noise” is present in the predicted regions. These discrepancies from the ground truth become detrimental during the stitching procedure.



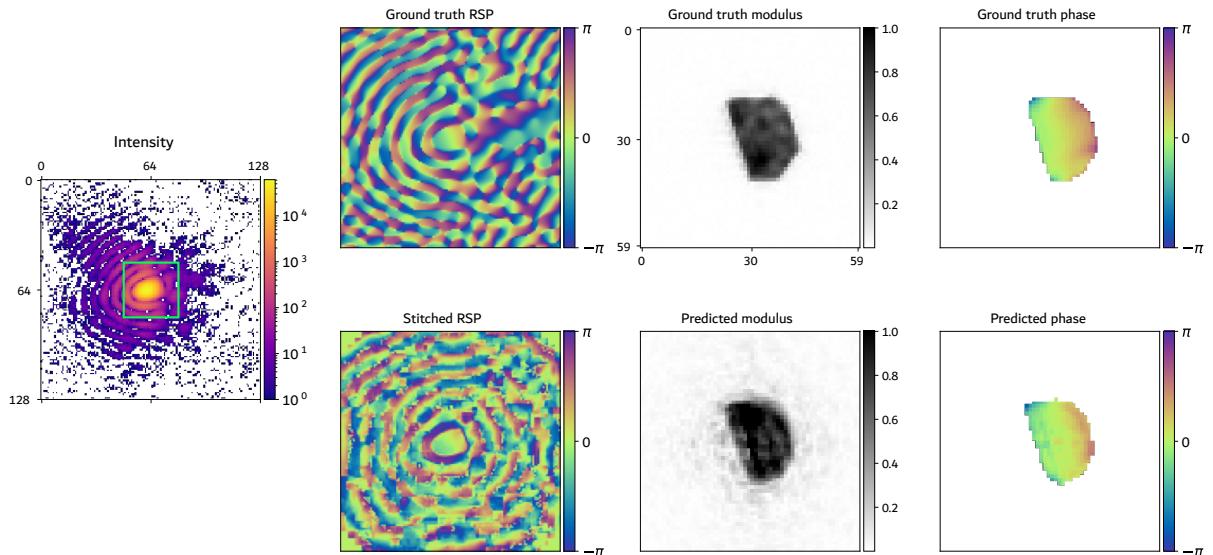
**Figure 6.26:** Examples of RSP prediction for outer patches. The first row shows the central slice of some patches extracted far from the central part of the BCDI peak. The second row shows the ground truth RSP, the third row the RSP initial guess obtained from the overlap with the nearest neighbor of the innermost shell, and the last row shows the DL output. The DL prediction is limited to the missing region (in red).

### 2207 6.6.6 Results on the full RSP - Stitching the patches

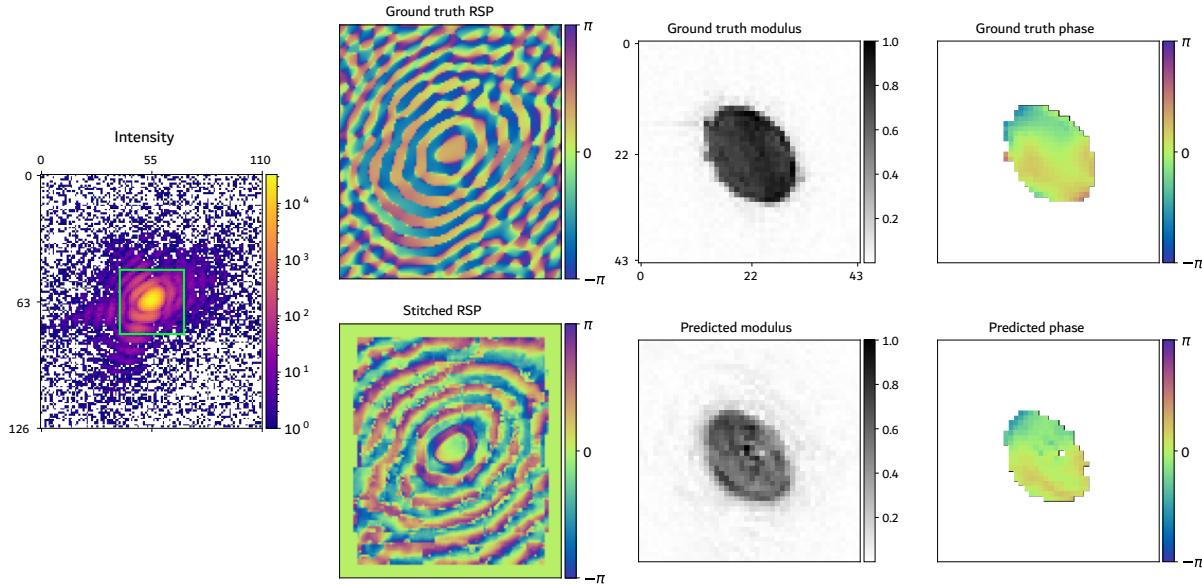
2208 Here the results of the combination of the two CNNs are presented for the low-strain case. Once  
 2209 completed the training the model was tested on full simulated and experimental BCDI pattern.  
 2210 In order to properly retrieve the full RSP corresponding to the diffracted intensity a stitching  
 2211 algorithm for the predicted patches was designed. The stitching takes place progressively start-  
 2212 ing from the central patch and updating the full RSP array shell by shell. Once the prediction  
 2213 of the central patch RSP is obtained from *Central-CNN*, the patches of intensity belonging to  
 2214 the first shell at distance 10 pixels are extracted and the overlapping regions between each  
 2215 patch with the central one are calculated. The predicted central patch RSP for each overlapping  
 2216 region is therefore located in the initial guess RSP array that is given to the model as input  
 2217 paired with the intensity patch. Subsequently, the full batch of pairs  $(I, \varphi_{\text{guess}})_{\text{shell}=1}$  is sent  
 2218 through *Outer-CNN* and the corresponding RSP output is obtained. At this stage two main  
 2219 issues have to be considered, namely: (i) during the training of *Outer-CNN* the initial guess RSP  
 2220 was taken from simulated *ground truth* RSPs while here it is taken from the model's previous  
 2221 prediction itself. It is for this reason that any small unavoidable discrepancies between the  
 2222 predicted and ground truth RSP can lead the model to further errors. (ii) When a round of RSP  
 2223 is predicted there are overlapping regions between patches of the same shell and patches of the  
 2224 previous batch. The most straightforward way to perform the stitching of the patches into the  
 2225 full RSP is to overwrite each time the results. However, although a better approach based on  
 2226 the average of the overlapping prediction was implemented, the issue did not seem to be fully  
 2227 resolved. A first test was conducted with this simpler solutions for both issues, and a more  
 2228 robust approach is briefly discussed at the end of the section.

2229

2230 The crop-predict-stitch method is hence repeated until the last shell. The number of shells  
 2231 and predictions scales with the size of the BCDI data, and it is however always restricted to a  
 2232 spherical region around the Bragg peak. It is therefore less accurate for non-cubic data. Here,  
 2233 Fig.6.27 shows an example of full RSP stitching for a cubic 128 pixel-sided simulated BCDI  
 2234 pattern, while Fig.6.28 reports the analogous result for an experimental data phased with PyNX.



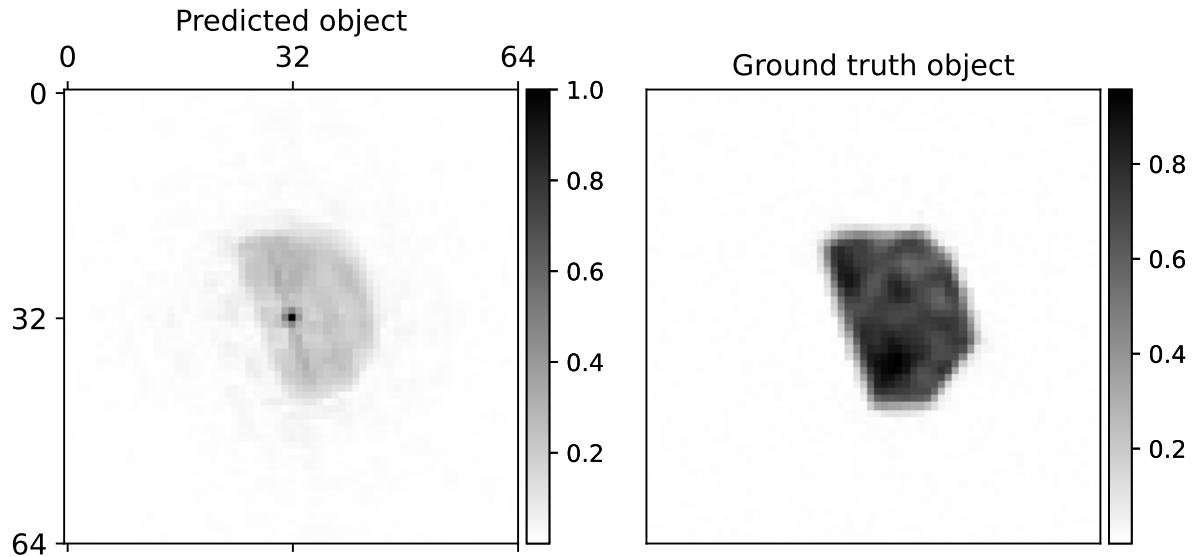
**Figure 6.27: Example 1.** Results of the stitching of RSP predicted patches for 3D simulated data (central slice displayed). The green square on the intensity figure represents the central patch.



**Figure 6.28: Example 2.** Results of the stitching of RSP predicted patches for 3D experimental data (central slice displayed). Here the ground truth quantities are obtained with PyNX phase retrieval of the diffraction pattern.

2235        What emerges from Figs. 6.27 - 6.28 can be summarized as follows:

- 2236        (i) the accurate prediction of the central RSP patch is fundamental to retrieve the low resolution  
2237        estimate of the object
- 2238        (ii) the stitching is problematic already from the first shell outside the central patch, most  
2239        likely because of the two issues pointed out above (initial guess from model prediction and  
2240        RSP averaging of overlapping regions)
- 2241        (iii) the outer RSP patches correctly infer the oversampling ratio as the “thickness” of the  
2242        RSP oscillations matches the one of the diffracted intensity, nonetheless, the preferred overall  
2243        orientation of the fringes seems to be circular.
- 2244        (iv) from the reconstructed object’s modulus a non-physical more intense spot in the center of  
2245        the array had to be filtered out. The occurrence of this spike indicates the presence of wrong  
2246        zero frequency components in reciprocal space, hence wrong zero RSP values. Fig. 6.29 shows  
2247        the object’s modulus before the removal of the high intensity spike located in the center.

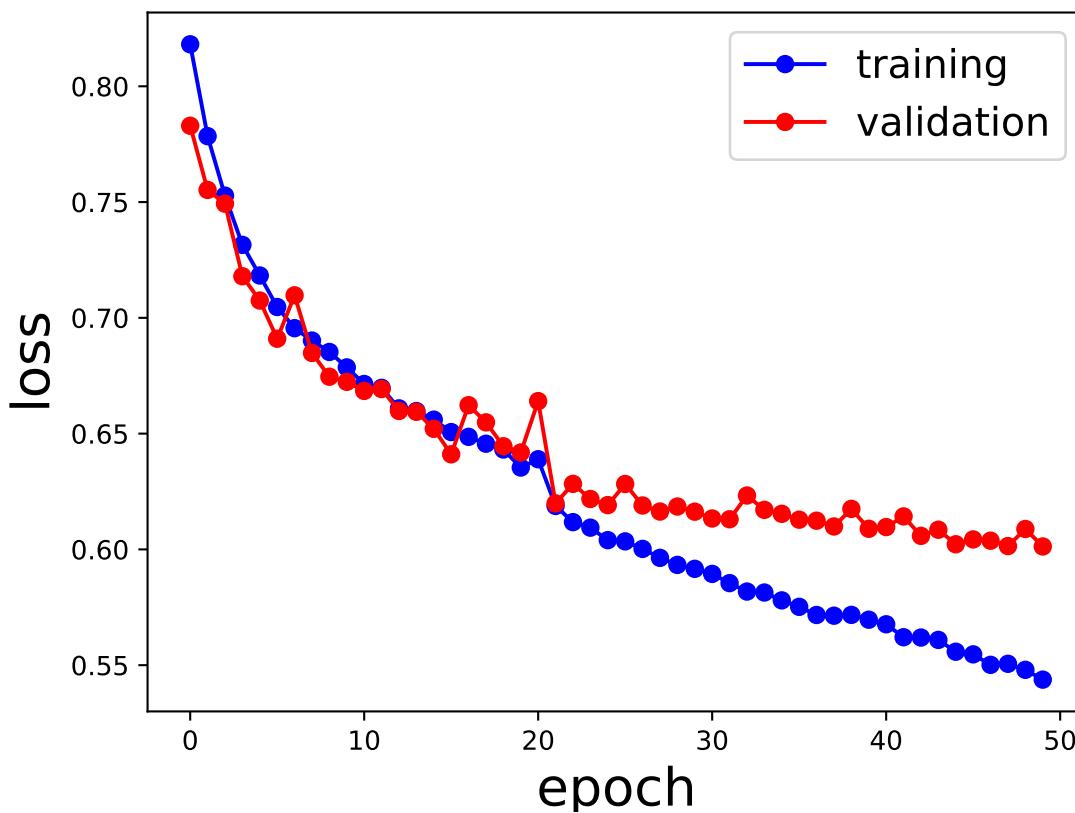


**Figure 6.29:** Predicted object's modulus before the filtering of the zero-frequency component, corresponding to noise in reciprocal space.

## 2248 6.7 Patches: 3D case high strain

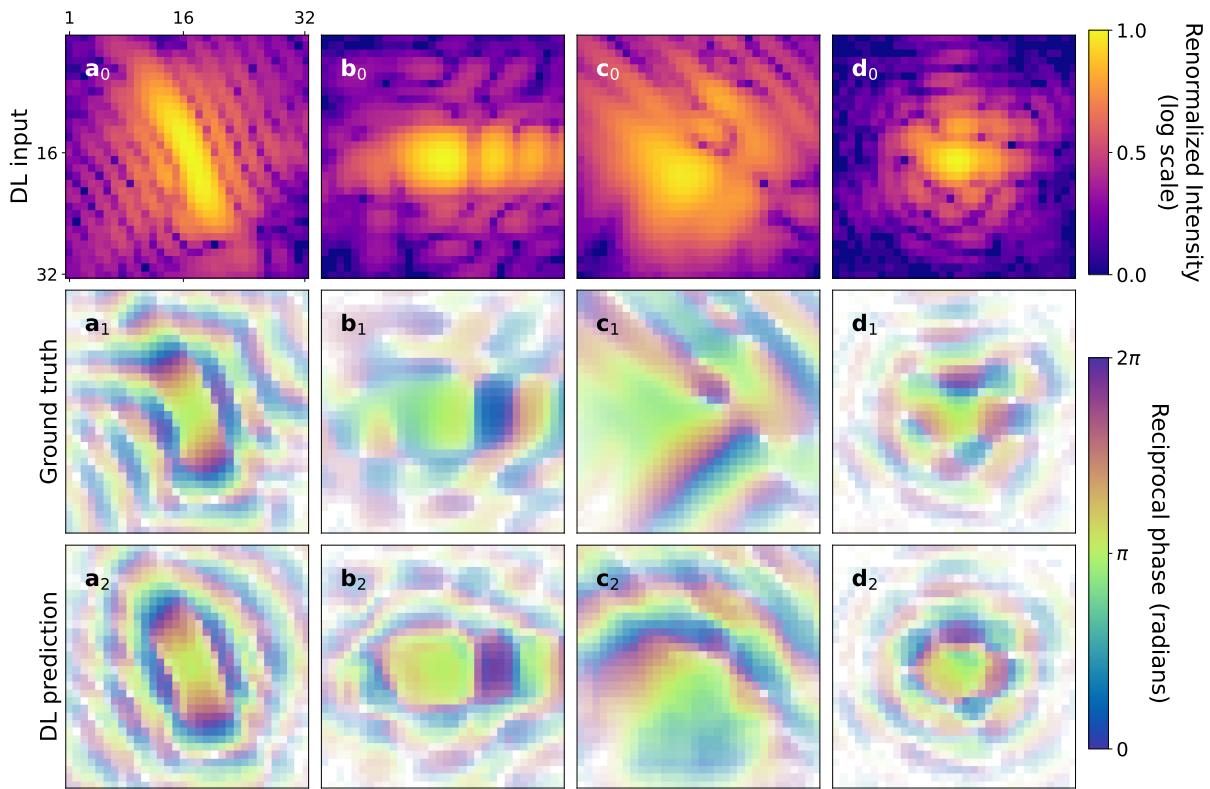
2249 For completeness, the same model has been trained on a dataset containing highly strained  
 2250 simulated diffraction patterns. Each BC DI pattern has been simulated following the procedure  
 2251 explained above in Sec. 5.4.1 and 6.6.2 for a total amount of 50000 for both central and outer  
 2252 patches. The two CNN have been trained for 50 epochs each and then tested on new simulated  
 2253 data.

2254 For what concerns the first CNN trained on central patches it was possible to deduce a  
 2255 more difficult learning from the loss curves. While in Fig. 6.23 the validation loss reaches 0.3 at  
 2256 the end of the training, for Fig. 6.30 it only drops to 0.6 with a significant divergence between  
 2257 training and validation starting from the 20th epoch. A higher loss value is nevertheless  
 2258 expected because of the more complex intensity-RSP mapping for the high strain cases, while  
 2259 the increasing gap between training and validation loss is a signature of poor generalization,  
 2260 early symptom of overfitting.

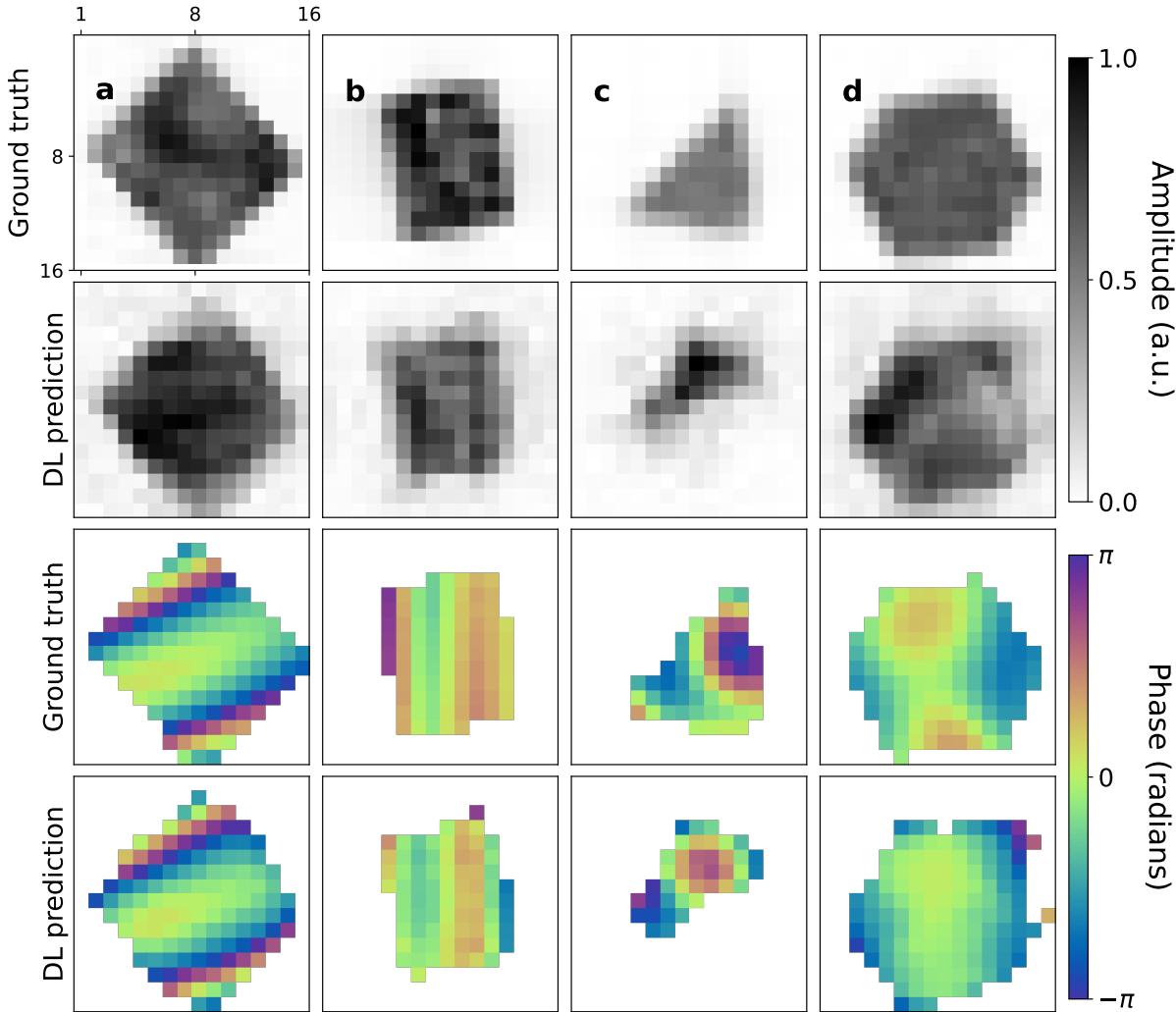


**Figure 6.30:** Training and validation loss trends for the training of *Central-CNN* on central patches of highly strained BCDI patterns. Higher loss values can be observed when comparing with the low strain case (Fig. 6.23) and a beginning of overfitting from the 20th epoch is also visible.

2261 The results on test data show that the model does not always manage to correctly predict  
2262 the RSP, especially when the iso-phase regions are not spherically symmetrical (Fig. 6.31c). It  
2263 however succeeds to retrieve the correct RSP oscillations (Fig. 6.31a) inside the central fringes  
2264 elongated by the high strain. The reconstructions in real space, shown in Fig. 6.32, confirm  
2265 satisfactory results for the first column and poor results for the third column.

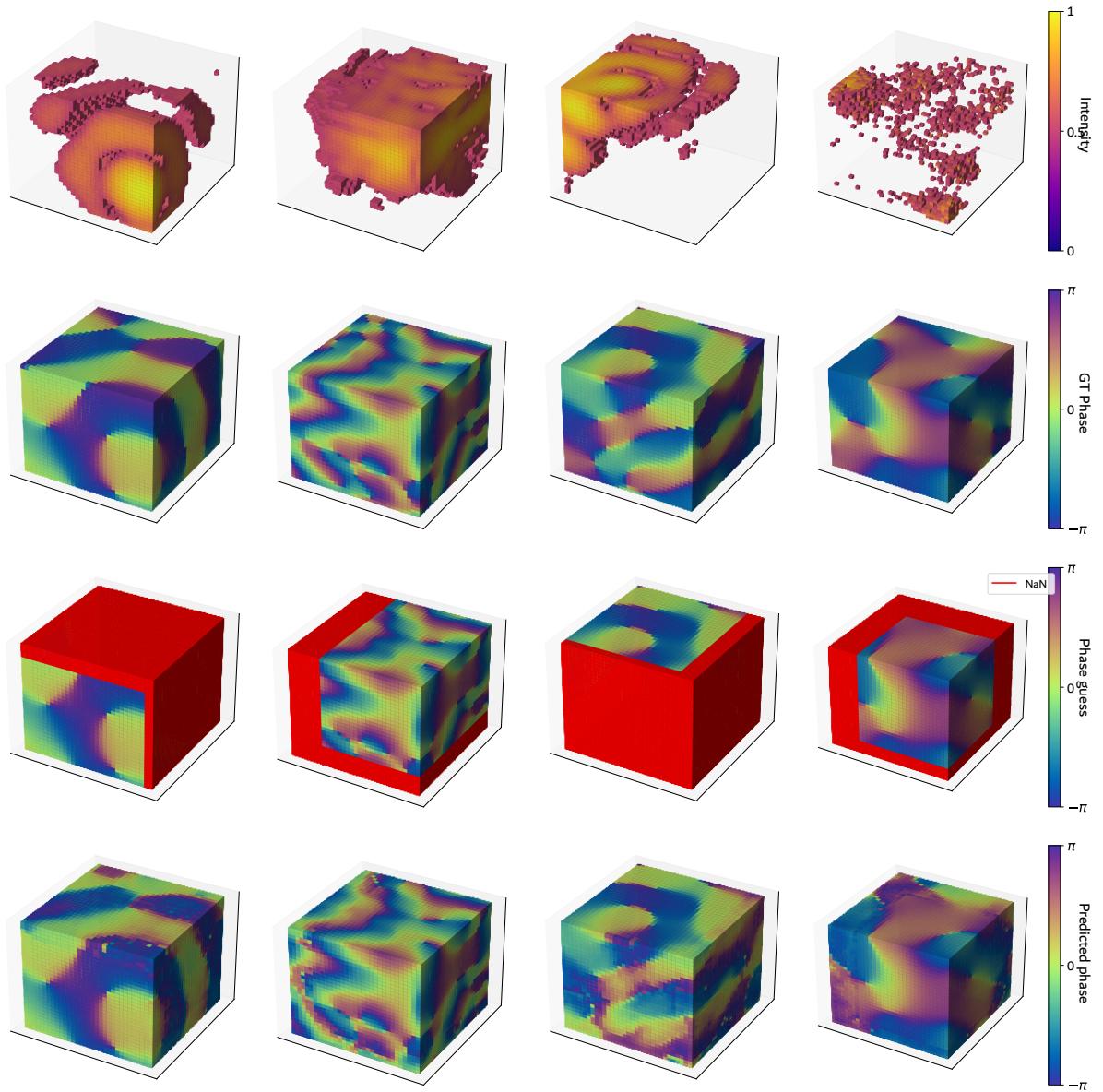


**Figure 6.31:** Slices of the central patches. First row shows four different examples of simulated high-strain BCDI patterns cropped around the center of the peak. Ground truth and predicted RSP are shown in second and third row respectively. The model manages to estimate the correct RSP when this shows spherically symmetric iso-phase regions while struggles more for asymmetries.



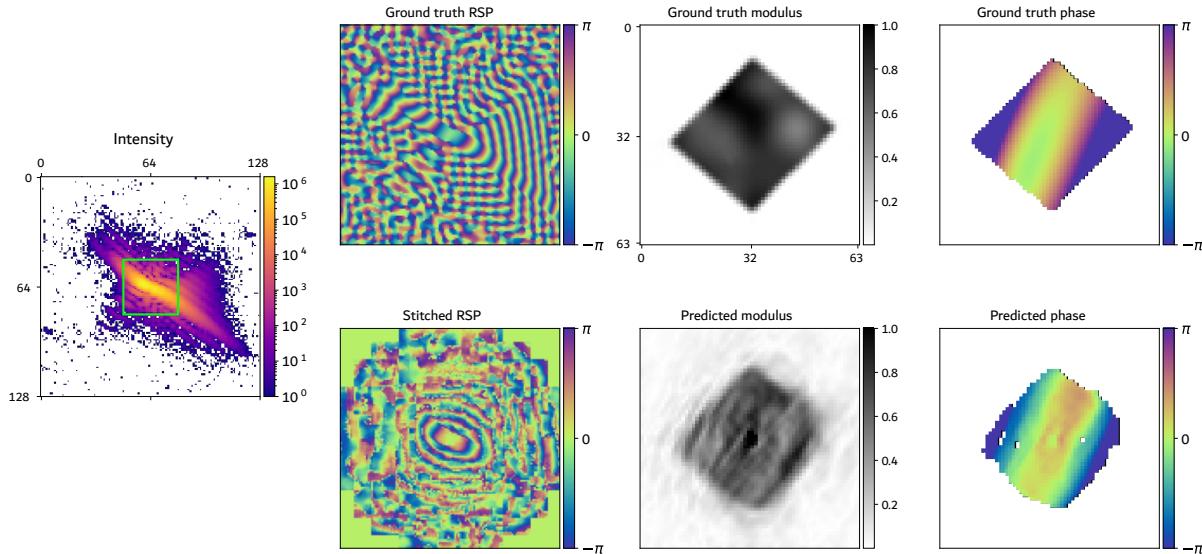
**Figure 6.32:** Corresponding reconstructed objects. Except for case **c** the predicted RSP is good enough to well estimate the size, shape and phase of the real space object.

Regarding the *Outer-CNN* trained on outer patches of highly strained patterns instead, one can observe that the performance is not severely affected by the high-strain. Strong discrepancies between predicted and ground truth RSP are mostly present where there is no intensity signal, thus not relevant (see Fig.6.33). The good accuracy of the outer patches predictions suggests that the crucial and more challenging map to retrieve is the central patch.



**Figure 6.33:** Examples of RSP prediction for outer patches cropped from simulated high-strain data. Similarly to the examples shown in Fig. 6.26 for the low-strain case, the model yields relatively correct outputs. Worse predictions are observed where low intensity signal is recorded, therefore less important during the reconstruction.

2271 For completeness, the result of the full RSP stitching and the corresponding reconstructed  
 2272 object for the high-strain case is shown in Fig. 6.34. A simulated test data has been used for  
 2273 the ground truth comparison. It is clear that the stitching algorithm is performing poorly as  
 2274 observed for the low-strain case. However, the central RSP patch is fairly similar to the ground  
 2275 truth and therefore the low resolution reconstructed object shows roughly the correct shape  
 2276 and phase.



**Figure 6.34:** Results of the stitching of RSP predicted outer patches for 3D simulated “high-strain” data (central slice displayed). Compared to Fig.6.28 it is visible how the stitched RSP is more “confused” outside the central patch. The less regular and symmetric fringes’ spatial arrangement is not correctly recovered during the stitching process. However, the discrete accuracy of the central patch allows for a close low-resolution estimate of the object’s modulus and phase.

### 2277 6.7.1 Discussion

2278 Although it failed, this study on the prediction of the RSP in smaller patches has led to a better  
 2279 understanding of the problem and nevertheless unveiled some interesting insights. For instance,  
 2280 it showed that the retrieval of the mapping between patches is possible with a CNN trained  
 2281 with the WCA loss function, untying the relationship with the real space object. Moreover, it  
 2282 emerged that the main difficulty of this approach is given by the stitching of the RSP patches  
 2283 into the full array. As mentioned above, the hypothesized reasons for this problem are (i) the  
 2284 fact that the model “sees” simulated ground truth RSP guesses during the training and predicted  
 2285 ones during inference, and (ii) the averaging of RSP predictions for overlapping voxels. In  
 2286 order to overcome these limitations other approaches were contemplated but never realized  
 2287 for lack of time. In particular, it was imagined a way to extract, analyze and stitch the patches  
 2288 inside the model into a sort of Recurrent Convolutional Neural Network (RCNN) that would  
 2289 keep track of the previous innermost shell thanks to a dedicated convolutional Long-Short  
 2290 Term Memory (LSTM) [189]. By doing this the model would always be exposed to its own  
 2291 RSP predictions as initial guess for outer patches and the RSP average over overlapping voxels  
 2292 could be replaced with a convolutional layer with non-linear activation function. While the  
 2293 attempts of setting up such model are not reported here, it is mentioned the idea as possible  
 2294 inspiration for future works.

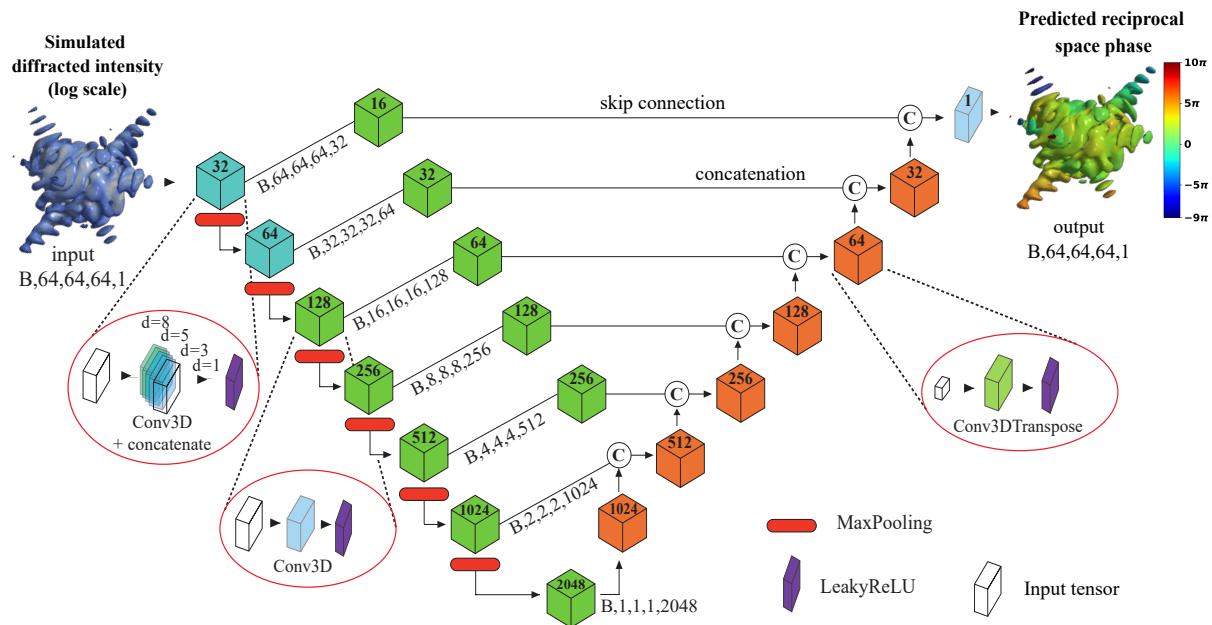
2295

2296 To conclude, main finding of this study on patches is that it is crucial for a good object  
 2297 estimate to accurately predict the RSP in the vicinity of the center of the Bragg peak, and that  
 2298 the CNN model trained with the WCA can accomplish this task for highly strained patterns as  
 2299 well. It was therefore decided to invest the efforts into a regular CNN for the prediction of the  
 2300 RSP of 3D highly strained BCDI patterns with the intermediate size of  $64 \times 64 \times 64$  pixels.

## 2301 6.8 Final model design: 3D case high-strain for central 2302 patch only

2303 Following from the results and considerations relative to the patching approach it was decided  
2304 to focus on the prediction of the central patch only. The size has been increased to  $64 \times 64 \times 64$   
2305 pixels to include more features. Moreover, this is the size employed by most of the recent  
2306 works presented in the beginning of the chapter [166, 181, 182]. Similarly to the previous  
2307 cases, the approach was to create a dataset to train the model with, in a supervised manner  
2308 with the WCA loss function. Only highly strained particles were considered, as it is for those  
2309 that a successful DL model can truly help the PR. The model here described and the results  
2310 obtained on simulated and experimental 3D BCDI patterns can be found entirely in the paper  
2311 "Phase Retrieval of Highly Strained Bragg Coherent Diffraction Patterns using Supervised  
2312 Convolutional Neural Network" [174].

2313 The architecture that was employed is an adaptation of the 3D U-Net employed for smaller  
2314 patches Fig. 6.35 As learned from the preliminary study on the 2D case, to better interpolate the  
2315 diverse distribution of distorted BCDI patterns, the number of trainable parameters and samples  
2316 in the training dataset was increased significantly. Additional encoder and corresponding  
2317 decoder blocks were added for a total of 145 million parameters. Moreover, similarly to the  
2318 model used for the gap inpainting, dilated convolutions were adopted in the first two encoder  
2319 blocks in order to improve the receptive field of the model.



**Figure 6.35:** Schematic of the 3D U-Net model employed for the RSP prediction of highly strained patterns. The model maintains the structure already shown in Fig. 6.22, adapted for  $64 \times 64 \times 64$  pixels data. Moreover additional convolutional blocks have been added between skip connections, to build a deeper layer configuration, with more trainable parameters.

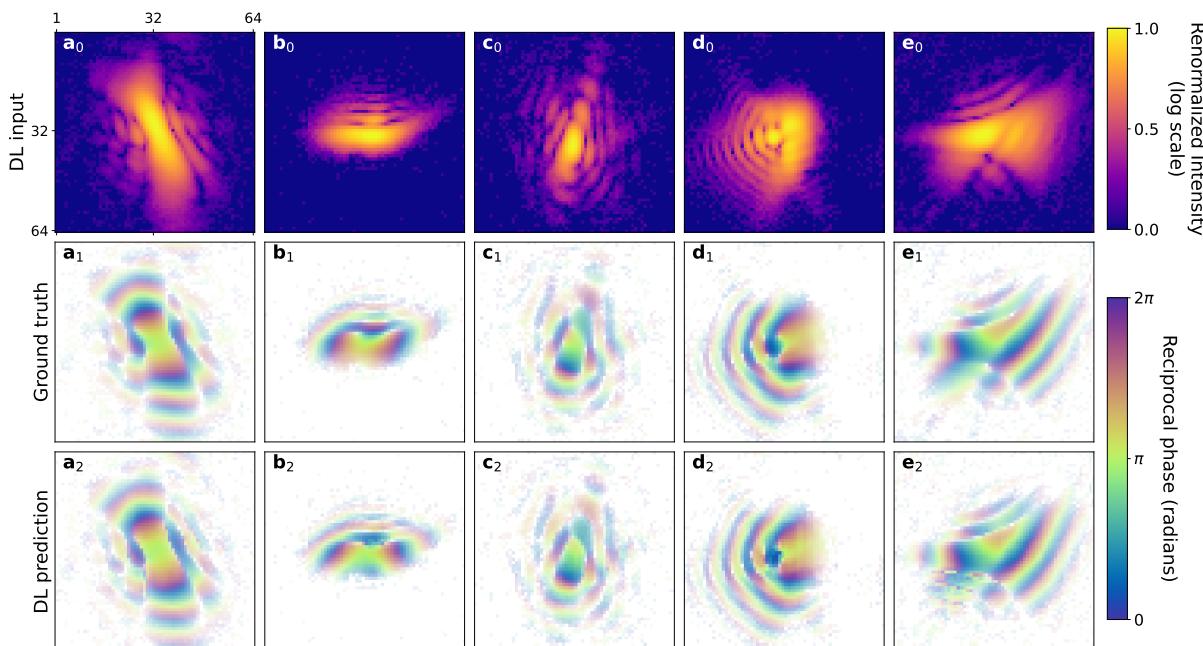
2320 95000 simulated BCDI patterns have been created following the procedure described in  
2321 Sec.5.4.1 on a cubic 64 pixel-sided grid for different particle's shape, strain distribution, over-

2322 sampling conditions and noise levels. Another smaller dataset containing 4000 samples was  
 2323 created instead for testing the model.

2324 The model was trained with the WCA loss function for 60 epochs with a learning rate of  
 2325  $10^{-4}$ . To speed up the process, the training has been conducted using two NVIDIA TeslaV100-  
 2326 SXM2-32GB GPUs using the MirroredStrategy feature for synchronous training across multiple  
 2327 devices provided by the Tensorflow library. This measure allowed to reduce the training time  
 2328 from 2 hours per epoch to 30 mins.

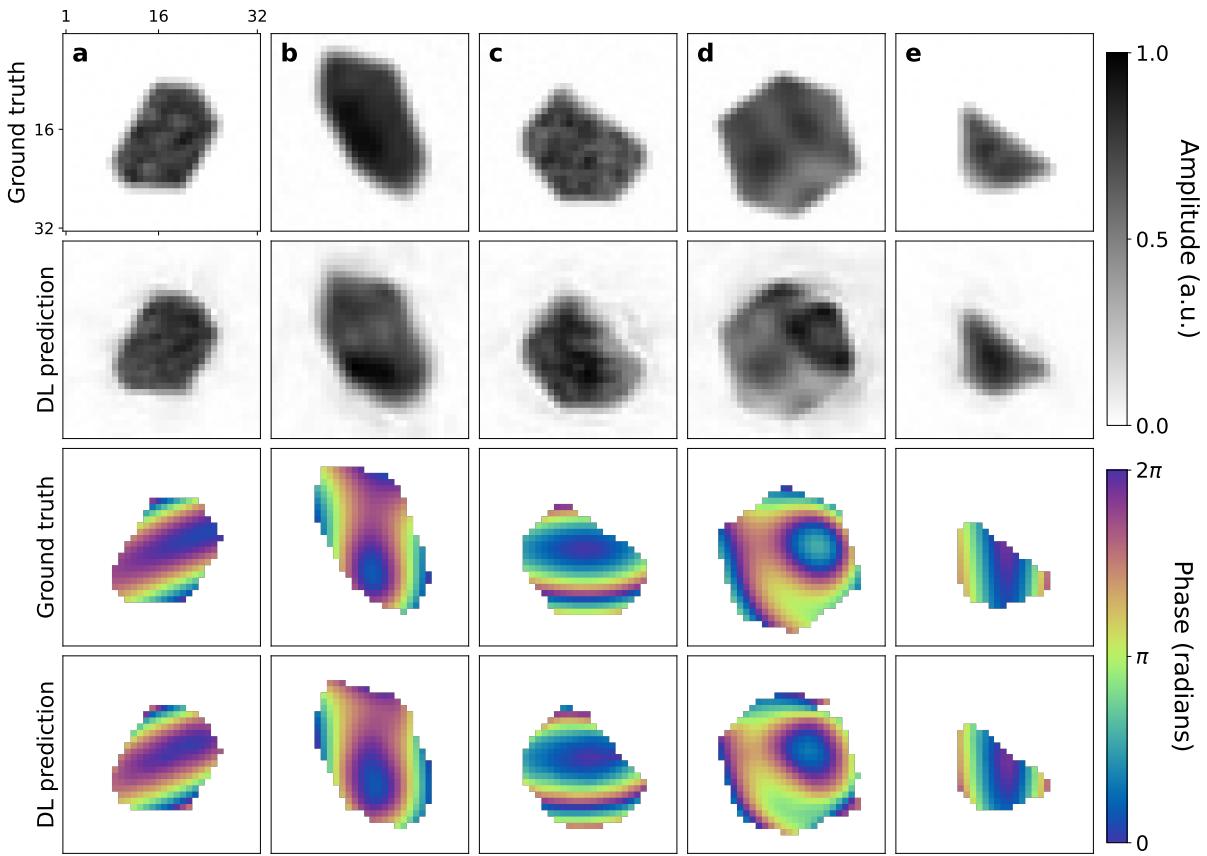
### 2329 6.8.1 Results: simulated data

2330 The model has been tested on simulated data. Figs.6.36 - 6.37 illustrate some examples of  
 2331 comparison between ground truth and predictions for both the RSP and the reconstructed  
 2332 objects respectively.



**Figure 6.36:**  $a_0 - e_0$ ) Central slices of simulated input intensities from the test dataset.  $a_1 - e_1$ ) Corresponding ground truth RSP.  $a_2 - e_2$ ) Corresponding slices taken from the Deep Learning (DL) model prediction of the RSP. The RSP are displayed with an opacity filter that highlights regions with diffracted intensity.

2333 The model correctly predicts the RSP oscillations inside the fringes even when they have  
 2334 been distorted or merged into a single continuous intensity stripe (Fig. 6.36 a-b) because of the  
 2335 high strain along the corresponding axis in real space.



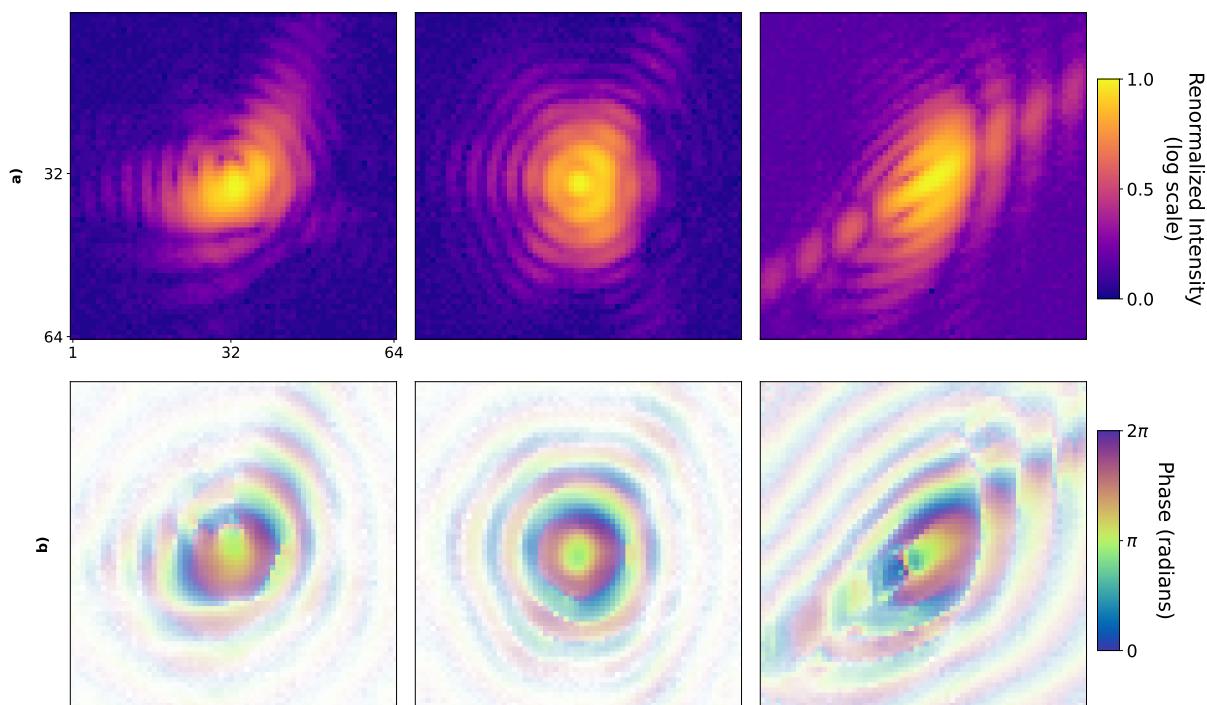
**Figure 6.37:** Reconstructed objects from the diffraction patterns in Fig. 6.36. **a-e)** Central slices of both the modulus and phase for the ground truths and the DL reconstructions.

2336     Except for some noise and inhomogeneities in the objects' moduli, the reconstructions from  
 2337     the predicted RSP achieve good accuracy on simulated data, for different particle shapes and  
 2338     strain distributions. It is worth mentioning that the presence of noise affecting the objects'  
 2339     shapes is an effect of the loss function that is not computed in real space. The model is never  
 2340     directly shown that the real space object are compactly supported, as it comes as consequence  
 2341     of the correct RSP prediction. However, while small discrepancies of the RSP prediction result  
 2342     in noise on the object shape, an overall accurate RSP prediction ensures the retrieval of both the  
 2343     correct shape and phase of the object. This fact is of primary importance when considering the  
 2344     use of the DL prediction as starting point of iterative refinement with conventional algorithms.  
 2345     It is indeed easier to reach convergence from a low resolution noisy but accurate estimate of  
 2346     the object rather than a clean but inaccurate one. Moreover, in experimental conditions the  
 2347     shape can be known via other techniques (Scanning Electron Microscopy) while the strain field,  
 2348     derived from the retrieved object's phase, is the ultimate interesting result of the reconstructions.

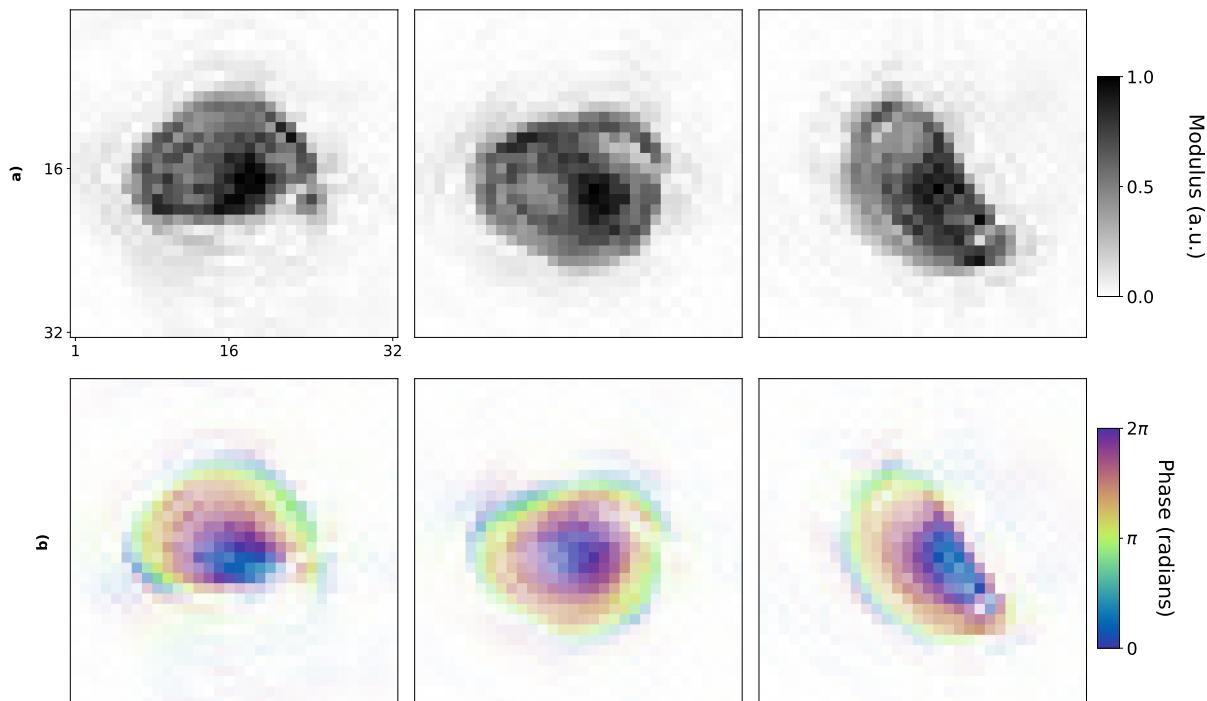
### 2349     6.8.2 Results: experimental data

2350     Given the satisfactory results obtained on simulated data the DL model was also tested on  
 2351     experimental data. Two relevant examples of BCDI patterns collected at the ID01 beamline  
 2352     of the ESRF-EBS are considered here. The first pattern (Fig. 6.38 a) is given by a platinum  
 2353     nanoparticle on Yttria-stabilized zirconia (YSZ) (Particle 1) while the second (Fig. 6.40 b) is a

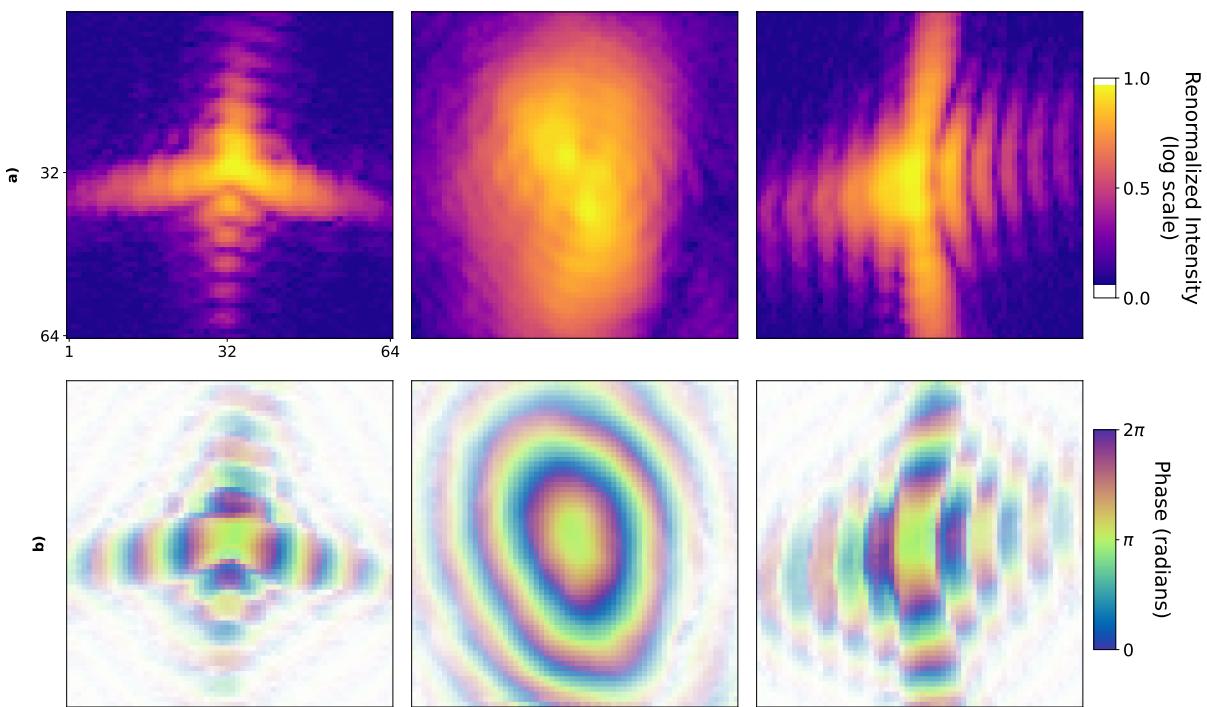
2354 dewetted platinum/palladium bilayer on a sapphire substrate (Particle 2).



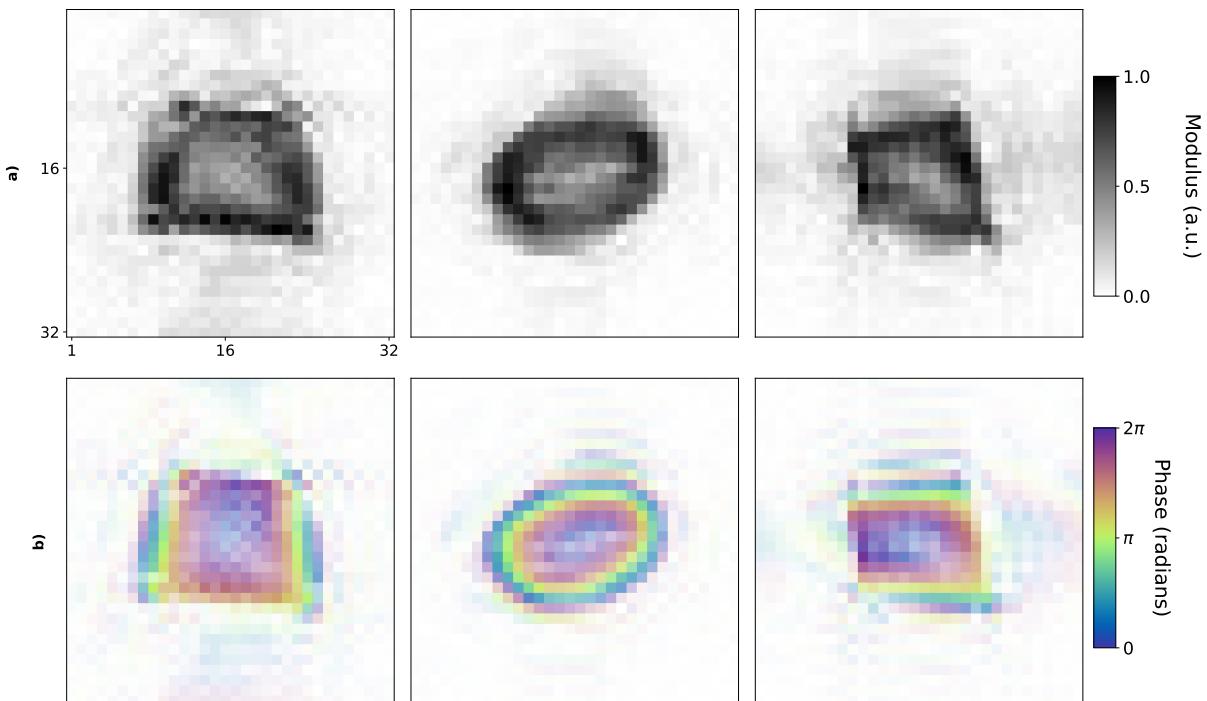
**Figure 6.38:** a) The three central slices of the 3D experimental diffraction pattern measured at ID01 from the Pt sample (Particle 1). The images have been transformed in log-scale and renormalized between 0 and 1, ready to be processed by the DL model. b) Corresponding slices of the DL model RSP predictions.



**Figure 6.39:** Corresponding three central slices of the reconstructed Particle 1 modulus (a) and phase (b). Each slice has been cropped around the particle for better visualization.



**Figure 6.40:** **a)** Central slices of the 3D experimental diffraction pattern measured at ID01 from the Pt/Pd sample (Particle 2). Note the shape of the diffraction pattern with broken centro-symmetry; signature of the high-strain. **b)** Corresponding slices of the DL model RSP predictions.



**Figure 6.41:** Central slices of the reconstructed Particle 2, modulus **(a)** and phase **(b)**

Fig. 6.39 and 6.41 show the reconstructed object obtained from the predicted RSP. Despite the low resolution and the presence of noise it is possible to recognize the shape of realistic particles (compact support and smooth phases).

2358 In these cases of PR of experimental data it is impossible to establish a comparison with  
2359 a ground truth because it is not available. However, the comparison has been made with the  
2360 results obtained with standard iterative algorithms.

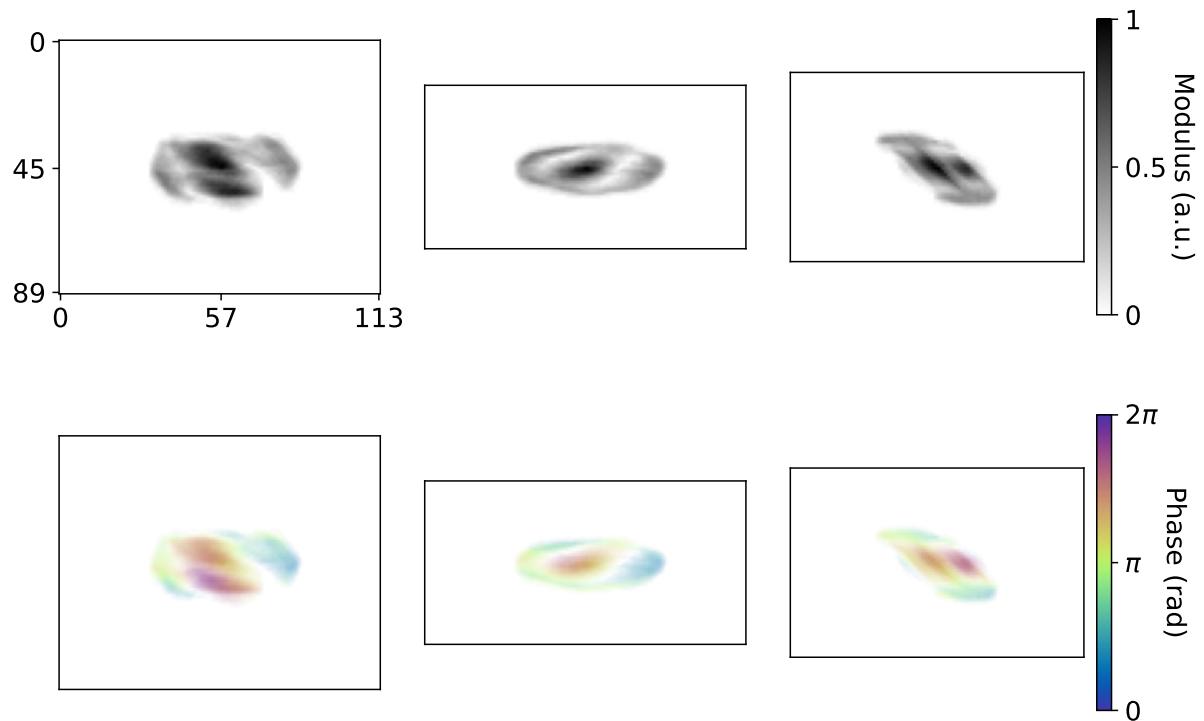
2361 For this, 60 independent and randomly initialized runs have been launched with PyNX in  
2362 order to reconstruct these datasets but no satisfactory results were obtained. Precisely, the  
2363 recipe of 400 HIO + 1000 RAAR + 300 ER and different thresholds for the support estimation  
2364 were used for each run (see Table 6.2). Moreover, the 3 best results according to the Mean-to-  
2365 Max metric [190, 191] were combined with mode decomposition [192] to improve the quality  
2366 of the reconstruction. This last passage, through a Singular Value Decomposition (SVD) of  
2367 the selected reconstructions, allows to estimate the reproducibility of the result by assessing  
2368 the magnitude of the first eigenvalue (first mode). Values above 90% typically imply good  
2369 reproducibility, often associated to good convergence.

2370 However, in both cases the obtained reconstructions were not satisfactory as holes were  
2371 present (Fig.6.42), or the support was excessively shrunk (Fig.6.44).

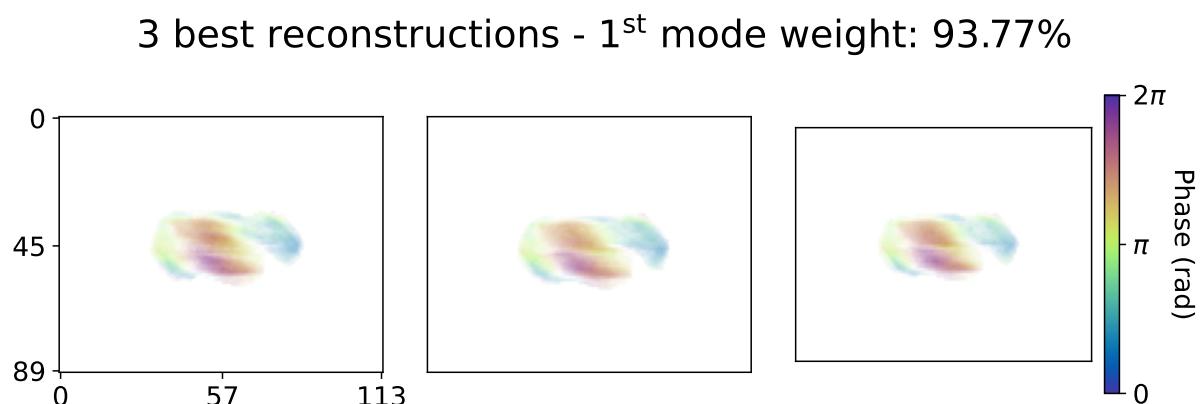
2372

Parameter	Value
obj	autocorrelation
support_autocorrelation_threshold	(0.09, 0.11)
recipe	400 HIO + 1000 RAAR + 300 ER
nb_runs	60
support_update_period	50
support_threshold	(0.15, 0.25)
update_border_n	3
post_expand	(1, -2, 1)

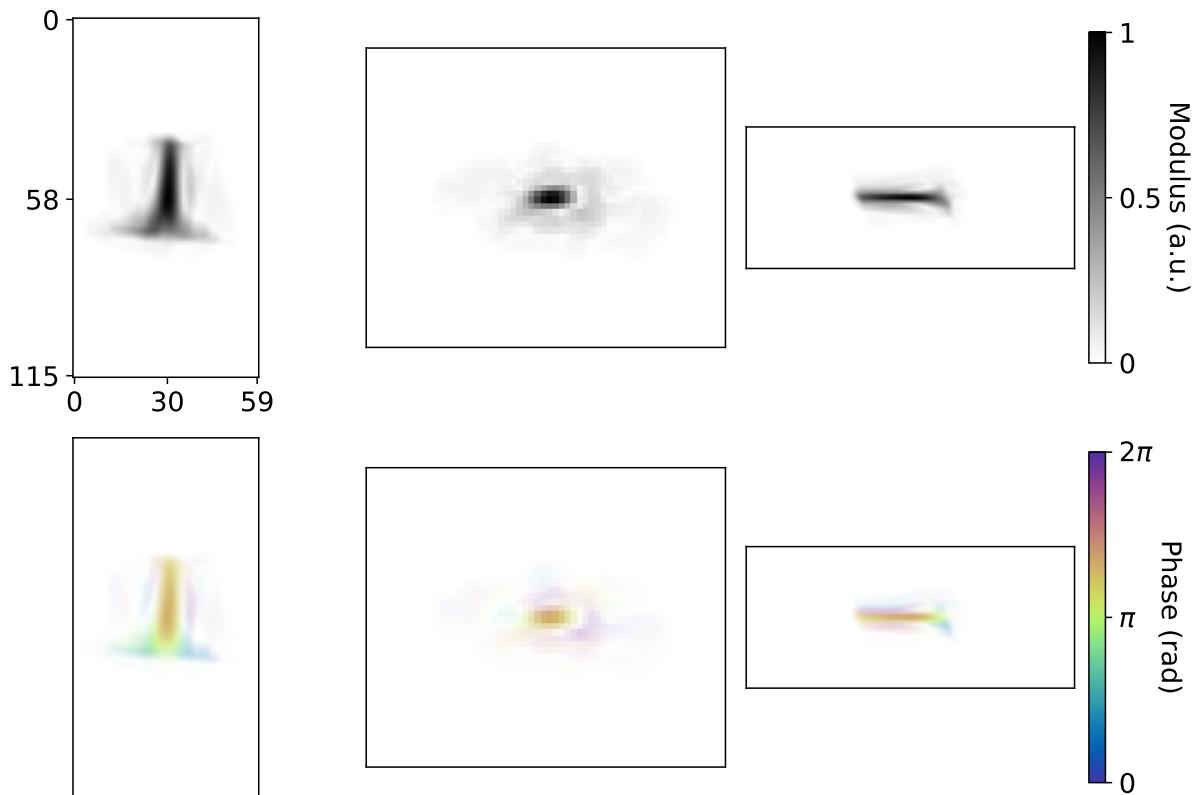
**Table 6.2: PyNX parameter settings for standard PR.** Each of the 60 independent runs starts from an object obtained from the inverse Fourier transform of the diffraction pattern (autocorrelation). Pixels above the autocorrelation's maximum multiplied by a random threshold between (0.09 - 0.2) different for each run, are included in the support. In each run 400 cycles of HIO are followed by 1000 of RAAR and 300 of ER. The support is updated every 50 iterations using the shrink-wrap algorithm [94]. During any of these updates the pixels above the object's modulus maximum multiplied by a random value between (0.15 and 0.25) are included in the support. Moreover, only a shell of +/- 3 pixels around the outer border of the support are updated to avoid too large variations, and a sequence of expansion, shrinking and re-expansion by (1,-2, 1) pixels respectively is evaluated at each update step, in order to make the estimate more robust.



**Figure 6.42:** Combination of the 3 best reconstructions of Particle 1 out of 60 independent runs with PyNX (central slices). Although the shape is guessed the presence of holes in the modulus and domains in the phase denotes a poor quality of the result.

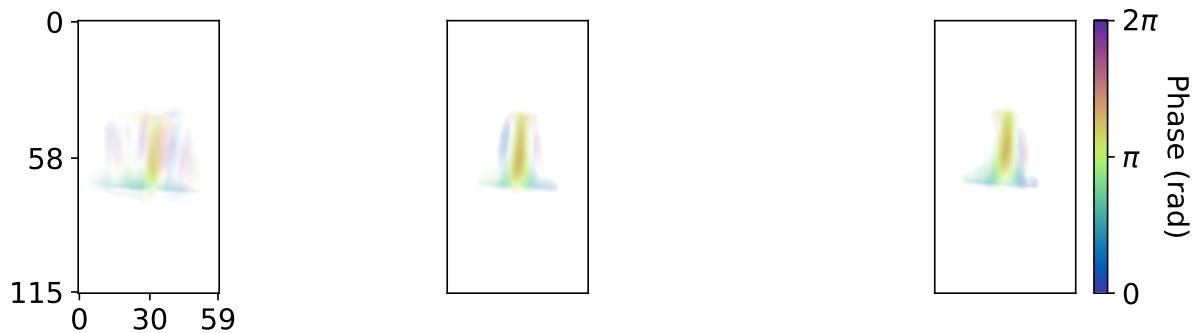


**Figure 6.43:** Phase of the three best reconstructions of Particle 1. Despite the similarity of the solutions, supported by the large weight of the first SVD eigenvalue [192], each of them and their combination presents amplitude dips and phase domains not expected from the studied Pt sample.



**Figure 6.44:** Combination of the 3 best reconstructions of Particle 2 out of 60 independent runs with PyNX (central slices). The high strain induced by the substrate deceives the standard PR algorithm that excessively shrinks the object's support.

3 best reconstructions - 1<sup>st</sup> mode weight: 72.68%



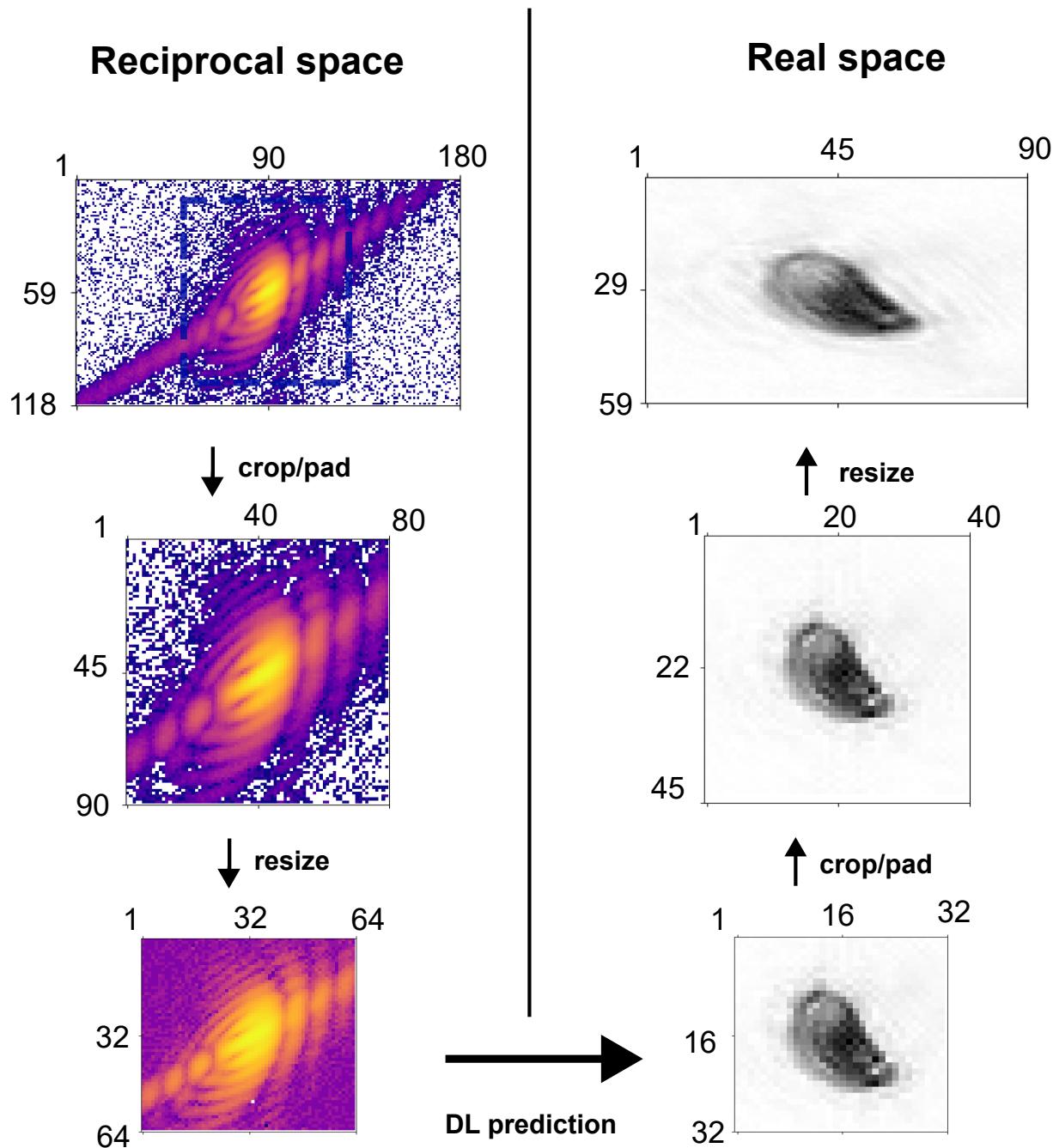
**Figure 6.45:** Phase of the three best reconstructions of Particle 2. Low reproducibility of the result is highlighted by the diversity of the reconstructions, supported by the relatively low first SVD eigenvalue.

## 2373 6.9 Refinement with iterative algorithms

2374 It has been shown in Figs. 6.42 - 6.44 how the PR of highly-strained particles with conventional  
 2375 iterative algorithms and standard routines can be challenging. In order to achieve successful  
 2376 reconstructions, fine parameter tuning operated by expert users as well as large numbers of

2377 runs are often required. On the contrary, the DL approach, while fast at inference, is limited  
2378 to low-resolution reconstructions and only generates outputs based on the approximation of  
2379 the statistical distribution of simulated BCDI patterns. For this reason, the DL model for the  
2380 phasing of experimental data needs be used as a preprocessing step that can provide a starting  
2381 point for further iterative refinement. Few iterations of ER steps can be used to polish the  
2382 DL estimate, simplifying the convergence, assumed that the DL estimate is located near the  
2383 solution, in the search space. In this way, the DL model can play a crucial role in guiding the  
2384 search toward the correct solution neighborhood, significantly reducing computational costs  
2385 and overall wall-clock time.

2386 At this stage it is worth mentioning that, since the DL model was trained on 64 pixel-size  
2387 cubic volumes, it was necessary to crop and resize the experimental diffraction patterns. In  
2388 fact, especially for large BCDI data, a resizing or binning step before cropping is often needed,  
2389 such that a sufficient amount of data is contained in the 64 pixel-size volume. This procedure  
2390 requires special attention as for different cropping/resizing parameters the DL prediction  
2391 varies. In particular, it is curious to notice that the cropping/padding operations in reciprocal  
2392 space correspond to interpolations in real space and vice versa. For this reason the procedure  
2393 illustrated in Fig.6.46 had to be adopted to adapt the data to the DL format and then bring back  
2394 the predicted object to the size relative to the original data one. This step is fundamental when  
2395 the DL object is then used as starting point for further refinement with classical algorithms.  
2396 Due to the large diversity of the region of interest (ROI) from one experimental BCDI pattern  
2397 to the other, the manual intervention is for the moment needed to adjust the cropping and  
2398 binning parameters such that the data fed into the DL model resembles the simulated data used  
2399 for training.



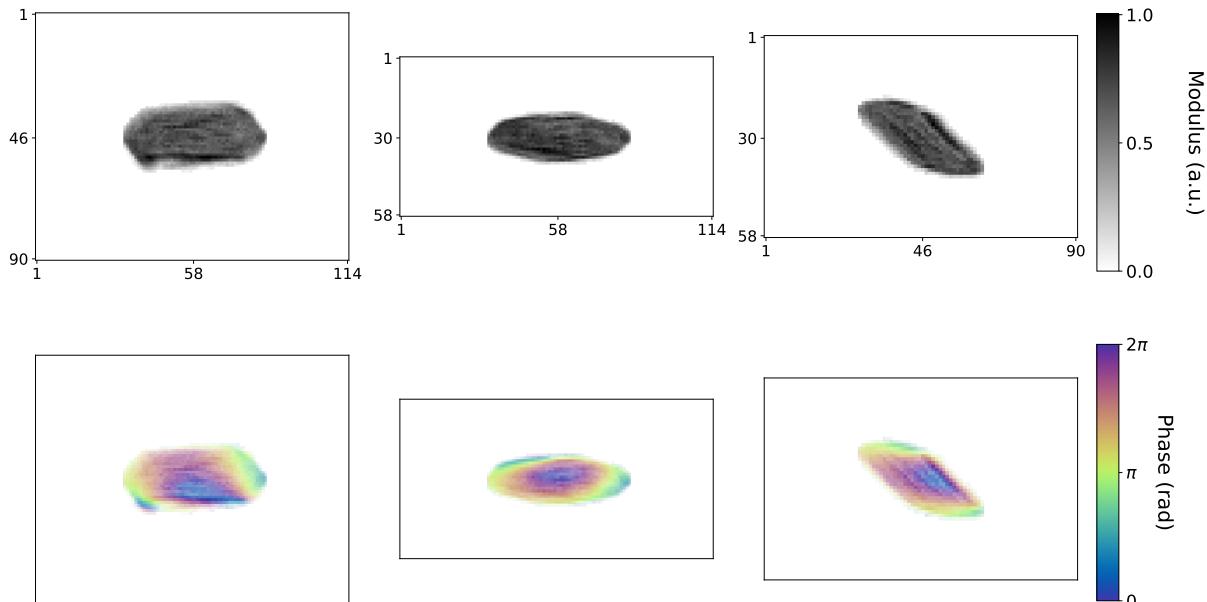
**Figure 6.46:** Manipulation of the datasets. Here the experimental BCDI pattern is firstly cropped around the COM of the Bragg peak, cutting out parts of the signal. The data is then interpolated into a 64 pixel sided cubic grid and transformed in normalized log-scale. The object obtained after the DL RSP prediction is then padded and resized back to the original shape such that it can directly be plugged into an iterative algorithm for refinement. Note that the size of the objects are already shown in boxes of half the size with respect to the corresponding diffraction patterns.

If the DL predicted object is a good low-resolution guess of the solution, few steps of ER are usually enough to reach convergence. In this specific case 300 iterations of ER were performed and, since the estimated DL object is assumed to be close to the solution, only the pixels at the border the support were allowed to be updated (see Table 6.3). Figures 6.47 and 6.48 show the results obtained after refinement. Theoretically, one could tune the number of

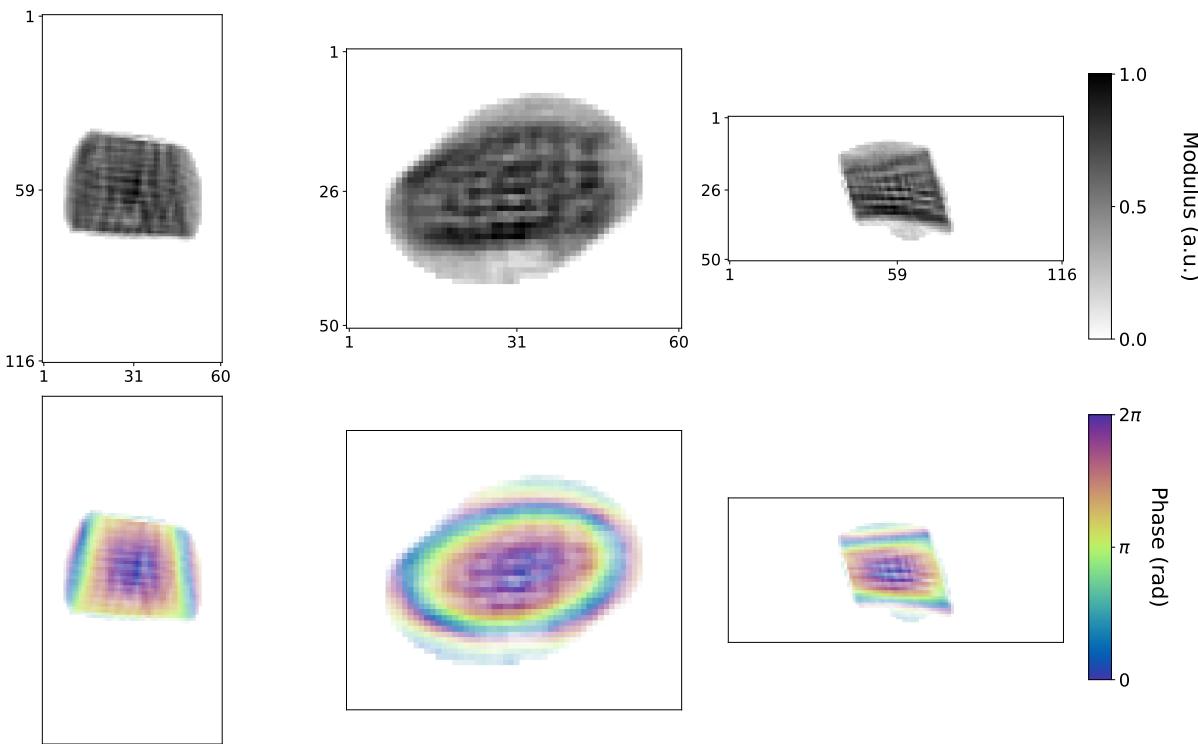
pixels updated at the border of the support depending on the resolution mismatch between the DL data ( $64 \times 64 \times 64$ ) and the full BCDI size. The larger the size difference the more the pixels at the border and vice-versa. However, it was found that a single pixel at the time is often the best choice.

Parameter	Value
obj	DL_obj
recipe	300 ER
nb_runs	1
support_threshold	0.3
support_update_period	50
update_border_n	1
post_expand	(1, -1)

**Table 6.3: PyNX parameter settings for the refinement after the DL prediction** A single run of 300 ER iterations using a support threshold of 0.3 with respect to the maximum of the object's modulus for the support update. This last is performed with the shrink-wrap algorithm, every 30 iterations. At this stage, the only pixels affected by the support updated lie within  $\pm 1$  pixels around the outer border of the support. Moreover, the support is also shrunk and expanded by 1 pixel at each update step.



**Figure 6.47:** Central slices of Particle 1 after 300 cycles of ER for refinement of the DL initial estimate.

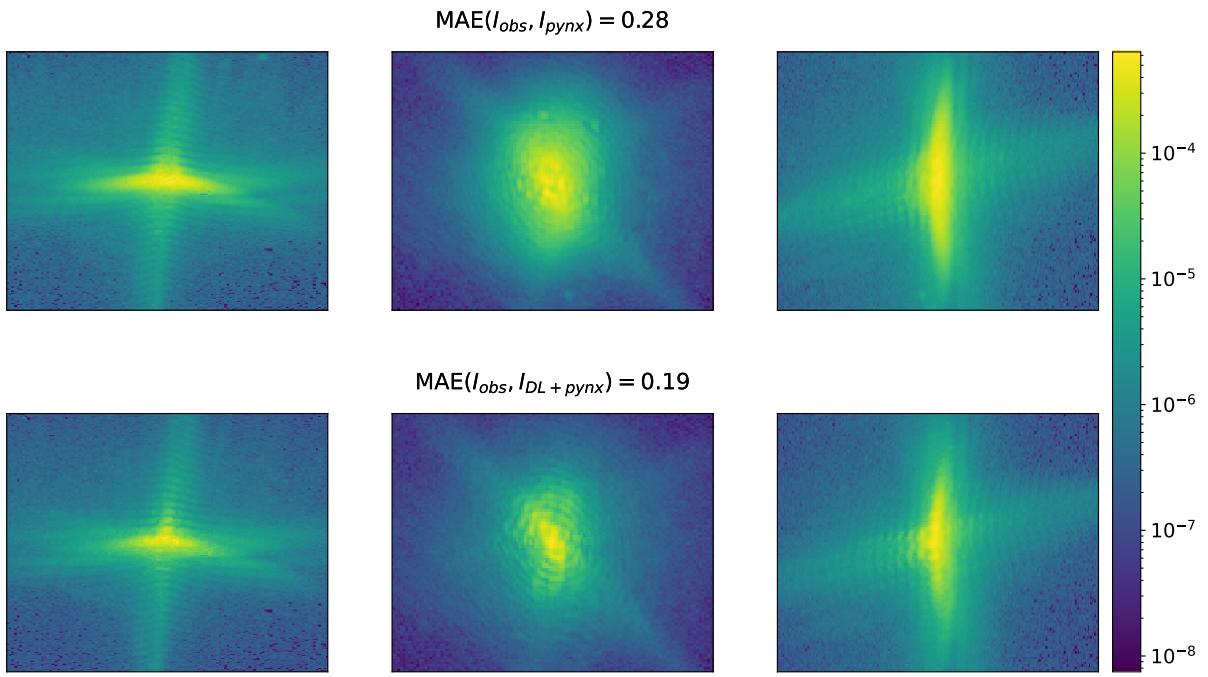


**Figure 6.48:** Central slices of Particle 2 after 300 cycles of ER for refinement of the DL initial estimate.

2409     The improved homogeneity of the modulus and the more physical shape of a Winterbottom  
2410     particle are a clear sign of better reconstructions for both cases. One can observe that the shape  
2411     and phase of the final reconstruction are not remarkably different from the DL estimates. This  
2412     proves that the DL model is able to generalize correctly to experimental data as well and that  
2413     the following ER steps are a good choice for refinement. It is worth noticing that launching  
2414     several independent runs with the same initial DL guess would not change the outcome as ER  
2415     converges to the local minimum by projecting each estimate on the constraint sets, therefore  
2416     never “exploring” other neighborhoods of the solution space.  
2417

2418     While for Particle 1 it is obvious that the DL + PyNX solution is better than the PyNX one  
2419     because of the absence of “holes” in the modulus, for Particle 2 it can be harder to believe that  
2420     the DL + PyNX approach actually yields an improved solution.

2421     Although greater homogeneity of the electron density, as quantified by the Mean-to-max  
2422     metric, often suffices to regard one solution as superior to another, rigorous assessment requires  
2423     validation against experimental data through evaluation of the discrepancy between observed  
2424     and calculated intensities. Here, Fig. 6.49 shows that the calculated intensity obtained from the  
2425     DL + PyNX is more accurate than the one obtained from the PyNX - only solution for both  
2426     Particles 1 and 2.



**Figure 6.49:** Projection along the three axis of the absolute difference between observed and calculated diffraction patterns relative to Particle 2. The mismatch (MAE) of the one calculated from the particle reconstructed using PyNX only, is higher than the one calculated from the particle obtained with the DL + ER method. This proves that the latter is the correct solution.

2427 In addition to the higher quality of the results it is worth mentioning the significant reduction  
 2428 of the wall-clock time. Table 6.4 summarizes the time taken for the 3 different methods. A 40X  
 2429 to 50X speed-up is recorded for these dataset sizes and the time saving increases for larger  
 2430 ones.

	Particle 1	Particle 2
<b>Data size</b>	(118,180,230)	(110,160,200)
<b>DL inference</b>	3.04 s	2.21 s
<b>PyNX: 60 runs</b>	227.41 s	123.55 s
<b>DL + PyNX: 1 run</b>	4.73 s	3.08 s

**Table 6.4:** Computation times for the three different methods. DL RSP prediction and inverse FT are considered for DL time. PyNX - only time includes 60 runs of 400 HIO + 1000 RAAR + 300 ER performed in parallel by PyNX. The best reconstructions selection and following mode decomposition is not considered. DL + PyNX time includes the full DL time plus the reshaping of the initial guess to the original size and a single run of 300 ER.

2431 For better comparison, Particle 2 has been reconstructed using a genetic PR approach. Simi-  
 2432 larly to the procedure illustrated in [193], from 15 randomly initialized guessed the following  
 2433 iterative chain was applied:

- 2434 • 400 iterations of HIO  
 2435 • 300 iterations of RAAR

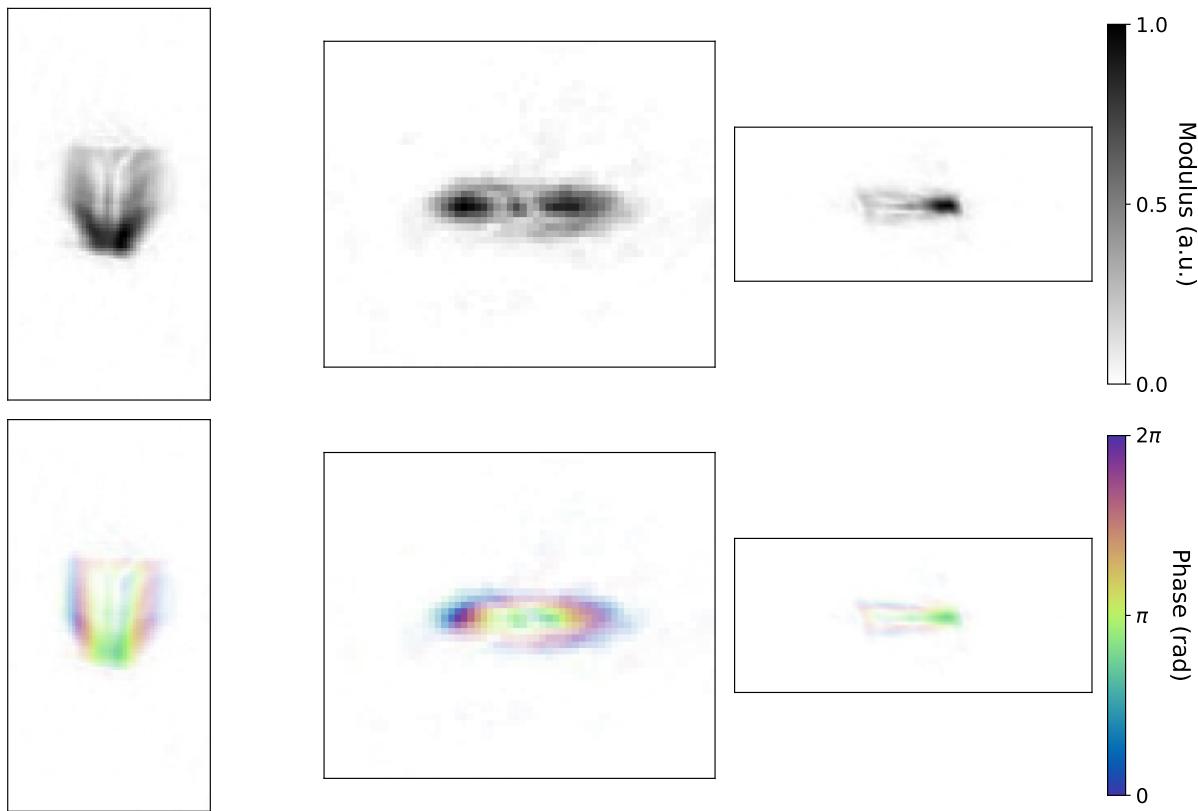
- 2436 • 100 iterations of ER

2437 One reconstruction was selected from the population based on a sharpness criterion, defined  
 2438 as the sum over all voxels of the absolute value of the reconstruction raised to the fourth power  
 2439 (see [193]).

2440 The modulus of this selected reconstruction was then used as a seed for the next 10  
 2441 generations of the base chain, with the phase kept unchanged. This update followed Wilkin's  
 2442 approach [103]:

$$\rho = \sqrt{\rho \times \rho_{\text{best}}} \quad (6.6)$$

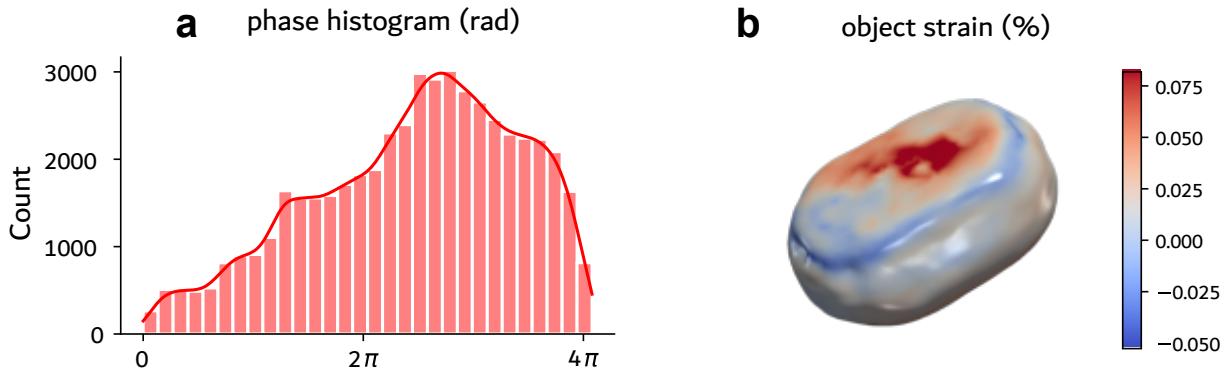
2443 where  $\rho$  is the modulus of an individual in the population being updated, and  $\rho_{\text{best}}$  is the  
 2444 modulus with the lowest sharpness from the previous generation. The results are shown in Fig.  
 2445 6.50



**Figure 6.50:** Central slices of Particle 2 reconstructed with genetic PR. Although better than Fig. 6.44 the result is far from what obtained with the DL-aided PR.

2446 At this point, it is evident that the diffraction pattern of Particle 2 cannot be reliably  
 2447 reconstructed without a DL-based initial estimate. This difficulty is commonly associated with  
 2448 the presence of local minima, which hinder the search for the correct solution and can trap  
 2449 iterative algorithms. An interesting observation is that this behavior can be linked to the phase  
 2450 range of the reconstructed object. Specifically, if the phase values are confined within a single  
 2451 phase wrap, i.e. the interval  $[0, 2\pi]$ , the phase difference between any two voxels is uniquely

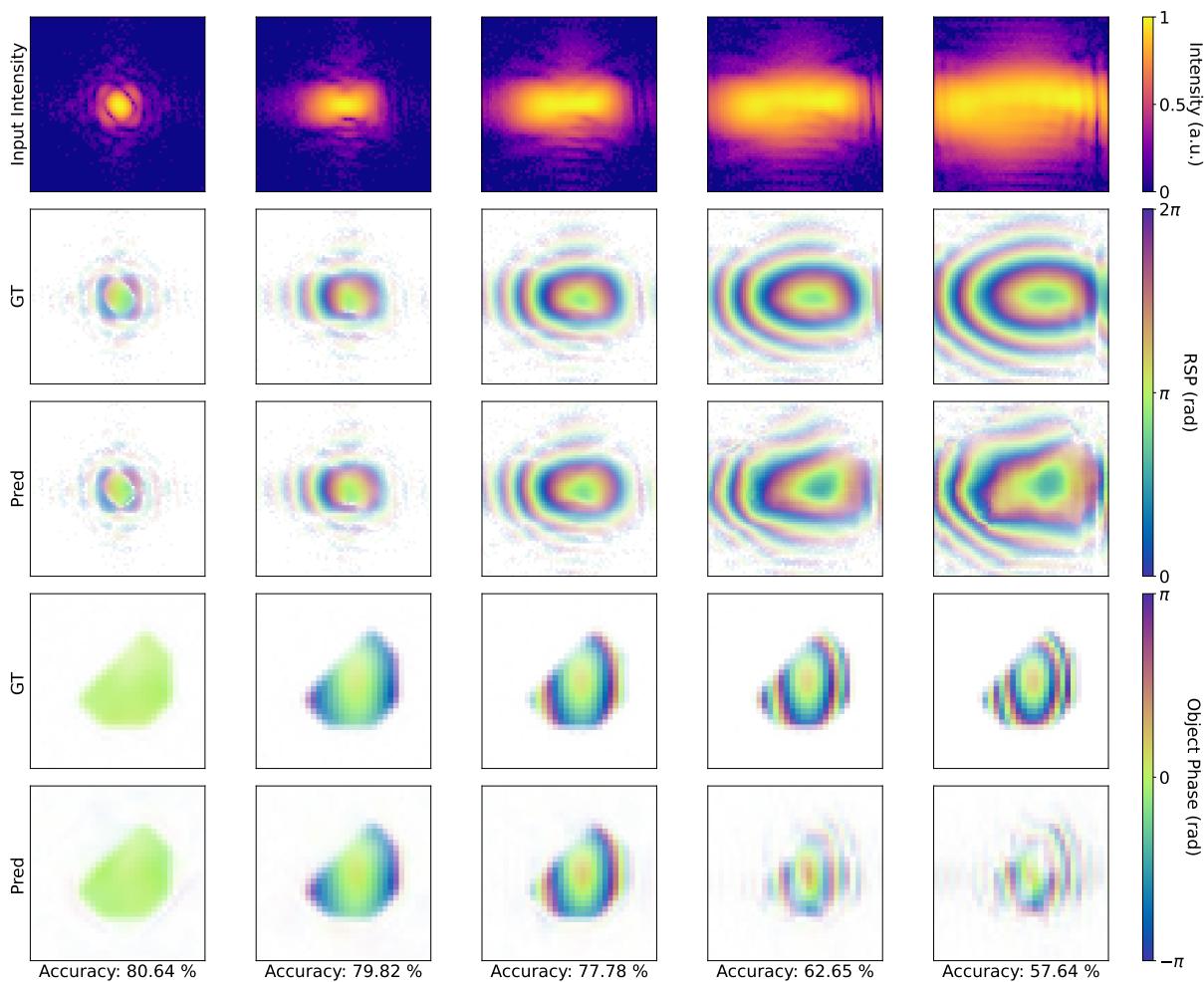
defined within that domain. In contrast, objects with broader phase ranges typically extend over multiple phase wraps, in which case the phase difference between voxels is not uniquely determined in  $[0, 2\pi)$ . It follows that the larger the range of the unwrapped phase, the higher the “population of local minima”, hence the more challenging the PR. These considerations were already proposed by Minkevich and coauthors in [100]. Fig. 6.51 shows the histogram of the phase range extending over the equivalent of two full wraps of Particle 2 (a) as well as with the strain distribution **b** on the 3D rendering of the orthogonalized reconstruction.



**Figure 6.51: Particle 2.** **a** Histogram representing the range of values of the reconstructed object’s phase. Such broad range is typically referred to as “strong-phase” or “high-strain” and implies challenging PR. **b** 3D rendering of the particle’s strain distribution after orthogonalization. The morphology confirms the Winterbottom shape expected from Pd/Pt on crystalline substrate.

## 2459 6.10 Performance assessment

2460 In this section the results of the DL model presented previously, tested against different  
 2461 strain configurations and magnitudes, are discussed. The scope of the study is to assess the  
 2462 model's performance for these different cases, to evaluate when it works best and ultimately to  
 2463 estimate the gain in accuracy when coupled with ER refinement. The first test was prepared by  
 2464 simulating a Winterbottom shaped particle, similar to the ones used for the training set, with an  
 2465 applied phase built with two Gaussian functions with two different and increasing amplitude  
 2466 ranges. The corresponding BCIDI pattern has been simulated for each case, keeping the same  
 2467 oversampling ratio and noise level across the simulations. For each calculated diffraction  
 2468 pattern the RSP has been predicted using the DL model and corresponding objects have been  
 2469 obtained with inverse FT. At this point the accuracy of the prediction was calculated using the  
 2470 formula in 6.5 and the results are shown in Fig. 6.52.

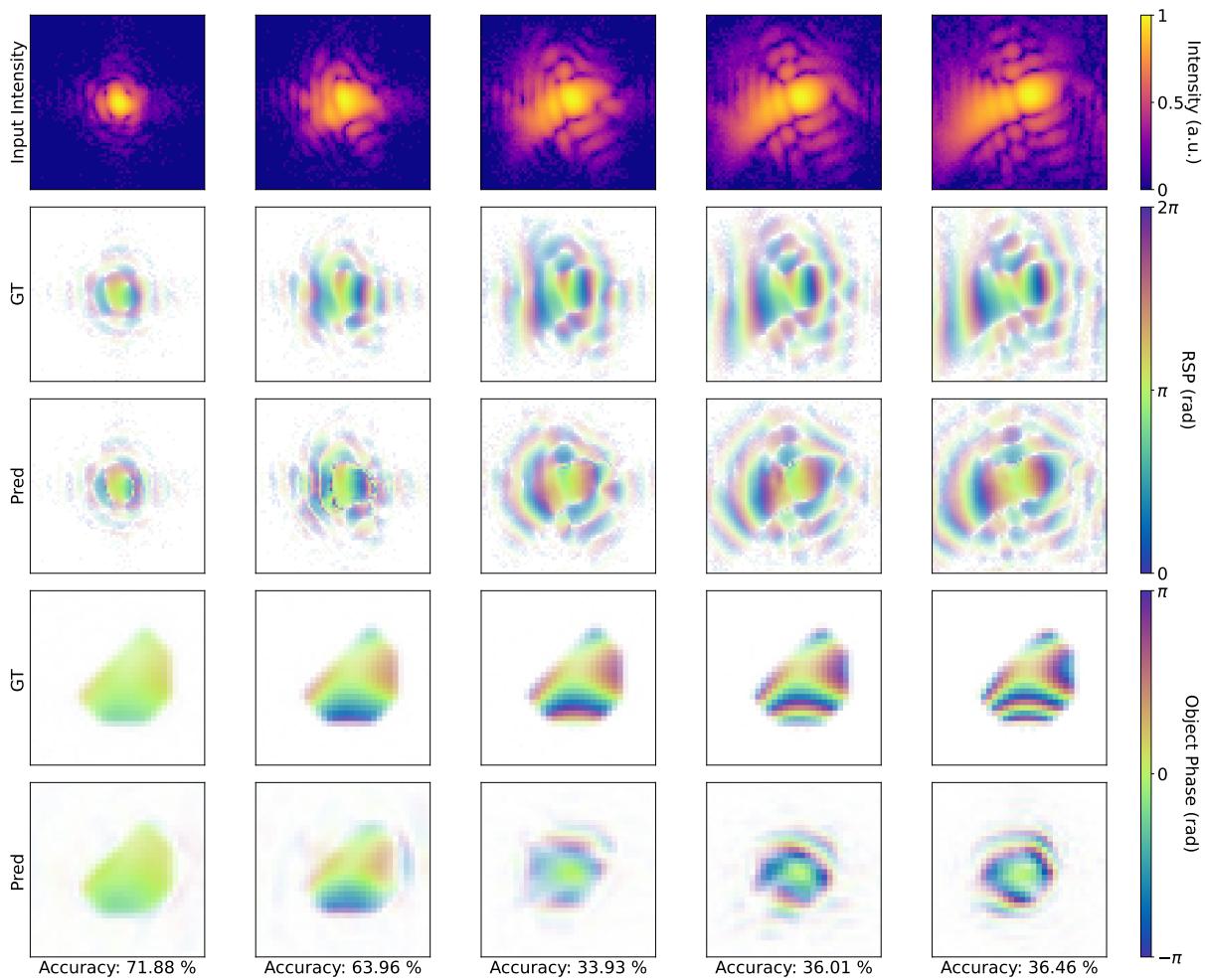


**Figure 6.52:** Evolution of the BCIDI pattern and corresponding RSP and object for higher strain calculated with the sum of two Gaussian functions of increasing amplitude. Third and fifth row show the DL predicted RSP and the reconstructed object's phase respectively. As expected the prediction worsens for high phase ranges but it overall maintains a similarity with the ground truth. One could notice the presence of aliasing due to the use of the Fourier transform and large oversampling. Although not observable in experimental conditions, this artifact does not affect severely the analysis and the reconstructions.

2471 As expected the accuracy drops as the phase range increases. It is interesting to notice that  
2472 despite the worsening of the prediction with the increasing strain the object's phase structure  
2473 inside the support resembles the ground truth phase. This detail is fundamental for the ER  
2474 refinement as the initial guess, notwithstanding the inhomogeneous and "shattered" support,  
2475 represents a good estimate of the solution. Objects with similar cleaner supports of incorrect  
2476 shapes and phases are much worse starting points for iterative refinement since they are far  
2477 from the solution.

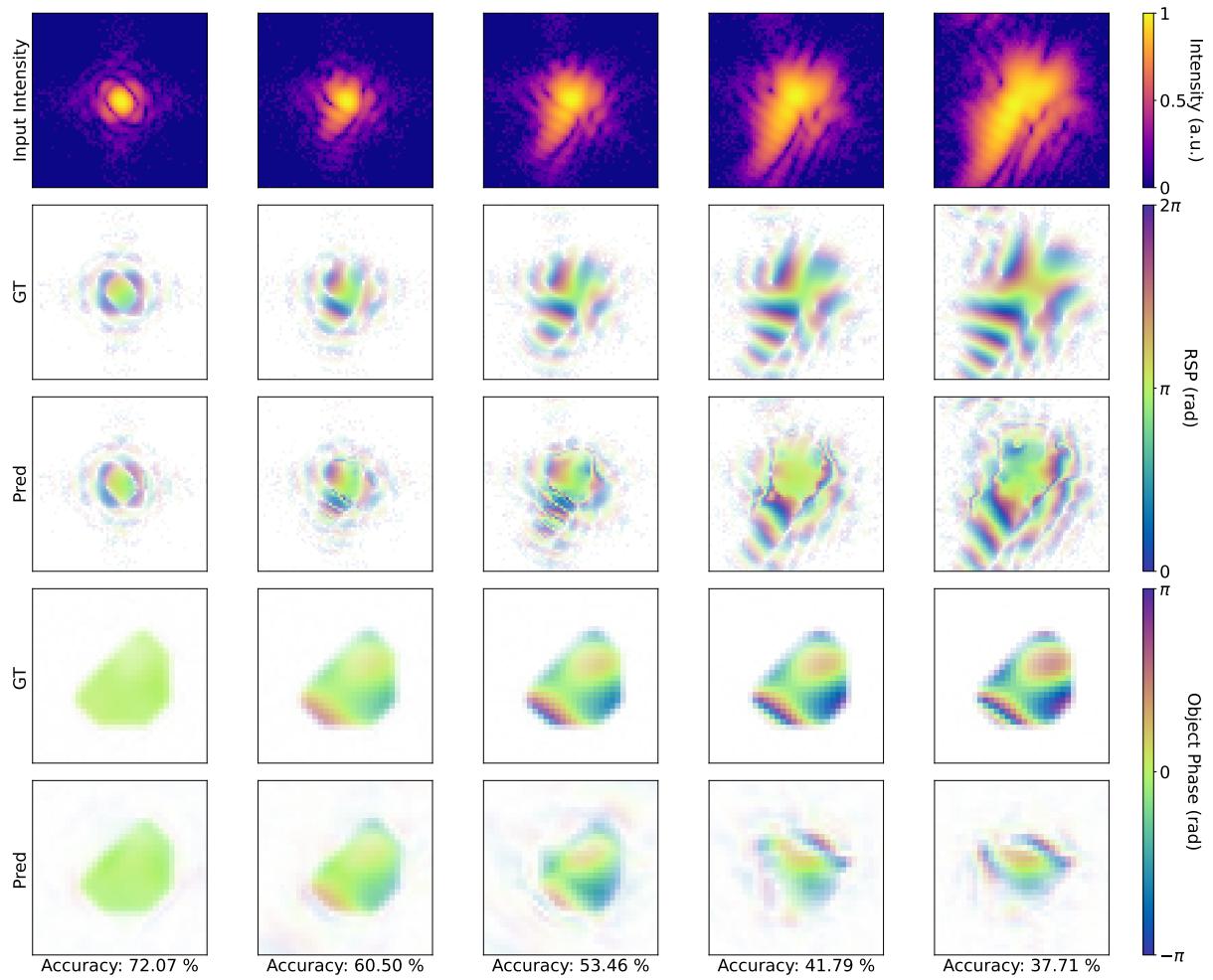
2478

2479 The same procedure has been repeated for a different strain configuration, this time con-  
2480 structed with two cosine functions with increasing amplitude. The phase distribution induced  
2481 by these functions is rather different, more in the spatial structure than the amplitude. Fig.  
2482 6.53 shows that the model accuracy is significantly poorer than the previous case. It is worth  
2483 recalling that the training set was composed of equal amounts of particles simulated with  
2484 Gaussian and cosine phase profiles, meaning that the lower accuracy scores are not due to  
2485 some possible imbalance of the training. The phase range inside the particles of Fig. 6.53, that  
2486 can be estimated visually by counting the number of phase wraps, is also in the same order  
2487 of magnitude as the one shown Fig. 6.52. This further suggests that the poorer performance  
2488 observed in Fig. 6.53 arises from other factors.



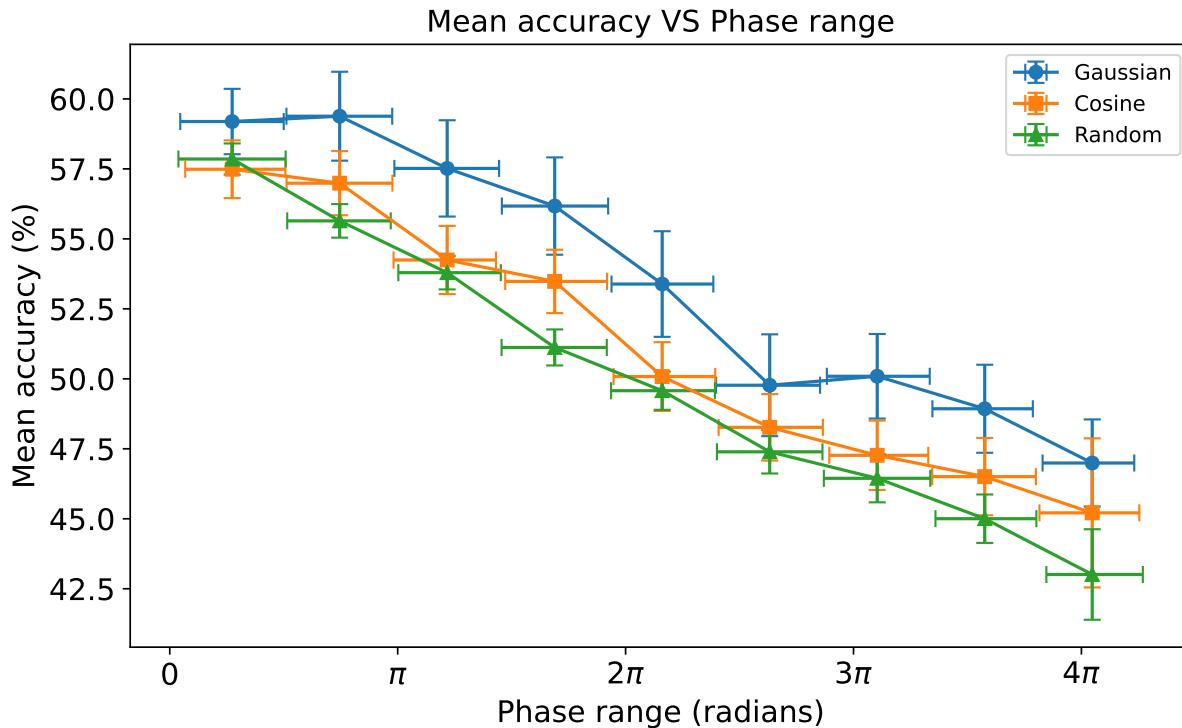
**Figure 6.53:** Evolution of the BCDI pattern and corresponding RSP and object for higher strain calculated with the sum of two cosine functions of increasing amplitude. Third and fifth row show the DL predicted RSP and the reconstructed object's phase respectively. The model rapidly struggles with this type of phase field.

2489 Similarly, the model seems to struggle more for phase fields simulated with using Gaussian  
 2490 correlated random field (same as in 5.4.1) than for the first case. Again the reason doesn't seem  
 2491 to be related to the different phase range nor population imbalance in the training set (Fig. 6.54)



**Figure 6.54:** Evolution of the BCDI pattern and corresponding RSP and object for higher strain applied by using a Gaussian correlated random field of increasing amplitude. Third and fifth row show the DL predicted RSP and the reconstructed object's phase respectively.

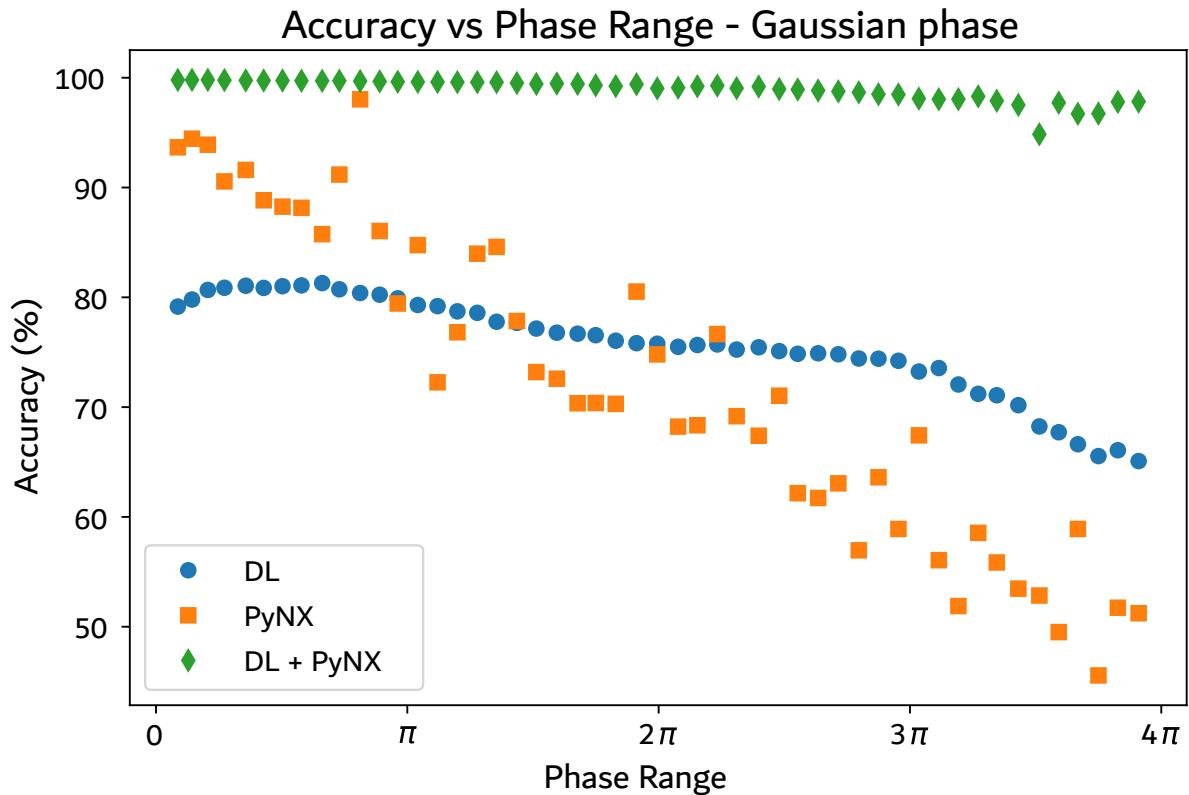
2492 To better assess the performance of the model on the three different cases, the same  
 2493 procedure has been repeated on a bigger and more diverse population. Namely, 50 randomly  
 2494 shaped particles have been simulated and for each of them an increasing strain field has been  
 2495 applied in 10 different steps, for the three types of distributions. The results of the model are  
 2496 shown in Fig. 6.55



**Figure 6.55:** Mean accuracy of the DL model for different phase ranges and types. The phase field simulated with two Gaussian functions yield RSPs that are better predicted by the DL model than the ones obtained with two cosine functions or random Gaussian fields. One can see a significant drop of the accuracy with respect to Figs. 6.52 - 6.53 - 6.54 which is attributable to the different random shapes used here instead of the Winterbottom shaped particle that was used for the previous tests. In fact, while the model was trained on equal numbers of Winterbottom-shaped and randomly shaped particles, the diversity among Winterbottom particles is inherently smaller. This lower variability makes them easier for the DL model to learn, which likely explains the higher accuracy obtained previously.

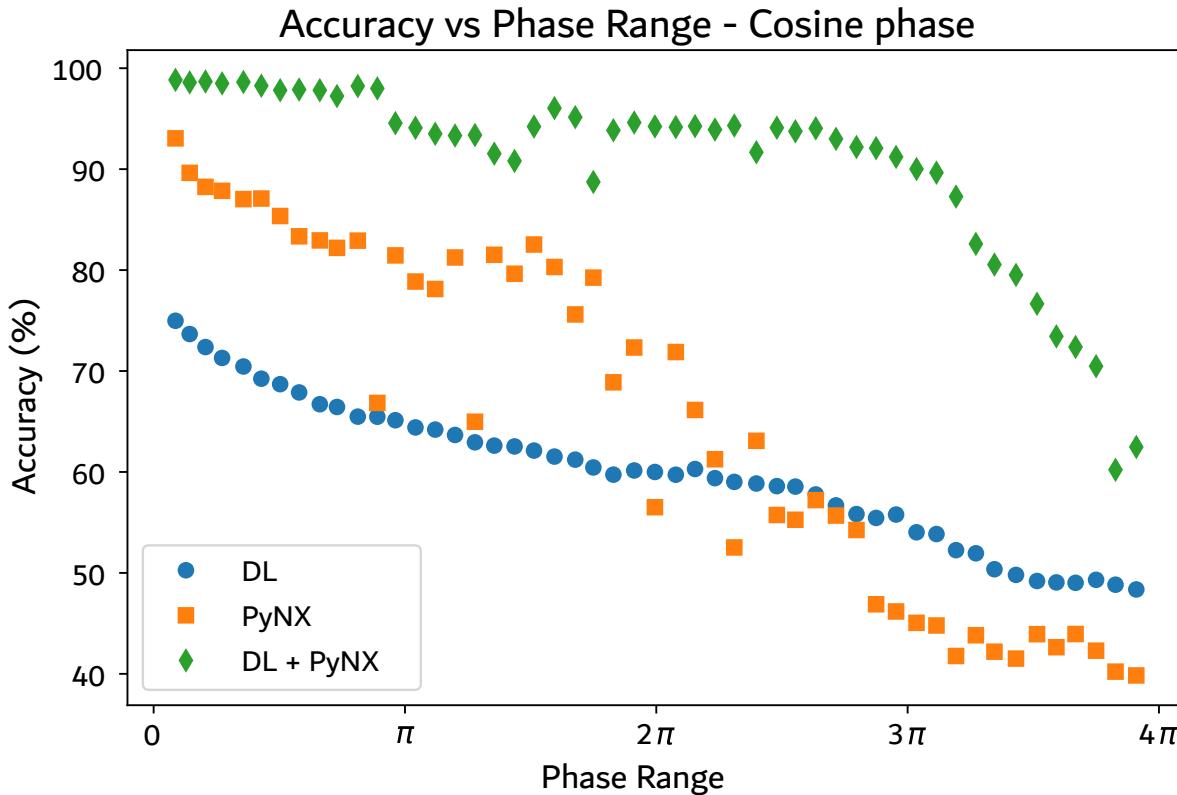
2497 Another study that has been conducted aimed at estimating the accuracy gain of the DL  
 2498 model coupled with ER refinement compared to the sole iterative algorithm. In this case a  
 2499 single Winterbottom particle has been selected for simplicity, and an increasing phase field was  
 2500 applied in 50 steps for the three different types. At this point the accuracy scores have been  
 2501 calculated for the DL model only. Separately, a single run of 400 HIO + 1000 RAAR + 300 ER  
 2502 has been performed using PyNX for each of the 50 diffraction patterns. The parameters are the  
 2503 same reported in Table 6.2, exception made for the autocorrelation and support thresholds fixed  
 2504 at 0.1 and 0.15 respectively. At last, the DL model predicted objects have been used as a starting  
 2505 point for 300 ER refinement, always with PyNX and the parameters in Table 6.3. In all cases, it  
 2506 was made sure that the solution found by either the DL or the iterative algorithm wasn't the  
 2507 twin of the ground truth. If so, the object was flipped before calculating the accuracy.

2508 The analysis has been repeated for the three different phase fields used in the above cases  
 2509 and the results are shown in Figs. 6.56, 6.57, 6.58.



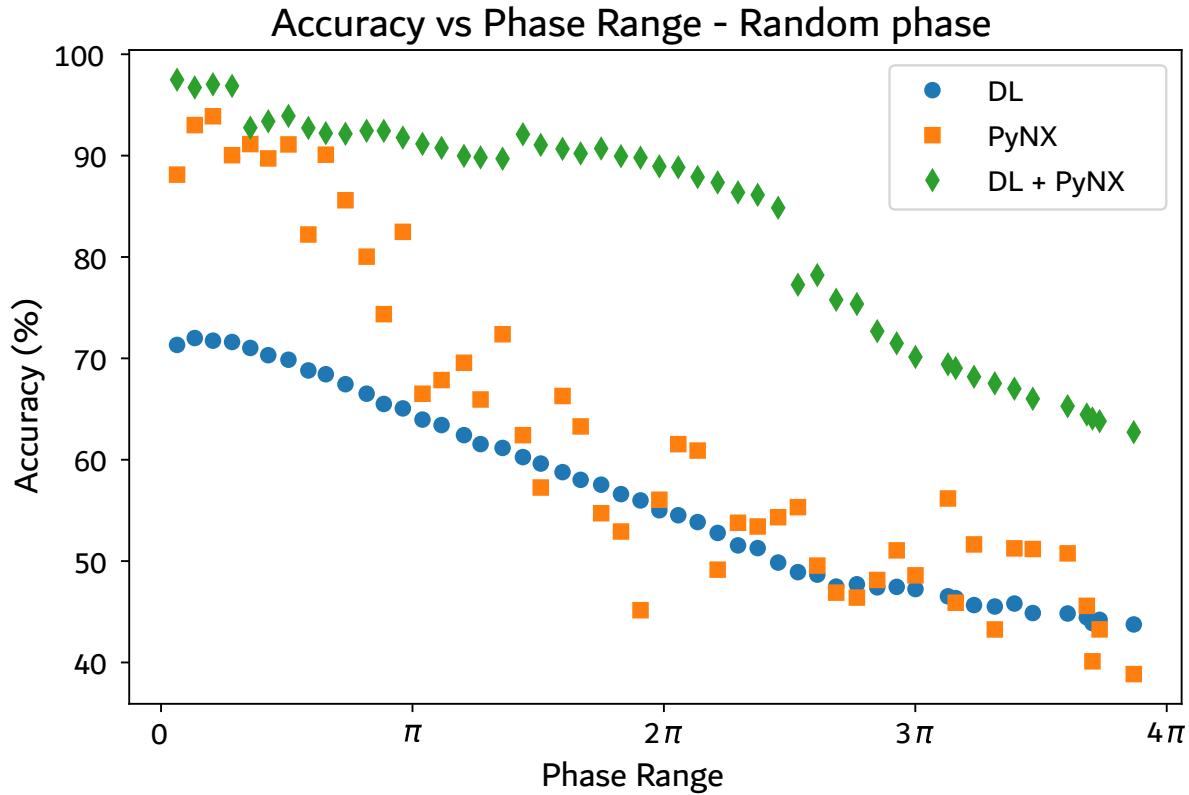
**Figure 6.56: Accuracy vs Gaussian phase range.** Comparison of the accuracy scores of the DL model prediction, iterative PR and DL + ER approaches for objects with phase fields simulated with two Gaussian functions with increasing amplitude. The DL + ER reconstructions always achieve higher accuracies than the other methods, also in those cases in which the DL ones are low.

2510 While for lower phase ranges the DL prediction is worse than the two other methods, the  
 2511 main result of Fig.6.56 is that in all cases the accuracy of the DL + ER method is always superior,  
 2512 meaning that the DL model can significantly aid the PR process also when the first estimate is  
 2513 not very close to the solution (down 50% accuracy in the figure.).



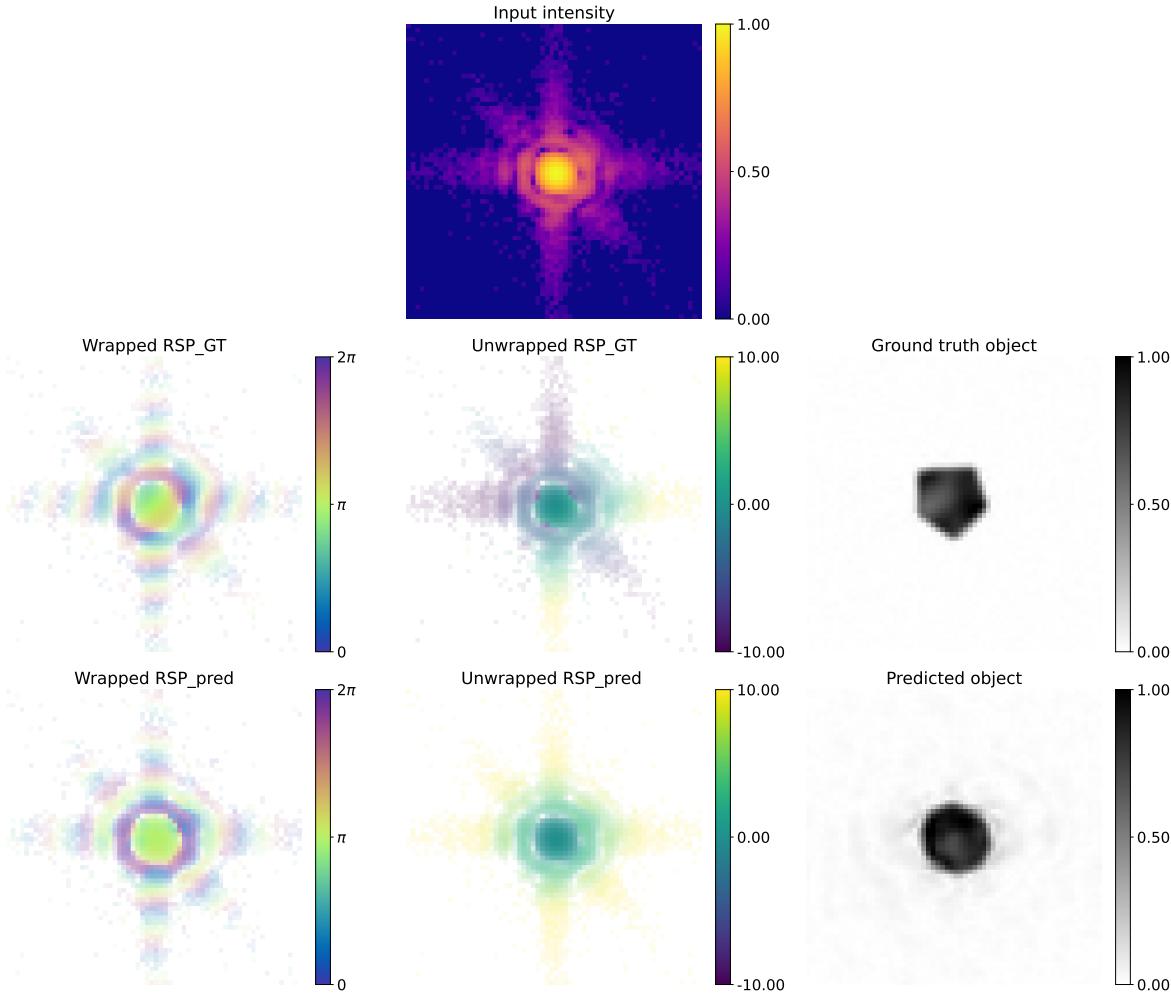
**Figure 6.57: Accuracy vs Cosine phase range.** Comparison of the accuracy scores of the DL model prediction, iterative PR and DL + ER approaches for objects with phase fields simulated with two cosine functions with increasing amplitude. Although PyNX shows better results for most of the phase ranges, the DL + ER combination outperforms PyNX across the whole range.

2514     Similarly, the DL model proves to be significantly helpful at providing a good initial guess  
2515     for the ER refinement when the object's phase is simulated with cosine functions. In this case as  
2516     well, the DL model, although when used alone yields relatively low accuracy scores, improves  
2517     the quality of the solution when coupled with some ER cycles for refinement.



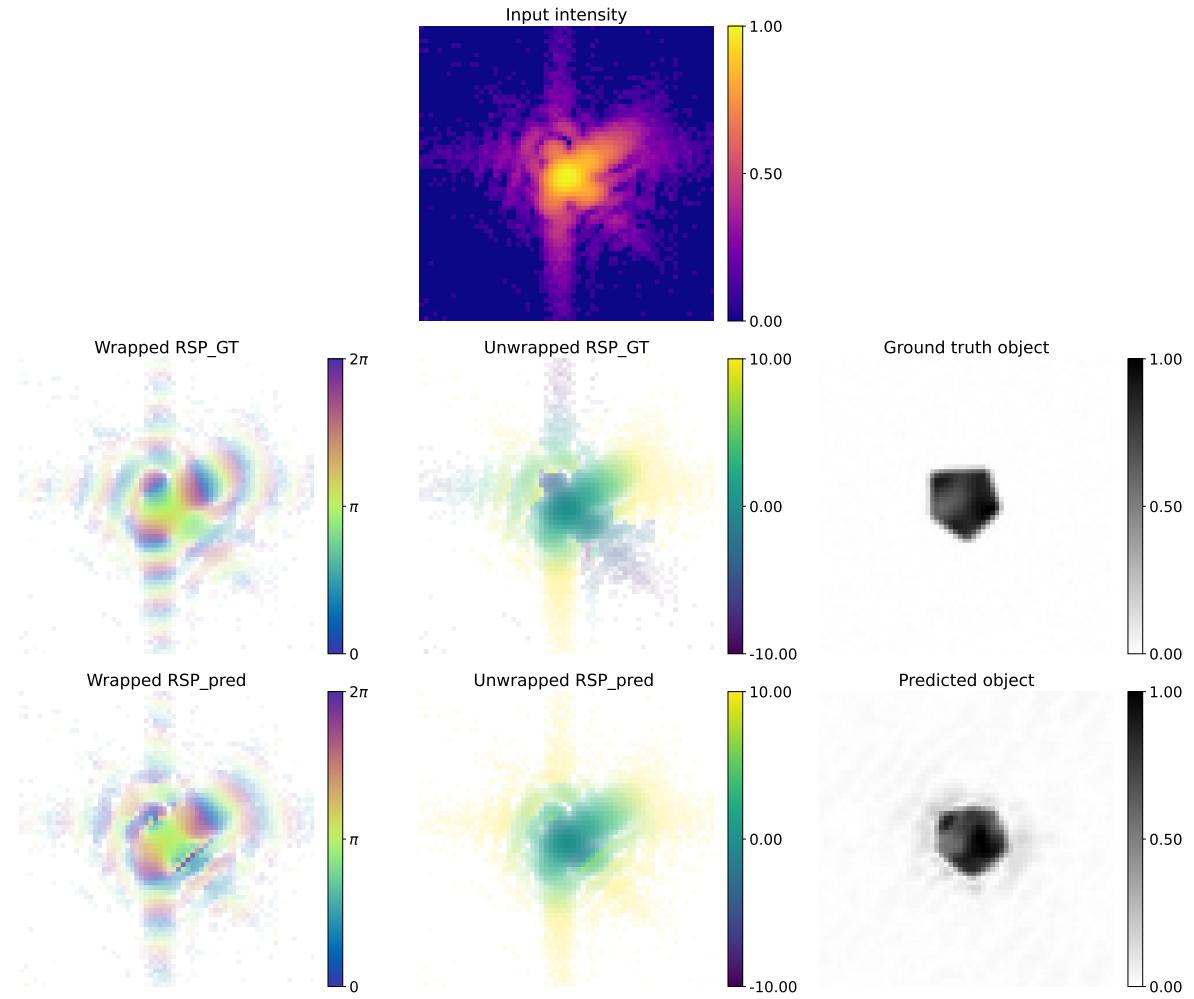
**Figure 6.58: Accuracy vs Random phase range.** Comparison of the accuracy scores of the DL model prediction, iterative PR and DL + ER approaches for objects with phase simulated with a Gaussian correlated random field with increasing amplitude. For small phases PyNX shows better accuracy scores than DL only, however for stronger phases the two methods yield similar results. However, most importantly, the DL + ER combination always outperforms both methods, meaning that the DL result is a close estimate of the solution for which few refinement iterations are enough to converge.

Finally the reasons behind the imbalance of accuracy scores for different phase fields are investigated on a qualitative level. From the visual assessment of the DL prediction one can observe that the model performs best when the RSP is characterized by “iso-phase” lines with spherical symmetry. This effect was already observed in the 2D case in which the WCA loss managed to overcome this limitation. In the 3D case this effect is more pronounced and the model still struggles to break this structural symmetry despite the large number of trainable parameters and training samples. A possible explanation is given by the symmetry of the diffraction pattern used as input. The averaged spatial intensity distribution of the ensemble of training samples possess a spherical symmetry, decaying with a power law radially. As a consequence, the model replicates a similar distribution in the predicted RSP. This result is clearer when the direct output of the model, in the form of an unwrapped RSP, is inspected.



**Figure 6.59:** Example of DL prediction for a low strain randomly shaped particle. The non-centrosymmetry of the object's shape and the small strain produce a symmetric diffraction pattern with a RSP that when unwrapped possesses a non-spherical symmetry. This deceives the model that wrongly predicts a symmetric RSP, resulting in an incorrect reconstructed object.

When comparing the unwrapped phases one can observe that while the ground truth, along the main streaks, grows monotonically from one to the other end, the predicted one, which is the direct output of the model, grows isotropically along all radial directions. The consequence is a wrong retrieved object as visible in Fig. 6.59. It is worth noticing that the example reported in the figure is a low-strain case, meaning that the main obstacle for the model is given by the structural symmetry of the RSP rather than the strain. To support this hypothesis, a stronger phase obtained with two Gaussian functions has been applied to the same object and the same test has been conducted. The results illustrated in Fig. 6.60 show that the RSP symmetry has changed to a more “spherical” one thanks to the strain. Therefore, the predicted RSP is more accurate as well as the reconstructed object’s shape. This would explain the initial bump in the accuracy scores present in both Figs. 6.56 - 6.58.

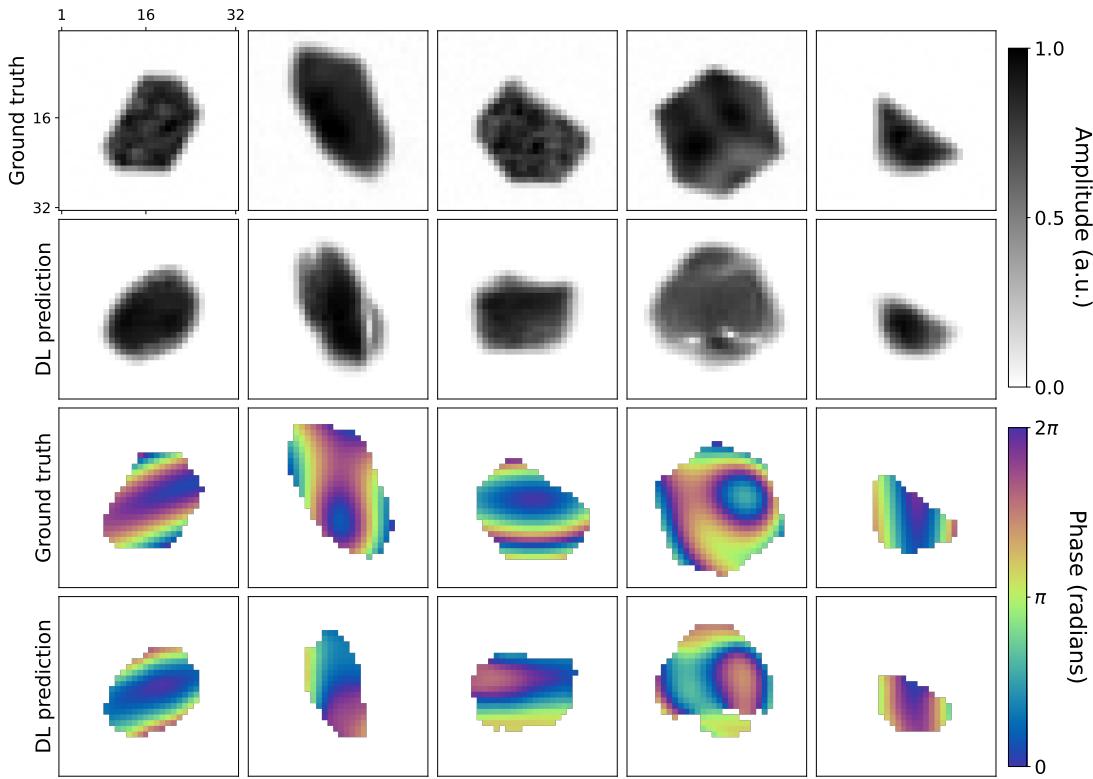


**Figure 6.60:** A stronger phase has been applied to the same object in Fig. 6.59. The strain field produces a more symmetric RSP that is correctly retrieved by the DL model, thus a better object is retrieved despite the higher strain.

## 2540 6.11 Other model test

2541 In this last section I would like to assess the real advantage of predicting the RSP rather than  
 2542 the complex real space object. In order to conduct this experiment, the most recent model in  
 2543 the literature of Deep Learning for BCDI Phase Retrieval was considered. In particular, the  
 2544 model presented by Yu and coauthors in [185], that makes use of complex convolutional layers  
 2545 seem to be a good candidate for the testing of our DL model. The 2D model that was presented  
 2546 in the paper was transformed into a similar one for 3D data, for a total amount of 36 million  
 2547 trainable parameters.

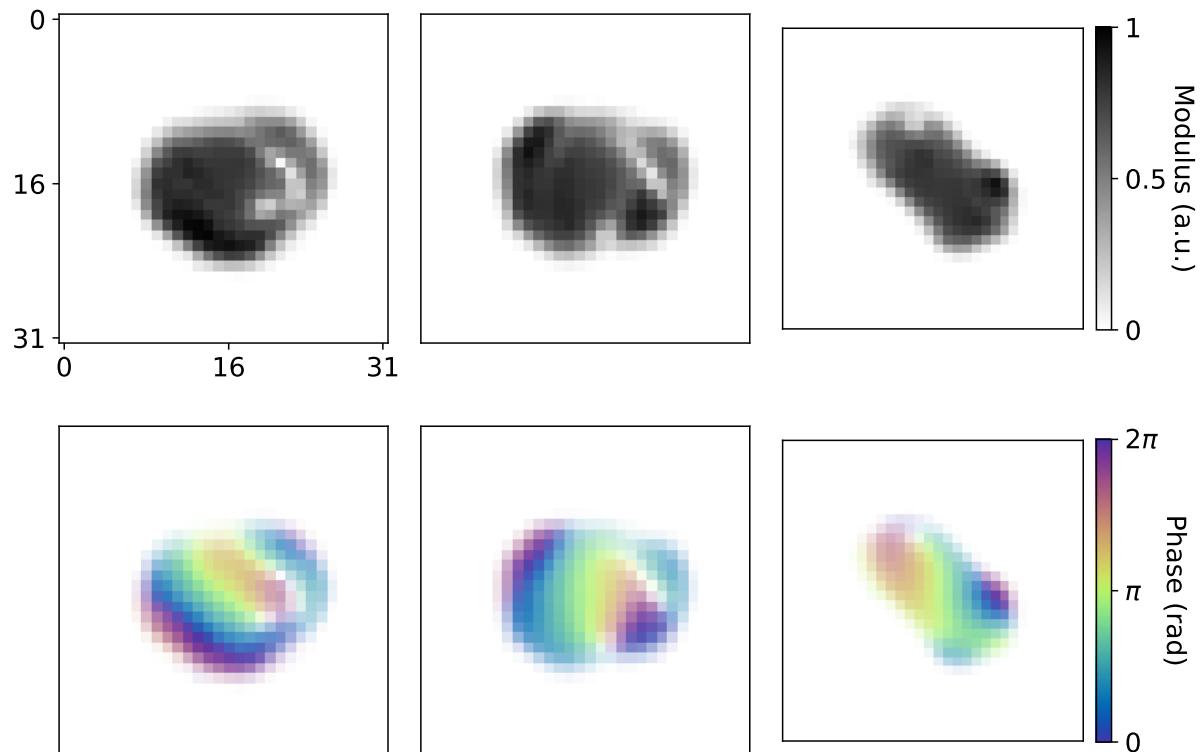
2548 The model has been trained for the same amount of epochs (60) on the same training dataset  
 2549 used for our DL model. Model structure and loss function were adopted like those presented  
 2550 in the paper, and tests on the same data shown in Fig. 6.37 have been performed with results  
 2551 shown in Fig. 6.61.



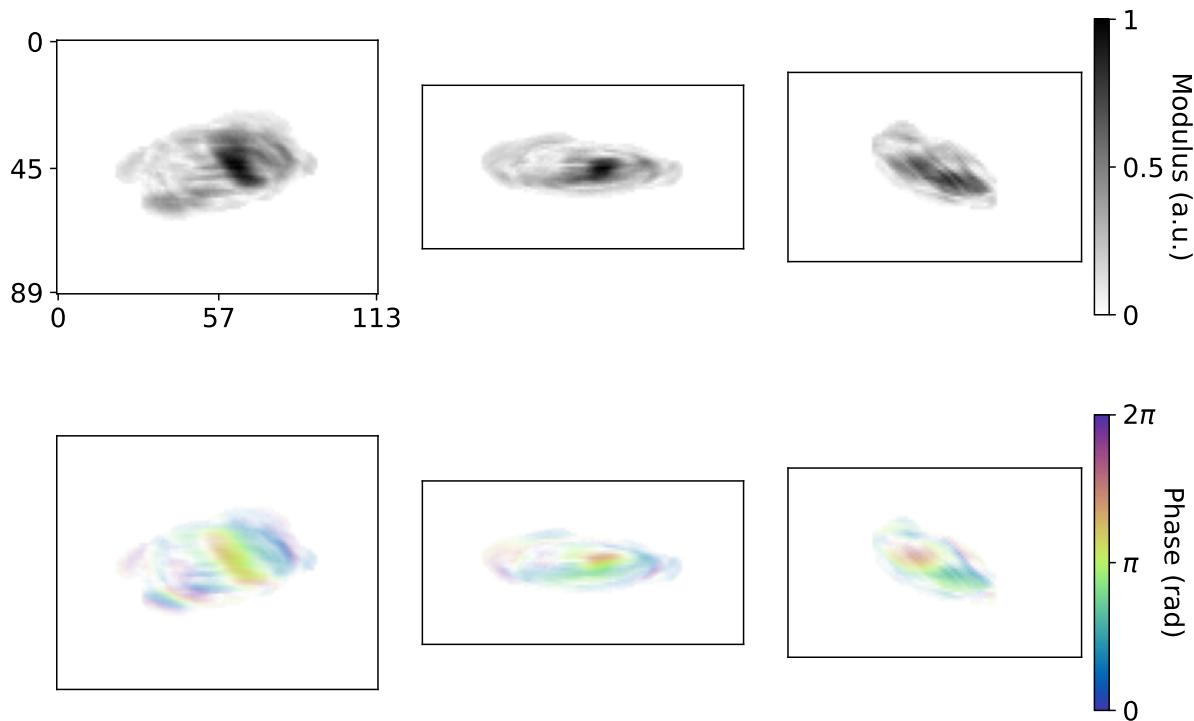
**Figure 6.61:** Predicted objects obtained by the 3D adaptation of the 2D Complex U-Net presented in [185]. The model returns a complex tensor that represents the real space solution. Although much less noisy than our predictions, the reconstructed objects possess incorrect shapes and phases.

2552     The model returns objects that have a better resolution and less noise levels because of the  
2553     loss function was calculated in real space. However, the shape and phase of the results are not  
2554     correct, thus unusable for ER refinement because they are misleading initial guesses. In fact,  
2555     the PR of the diffraction pattern of Particle 1 (Fig. 6.38) has been inferred with this model (see  
2556     Fig. 6.62) and the result used as a starting point for further refinement using PyNX and the  
2557     parameters of Table 6.3. The model prediction possesses clean and uniform modulus, and a  
2558     support which is almost correct. However the phase is quite different from what obtained in  
2559     6.47.

2560     For this reason, overall, the predicted object does not represent a close enough estimate  
2561     for ER to converge to the solution. The result instead shows inhomogeneous modulus with  
2562     amplitude dips and domains in the phase, therefore far from the correct solution.



**Figure 6.62:** Three central slices of the prediction of Particle 1 using the DL model presented in [185] adopted for the 3D case and trained on highly strained BCDI data. The retrieved object is clean and homogeneous but the phase has been revealed to be incorrect.



**Figure 6.63:** PR of Particle 1 obtained with PyNX refinement following Table 6.3 of the DL prediction in Fig. 6.62. The results show that the convergence to the correct solution was not achieved because the DL estimate is too far from it.

2563 This simple example shows that the prediction of the RSP suites better the PR of BCDI  
2564 patterns with convolutional neural networks, especially for highly strained particles.

## 2565 6.12 Conclusion

2566 To conclude this journey through the developments, results and interpretations of the DL model  
2567 for the BCDI RSP prediction the main points can be summarized as follows.

- 2568 • A novel approach for the DL-based PR of BCDI patterns, focused on the prediction of  
2569 the RSP has been investigated. The method proved to be advantageous since (i) it entails  
2570 the inference of a single array rather than two coupled ones, (ii) it exploits the exact  
2571 calculation of the IFT, directly using the diffraction pattern in the transform, rather than  
2572 a learned map from the diffraction pattern and the solution in real space. (iii) it takes  
2573 advantage of the similarity between the input and the output variables, making the best  
2574 use of the skip connections of the U-Net architecture. The superiority of this approach  
2575 over the more conventional one found in literature has been demonstrated as discussed  
2576 in 6.11.
- 2577 • The training of the DL model for the prediction of the RSP can be optimized with the  
2578 use of the custom WCA loss function, designed specifically for the handling of complex  
2579 phases. This loss function naturally resolves the wrap, the sign and the offset symmetries  
2580 inherent to the lost phase problem. Adopting the WCA loss not only improves the results

2581 but also speeds up the training as it takes less time than a loss computed in real space.  
 2582 Moreover, the use of the WCA avoids any reference to the real space object during the  
 2583 training, opening the door for a patching approach, where the RSP is predicted from a  
 2584 sub-volume of diffracted intensity.

- 2585 • Such a tailored model is trained in supervised fashion on simulated data only. This can be  
  2586 at first a limiting factor as it requires the simulation of big and diverse datasets. However,  
  2587 as the BCDI technique is restricted to single crystals, the population of shapes, though  
  2588 being potentially infinite, is physically confined by nature. It is thus easier for the DL  
  2589 model to generalize for new particle shapes.
- 2590 • The DL model trained as presented is able to perform on highly-strained particles as  
  2591 well, showing promising results in 2D and 3D. It has proven to be successful at inverting  
  2592 experimental data that happened to be extremely challenging with conventional iterative  
  2593 methods only. This achievement marks a milestone for the BCDI community, upgrading  
  2594 the DL studies on BCDI PR from a “*proof of concept*” level to an actual *practical use* in the  
  2595 analysis of experimental data.
- 2596 • The DL model proved to be compatible with iterative refinement using conventional  
  2597 algorithms, contributing to a faster and more robust pipeline for PR. The computational  
  2598 time and costs typically devoted to the search of the global minimum with regular  
  2599 algorithms is now significantly reduced by the DL estimate. This can potentially broaden  
  2600 the field of application of BCDI to many more samples in which high-strain is involved  
  2601 as cause or consequence of physically relevant mechanism.
- 2602 • The DL model exhibits certain shortcomings attributable to the intrinsic symmetry of  
  2603 RSP. It in fact “prefers” RSPs with a spherically symmetric distribution and the major  
  2604 hypothesis for this fact is that such symmetry is typical of the intensity distribution of  
  2605 the average diffraction pattern. Hence, the increased difficulty of the model to escape  
  2606 this bias for 3D data as compared to 2D data. Additionally, it is curious to notice that  
  2607 this symmetry seems to be the strongest limitation for the DL model, lying above the  
  2608 sign symmetry and the high strain, which are both resolved when in many cases. Future  
  2609 investigations that elucidate the root causes of this phenomenon may yield improved  
  2610 performance on more heterogeneous data sets.
- 2611 • At last, the substrate-induced strain imposed on Winterbottom particles often manifests a  
  2612 symmetry that is faithfully reproduced by the pair of Gaussian functions used to generate  
  2613 the phase of the training ensembles. This symmetry gives rise to a spherically symmetric  
  2614 RSP, which the DL model, by design, generally predicts with high accuracy.



2616 AUTOMATIC DIFFERENTIATION FOR PHASE RE-

2617 TRIEVAL

“An approach that would be superior to the ones considered here would be one that minimizes the Fourier-domain error while inherently satisfying the object-domain constraints, or one that minimizes an error metric that combines the Fourier- and object-domain constraints [...]. Something along these lines would be very useful for the problem of a single intensity measurement; clearly, more could be done in this area”

---

J.R.Fienup [2]

2618 In this chapter a different approach to the BCDI phase retrieval will be presented. It origi-  
 2619 nated from the need to resolve those cases in which neither standard iterative algorithms, nor  
 2620 the DL assisted PR can succeed to converge to a satisfactory reconstruction. The developed  
 2621 approach differs from the alternating projections algorithms classically used for the Fourier PR,  
 2622 as it is formulated as minimization problem solved with gradient descent (GD). The gradients  
 2623 however are computed through the efficient automatic differentiation (AD) enabled by graph-  
 2624 based differentiable programming packages like Tensorflow and PyTorch, accelerated on GPU.  
 2625 For this reason one could see the AD approach as unsupervised machine learning on a single  
 2626 training dataset.  
 2627 Gradient descent-based optimization differs fundamentally from fixed-point alternating pro-  
 2628 jections. In qualitative terms, while the latter alternates between real and reciprocal space,  
 2629 imposing constraints in each, the former begins with an initial complex object and iteratively  
 2630 updates its modulus and phase using the gradients of the mismatch between the measured and  
 2631 calculated diffraction intensities. In this way, the knowledge on the particle can be implemented

2632 by initializing the object with some physical constraints or adding regularization terms that  
2633 will drive the updates towards more reasonable solutions.

2634

## 2635 7.1 State of the Art

2636 AD methods for PR have been investigated already in 2014 by Jurling and Fienup [194] who  
2637 first considered the use of AD for GD-based PR. In this theoretical work the authors proposed  
2638 a pedagogical “manual automatic differentiation” approach for the phase problem, extended to  
2639 complex-valued variables. The authors also denounced the lack of suitable softwares as major  
2640 limitation to the use of AD-based PR. The advent of high-level, GPU oriented, libraries such as  
2641 Tensorflow, PyTorch, JAX and Autograd has opened the opportunity to efficiently exploit AD  
2642 algorithms for the phase problem. The first implementations in the CDI field have considered  
2643 mostly ptychography in forward and Bragg geometries [195, 196] and multi-Bragg CDI [197].  
2644 Chronologically, it was firstly Nashed and coauthors in 2017 [195] who opened the field using  
2645 a Tensorflow AD model for ptychography using the ADAM optimizer. From the same group,  
2646 Kandel *et al.* in 2019, showed the competitive performance of the AD model when compared  
2647 to conventional algorithms and extended the model to multi-angle Bragg ptychography on  
2648 simulated data. In 2023 Maddali and coauthors [197] explored the use of AD methods for multi  
2649 reflection Bragg CDI. The authors leverage the flexibility of the GD approach by designing a  
2650 global optimization function that simultaneously accounts for the geometrical and physical  
2651 constraints related to multi reflection BCDI. More recently Zhou in 2024 [198] and Wu in 2025  
2652 [199] developed AD-based PR algorithms that are able to reconstruct large particles, for which  
2653 a dynamical description of the scattering processes is required. These works again exploit  
2654 the flexibility of AD-based models for the implementation of a forward model tailored to the  
2655 specific problem of mixed kinematic and dynamic x-ray scattering taking place in large crystals.

2656 At the moment of writing, there aren’t any published works that aim at solving the phase  
2657 problem for BCDI datasets considered hard to invert.

## 2658 7.2 Model implementation

2659 In an AD-driven optimization problem some trainable parameters are initialized. In the first  
2660 basic formulation these trainable parameters can be the values of the voxels corresponding  
2661 to the modulus  $\rho$  and the phase  $\phi$  of the complex object that represents the solution of the  
2662 PR problem. All of these voxels contribute to the creation of a simulated diffracted intensity  
2663 pattern via the forward model  $I_{calc} = |\mathcal{F}\{\rho e^{i\phi}\}|^2$ . Subsequently, the gradients of a metric  
2664 (loss function) that estimates the distance between the observed BCDI pattern  $I_{obs}$  and  $I_{calc}$  are  
2665 calculated with respect to each of the trainable variables with automatic differentiation. At  
2666 this point the value of each of these voxels is updated using a chosen optimizer (SGD, ADAM,  
2667 etc.) and a given learning rate. The Tensorflow library allows for an easy implementation of  
2668 the trainable variables, and loss function, and handles gradient operations with predefined  
2669 methods. It is therefore straightforward to run the optimization as it follows the same structure  
2670 of a deep learning model, with less trainable parameters and for a single dataset.

2671 However, such simple formulation of the complex object as mere real-valued variables is

not optimal for a non-linear and non-convex inverse problem such as Fourier phase retrieval. In fact, many non-physical modulus-phase configurations could yield an  $I_{calc}$  that is close to  $I_{obs}$  [192]. The presence of these local minima is the reason why, in conventional PR, algorithms like hybrid input-output, capable of escaping them, are employed. Moreover, it was shown by Marchesini in [84] that steepest GD and even more sophisticated conjugate GD are more prone to get trapped in local minima, which is why they are not commonly utilized for Fourier PR. However, the active research field of machine learning has brought important advancements in the formulation of efficient and robust optimizers based on stochastic gradient descent with powerful features like Nesterov or adaptive momentum (ADAM [121]). These GD techniques are more robust to local minima, since the gradient is computed on mini-batches of trainable variables rather all of them (stochastic rather than classical steepest GD), and converge faster thanks to the “memory” of previous steps. Additionally, they are often wrapped into handy classes, ready to use, in Tensorflow and Pytorch libraries. The current model design however, does not include any stochastic gradient since the derivatives are calculated for all the data points at each step. The use of ADAM optimizer in this case is therefore advantageous for the first and second order momenta which direct the updates. Future developments of the model could explore the training using structured mini-batches similarly to what Wu and coauthors have done in [199].

However, to facilitate the convergence the formulation of the complex object to be optimized has embedded some physical considerations that helped to restrict the solution space. First of all, both support and phase built on a 3D grid occupying half the volume of the input BCDI data to account for the oversampling ratio which has to be at least 2 in all directions to ensure invertibility. Additionally, other constraints specifically designed for the object shape and phase were considered.

### 7.2.1 Object’s shape

The formulation of the object’s shape starts considering the typical crystalline samples that are studied with the BCDI technique and the requirements the modulus of the reconstructed object need to fulfill to be considered a “good solution”. Usually, a successful reconstruction shows a *homogeneous* modulus, sometimes quantitatively assessed through the mean-to-max metric<sup>1</sup> [190, 191]. Enforcing a homogeneous modulus by construction limits the search space and helps the convergence to the solution<sup>2</sup>. It follows that parametrizing the *surface* of the support, and setting to 1 the inside, is much more advantageous than optimizing the full 3D volume. This approach, already proposed by Scheinker and Pokharel in [178], also significantly reduces the number of variables to optimize.

An additional consideration is that the probed samples are crystalline, thus often *faceted* and *convex*. Therefore, one could simplify even more the construction of the object shape by building a certain amount of planes in the 3D space and obtain the support from the volume that lies inside the intersections of all them. This would remove the possibility to have spikes or rough surfaces that might satisfy some local minimum but wouldn’t represent a crystal. Moreover, with this representation the number of trainable variables would be further reduced.

---

<sup>1</sup>as in standard BCDI the form factor is approximated uniform across all the scattering sites

<sup>2</sup>This approach of constraining the modulus of the object to be homogeneous was already considered in the literature (see [58, 100, 101])

According to this scheme the relevant parameters to be optimized are the angles  $\theta$  and  $\varphi$  of the spherical coordinates and the length  $d$  of a given number  $N$  of the so-called *half-spaces*. More formally, the normals  $n_i$  for each of the  $N$  half-spaces are defined with a pair of  $(\theta, \varphi)$  that its orientation in space (Eq. 7.1). Subsequently, only the intersection of those  $(x, y, z)$  coordinates for which the dot product with each  $n_i$  is smaller than the length  $d_i$  is considered as support (Eq. 7.2).

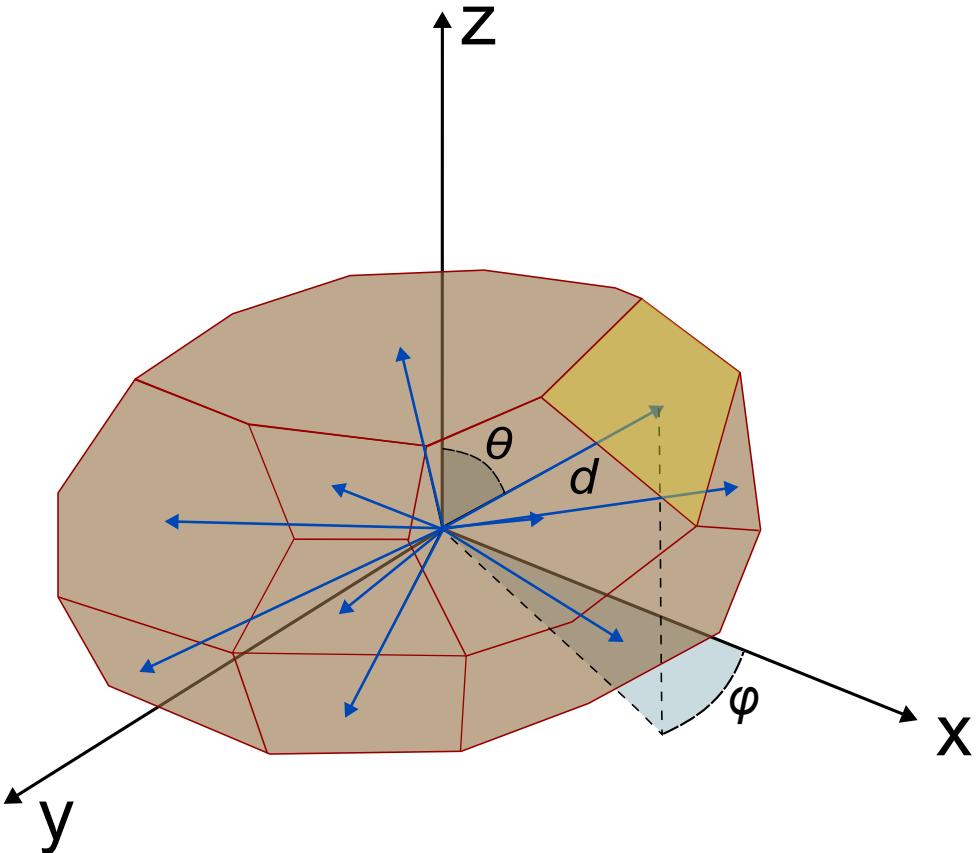
$$\mathbf{n}_i = \begin{pmatrix} \sin \varphi_i \cos \theta_i \\ \sin \varphi_i \sin \theta_i \\ \cos \varphi_i \end{pmatrix}, \quad (7.1)$$

$$\mathcal{S} = \bigcap_{i=1}^N \left\{ \mathbf{x} = (x, y, z) \in \mathbb{R}^3 : \mathbf{n}_i \cdot \mathbf{x} \leq d_i \right\}, \quad (7.2)$$

A schematic representation of this construction is provided by Fig. 7.1.

With this approach the user needs to provide a number of half-spaces as hyperparameter meaning that a sort of prior knowledge on the sample can be leveraged in these regards as well. However, this number doesn't have to be precisely the number of facets expected. In fact, a large  $N$  is often advised for unknown sample shape such that even roundish objects can be retrieved. In case of well faceted samples the large  $N$  is a minor problem as many  $n_i$  will be automatically aligned to the same  $(\theta_i, \varphi_i, d_i)$  at the cost of some more trainable parameters.

The first drawback of this convex-hull parametrization is that concave objects can't be retrieved. Although these cases are less frequent in typical BCDI experiment, a future development of the models should include the parametrization of concave supports as well. The second limitation is that this formulation is incapable of modeling defects that would zero the contribution of the object's modulus to the diffraction pattern [200]. A correct BCDI reconstruction of particles affected by this type of defects presents "holes" inside the hull in correspondence of the defect. However, the current model cannot address this type of features as the support is by construction fully homogeneous inside the borders. Further developments of the algorithm could indeed aim at a more complete formulation of the construction of the object modulus.



**Figure 7.1:** Construction of the convex hull with half-spaces expressed with spherical coordinates. A number  $N$  is given by the user, and a set of  $N$  half-space vectors (blue arrows) are initialized with random length  $d$  and orientation in space ( $\theta, \varphi$ ). The convex-hull obtained with the intersections of the planes built from each half-space vector makes the object support.

2734     The last important consideration of this parametrization is that the support  $\mathcal{S}$  is sharply  
 2735     divided into a binary variable (1 inside and 0 outside) thus leading to differentiability problems.  
 2736     In fact, in such a way the gradients, essential for the support update, are not defined. For  
 2737     this reason  $\mathcal{S}$  is first passed through a sigmoid function controlled by a hyperparameter  $\epsilon$   
 2738     responsible for the smoothening of the support borders. This measure can also be seen as a  
 2739     control of the resolution of the object. Additionally, a mildly steep sigmoid in the early stage of  
 2740     the optimization can help retrieving a low resolution estimate of the support, that can  
 2741     be further refined adjusting the  $\epsilon$  parameter.

2742

### 2743     7.2.2 Object's phase

2744     The parametrization of the object's phase is more challenging. From a qualitative point of  
 2745     view, the prior knowledge that can be exploited for a tailored implementation, is limited to  
 2746     the awareness that a physically meaningful atomic displacement field cannot have "too many"  
 2747     sharp variations. This observation is translated into code by smooth functions parametrization  
 2748     or total variation (TV) regularization [201] of the object's phase. While the former would en-  
 2749     force smoothness by construction the latter would operate adding a penalty to the data-fidelity  
 2750     term of the loss function for non-smooth solutions. Both approaches have been explored and

2751 are reported here.

2752

2753 Forcing a scalar field defined on an  $L \times H \times W$  grid to exhibit smooth behavior is equivalent  
2754 to seeking a sparse representation of that field—that is, to concentrating its essential information  
2755 into far fewer degrees of freedom than the original  $L \times H \times W$  samples. Concretely, one looks  
2756 for a change of basis in which the field can be written as a linear combination of a hierarchy  
2757 of modes or atoms, ordered by “importance”. In a Fourier or wavelet expansion, for instance,  
2758 the expansion coefficients are naturally sorted from largest (low-frequency or coarse-scale  
2759 modes) to smallest (high-frequency or fine-scale modes). Retaining only the largest coefficients  
2760 both compresses the data and removes rapid oscillations, yielding an inherently smoother  
2761 reconstruction. Equivalently, in the matrix case a Singular Value Decomposition (SVD) identifies  
2762 an orthonormal basis in which only a few singular values are nonzero; by truncating to the  
2763 top singular values one obtains a low-rank—and thus smoother—approximation [202]. For  
2764 higher dimensional data, this same principle underlies higher-order generalizations of the  
2765 SVD—Tucker/HOSVD, CP, Tensor-Train, and T-SVD—each of which orders multilinear “modes”  
2766 by their singular-value (or eigenvalue) strength, and truncating to a small subset produces both  
2767 compression and smoothness [203].

2768 In this case the Tucker decomposition was chosen, among the several possible methods,  
2769 for its simplicity of implementation with the Tensorflow library and for the suitability for  
2770 moderately low dimensions [204]. For a 3D tensor the Tucker decomposition is done as follows:

2771 Considering  $\phi \in \mathbb{R}^{L \times H \times W}$  the 3D object’s phase. The Tucker decomposition expresses  $\phi$   
2772 as:

$$\phi = \mathcal{G} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)},$$

2773 where:

- 2774 •  $\mathcal{G} \in \mathbb{R}^{R_1 \times R_2 \times R_3}$  is the **core tensor**,  
2775 •  $U^{(1)} \in \mathbb{R}^{I \times R_1}$ ,  $U^{(2)} \in \mathbb{R}^{J \times R_2}$ , and  $U^{(3)} \in \mathbb{R}^{K \times R_3}$  are the **factor matrices**,  
2776 •  $\times_n$  denotes the mode- $n$  tensor-matrix product.

2777 In index notation, this becomes:

$$\phi_{i,j,k} = \sum_{\alpha=1}^{R_1} \sum_{\beta=1}^{R_2} \sum_{\gamma=1}^{R_3} \mathcal{G}_{\alpha,\beta,\gamma} \cdot U_{i,\alpha}^{(1)} \cdot U_{j,\beta}^{(2)} \cdot U_{k,\gamma}^{(3)}.$$

2778 With this formulation the parameters  $R_1, R_2, R_3$  are set by the user and define the “storage  
2779 space” in which the information required to represent  $\phi$  has to be condensed. The trainable  
2780 parameters are all the components of  $\mathcal{G}, U^{(1)}, U^{(2)}, U^{(3)}$  therefore scaling from  $\mathcal{O}(L \times H \times W)$   
2781 of a grid-like parametrization to  $\mathcal{O}(R_1 \times R_2 \times R_3 + R_1 \times L + R_2 \times H + R_3 \times W)$ , which is  
2782 also convenient in terms of memory for small core tensors. It is proven that for  $R_i = L, H, W$   
2783 respectively, the tensor  $\phi$  is exactly represented but in practice, for typical BCDI datasets of  
2784  $200 \times 200 \times 200$  pixels a cubic 8 pixels sided  $\mathcal{G}$  is sufficient to represent the phase  $\phi$ . Note,  
2785 that in the general formulation of the Tucker decomposition the  $U^{(1)}, U^{(2)}, U^{(3)}$  are not strictly  
2786 unitary as for the matrices  $U, V$  in the SVD decomposition. The Tensorflow implementation

2787 of the Tucker decomposition is thus rather straightforward as the function `tf.einsum()`  
 2788 takes care of the tensor contraction.

2789

2790 A different approach that has been considered, leverages the TV regularization to push the  
 2791 algorithm towards a smooth object phase. The full  $L \times H \times W$  tensor is therefore optimized  
 2792 and a penalty on the sum of the absolute value of the gradients of the phase is added to the loss  
 2793 function. Precisely, the formula that has been used calculates the sum of the *squared* gradients,  
 2794 since the square root operation, necessary to obtain the correct formula, creates a problem  
 2795 around zero because of the infinite gradient. The final equation is therefore:

$$TV = \alpha \sum_{i=1}^L \sum_{j=1}^H \sum_{k=1}^W \mathcal{S}[(\varphi_i - \varphi_{i-1})^2 + (\varphi_j - \varphi_{j-1})^2 + (\varphi_k - \varphi_{k-1})^2] \quad (7.3)$$

2796 where  $\alpha$  is a hyperparameter that acts as a scaling factor,  $(i, j, k)$  are the indices running  
 2797 over the coordinates of the  $L \times H \times W$  grid and  $\mathcal{S}$  is the object support. The parameter  $\alpha$  in  
 2798 this case was chosen to be assigned relative to a fraction  $\alpha_0$ , chosen by the user, of the TV loss  
 2799 with respect to the data fidelity loss. In such a way, the TV penalty is rescaled with respect to  
 2800 the data loss. Typical values of  $\alpha_0$  range between (0.01, 0.9). Further developments could aim  
 2801 at finding adaptive formulations for the magnitude of  $\alpha$ .

### 2802 7.2.3 Loss function

2803 Another important aspect of the model is the loss function. Typically, for inverse problems there  
 2804 is a *data fidelity* term that in this case measures the distance between  $I_{obs}$  and  $I_{calc}$  according  
 2805 to some metric, and other additional *regularization* terms that guide the optimization process  
 2806 with physical constraints.

2807 **Data fidelity:** The most common and intuitive metrics are the Mean Squared Error (MSE)  
 2808 and the Mean Absolute Error (MAE) that evaluate the Euclidean distance between the observed  
 2809 and calculated intensities. Practically, because of the large dynamic range of typical BCIDI data,  
 2810 the MAE performs better as it doesn't focus on bright pixels only, but manages to correct for  
 2811 lower intensity tails as well. A more faithful metric for BCIDI experimental data is the Poisson  
 2812 Negative Log-Likelihood (P-NLLK). This metric assumes indeed the handling of count data,  
 2813 like the type obtained by photon counting detectors, and that the stochasticity of physical  
 2814 process is Poisson distributed. When summed over the full dataset, the discrepancies between  
 2815 calculated and observed intensities are not intended as Euclidean distances but like divergences  
 2816 between two probability distributions. In other words, the P-NLLK estimates the likelihood  
 2817 that  $I_{calc}$  belongs to the same Poisson distribution of  $I_{obs}$  [205]. Derived from the equation for  
 2818 the probability for Poissonian events, the formula of the averaged P-NLLK, in the form of a  
 2819 Kullback-Liebler divergence is:

$$\langle P - LLK \rangle = \frac{2}{N_{obs}} \left[ \sum_{I_{obs} > 0} \left( I_{calc} - I_{obs} + I_{obs} \ln \frac{I_{obs}}{I_{calc}} \right) + \sum_{I_{obs} = 0} I_{calc} \right]. \quad (7.4)$$

2820 Both the MAE and the P-NLLK have been tested on several simulated and experimental

2821 datasets and the MAE has always shown better convergence. An explanation for this unexpected  
2822 result is yet to be found, but the main suspect is that the gradients calculated during the back-  
2823 propagation can have instabilities because of the logarithm.

2824 **Regularizations:** Beside the TV on the object's phase to ensure smoothness, another term  
2825 that was considered concerns the size of the support. For a given data fidelity value, it is known  
2826 that the object with the smallest support represents the optimal solution [192]. Large supports  
2827 can induce the model to overfit the observed intensity. Intuitively this could be explained with  
2828 the fact that there are many more object phase configurations that would combine constructive  
2829 and destructive interferences to match the observed intensity in reciprocal space. The analogous  
2830 measure is the shrink-wrap algorithm [94] used in alternating projections algorithms. For  
2831 this reason a penalty  $P$  on the size of the support can be added to the loss function with the  
2832 formula:

$$P = \beta \sum_{i,j,k} \mathcal{S} \quad (7.5)$$

2833 where  $\beta$  is a hyperparameter that similarly to  $\alpha$  is calculated from the parameter  $\beta_0$  chosen  
2834 by the user with respect to the data fidelity loss. Typical values of  $\beta_0$  range between (0.01 - 0.5).  
2835 Both the  $\alpha_0, \beta_0$  hyperparameters can be tuned manually during the optimization, to adjust in  
2836 case of need, the strength of the regularization terms. The final formula for the loss function  
2837 can be ultimately expressed as:

$$L = \frac{1}{N_{\text{obs}}} \sum_{i=1}^{N_{\text{obs}}} |I_{\text{calc},i} - I_{\text{obs},i}| + \alpha TV(\varphi) + \beta P(\mathcal{S}) \quad (7.6)$$

2838 For the optimization, a tolerance on the MAE value or a fixed number of steps can be set  
2839 to stop the algorithm. Empirical observations have shown that a MAE value around 0.2 is  
2840 sufficiently low for the result to be considered good. However, an additional refinement with a  
2841 few iterations ( $\sim 300$ ) is recommended for cross-validation.  
2842

2843 Before concluding the section, it is worth highlighting that this AD implementation offers  
2844 the possibility to simultaneously run multiple reconstructions in parallel, efficiently on the GPU.  
2845 One can create a 4D tensor by stacking several copies of the 3D intensity data, creating therefore  
2846 a batch. For each element in the batch a different initial support and phase configuration can  
2847 be chosen, hence increasing the likelihood to converge to the solution.

## 2848 7.3 Results: simulated data

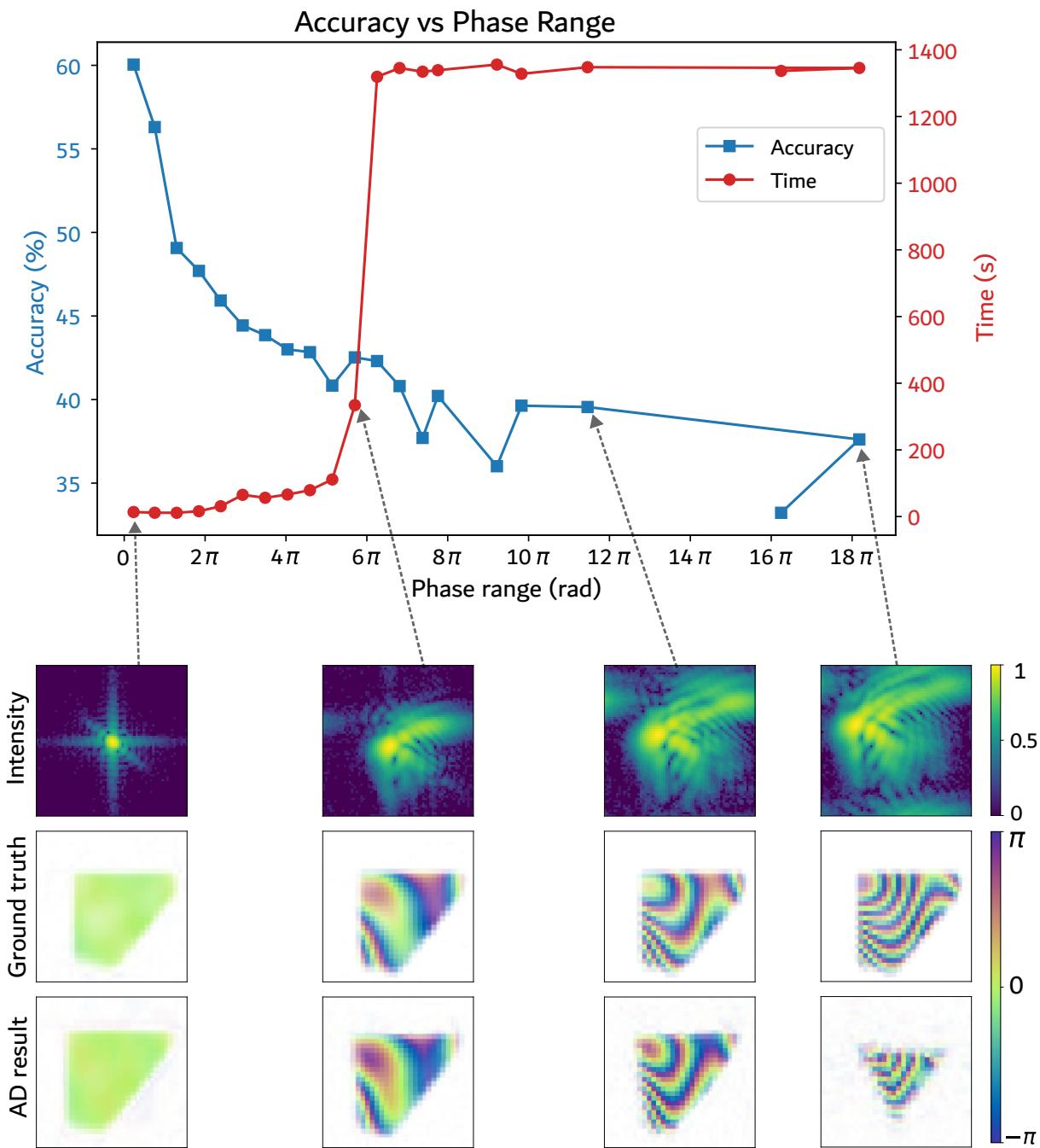
2849 Some tests have been conducted on simulated data. Similar to that described in Section 6.10,  
2850 a 3D randomly shaped faceted particle has been created with an applied Gaussian correlated  
2851 random phase. Subsequently, the diffraction pattern has been simulated on a  $64 \times 64 \times 64$  grid.  
2852 The amplitude of the phase field has been gradually increased from 0 to  $19\pi$  inside the particle  
2853 and for each of the 20 steps the corresponding diffraction pattern was simulated with DFT,  
2854 keeping a constant oversampling ratio and noise level.

2855      The PR has been conducted with the following parameters:

Parameter	Value
batch_size	20
nb_half_spaces	128
sigmoid_eps	0.6
phase_type	'tucker'
ranks	(8, 8, 8)
alpha_TV	0.0
beta_small	0.01
initial_lr	0.05
tol	0.09
max_iter	10000

**Table 7.1:** Parameters initialization for the AD model. In order they represent: (i) the number of copies optimized in parallel, (ii) the number of half-spaces used to build each of the optimized objects' shape, (iii) the  $\epsilon$  parameter controlling the "spatial resolution" of the support, (iv) the object phase is parametrized with the Tucker decomposition, and (v) is the size of the 3D core tensor (vi) the coefficient multiplying the TV loss on the phase (vii) the coefficient multiplying the penalty on the support size, (viii) the initial learning rate for the ADAM optimizer and (ix) tolerance on the MAE error and (x) the number of maximum iterations.

2856      The results are shown in Fig.7.2. Objects with small phases are reconstructed within a few  
 2857      seconds while for larger phases the computational time rapidly increases to the maximum fixed  
 2858      by the limit of iterations.



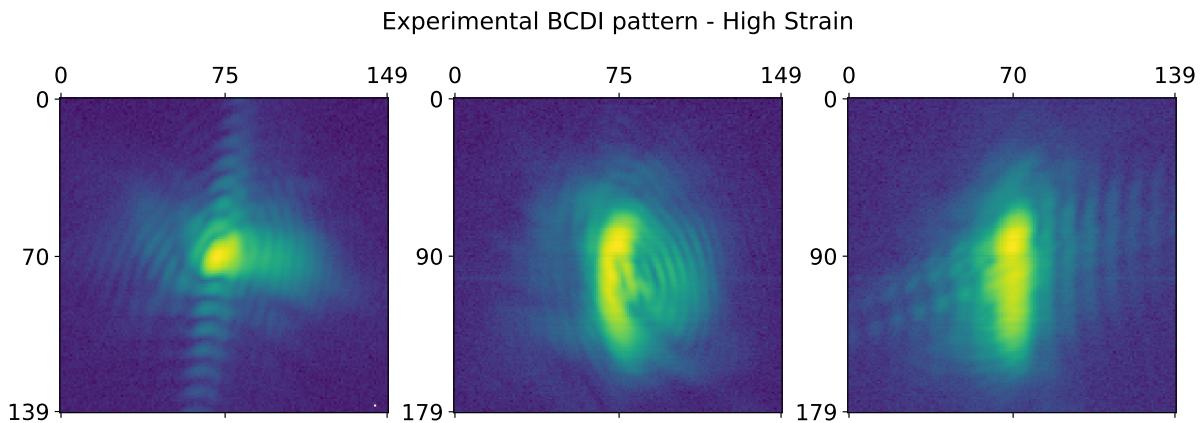
**Figure 7.2:** Results of the PR on simulated data for increasing applied phase magnitude. The accuracy has been calculated with the formula in Eq.6.5 and the time is measured from the start of each run until it ends, either when the tolerance is surpassed or when the maximum number of iterations is reached. For the smallest phase range (0.72 radians) the PR is completed within 11 seconds. Starting from phase ranges exceeding  $6\pi$  the model runs over all the allowed iterations. The PR convergence is however dependent on the initialization of the support (randomly oriented half-spaces) and the phase (random uniform between  $-\pi$  and  $\pi$ ), thus the oscillations of accuracy scores. Notice that the heavily distorted diffraction patterns here depicted show a significant amount of aliasing. Although never occurring in experimental data, this is not a problem to the scope of this analysis.

2860 faceted crystals.

## 2861 7.4 Results: experimental data

2862 In this section two relevant results will be presented. The first example is a highly strained  
 2863 palladium particle on sapphire substrate measured at the ID01 beamline of the ESRF [206]. The  
 2864 large strain inside the particle distorts the BCDI pattern and makes the reconstruction with  
 2865 conventional iterative algorithms overly challenging.

### 2866 7.4.1 Hardly-invertible BCDI patterns with high strain



**Figure 7.3:** Projections along the three axes of a BCDI pattern of the highly strained Pd particle.

2867 Here a comparison between different reconstruction methods will follow. Namely, (i) the results  
 2868 obtained from 60 independent runs of standard PR using PyNX software. Each run consisted of  
 2869 400 HIO iterations followed by 1000 RAAR and 300 ER ones. The parameters are displayed in  
 2870 Table 6.2 already mentioned in the previous chapter. At the end of the process, the two best  
 2871 reconstructions among the 60 runs according to the “mean to max” metric were combined  
 2872 using the mode decomposition technique proposed in [192]. This method is referred in the text  
 2873 to as “PyNX”.

2874 The second method makes use of the DL model presented in the previous chapter. The  
 2875 large original data is firstly binned to a (196, 140, 140) shape and then cropped in a (80,90,110)  
 2876 shaped ROI and resized to a 64 pixel-sided cube for the DL model. The DL predicted object is  
 2877 then interpolated back to the original size and refined with PyNX using a single run of 300 ER,  
 2878 with the parameters listed in Table 6.3. This combined method is referred in the text to as “DL  
 2879 + PyNX”.

2880 The third method employs the AD model presented above with the parameters listed in  
 2881 Table 7.2. Additionally, as last step, the final object is obtained computing the IFFT of the  
 2882 complex diffracted amplitude built using the experimental diffracted measurement as modulus  
 2883 and the RSP extracted from the FFT of the object itself. This last step that can also be seen as a  
 2884 “halfER step”. The method described here is referred in the text to as “AD”.

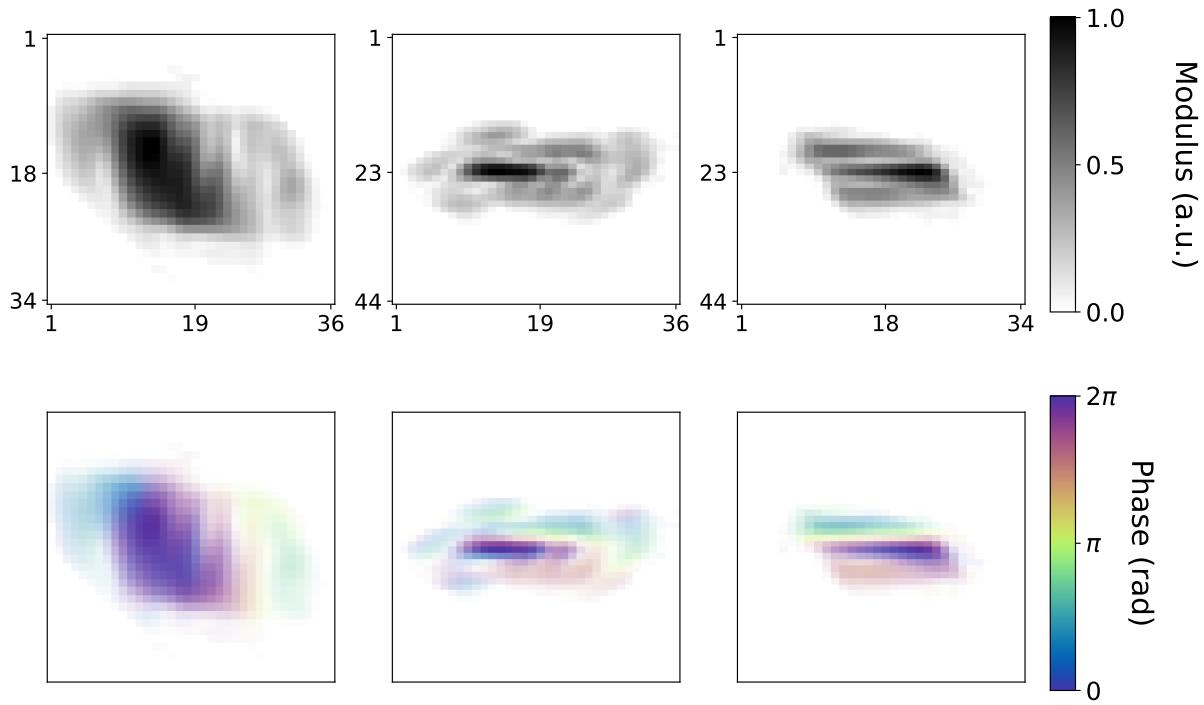
2885

Furthermore, similarly to the DL case, the AD retrieved object can be refined with 300 cycles of ER using PyNX. In this case, this last passage can be used to verify the credibility of the found solution. In fact, such defined AD model can always yield a faceted crystal with homogeneous amplitude that is however far from the actual solution. This method, that runs PyNX with the parameters shown in Table 6.3 using as initial guess the object found with the AD, is in the text referred to as “AD + PyNX”.

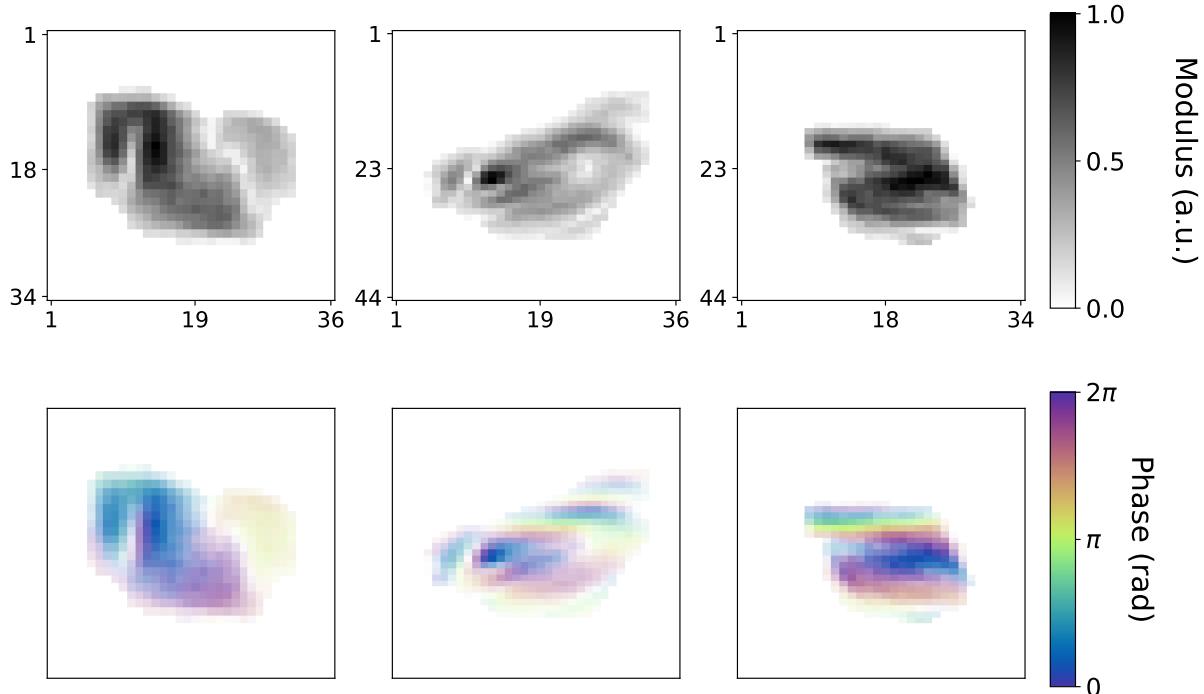
Parameter	Value
batch_size	5
nb_half_spaces	128
sigmoid_eps	0.9
phase_type	‘tucker’
ranks	(16, 16, 16)
alpha_TV	0.0
beta_small	0.01
initial_lr	0.05
tol	0.2

**Table 7.2:** Parameters initialization for the AD model for the phasing of experimental data. Given the larger size of the data as compared to the simulated ones, some parameters have been adjusted with respect to Table 7.1. Namely, the number of copies has been reduced for memory issues, the core tensor for the Tucker decomposition has been increased and the tolerance and maximum iterations constraints have been relaxed.

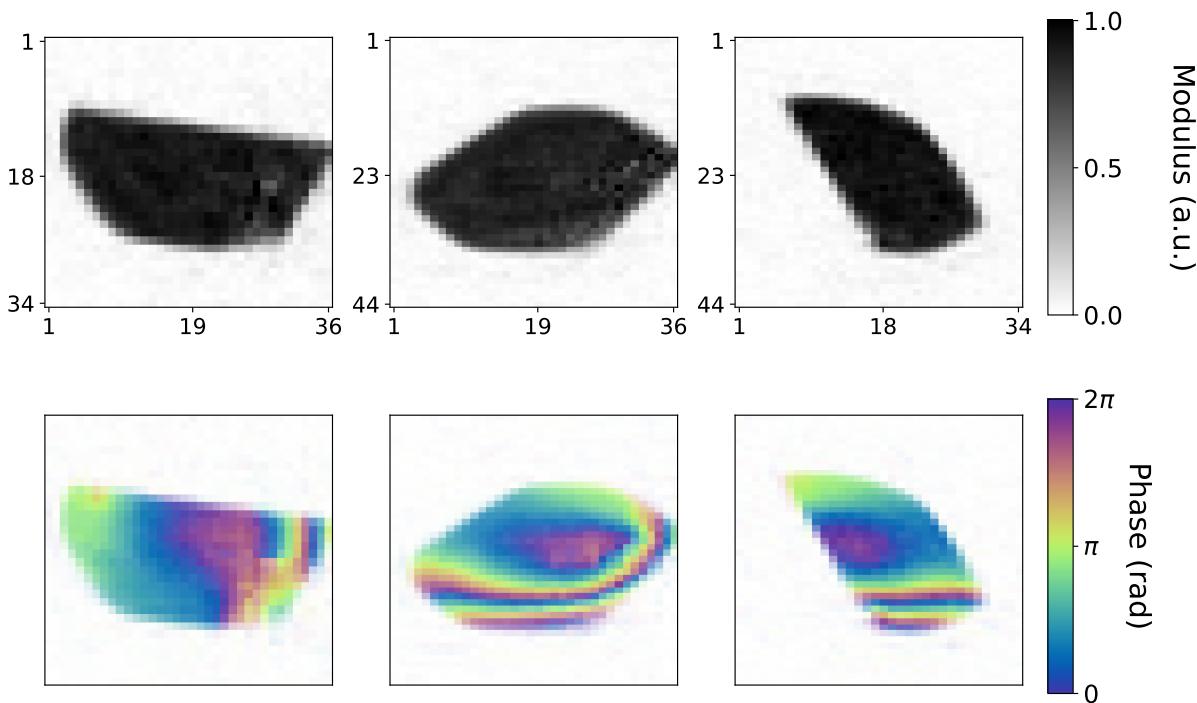
The following Figures 7.4 - 7.5 - 7.6 - 7.7 show the results of the reconstructions obtained with the “PyNX”, “DL + PyNX”, “AD” and “AD + PyNX” methods respectively, while Table 7.3 displays the scores of the Poisson LLk (calculated from Eq.7.4), MAE and MSE metrics evaluated for the  $I_{obs}$  and  $I_{calc}$  in each case.



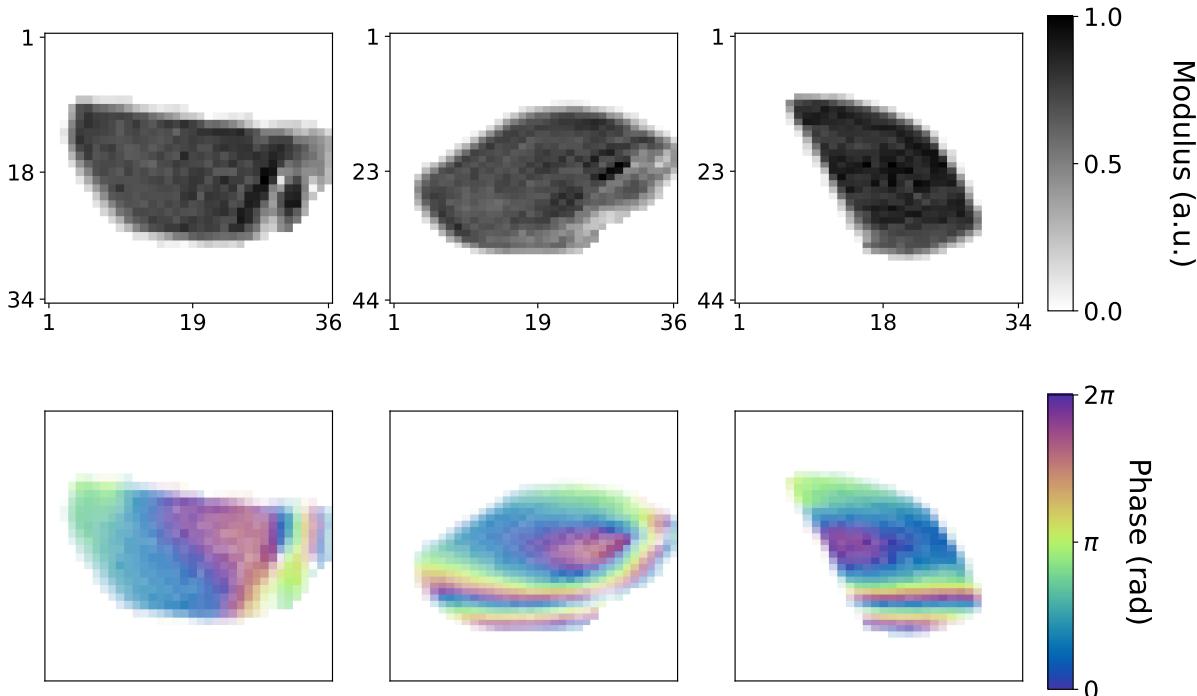
**Figure 7.4:** Central slices for modulus (first row) and phase (second row) of the reconstruction obtained with the PyNX method. The presence of holes and inhomogeneous object's electron density suggests a poor quality reconstruction.



**Figure 7.5:** Central slices for modulus (first row) and phase (second row) of the reconstruction obtained with the DL + PyNX method. Although the increased quality, the result cannot be considered a good reconstruction.



**Figure 7.6:** Central slices for modulus (first row) and phase (second row) of the reconstruction obtained with the AD method. The model converges to a reasonable result for Winterbottom shaped particle with visible high-strain given by the large phase ramp wrapped multiple times. The model overcomes the tolerance level within 1000 steps, for an elapsed time of 325 seconds



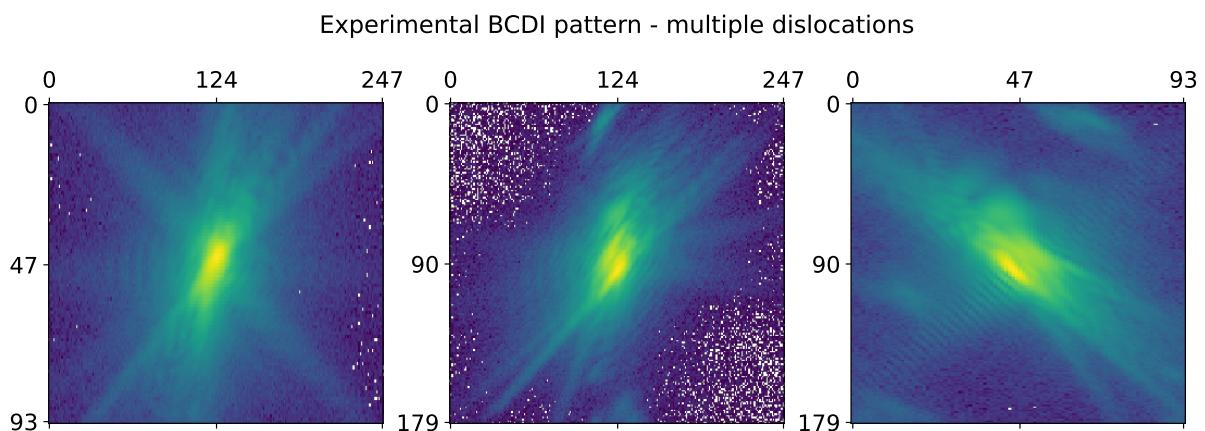
**Figure 7.7:** Central slices for modulus (first row) and phase (second row) of the reconstruction obtained with the AD + PyNX method. The 300 iterations of ER refinement do not alter the shape nor the phase found by the AD model, therefore validating the solution.

	Poisson LLK	MAE	MSE
AD	3.3083e-01	4.1249e-02	7.3738e-03
AD + PyNX	2.9259e-01	1.1893e-01	5.4739e-03
DL + PyNX	3.5195e-01	1.5866e-01	1.5841e-02
PyNX	6.3946e-01	3.1050e-01	3.7996e-02

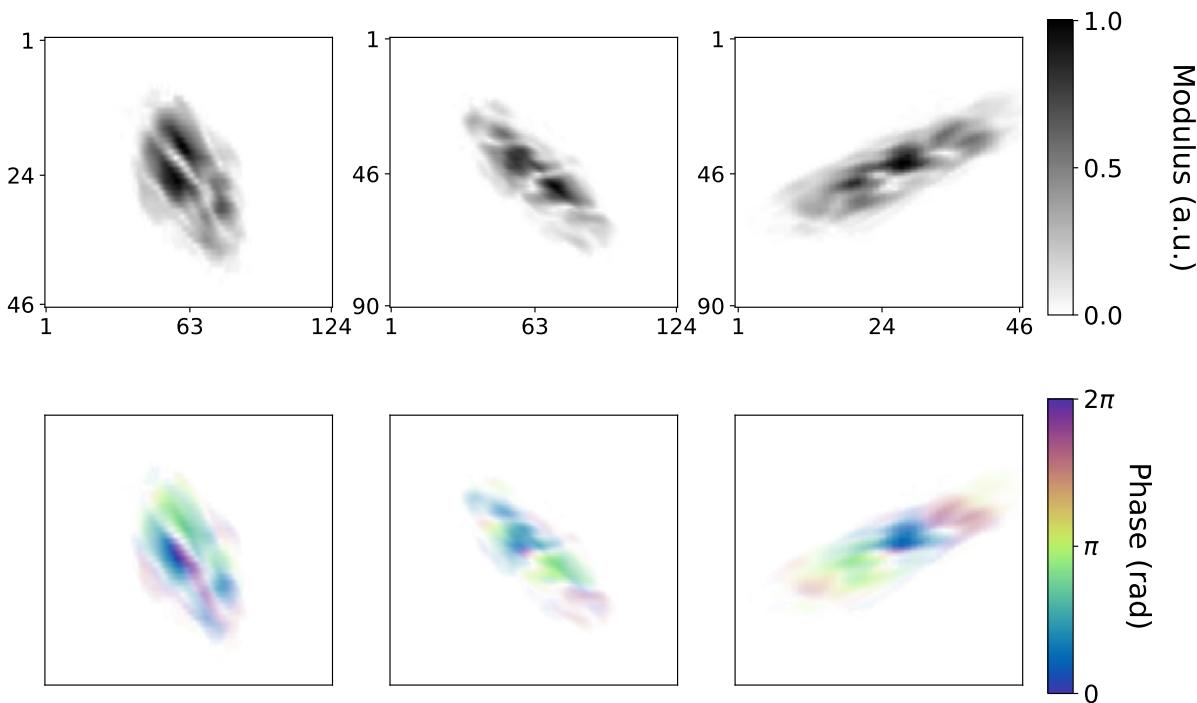
**Table 7.3:** Performance metrics between  $I_{calc}$  and  $I_{obs}$  for the Pd particle calculated for normalized magnitudes. The lowest scores are highlighted

#### 2896 7.4.2 Hardly-invertible BCDI patterns with multiple dislocations

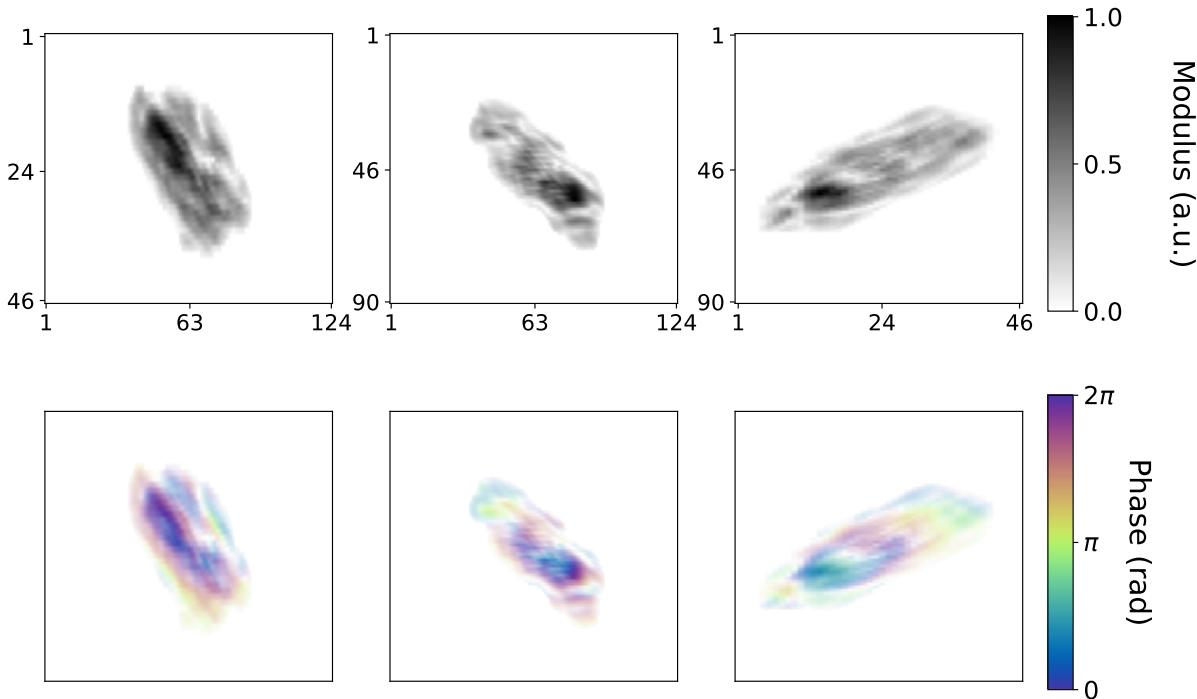
2897 In this paragraph another illustrative example is presented. This time a platinum particle on  
 2898 YSZ substrate. The four different methods presented above are repeated here for this dataset in  
 2899 the same way, except for the AD method, for which the phase has been parametrized with the  
 2900 full grid (phase\_type: grid) and  $\alpha_{TV} = 0.1$  parameter. Although more expensive in  
 2901 terms of memory, this parametrization is more suitable for particles with dislocations, where  
 2902 large phase variations are concentrated in a few voxels, and a built-in sparse formulation can  
 2903 struggle with these high-frequency features. The results are shown in the following Figures 7.9  
 2904 - 7.10 - 7.11 - 7.12 and Table 7.4



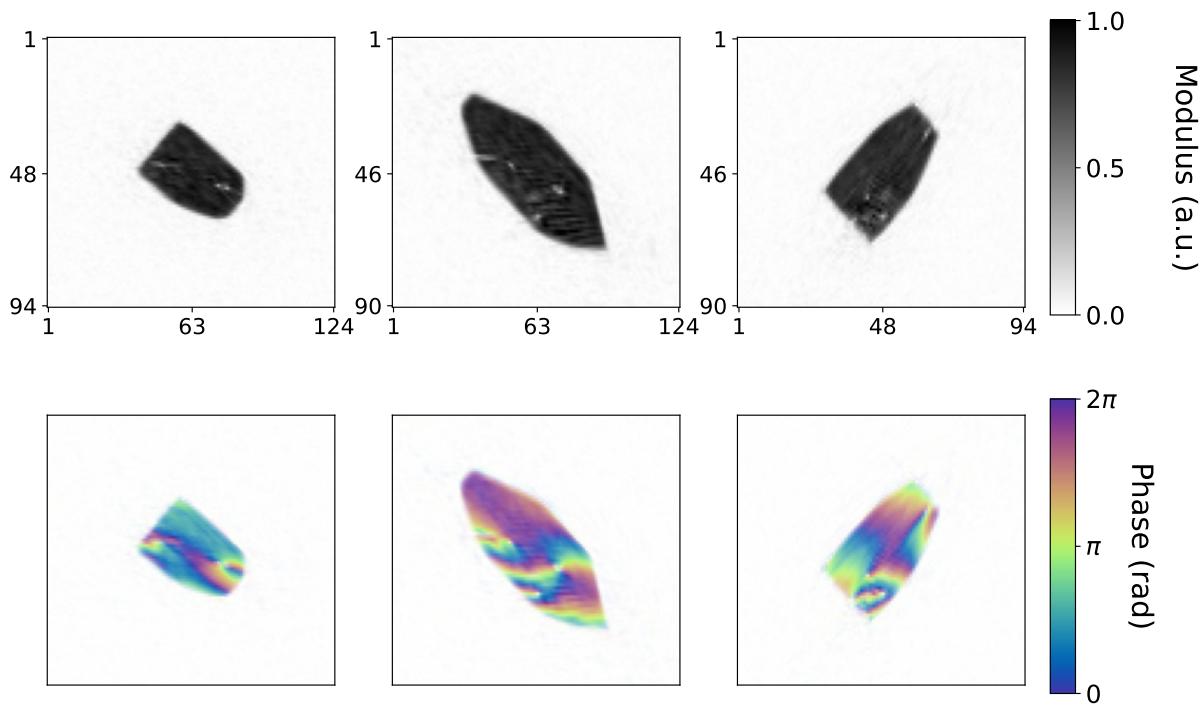
**Figure 7.8:** Projections along the three axes of a diffraction pattern from a Pt particle with multiple dislocations.



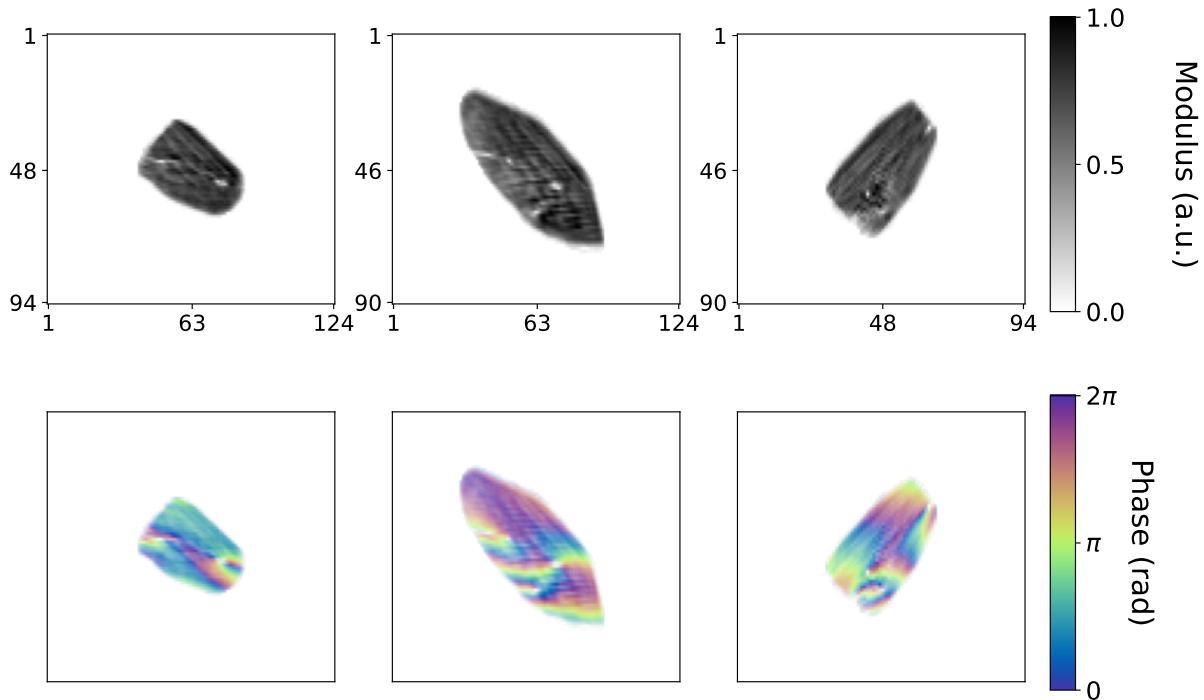
**Figure 7.9:** Central slices for modulus (first row) and phase (second row) of the reconstruction obtained with the PyNX method. The presence of holes and inhomogeneous object's electron density suggests a poor quality reconstruction.



**Figure 7.10:** Central slices for modulus (first row) and phase (second row) of the reconstruction obtained with the DL + PyNX method. Here, a failure of the DL prediction is expected also because it hadn't been trained on datasets with dislocations.

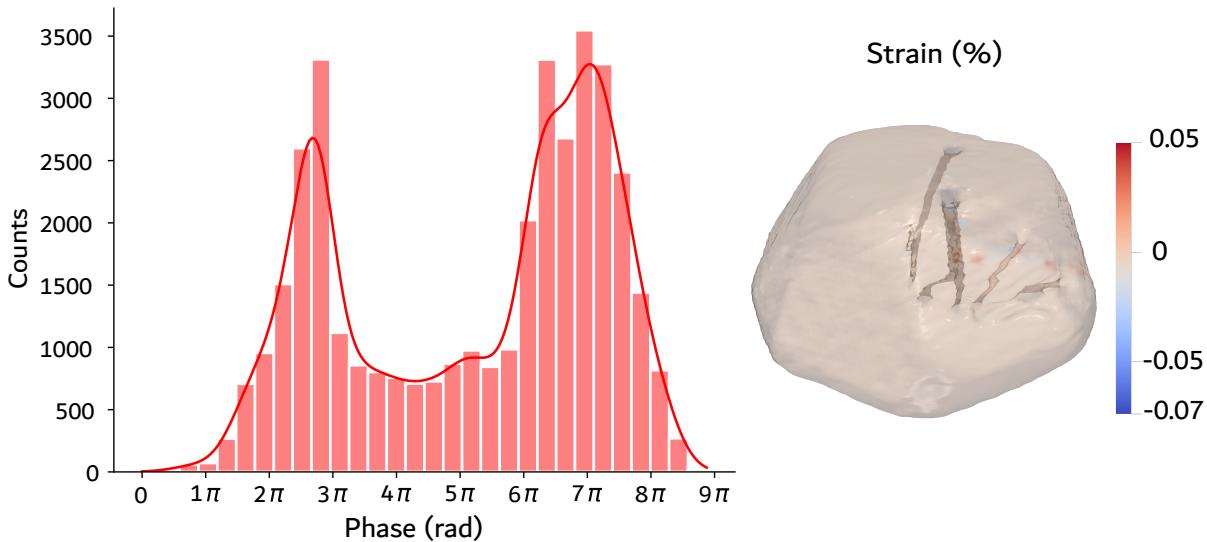


**Figure 7.11:** Central slices for modulus (first row) and phase (second row) of the reconstruction obtained with the AD method. The AD model converges to a reasonably faceted crystal with several dislocations. The model overcomes the tolerance level within 1500 steps, for an elapsed time of 930 seconds



**Figure 7.12:** Central slices for modulus (first row) and phase (second row) of the reconstruction obtained with the AD + PyNX method. The 300 iterations of ER refinement do not alter the shape nor the phase found by the AD model, therefore validating the solution. Some ripples appear instead in the objects' modulus and phase, probably caused by the presence of some neighbor crystals illuminated by the x-ray beam, scattering on the same detector ROI (*aliens*).

	Poisson LLK	MAE	MSE
AD	2.5650e-01	1.3828e-02	1.0402e-02
AD + PyNX	1.5138e-01	8.5945e-02	3.0269e-03
DL + PyNX	2.9194e-01	1.6544e-01	1.4580e-02
PyNX	3.8114e-01	2.4265e-01	1.7078e-02

**Table 7.4:** Performance metrics between  $I_{calc}$  and  $I_{obs}$  for the Pt particle.**Figure 7.13:** Histogram of the phase range and 3D strain visualization for the Pt particle in Fig. 7.12. The large phase range extending over  $8\pi$  confirms the challenging PR. The 3D rendering with reduced opacity highlights the multiple dislocation cores traversing the particle (orthogonalized).

## 2905 7.5 Conclusions

2906 In this chapter a novel PR method for BCDI based on a physics-informed AD model has been  
 2907 presented. The goal of this section was to explore GD-based methods for PR, aiming at resolving  
 2908 those difficult cases for which both conventional iterative algorithms and DL assisted methods  
 2909 struggle. To sum up, the major factors that made the model successful are here listed briefly:

- 2910 • The efficient exact gradient calculation offered by the automatic differentiation, already  
 2911 identified by Jurling and Fienup as potential alternative to alternating projections for PR, is  
 2912 today easily accessible and GPU accelerated by common machine learning programming  
 2913 libraries like Tensorflow and PyTorch. This ingredient is fundamental for PR of 3D  
 2914 datasets in competitive computational times, comparable to standard PR algorithms  
 2915 optimized for GPUs.
- 2916 • The ADAM optimizer offers adaptive step-size tuning (eliminating the need to set a single  
 2917 learning rate for all parameters) and first and second order moment estimation (thus  
 2918 providing both direction smoothing and per-parameter scaling), leading to more stable

and faster convergence. Additionally, it allows computing gradients on sub-volumes or slices of the diffraction pattern at each step, hence introducing stochasticity that could be beneficial for escaping local minima. This feature, could be developed in future versions of the model, and strategic batches organization (i.e. concentric sub-volumes from the center of the Bragg peak) could be explored as well.

- The possibility to easily embed physical constraints in the forward model. In this case the prior knowledge on the homogeneous and compactly supported nature of the crystal electronic density can be easily implemented with the half-spaces method, thus restricting the solution space without need for additional regularization. In the same way, the Tucker decomposition of the object's phase tensor facilitates the finding of physical solutions.
- The multidimensional tensor-based computations typical of modern machine learning libraries make it straightforward to extend the optimization to multiple parallel instances of the same phase retrieval problem, with only a modest increase in computational time. The main limitation, however, lies in the substantial GPU memory requirements, as this approach can be highly memory-intensive. To give some useful numbers, with the parameters shown in Table 7.2 a GPU with 32GB of RAM can sustain the optimization of maximum 30 copies of a  $64 \times 64 \times 64$  diffraction pattern.
- The loss function definition gives the user high flexibility of implementing the most suitable metric for the specific problem, allowing for parameter tuning during the optimization as well. In this case the MAE metric was found to be the best one for the BCDI problem.

This study has shown that AD-based PR for BCDI is a valuable alternative to conventional or data-driven PR algorithms. It offers an additional tool that can extend the range of applicability of the BCDI technique to highly defective or strained crystals, the reconstruction of which is problematic. The flexibility of the programming enables the implementation of targeted models, and it is suitable for tailored constraints, important for PR of challenging datasets.

The current implementation however still requires significant amounts of time for particularly ill-posed PR, yet it achieved promising results on experimental datasets whose phase retrieval had previously been discarded due to the inability of conventional algorithms to yield a reconstruction. Lastly, for future developments, the model can be extended to non-convex objects or separated ones (e.g. in case of twin boundaries) as well as different forward models beyond the kinematic approximation or further embedding of prior knowledge.



---

2952 CONCLUSIONS AND PERSPECTIVES

2953 In this PhD dissertation the use of deep convolutional neural networks and algorithmic dif-  
2954 ferentiation has been explored for the processing of BCDI data. Chapter 2 has presented the  
2955 physics of coherent x-ray diffraction on single crystal, highlighting the effect of internal lattice  
2956 displacement on the data. Moreover, a short practical overview of the typical BCDI experiments  
2957 was given as well. Chapter 3 was dedicated to the fascinating Fourier phase problem. The  
2958 uniqueness conditions, the PR iterative algorithms and some insights on the high-strain case  
2959 were discussed. In particular, the link between the effects of the high-strain on the diffraction  
2960 pattern, illustrated in Chapter 2 and the relative increased difficulty of the PR shown in Chapter  
2961 3 has been emphasized. What emerged in the discussion is that improved performance of the  
2962 PR is obtained when the problem is *regularized* with some prior knowledge that constrains  
2963 or guides the search of the solution. Here, the connection to neural networks introduced in  
2964 Chapter 4 becomes apparent. The strong inductive prior of convolutional neural networks for  
2965 structured images, combined with a targeted data-driven strategy, was explored in Chapters 5  
2966 -6, yielding satisfactory results on two different kinds of inverse problems.

2967 Specifically, in Chapter 5 the preliminary investigations and the development of a novel  
2968 patching-based model for addressing detector gaps in BCDI data were presented. The results  
2969 obtained on new simulated and experimental datasets confirmed the capability of convolutional  
2970 neural networks to extrapolate information from structured images and to accurately predict  
2971 the continuation of patterns within missing data regions. In Chapter 6 the goal of assisting  
2972 standard iterative algorithms during the PR of highly-strained BCDI patterns has been achieved  
2973 with the use of a convolutional neural network trained with the novel WCA loss function  
2974 for the prediction of the *reciprocal space phase*, unlike what is present in the literature. The  
2975 successful results attained by the DL + PyNX refinement method can significantly improve the  
2976 BCDI technique by drastically reducing the computational time needed for the reconstructions  
2977 of experimental data. Over the long term, the computational resources required to train the DL  
2978 model are expected to be minimal relative to the substantial time and energy savings achieved  
2979 through DL-based initialization of the PR.

2980 Chapter 7 instead moves away from the data-driven approach. Here, the computational  
2981 framework for automatic differentiation is leveraged for a physics constrained PR solved with  
2982 gradient descent. The prior knowledge of uniform electron density inside the object support  
2983 and the faceted nature of crystals' surfaces does not come from the data, nor from penalty  
2984 terms but from construction constraints.

2985 Integrating the presented algorithms into the standard BCDI data analysis pipeline should  
2986 be foreseen to enable more systematic usage. So far, all three main models have been employed  
2987 by ID01 users for inpainting and phase retrieval tasks, with experimental results soon to be  
2988 published. However, before this integration some further developments could be considered. I  
2989 will provide an outlook on potential directions with the following.

2990 • **DL-based Gap Inpainting.** The patching approach has proven to have many benefits,  
2991 including faster training, larger training datasets and possibility of targeted fine-tuning.  
2992 The bottleneck of this method is however given by the size of the gaps. Detectors like  
2993 EIGER, have large gaps (12 - 38 pixels) for which the patching approach is not suitable.  
2994 One could get around this limitation increasing the patch size, however, in those cases  
2995 other methods leveraging some type of regularization during the reconstruction [146]  
2996 can offer a better alternative to a DL approach. Another interesting development could  
2997 aim to reduce the number of repeated applications of the DL model along the gap. In  
2998 fact, given the typical structure of BCDI patterns and experimental conditions, the gap  
2999 only affects a small region of signal, often on long truncation rods streaks. An adaptive  
3000 algorithm for the inpainting of those regions only, could save time and computations  
3001 discarding the inpainting of dark regions. More in general, it would be interesting to test  
3002 the patching-based inpainting on data from other imaging techniques.

3003 • **DL-based Phase Retrieval.** The RSP prediction on patches is at the same time appealing  
3004 and challenging. The results obtained on independent patches are promising, but the  
3005 stitching of the patches still represents a challenge. As anticipated in the conclusion  
3006 Chapter 6, the design of a Recurrent convolutional neural network could better address  
3007 the problem. In particular, I would suggest an approach similar to what proposed by  
3008 Pinheiro and Collobert in [207] could be considered. There, instead of stacking many  
3009 different layers to increase receptive fields, they used *weight-sharing* recurrence, i.e. the  
3010 same convolutional layer is applied multiple times to its own output. This allows the  
3011 network to iteratively refine predictions and integrate increasingly larger context without  
3012 adding new parameters.

3013 This approach could also break the symmetry problem that was evidenced in the end  
3014 of Chapter 6 (see Fig. 6.59) for which the predicted RSP always tend to show a radial  
3015 symmetry like the average, over the whole training dataset, of the diffraction intensity  
3016 that the model receives as inputs.

3017 • **AD-based Phase Retrieval.** Concerning this project, in my opinion the potential is high  
3018 and several developments, additions, integrations can be foreseen. The high flexibility  
3019 and efficiency of gradient-based optimizations provided by modern machine learning  
3020 libraries allows for relatively easy implementation of tailored models. The first important  
3021 extension to the current formulation should include the modeling of non-convex surfaces  
3022 as well. An idea that would maintain the half-spaces approach would be to define two  
3023 or more convex volumes built with the half-spaces method and find the final support  
3024 with union, subtraction and intersection operations. Additionally, other upgrades to the  
3025 current model could make use of the observed intensity projection (the projection on the  
3026 modulus constraint set presented in Chapter 3), combined with the gradient descent, as  
3027 well as the use of mini-batches to enable stochastic gradient descent, for faster and more  
3028 robust convergence.

3029 More in general, it is clear that the utilization of GPU-accelerated AD will become  
3030 a pivotal tool for efficient gradient-based optimization in various scientific disciplines  
3031 characterized by intensive computational demands. This approach is anticipated to gain  
3032 prominence in the coming years, potentially surpassing data-driven methodologies in  
3033 certain applications [127, 208, 209].



# BIBLIOGRAPHY

- 3036 1. Fienup, J. R. *Reconstruction of an object from the modulus of its Fourier transform* Publication  
3037 Title: OPTICS LETTERS Volume: 3 Issue: 1 (1978).
- 3038 2. Fienup, J. R. *Phase retrieval algorithms: a comparison* (1982).
- 3039 3. Luke, D. R. Relaxed averaged alternating reflections for diffraction imaging. *Inverse*  
3040 *Problems* **21**, 37–50. ISSN: 1361-6420. <http://dx.doi.org/10.1088/0266-5611/21/1/004> (Nov.  
3041 2004).
- 3042 4. Miao, J., Sayre, D. & Chapman, H. N. Phase retrieval from the magnitude of the Fourier  
3043 transforms of nonperiodic objects. *J. Opt. Soc. Am. A* **15**, 1662–1669. <https://opg.optica.org/josaa/abstract.cfm?URI=josaa-15-6-1662> (June 1998).
- 3045 5. Neutze, R., Wouts, R., van der Spoel, D., Weckert, E. & Hajdu, J. Potential for biomolecular  
3046 imaging with femtosecond X-ray pulses. *Nature* **406**, 752–757 (2000).
- 3047 6. Chapman, H. N. *et al.* High-resolution ab initio three-dimensional x-ray diffraction  
3048 microscopy. *J. Opt. Soc. Am. A* **23**, 1179–1200. <https://opg.optica.org/josaa/abstract.cfm?URI=josaa-23-5-1179> (May 2006).
- 3050 7. Schroer, C. G. *et al.* Coherent X-Ray Diffraction Imaging with Nanofocused Illumination.  
3051 *Physical Review Letters* **101**, 090801 (2008).
- 3052 8. Rodriguez, J. A. *et al.* Three-dimensional coherent X-ray diffractive imaging of whole  
3053 frozen-hydrated cells. *IUCrJ* **2**, 575–583 (2015).
- 3054 9. Robinson, I. K., Vartanyants, I. A., Williams, G. J., Pfeifer, M. A. & Pitney, J. A. Recon-  
3055 struction of the Shapes of Gold Nanocrystals Using Coherent X-Ray Diffraction. *Phys.*  
3056 *Rev. Lett.* **87**, 195505. <https://link.aps.org/doi/10.1103/PhysRevLett.87.195505> (19 Oct.  
3057 2001).
- 3058 10. Pfeifer, M., Williams, G., Vartanyants, I., Harder, R. & Robinson, I. Three-dimensional  
3059 mapping of a deformation field inside a nanocrystal. *Nature* **442**, 63–66 (July 2006).
- 3060 11. Robinson, I. & Harder, R. Coherent X-ray diffraction imaging of strain at the nanoscale.  
3061 *Nature materials* **8**, 291–8 (May 2009).
- 3062 12. Newton, M. C., Leake, S. J., Harder, R. & Robinson, I. K. Three-dimensional imaging  
3063 of strain in a single ZnO nanorod. *Nature Materials* **9**, 120–124. ISSN: 1476-4660. <https://doi.org/10.1038/nmat2607> (Feb. 2010).
- 3065 13. Favre-Nicolin, V. *et al.* Analysis of strain and stacking faults in single nanowires using  
3066 Bragg coherent diffraction imaging. *New Journal of Physics* **12**, 035013. <https://dx.doi.org/10.1088/1367-2630/12/3/035013> (Mar. 2010).
- 3068 14. Labat, S. *et al.* Inversion Domain Boundaries in GaN Wires Revealed by Coherent Bragg  
3069 Imaging. *ACS Nano* **9**, 9210–9216 (2015).
- 3070 15. Dupraz, M. *et al.* 3D Imaging of a Dislocation Loop at the Onset of Plasticity in an  
3071 Indented Nanocrystal. *Nano Letters* **17**, 6696–6701 (2017).

- 3072 16. Carnis, J. *et al.* Twin boundary migration in an individual platinum nanocrystal during  
3073 catalytic CO oxidation. *Nature Communications* **12**, 5385 (2021).
- 3074 17. Bellec, E. *et al.* Imaging the facet surface strain state of supported multi-faceted Pt  
3075 nanoparticles during reaction. *Nature Communications* **13**, 3003 (2022).
- 3076 18. Chatelier, C. *et al.* Unveiling Core–Shell Structure Formation in a Ni<sub>3</sub>Fe Nanoparticle with  
3077 In Situ Multi-Bragg Coherent Diffraction Imaging. *ACS Nano* **18**, 13517–13527 (2024).
- 3078 19. Grimes, M. *et al.* Capturing Catalyst Strain Dynamics during Operando CO Oxidation.  
3079 *ACS Nano* (2024).
- 3080 20. Singer, A. *et al.* Nucleation of dislocations and their dynamics in layered oxide cathode  
3081 materials during battery charging. *Nature Energy* **3**, 641–647. ISSN: 2058-7546. <https://doi.org/10.1038/s41560-018-0184-2> (Aug. 2018).
- 3083 21. Serban, D. *et al.* Imaging in-operando LiCoO<sub>2</sub> nanocrystallites with Bragg coherent X-ray  
3084 diffraction. *Communications Chemistry* **7**, 243. ISSN: 2399-3669. <https://doi.org/10.1038/s42004-024-01331-y> (2024).
- 3086 22. Atlan, C. *et al.* Imaging the strain evolution of a platinum nanoparticle under electrochem-  
3087 ical control. *Nature Materials* **22**. Publisher: Nature Research, 754–761. ISSN: 14764660  
3088 (June 1, 2023).
- 3089 23. Grünewald, T. A., Liebi, M. & Birkedal, H. Crossing length scales: X-ray approaches to  
3090 studying the structure of biological materials. *IUCrJ* **11**, 708–722. <https://doi.org/10.1107/S2052252524007838> (Sept. 2024).
- 3092 24. Björling, A. *et al.* Coherent Bragg imaging of 60nm Au nanoparticles under electrochem-  
3093 ical control at the NanoMAX beamline. *Journal of Synchrotron Radiation* **26**, 1830–1834.  
3094 <https://doi.org/10.1107/S1600577519010385> (Sept. 2019).
- 3095 25. Richard, M.-I. *et al.* Bragg coherent diffraction imaging of single 20 nm Pt particles at the  
3096 ID01-EBS beamline of ESRF. *Journal of Applied Crystallography* **55**, 621–625 (May 2022).
- 3097 26. Cherukara, M. J., Cha, W. & Harder, R. J. Anisotropic nano-scale resolution in 3D Bragg  
3098 coherent diffraction imaging. *Applied Physics Letters* **113**. Publisher: American Institute  
3099 of Physics Inc. ISSN: 00036951. arXiv: [1806.02898](https://arxiv.org/abs/1806.02898) (Nov. 12, 2018).
- 3100 27. Leake, S. J. *et al.* The nanodiffraction beamline ID01/ESRF: A microscope for imaging  
3101 strain and structure. *Journal of Synchrotron Radiation* **26**. Publisher: Wiley-Blackwell,  
3102 571–584. ISSN: 16005775 (Mar. 1, 2019).
- 3103 28. Pateras, A. *et al.* Combining Laue diffraction with Bragg coherent diffraction imaging  
3104 at 34-ID-C. *Journal of Synchrotron Radiation* **27**, 1430–1437. <https://doi.org/10.1107/S1600577520009844> (Sept. 2020).
- 3106 29. Tolentino, H. C. N. *et al.* CARNAÚBA: The Coherent X-Ray Nanoprobe Beamline for the  
3107 Brazilian Synchrotron SIRIUS/LNLS. *Journal of Physics: Conference Series* **849**, 012057.  
3108 <https://dx.doi.org/10.1088/1742-6596/849/1/012057> (June 2017).
- 3109 30. Newton, M. C., Nishino, Y. & Robinson, I. K. Bonsu: the interactive phase retrieval suite.  
3110 *Journal of Applied Crystallography* **45**, 840–843 (2012).

- 3111 31. Favre-Nicolin, V. *et al.* PyNX: High-performance computing toolkit for coherent X-ray  
 3112 imaging based on operators. *Journal of Applied Crystallography* **53**. Publisher: Interna-  
 3113 tional Union of Crystallography, 1404–1413. ISSN: 16005767. arXiv: 2008.11511 (Oct. 1,  
 3114 2020).
- 3115 32. Simonne, D. *et al.* Gwaihir: Jupyter Notebook graphical user interface for Bragg coherent  
 3116 X-ray diffraction data. *Journal of Applied Crystallography* **55**, 1325–1335 (2022).
- 3117 33. Atlan, C., Chatelier, C. & Olson, K. *cdiutils: A Python package for X-ray Bragg coherent*  
 3118 *diffraction imaging processing, analysis and visualisation workflows version 0.1.4-beta.*  
 3119 Zenodo software release. 2023. <https://github.com/clatlan/cdiutils>.
- 3120 34. Colombo, A. *et al.* SPRING: an effective and reliable framework for image reconstruction  
 3121 in single-particle Coherent Diffraction Imaging. *npj Computational Materials / Nature*  
*Computational Materials* “to be assigned”, … (2025).
- 3123 35. Li, P. *et al.* 4th generation synchrotron source boosts crystalline imaging at the nanoscale.  
 3124 *Light: Science & Applications* **11**, 73. ISSN: 2047-7538. <https://doi.org/10.1038/s41377-022-00758-z> (Mar. 2022).
- 3126 36. Kim, C. *et al.* Three-Dimensional Imaging of Phase Ordering in an Fe-Al Alloy by Bragg  
 3127 Ptychography. *Phys. Rev. Lett.* **121**, 256101. <https://link.aps.org/doi/10.1103/PhysRevLett.121.256101> (25 Dec. 2018).
- 3129 37. Chamard, V. *et al.* Strain in a silicon-on-insulator nanostructure revealed by 3D x-ray  
 3130 Bragg ptychography. *Scientific Reports* **5**, 9827. ISSN: 2045-2322. <https://doi.org/10.1038/srep09827> (May 2015).
- 3132 38. Als-Nielsen, J. & McMorrow, D. *Elements of Modern X-ray Physics* 2nd ed. ISBN: 978-0-  
 3133 470-97376-9 (Wiley, Chichester, UK, 2011).
- 3134 39. Guinier, A. *X-ray Diffraction in Crystals, Imperfect Crystals, and Amorphous Bodies* Reprint  
 3135 Edition. ISBN: 978-0-486-68011-4 (Dover Publications, New York, 1994).
- 3136 40. Paganin, D. M. *Coherent X-Ray Optics* 1st ed. English. ISBN: 0198567286 (Oxford University  
 3137 Press, Oxford, United Kingdom, 2006).
- 3138 41. Vartanyants, I. A. & Yefanov, O. M. *Coherent X-ray Diffraction Imaging of Nanostructures*  
 3139 2013. arXiv: 1304.5335 [cond-mat.mes-hall]. <https://arxiv.org/abs/1304.5335>.
- 3140 42. Dupraz, M. *Coherent X-ray diffraction applied to metal physics* Theses (Université Grenoble  
 3141 Alpes, Nov. 2015). <https://theses.hal.science/tel-01285735>.
- 3142 43. Girard, G. *Development of Bragg coherent X-ray diffraction and Ptychography methods,*  
 3143 *application to the study of semiconductor nanostructures* Theses (Université Grenoble  
 3144 Alpes, Mar. 2020). <https://theses.hal.science/tel-02906931>.
- 3145 44. Atlan, C. *Sur les déformations de nanoparticules de Platine et de Palladium dans un*  
 3146 *environnement électrochimique* Dirigée par Marie-Ingrid Richard et Frédéric Maillard.  
 3147 Soutenue le 12 décembre 2023. Thèse de doctorat (Université Grenoble Alpes, 2023).  
 3148 <https://theses.fr/2023GRALY092>.
- 3149 45. Takagi, S. A Dynamical Theory of Diffraction for a Distorted Crystal. *Journal of the*  
 3150 *Physical Society of Japan* **26**, 1239–1253 (1969).

- 3151 46. Gorobtsov, O. Y. & Vartanyants, I. Phase of transmitted wave in dynamical theory and  
3152 quasi-kinematical approximation. *Physical Review B* **93**, 184107 (2016).
- 3153 47. Shabalin, A. G., Yefanov, O. M., Nosik, V. L., Bushuev, V. A. & Vartanyants, I. A. Dynamical  
3154 effects in Bragg coherent x-ray diffraction imaging of finite crystals. *Phys. Rev. B* **96**,  
3155 **064111**. <https://link.aps.org/doi/10.1103/PhysRevB.96.064111> (6 Aug. 2017).
- 3156 48. Ashcroft, N. W. & Mermin, N. D. *Solid State Physics* ISBN: 978-0-03-083993-1 (Holt,  
3157 Rinehart and Winston, New York, 1976).
- 3158 49. Ponchut, C. *et al.* MAXIPIX, a fast readout photon-counting X-ray area detector for syn-  
3159 chrotron applications in *Journal of Instrumentation* **6**. Issue: 1 ISSN: 17480221 (Jan. 2011).
- 3160 50. Friedrich, W., Knipping, P. & von Laue, M. Interferenz-Erscheinungen bei Röntgen-  
3161 strahlen. *Sitzungsberichte der Königlich Bayerischen Akademie der Wissenschaften, Mathematisch-*  
3162 *Physikalische Klasse*, 303–322 (1912).
- 3163 51. Bragg, W. L. The reflection of X-rays by crystals. *Proceedings of the Cambridge Philosoph-  
3164 ical Society* **17**, 43–57 (1913).
- 3165 52. Ewald, P. P. X-ray diffraction by finite and imperfect crystal lattices. *Proceedings of the  
3166 Physical Society* **52**, 167. <https://dx.doi.org/10.1088/0959-5309/52/1/323> (Jan. 1940).
- 3167 53. Robinson, I. K. Crystal truncation rods and surface roughness. *Physical Review B* **33**,  
3168 **3830–3836** (1986).
- 3169 54. Aubert, E. & Lecomte, C. Illustrated Fourier transforms for crystallography. *Journal of  
3170 Applied Crystallography* **40**, 1153–1165. <https://doi.org/10.1107/S0021889807043622> (Dec.  
3171 2007).
- 3172 55. Favre-Nicolin, V., Coraux, J., Richard, M.-I. & Renevier, H. Fast computation of scattering  
3173 maps of nanostructures using graphical processing units. *Journal of Applied Crystallography*  
3174 **44**, 635–640. <https://doi.org/10.1107/S0021889811009009> (June 2011).
- 3175 56. Godard, P. On the use of the scattering amplitude in coherent X-ray Bragg diffraction  
3176 imaging. *Journal of Applied Crystallography* **54**, 797–802. <https://doi.org/10.1107/S1600576721003113> (June 2021).
- 3178 57. Haag, S. T. *et al.* Anomalous coherent diffraction of core-shell nano-objects: A methodol-  
3179 ogy for determination of composition and strain fields. *Phys. Rev. B* **87**, 035408. <https://link.aps.org/doi/10.1103/PhysRevB.87.035408> (3 Jan. 2013).
- 3181 58. Kim, C., Scholz, M. & Madsen, A. The influence of strain on image reconstruction in  
3182 Bragg coherent X-ray diffraction imaging and ptychography. *Journal of Synchrotron  
3183 Radiation* **28**, 1159–1165. <https://doi.org/10.1107/S160057752100477X> (July 2021).
- 3184 59. Friedel, G. Sur les symétries cristallines que peut révéler la diffraction des rayons Röntgen.  
3185 *Comptes Rendus de l'Académie des Sciences, Paris* **157**, 1533–1536 (1913).
- 3186 60. Poisson, S.-D. *Recherches sur la probabilité des jugements en matière criminelle et en  
3187 matière civile* Contains the first appearance of what is now called the Poisson distribution  
3188 (Bachelier, Paris, 1837).
- 3189 61. Griffiths, D. J. *Introduction to Electrodynamics* 3rd ed. ISBN: 9780138053260 (Prentice Hall,  
3190 1999).

- 3191 62. Margaritondo, G. *Introduction to Synchrotron Radiation* English. Accessed: 27 August 2025,  
 3192 via National Library of Australia. ISBN: 0195045246. <https://nla.gov.au/nla.cat-vn1060674>  
 3193 (Oxford University Press, 1988).
- 3194 63. Shabalin, A. *Coherent X-ray diffraction studies of mesoscopic materials* (Jan. 2015).
- 3195 64. European Synchrotron Radiation Facility. *ID01 - Nano/Micro diffraction imaging* <https://www.esrf.fr/UsersAndScience/Experiments/XNP/ID01>. Accessed: 2025-08-27. ESRF -  
 3196 The European Synchrotron, 2025.
- 3197 65. Pelzer, K., Schwarz, N. & Harder, R. Removal of spurious data in Bragg coherent diffraction  
 3198 imaging: an algorithm for automated data preprocessing. *Journal of Applied Crystallography* **54**, 523–532. <https://doi.org/10.1107/S1600576721000819> (Apr. 2021).
- 3200 66. Leake, S. J., Newton, M. C., Harder, R. & Robinson, I. K. Longitudinal coherence function  
 3201 in X-ray imaging of crystals. *Optics Express* **17**, 15853–15859 (Aug. 2009).
- 3203 67. Sayre, D. Some implications of a theorem due to Shannon. *Acta Crystallographica* **5**,  
 3204 843–843. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1107/S0365110X52002276>. <https://onlinelibrary.wiley.com/doi/abs/10.1107/S0365110X52002276> (1952).
- 3206 68. Shannon, C. Communication in the Presence of Noise. *Proceedings of the IRE* **37**, 10–21  
 3207 (1949).
- 3208 69. Gerchberg, R. W. & Saxton, W. O. A practical algorithm for the determination of the  
 3209 phase from image and diffraction plane pictures. *Optik* **35**, 237–246 (1972).
- 3210 70. Bruck, Y. M. & Sodin, L. G. On the Ambiguity of the Image Reconstruction Problem.  
 3211 *Optics Communications* **30**, 304–308 (1979).
- 3212 71. Bates, R. H. T. Fourier phase problems are uniquely solvable in more than one dimension.  
 3213 I: Underlying theory. *Optik* **61**, 247–262 (1982).
- 3214 72. Fienup, J. R. Reconstruction of a Complex-Valued Object from the Modulus of its Fourier  
 3215 Transform Using a Support Constraint. *JOSA A* **4**, 118–123 (1987).
- 3216 73. Sayre, D. in *Direct Methods of Solving Crystal Structures* (ed Schenk, H.) 353–356 (Plenum  
 3217 Press, New York, 1991).
- 3218 74. Miao, J., Charalambous, P., Kirz, J. & Sayre, D. Extending the methodology of X-ray  
 3219 crystallography to allow imaging of micrometre-sized non-crystalline specimens. *Nature*  
 3220 **400**, 342–344 (1999).
- 3221 75. Miao, J., Kirz, J. & Sayre, D. The oversampling phasing method. *Acta Crystallographica*  
 3222 *Section D* **56**, 1312–1315. <https://doi.org/10.1107/S0907444900008970> (Oct. 2000).
- 3223 76. Elser, V. Phase retrieval by iterated projections. *JOSA A* **20**, 40–55 (2003).
- 3224 77. Bauschke, H. H., Combettes, P. L. & Luke, D. Finding best approximation pairs relative to  
 3225 two closed convex sets in Hilbert spaces. *Journal of Approximation Theory* **127**, 178–192.  
 3226 ISSN: 0021-9045. <https://www.sciencedirect.com/science/article/pii/S002190450400036X>  
 3227 (2004).
- 3228 78. Rodenburg, J. M. & Faulkner, H. M. L. A phase retrieval algorithm for shifting illumination.  
 3229 *Ultramicroscopy* **103**, 153–164 (2004).

- 3230 79. Godard, P. *et al.* Three-dimensional high-resolution quantitative microscopy of extended  
3231 crystals. *Nature Communications* **2**, 568. ISSN: 2041-1723. <https://doi.org/10.1038/ncomms1569> (2011).
- 3232 80. Candès, E. J., Strohmer, T. & Voroninski, V. PhaseLift: Exact and Stable Signal Recovery  
3233 from Magnitude Measurements via Convex Programming. *Communications on Pure and  
3234 Applied Mathematics* **66**, 1241–1274 (2013).
- 3235 81. Wirtinger, W. Zur formalen Theorie der Funktionen von mehr komplexen Veränderlichen.  
3236 *Mathematische Annalen* **97**, 357–375. ISSN: 1432-1807. <https://doi.org/10.1007/BF01447872>  
3237 (1927).
- 3238 82. Shechtman, Y. *et al.* Phase Retrieval with Application to Optical Imaging: A contemporary  
3239 overview. *IEEE Signal Processing Magazine* **32**, 87–109 (2015).
- 3240 83. Fannjiang, A. & Strohmer, T. The numerics of phase retrieval. *Acta Numerica* **29**, 125–228.  
3241 <https://api.semanticscholar.org/CorpusID:215744989> (2020).
- 3242 84. Marchesini, S. A unified evaluation of iterative projection algorithms for phase retrieval.  
3243 *Review of Scientific Instruments* **78**. ISSN: 00346748. arXiv: [physics/0603201](https://arxiv.org/abs/physics/0603201) (2007).
- 3244 85. Patterson, A. L. The Diffraction of X-Rays by Small Crystalline Particles. *Phys. Rev.* **56**,  
3245 972–977. <https://link.aps.org/doi/10.1103/PhysRev.56.972> (10 Nov. 1939).
- 3246 86. Latychevskaia, T. Iterative phase retrieval in coherent diffractive imaging: practical issues.  
3247 *Appl. Opt.* **57**, 7187–7197. <https://opg.optica.org/ao/abstract.cfm?URI=ao-57-25-7187>  
3248 (Sept. 2018).
- 3249 87. Van der Veen, F. & Pfeiffer, F. Coherent x-ray scattering. *Journal of Physics: Condensed  
3250 Matter* **16**, 5003. <https://dx.doi.org/10.1088/0953-8984/16/28/020> (July 2004).
- 3251 88. Öztürk, H. *et al.* Performance evaluation of Bragg coherent diffraction imaging. *New  
3252 Journal of Physics* **19**, 103001. <https://dx.doi.org/10.1088/1367-2630/aa83a9> (Oct. 2017).
- 3253 89. Millane, R. P. Phase retrieval in crystallography and optics. *Journal of the Optical Society  
3254 of America A* **7**, 394–411 (1990).
- 3255 90. Guizar-Sicairos, M. & Fienup, J. R. *Understanding the twin-image problem in phase retrieval*  
3256 (2012).
- 3257 91. Gaur, C., Mohan, B. & Khare, K. Sparsity-assisted solution to the twin image problem in  
3258 phase retrieval. *J. Opt. Soc. Am. A* **32**, 1922–1927. <https://opg.optica.org/josaa/abstract.cfm?URI=josaa-32-11-1922> (Nov. 2015).
- 3259 92. Cegielski, A. *Iterative Methods for Fixed Point Problems in Hilbert Spaces* ISBN: 978-3-642-  
3260 30900-7 (Jan. 2012).
- 3261 93. Luke, D. R., Burke, J. V. & Lyon, R. G. Optical Wavefront Reconstruction: Theory and  
3262 Numerical Methods. *SIAM Review* **44**, 169–224. eprint: <https://doi.org/10.1137/S003614450139075> (2002).  
3263 94. Marchesini, S. *et al.* X-ray image reconstruction from a diffraction pattern alone. *Phys.  
3264 Rev. B* **68**, 140101. <https://link.aps.org/doi/10.1103/PhysRevB.68.140101> (14 Oct. 2003).
- 3265

- 3268 95. Cha, W. *et al.* Exploration of crystal strains using coherent x-ray diffraction. *New Journal*  
3269 *of Physics* **12**, 035022. <https://dx.doi.org/10.1088/1367-2630/12/3/035022> (Mar. 2010).
- 3270 96. Newton, M. C., Harder, R., Huang, X., Xiong, G. & Robinson, I. K. Phase retrieval of  
3271 diffraction from highly strained crystals. *Phys. Rev. B* **82**, 165436. <https://link.aps.org/doi/10.1103/PhysRevB.82.165436> (16 Oct. 2010).
- 3273 97. Pavlov, K. M., Punegov, V. I., Morgan, K. S., Schmalz, G. & Paganin, D. M. Deterministic  
3274 Bragg Coherent Diffraction Imaging. *Scientific Reports* **7**, 1132. ISSN: 2045-2322. <https://doi.org/10.1038/s41598-017-01164-x> (Apr. 2017).
- 3276 98. Calvo-Almazán, I., Chamard, V., Grünewald, T. A. & Allain, M. Inhomogeneous probes  
3277 for Bragg coherent diffraction imaging: Toward the imaging of dynamic and distorted  
3278 crystals. *Phys. Rev. B* **110**, 134117. <https://link.aps.org/doi/10.1103/PhysRevB.110.134117>  
3279 (13 Oct. 2024).
- 3280 99. Zhao, J., Vartanyants, I. A. & Zhang, F. Bragg coherent modulation imaging for highly  
3281 strained nanocrystals: a numerical study. *Journal of Applied Crystallography* **56**, 1528–  
3282 1536. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1107/S1600576723007720>. <https://onlinelibrary.wiley.com/doi/abs/10.1107/S1600576723007720> (2023).
- 3284 100. Minkevich, A. A. *et al.* Inversion of the diffraction pattern from an inhomogeneously  
3285 strained crystal using an iterative algorithm. *Phys. Rev. B* **76**, 104106. <https://link.aps.org/doi/10.1103/PhysRevB.76.104106> (10 Sept. 2007).
- 3287 101. Minkevich, A. A., Baumbach, T., Gailhanou, M. & Thomas, O. Applicability of an iterative  
3288 inversion algorithm to the diffraction patterns from inhomogeneously strained crystals.  
3289 *Phys. Rev. B* **78**, 174110. <https://link.aps.org/doi/10.1103/PhysRevB.78.174110> (17 Nov.  
3290 2008).
- 3291 102. Ulvestad, A. *et al.* Identifying Defects with Guided Algorithms in Bragg Coherent Diffrac-  
3292 tive Imaging. *Scientific Reports* **7**, 9920 (Aug. 2017).
- 3293 103. Wilkin, M. J. *et al.* Experimental demonstration of coupled multi-peak Bragg coherent  
3294 diffraction imaging with genetic algorithms. *Phys. Rev. B* **103**, 214103. <https://link.aps.org/doi/10.1103/PhysRevB.103.214103> (21 June 2021).
- 3296 104. Gao, Y., Huang, X., Yan, H. & Williams, G. J. Bragg coherent diffraction imaging by  
3297 simultaneous reconstruction of multiple diffraction peaks. *Phys. Rev. B* **103**, 014102.  
3298 <https://link.aps.org/doi/10.1103/PhysRevB.103.014102> (1 Jan. 2021).
- 3299 105. Wang, Z., Gorobtsov, O. & Singer, A. An algorithm for Bragg coherent x-ray diffractive  
3300 imaging of highly strained nanocrystals. *New Journal of Physics* **22**, 013021. <https://dx.doi.org/10.1088/1367-2630/ab61db> (Jan. 2020).
- 3302 106. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning* <http://www.deeplearningbook.org> (MIT Press, 2016).
- 3304 107. Prince, S. J. *Understanding Deep Learning* <http://udlbook.com> (The MIT Press, 2023).
- 3305 108. Jagtap, A. D. & Karniadakis, G. E. *How important are activation functions in regression*  
3306 *and classification? A survey, performance comparison, and future directions* 2022. arXiv:  
3307 2209.02681 [cs.LG]. <https://arxiv.org/abs/2209.02681>.

- 3308 109. Kunc, V. & Kléma, J. *Three Decades of Activations: A Comprehensive Survey of 400 Activation*  
3309 *Functions for Neural Networks* 2024. arXiv: 2402.09092 [cs.LG]. <https://arxiv.org/abs/2402.09092>.
- 3311 110. Maas, A. L. *Rectifier Nonlinearities Improve Neural Network Acoustic Models* in (2013).  
3312 <https://api.semanticscholar.org/CorpusID:16489696>.
- 3313 111. Haykin, S. S. *Neural networks and learning machines* Third (Pearson Education, Upper  
3314 Saddle River, NJ, 2009).
- 3315 112. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Mathematics*  
3316 *of Control, Signals and Systems* 2, 303–314. ISSN: 1435-568X. <https://doi.org/10.1007/BF02551274> (1989).
- 3318 113. Hornik, K., Stinchcombe, M. & White, H. Multilayer feedforward networks are universal  
3319 approximators. *Neural Networks* 2, 359–366. ISSN: 0893-6080. <https://www.sciencedirect.com/science/article/pii/0893608089900208> (1989).
- 3321 114. Leshno, M., Lin, V. Y., Pinkus, A. & Schocken, S. Multilayer feedforward networks with  
3322 a nonpolynomial activation function can approximate any function. *Neural Networks*  
3323 6, 861–867. ISSN: 0893-6080. <https://www.sciencedirect.com/science/article/pii/S0893608005801315> (1993).
- 3325 115. Zhou, D.-X. Universality of deep convolutional neural networks. *Applied and Computational*  
3326 *Harmonic Analysis* 48, 787–794. ISSN: 1063-5203. <https://www.sciencedirect.com/science/article/pii/S1063520318302045> (2020).
- 3328 116. Hornik, K., Stinchcombe, M. & White, H. Universal approximation of an unknown  
3329 mapping and its derivatives using multilayer feedforward networks. *Neural Networks*  
3330 3, 551–560. ISSN: 0893-6080. <https://www.sciencedirect.com/science/article/pii/0893608090900056> (1990).
- 3332 117. Zhao, S.-Y., Xie, Y.-P. & Li, W.-J. On the convergence and improvement of stochastic  
3333 normalized gradient descent. *Science China Information Sciences* 64, 132103. ISSN: 1869-  
3334 1919. <https://doi.org/10.1007/s11432-020-3023-7> (Feb. 2021).
- 3335 118. Polyak, B. T. Some methods of speeding up the convergence of iteration methods. *USSR*  
3336 *Computational Mathematics and Mathematical Physics* 4, 1–17 (1964).
- 3337 119. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning representations by back-  
3338 propagating errors. *Nature* 323, 533–536. ISSN: 1476-4687. <https://doi.org/10.1038/323533a0> (1986).
- 3340 120. Duchi, J., Hazan, E. & Singer, Y. Adaptive Subgradient Methods for Online Learning and  
3341 Stochastic Optimization. *J. Mach. Learn. Res.* 12, 2121–2159. ISSN: 1532-4435 (July 2011).
- 3342 121. Kingma, D. P. & Ba, J. *Adam: A Method for Stochastic Optimization* 2017. arXiv: 1412.6980  
3343 [cs.LG]. <https://arxiv.org/abs/1412.6980>.
- 3344 122. Yao, X. Evolving artificial neural networks. *Proceedings of the IEEE* 87, 1423–1447 (1999).

- 3345 123. Stoudenmire, E. & Schwab, D. J. *Supervised Learning with Tensor Networks* in *Advances in*  
 3346 *Neural Information Processing Systems* (eds Lee, D., Sugiyama, M., Luxburg, U., Guyon, I.  
 3347 & Garnett, R.) **29** (Curran Associates, Inc., 2016). [https://proceedings.neurips.cc/paper\\_files/paper/2016/file/5314b9674c86e3f9d1ba25ef9bb32895-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2016/file/5314b9674c86e3f9d1ba25ef9bb32895-Paper.pdf).
- 3349 124. Paszke, A. *et al.* *PyTorch: An Imperative Style, High-Performance Deep Learning Library*  
 3350 2019. arXiv: **1912.01703** [cs.LG]. <https://arxiv.org/abs/1912.01703>.
- 3351 125. Abadi, M. *et al.* *TensorFlow: A system for large-scale machine learning* 2016. arXiv: **1605.08695**  
 3352 [cs.DC]. <https://arxiv.org/abs/1605.08695>.
- 3353 126. Griewank, A. & Walther, A. *Evaluating derivatives - principles and techniques of algorithmic*  
 3354 *differentiation, Second Edition* in *Frontiers in applied mathematics* (2000). <https://api.semanticscholar.org/CorpusID:35253217>.
- 3356 127. Baydin, A. G., Pearlmutter, B. A., Radul, A. A. & Siskind, J. M. *Automatic differentiation*  
 3357 *in machine learning: a survey* 2018. arXiv: **1502.05767** [cs.SC]. <https://arxiv.org/abs/1502.05767>.
- 3359 128. Fukushima, K. & Miyake, S. *Neocognitron: A Self-Organizing Neural Network Model for a*  
 3360 *Mechanism of Visual Pattern Recognition* in *Competition and Cooperation in Neural Nets*  
 3361 (eds Amari, S.-i. & Arbib, M. A.) (Springer Berlin Heidelberg, Berlin, Heidelberg, 1982),  
 3362 267–285. ISBN: 978-3-642-46466-9.
- 3363 129. LeCun, Y. *et al.* *Handwritten Digit Recognition with a Back-Propagation Network* in *Neural*  
 3364 *Information Processing Systems* (1989). <https://api.semanticscholar.org/CorpusID:2542741>.
- 3365 130. LeCun, Y. *et al.* Backpropagation Applied to Handwritten Zip Code Recognition. *Neural*  
 3366 *Computation* **1**, 541–551. <https://api.semanticscholar.org/CorpusID:41312633> (1989).
- 3367 131. Ongie, G. *et al.* Deep Learning Techniques for Inverse Problems in Imaging. *IEEE Journal*  
 3368 *on Selected Areas in Information Theory* **1**, 39–56 (2020).
- 3369 132. Jin, K. H., McCann, M. T., Froustey, E. & Unser, M. Deep Convolutional Neural Network  
 3370 for Inverse Problems in Imaging. *IEEE Transactions on Image Processing* **26**, 4509–4522  
 3371 (2017).
- 3372 133. Ulyanov, D., Vedaldi, A. & Lempitsky, V. Deep Image Prior. *International Journal of*  
 3373 *Computer Vision* **128**, 1867–1888. ISSN: 1573-1405. <http://dx.doi.org/10.1007/s11263-020-01303-4> (Mar. 2020).
- 3375 134. LeCun, Y., Bottou, L., Bengio, Y. & Haffner, P. Gradient-based learning applied to docu-  
 3376 ment recognition. *Proceedings of the IEEE* **86**, 2278–2324 (1998).
- 3377 135. Krizhevsky, A., Sutskever, I. & Hinton, G. E. *ImageNet classification with deep convolutional*  
 3378 *neural networks* in *Advances in Neural Information Processing Systems (NeurIPS)* (2012),  
 3379 1097–1105.
- 3380 136. Hinton, G. E. & Salakhutdinov, R. R. Reducing the Dimensionality of Data with Neural  
 3381 Networks. *Science* **313**, 504–507. eprint: <https://www.science.org/doi/pdf/10.1126/science.1127647>. <https://www.science.org/doi/abs/10.1126/science.1127647> (2006).
- 3383 137. Goodfellow, I. J. *et al.* *Generative Adversarial Networks* 2014. arXiv: **1406.2661** [stat.ML].  
 3384 <https://arxiv.org/abs/1406.2661>.

- 3385 138. Dosovitskiy, A. *et al.* *An Image is Worth 16x16 Words: Transformers for Image Recognition*  
3386 *at Scale* 2021. arXiv: 2010.11929 [cs.CV]. <https://arxiv.org/abs/2010.11929>.
- 3387 139. He, K., Zhang, X., Ren, S. & Sun, J. *Deep Residual Learning for Image Recognition* 2015.  
3388 arXiv: 1512.03385 [cs.CV]. <https://arxiv.org/abs/1512.03385>.
- 3389 140. Ronneberger, O., Fischer, P. & Brox, T. U-Net: Convolutional Networks for Biomedical  
3390 Image Segmentation. arXiv: 1505.04597. <http://arxiv.org/abs/1505.04597> (May 18, 2015).
- 3391 141. Balduzzi, D. *et al.* *The Shattered Gradients Problem: If resnets are the answer, then what is*  
3392 *the question?* 2018. arXiv: 1702.08591 [cs.NE]. <https://arxiv.org/abs/1702.08591>.
- 3393 142. Li, H., Xu, Z., Taylor, G., Studer, C. & Goldstein, T. Visualizing the Loss Landscape of  
3394 Neural Nets. arXiv: 1712.09913. <http://arxiv.org/abs/1712.09913> (Dec. 28, 2017).
- 3395 143. Johnson, I. *et al.* Eiger: a single-photon counting x-ray detector. *Journal of Instrumentation*  
3396 **9**, C05032. <https://dx.doi.org/10.1088/1748-0221/9/05/C05032> (May 2014).
- 3397 144. Carnis, J. *et al.* Towards a quantitative determination of strain in Bragg Coherent X-ray  
3398 Diffraction Imaging: artefacts and sign convention in reconstructions. *Scientific Reports*  
3399 **9**. Publisher: Nature Research. ISSN: 20452322 (Dec. 1, 2019).
- 3400 145. He, K., Sharma, M. K. & Cossairt, O. High dynamic range coherent imaging using  
3401 compressed sensing. *Opt. Express* **23**, 30904–30916. <https://opg.optica.org/oe/abstract.cfm?URI=oe-23-24-30904> (Nov. 2015).
- 3403 146. Malm, E. & Chushkin, Y. Reduction of artifacts associated with missing data in coherent  
3404 diffractive imaging. *Journal of Synchrotron Radiation* **32**, 210–216. <https://doi.org/10.1107/S1600577524010956> (Jan. 2025).
- 3406 147. Elharrouss, O., Almaadeed, N., Al-Maadeed, S. & Akbari, Y. Image Inpainting: A Review.  
3407 *Neural Processing Letters* **51**, 2007–2028. ISSN: 1573-773X. <http://dx.doi.org/10.1007/s11063-019-10163-0> (Dec. 2019).
- 3409 148. Jam, J. *et al.* A comprehensive review of past and present image inpainting methods.  
3410 *Computer Vision and Image Understanding* **203**, 103147. ISSN: 1077-3142. <https://www.sciencedirect.com/science/article/pii/S1077314220301661> (2021).
- 3412 149. Efros, A. & Leung, T. *Texture synthesis by non-parametric sampling* in *Proceedings of the*  
3413 *Seventh IEEE International Conference on Computer Vision* **2** (1999), 1033–1038 vol.2.
- 3414 150. Bertalmio, M., Bertozzi, A. & Sapiro, G. *Navier-stokes, fluid dynamics, and image and*  
3415 *video inpainting* in *Proceedings of the 2001 IEEE Computer Society Conference on Computer*  
3416 *Vision and Pattern Recognition. CVPR 2001* **1** (2001), I–I.
- 3417 151. Mairal, J., Elad, M. & Sapiro, G. Sparse Representation for Color Image Restoration. *IEEE*  
3418 *Transactions on Image Processing* **17**, 53–69 (2008).
- 3419 152. Allene, C. & Paragios, N. *Image Renaissance Using Discrete Optimization* in *18th Interna-*  
3420 *tional Conference on Pattern Recognition (ICPR'06)* **3** (2006), 631–634.
- 3421 153. Jiang, Y., Xu, J., Yang, B., Xu, J. & Zhu, J. Image Inpainting Based on Generative Adversarial  
3422 Networks. *IEEE Access* **8**, 22884–22892 (2020).

- 3423 154. Xiang, H. *et al.* Deep learning for image inpainting: A survey. *Pattern Recognition*  
 3424 **134**, 109046. ISSN: 0031-3203. <https://www.sciencedirect.com/science/article/pii/S003132032200526X> (2023).
- 3426 155. Xu, Z. *et al.* A Review of Image Inpainting Methods Based on Deep Learning. *Applied  
 3427 Sciences* **13**. ISSN: 2076-3417. <https://www.mdpi.com/2076-3417/13/20/11189> (2023).
- 3428 156. Sogancioglu, E., Hu, S., Belli, D. & van Ginneken, B. *Chest X-ray Inpainting with Deep  
 3429 Generative Models* 2018. arXiv: [1809.01471 \[cs.GR\]](https://arxiv.org/abs/1809.01471). <https://arxiv.org/abs/1809.01471>.
- 3430 157. MA, B. *et al.* Deep learning-based automatic inpainting for material microscopic images.  
 3431 *Journal of Microscopy* **281**, 177–189. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/jmi.12960>. <https://onlinelibrary.wiley.com/doi/abs/10.1111/jmi.12960> (2021).
- 3433 158. Chavez, T., Roberts, E. J., Zwart, P. H. & Hexemer, A. A comparison of deep-learning-based  
 3434 inpainting techniques for experimental X-ray scattering. *Journal of Applied Crystallography* **55**. Publisher: International Union of Crystallography (IUCr), 1277–1288. ISSN:  
 3435 1600-5767. <https://scripts.iucr.org/cgi-bin/paper?S1600576722007105> (Oct. 1, 2022).
- 3437 159. Bellisario, A., Maia, F. R. & Ekeberg, T. Noise reduction and mask removal neural network  
 3438 for X-ray single-particle imaging. *Journal of Applied Crystallography* **55**. Publisher: International Union of Crystallography, 122–132. ISSN: 16005767 (Feb. 1, 2022).
- 3440 160. Garcia, N. & Stoll, E. Monte Carlo Calculation for Electromagnetic-Wave Scattering from  
 3441 Random Rough Surfaces. *Phys. Rev. Lett.* **52**, 1798–1801. <https://link.aps.org/doi/10.1103/PhysRevLett.52.1798> (20 May 1984).
- 3443 161. Lecun, Y., Bottou, L., Orr, G. & Müller, K.-R. Efficient BackProp (Aug. 2000).
- 3444 162. Wang, Z., Bovik, A., Sheikh, H. & Simoncelli, E. Image quality assessment: from error  
 3445 visibility to structural similarity. *IEEE Transactions on Image Processing* **13**, 600–612  
 3446 (2004).
- 3447 163. Pelt, D. M. & Sethian, J. A. A mixed-scale dense convolutional neural network for image  
 3448 analysis. *Proceedings of the National Academy of Sciences of the United States of America*  
 3449 **115**, 254–259. ISSN: 10916490 (2 Jan. 2017).
- 3450 164. Yu, F. & Koltun, V. *Multi-Scale Context Aggregation by Dilated Convolutions* 2016. arXiv:  
 3451 [1511.07122 \[cs.CV\]](https://arxiv.org/abs/1511.07122). <https://arxiv.org/abs/1511.07122>.
- 3452 165. Blog, N. D. *Optimizing GPU Performance with Tensor Cores* Accessed: 2025-03-25. 2020.  
 3453 <https://developer.nvidia.com/blog/optimizing-gpu-performance-tensor-cores/>.
- 3454 166. Lim, B. *et al.* A convolutional neural network for defect classification in Bragg coherent X-  
 3455 ray diffraction. *npj Computational Materials* **7**. Publisher: Nature Research. ISSN: 20573960.  
 3456 arXiv: [2106.16179](https://arxiv.org/abs/2106.16179) (Dec. 1, 2021).
- 3457 167. Thompson, A. P. *et al.* LAMMPS – a flexible simulation tool for particle-based materials  
 3458 modeling at the atomic, meso, and continuum scales. *Computer Physics Communications*  
 3459 **271**, 108171 (2022).
- 3460 168. Krull, A., Buchholz, T.-O. & Jug, F. *Noise2Void - Learning Denoising from Single Noisy  
 3461 Images* 2019. arXiv: [1811.10980 \[cs.CV\]](https://arxiv.org/abs/1811.10980). <https://arxiv.org/abs/1811.10980>.

- 3462 169. Ulvestad, A. *et al.* ARTICLE Avalanching strain dynamics during the hydriding phase  
3463 transformation in individual palladium nanoparticles. [www.nature.com/naturecommunications](http://www.nature.com/naturecommunications)  
3464 (2015).
- 3465 170. Kim, D. *et al.* Active site localization of methane oxidation on Pt nanocrystals. *Nature  
3466 Communications* **9**, 3422. ISSN: 2041-1723. <https://doi.org/10.1038/s41467-018-05464-2>  
3467 (2018).
- 3468 171. Kawaguchi, T. *et al.* Gas-Induced Segregation in Pt-Rh Alloy Nanoparticles Observed  
3469 by In Situ Bragg Coherent Diffraction Imaging. *Phys. Rev. Lett.* **123**, 246001. <https://link.aps.org/doi/10.1103/PhysRevLett.123.246001> (24 Dec. 2019).
- 3471 172. Abuin, M. *et al.* Coherent X-ray Imaging of CO-Adsorption-Induced Structural Changes  
3472 in Pt Nanoparticles: Implications for Catalysis. *ACS Applied Nano Materials* **2**, 4818–4824.  
3473 <https://doi.org/10.1021/acsanm.9b00764> (2019).
- 3474 173. Dupraz, M. *et al.* Imaging the facet surface strain state of supported multi-faceted Pt  
3475 nanoparticles during reaction. *Nature communications* **3003**. <https://doi.org/10.1038/s41467-022-30592-1> (May 2022).
- 3477 174. Masto, M. *et al.* Phase Retrieval of Highly Strained Bragg Coherent Diffraction Patterns using  
3478 Supervised Convolutional Neural Network 2025. arXiv: [2507.06644](https://arxiv.org/abs/2507.06644) [cond-mat.mtrl-sci].  
3479 <https://arxiv.org/abs/2507.06644>.
- 3480 175. Sinha, A., Lee, J., Li, S. & Barbastathis, G. Lensless computational imaging through deep  
3481 learning. *Optica* **4**, 1117–1125. <https://opg.optica.org/optica/abstract.cfm?URI=optica-4-9-1117> (Sept. 2017).
- 3483 176. Rivenson, Y., Zhang, Y., Günaydin, H., Teng, D. & Ozcan, A. Phase recovery and holo-  
3484 graphic image reconstruction using deep learning in neural networks. *Light: Science &  
3485 Applications* **7**, 17141. ISSN: 2047-7538. <https://doi.org/10.1038/lsa.2017.141> (2018).
- 3486 177. Cherukara, M. J., Nashed, Y. S. & Harder, R. J. Real-time coherent diffraction inversion  
3487 using deep generative networks. *Scientific Reports* **8**. Publisher: Nature Publishing Group.  
3488 ISSN: 20452322 (Dec. 1, 2018).
- 3489 178. Scheinker, A. & Pokharel, R. Adaptive 3D convolutional neural network-based recon-  
3490 struction method for 3D coherent diffraction imaging. *Journal of Applied Physics* **128**.  
3491 Publisher: American Institute of Physics Inc. ISSN: 10897550. arXiv: [2008.10094](https://arxiv.org/abs/2008.10094) (Nov. 14,  
3492 2020).
- 3493 179. Wu, L., Juhas, P., Yoo, S. & Robinson, I. Complex imaging of phase domains by deep  
3494 neural networks. *IUCrJ* **8**, 12–21. ISSN: 20522525 (2021).
- 3495 180. Chan, H. *et al.* Rapid 3D nanoscale coherent imaging via physics-aware deep learning.  
3496 *Applied Physics Reviews* **8**. Publisher: American Institute of Physics Inc. ISSN: 19319401.  
3497 arXiv: [2006.09441](https://arxiv.org/abs/2006.09441) (June 1, 2021).
- 3498 181. Wu, L. *et al.* Three-dimensional coherent X-ray diffraction imaging via deep convolu-  
3499 tional neural networks. *npj Computational Materials* **7**. Publisher: Nature Research. ISSN:  
3500 20573960. arXiv: [2103.00001](https://arxiv.org/abs/2103.00001) (Dec. 1, 2021).

- 3501 182. Yao, Y. *et al.* AutoPhaseNN: unsupervised physics-aware deep learning of 3D nanoscale  
 3502 Bragg coherent diffraction imaging. *npj Computational Materials* **8**. Publisher: Nature  
 3503 Research. ISSN: 20573960. arXiv: [2109.14053](https://arxiv.org/abs/2109.14053) (Dec. 1, 2022).
- 3504 183. Zhuang, Z., Yang, D., Hofmann, F., Barmherzig, D. A. & Sun, J. Practical Phase Retrieval  
 3505 Using Double Deep Image Priors. *ArXiv* **abs/2211.00799**. <https://api.semanticscholar.org/CorpusID:253255048> (2022).
- 3507 184. Tayal, K., Lai, C.-H., Kumar, V. & Sun, J. *Inverse Problems, Deep Learning, and Symmetry  
 3508 Breaking* 2020. arXiv: [2003.09077 \[cs.LG\]](https://arxiv.org/abs/2003.09077). <https://arxiv.org/abs/2003.09077>.
- 3509 185. Yu, X. *et al.* Ultrafast Bragg coherent diffraction imaging of epitaxial thin films using deep  
 3510 complex-valued neural networks. *npj Computational Materials* **10**. Publisher: Nature  
 3511 Research. ISSN: 20573960 (Dec. 1, 2024).
- 3512 186. Diaz, A. *et al.* Imaging the displacement field within epitaxial nanostructures by coherent  
 3513 diffraction: a feasibility study. *New Journal of Physics* **12**, 035006. <https://dx.doi.org/10.1088/1367-2630/12/3/035006> (Mar. 2010).
- 3515 187. Zhang, W., Wan, Y., Zhuang, Z. & Sun, J. What is Wrong with End-to-End Learning for  
 3516 Phase Retrieval? arXiv: [2403.15448](https://arxiv.org/abs/2403.15448). [http://arxiv.org/abs/2403.15448](https://arxiv.org/abs/2403.15448) (Mar. 17, 2024).
- 3517 188. Miao, J., Sayre, D. & Chapman, H. N. Phase retrieval from the magnitude of the Fourier  
 3518 transforms of nonperiodic objects. *J. Opt. Soc. Am. A* **15**, 1662–1669. <https://opg.optica.org/josaa/abstract.cfm?URI=josaa-15-6-1662> (June 1998).
- 3520 189. Shi, X. *et al.* *Convolutional LSTM Network: A Machine Learning Approach for Precipitation  
 3521 Nowcasting* 2015. arXiv: [1506.04214 \[cs.CV\]](https://arxiv.org/abs/1506.04214). <https://arxiv.org/abs/1506.04214>.
- 3522 190. Frisch, M. L. *et al.* Unraveling the synergistic effects of Cu–Ag tandem catalysts during  
 3523 electrochemical CO<sub>2</sub> reduction using nanofocused X-ray probes. *Nature Communications*  
 3524 **14**. Published: 2023-11-29, 7833. ISSN: 2041-1723. <https://doi.org/10.1038/s41467-023-43693-2> (2023).
- 3526 191. Grimes, M. *et al.* Capturing Catalyst Strain Dynamics during Operando CO Oxidation.  
 3527 *ACS Nano* **18**. Published: 2024-07-30, 19608–19617. ISSN: 1936-0851. <https://doi.org/10.1021/acsnano.4c04127> (2024).
- 3529 192. Favre-Nicolin, V., Leake, S. & Chushkin, Y. Free log-likelihood as an unbiased metric  
 3530 for coherent diffraction imaging. *Scientific Reports* **10**. Publisher: Nature Research. ISSN:  
 3531 20452322. arXiv: [1904.07056](https://arxiv.org/abs/1904.07056) (Dec. 1, 2020).
- 3532 193. Ulvestad, A. *et al.* Three-dimensional imaging of dislocation dynamics during the hy-  
 3533 driding phase transformation. *Nature Materials* **16**, 565–571. ISSN: 1476-4660. <https://doi.org/10.1038/nmat4842> (2017).
- 3535 194. Jurling, A. & Fienup, J. Applications of algorithmic differentiation to phase retrieval  
 3536 algorithms. *Journal of the Optical Society of America A* **31**, 1348–1359 (June 2014).
- 3537 195. Nashed, Y. S., Peterka, T., Deng, J. & Jacobsen, C. Distributed Automatic Differentiation  
 3538 for Ptychography. *Procedia Computer Science* **108**. International Conference on Compu-  
 3539 tational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland, 404–414. ISSN: 1877-0509.  
 3540 <https://www.sciencedirect.com/science/article/pii/S1877050917306531> (2017).

- 3541 196. Kandel, S. *et al.* Using automatic differentiation as a general framework for ptychographic  
3542 reconstruction. *Optics Express* **27**, 18653–18672. ISSN: 1094-4087 (2019).
- 3543 197. Maddali, S. *et al.* Concurrent multi-peak Bragg coherent x-ray diffraction imaging of 3D  
3544 nanocrystal lattice displacement via global optimization. *npj Computational Materials* **9**,  
3545 77. ISSN: 2057-3960. <https://doi.org/10.1038/s41524-023-01022-7> (2023).
- 3546 198. Zhou, T. *et al.* Extending the Takagi-Taupin equations for x-ray nanobeam Bragg coherent  
3547 diffraction. *Phys. Rev. B* **110**, 054116. <https://link.aps.org/doi/10.1103/PhysRevB.110.054116> (5 Aug. 2024).
- 3549 199. Wu, L. *et al.* *Unveiling Nano-scale Crystal Deformation using Coherent X-ray Dynamical*  
3550 *Diffraction* 2025. arXiv: 2506.15939 [physics.app-ph]. <https://arxiv.org/abs/2506.15939>.
- 3552 200. Favre-Nicolin, V. *et al.* Analysis of strain and stacking faults in single nanowires using  
3553 Bragg coherent diffraction imaging. *New Journal of Physics* **12**. Publisher: Institute of  
3554 Physics Publishing. ISSN: 13672630 (Mar. 31, 2010).
- 3555 201. Rudin, L. I., Osher, S. & Fatemi, E. Nonlinear total variation based noise removal algo-  
3556 rithms. *Physica D: Nonlinear Phenomena* **60**, 259–268. ISSN: 0167-2789. <https://www.sciencedirect.com/science/article/pii/016727899290242F> (1992).
- 3558 202. Golub, G. H. & Van Loan, C. F. *Matrix Computations* 3rd. ISBN: 0-8018-5414-8 (Johns  
3559 Hopkins University Press, Baltimore, MD, 1996).
- 3560 203. Kolda, T. G. & Bader, B. W. Tensor Decompositions and Applications. *SIAM Review* **51**,  
3561 455–500. eprint: <https://doi.org/10.1137/07070111X>. <https://doi.org/10.1137/07070111X>  
3562 (2009).
- 3563 204. Oseledets, I. V. Tensor-Train Decomposition. *SIAM Journal on Scientific Computing* **33**,  
3564 2295–2317. eprint: <https://doi.org/10.1137/090752286>. <https://doi.org/10.1137/090752286>  
3565 (2011).
- 3566 205. Thibault, P. & Guizar-Sicairos, M. Maximum-likelihood refinement for coherent diffrac-  
3567 tive imaging. *New Journal of Physics* **14**, 063004. <https://dx.doi.org/10.1088/1367-2630/14/6/063004> (June 2012).
- 3569 206. Bellec, E. *et al.* *Ultra-fast BCDI during acetylene hydrogenation of Pd nanoparticles* Dataset  
3570 (European Synchrotron Radiation Facility, 2026). <https://doi.org/10.15151/ESRF-ES-1340529140>.
- 3572 207. Pinheiro, P. H. O. & Collobert, R. *Recurrent Convolutional Neural Networks for Scene Label-*  
3573 *ing in International Conference on Machine Learning* (2014). <https://api.semanticscholar.org/CorpusID:6613124>.
- 3575 208. Stierle, R. *et al.* Classical density functional theory in three dimensions with GPU-  
3576 accelerated automatic differentiation: Computational performance analysis using the  
3577 example of adsorption in covalent-organic frameworks. *Chemical Engineering Science*  
3578 **298**, 120380. ISSN: 0009-2509. <https://www.sciencedirect.com/science/article/pii/S0009250924006808> (2024).

- 3580 209. Zubair, M. *et al.* *Efficient GPU Implementation of Automatic Differentiation for Compu-*  
3581 *tational Fluid Dynamics* in *2023 IEEE 30th International Conference on High Performance*  
3582 *Computing, Data, and Analytics (HiPC)* (2023), 377–386.