MACHINE LEARNING
Science and Technology

PURPOSE-LED PUBLISHING™

PAPER • OPEN ACCESS

Deep learning phase retrieval in x-ray single-particle imaging for biological macromolecules

To cite this article: Alfredo Bellisario and Tomas Ekeberg 2024 *Mach. Learn.: Sci. Technol.* **5** 045022

IOP Publishing Mach. Learn.: Sci. Technol. 5 (2024) 045022 https://doi.org/10.1088/2632-2153/ad7f22

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
24 April 2024

REVISED
21 August 2024

ACCEPTED FOR PUBLICATION
24 September 2024

PUBLISHED
24 October 2024

PAPER

Deep learning phase retrieval in x-ray single-particle imaging for biological macromolecules

Alfredo Bellisario and Tomas Ekeberg*

Laboratory of Molecular Biophysics, Department of Cell and Molecular Biology, Uppsala University, Husargatan 3 (Box 596), SE-751 24 Uppsala, Sweden

* Author to whom any correspondence should be addressed.

E-mail: tomas.ekeberg@icm.uu.se

Keywords: Convolutional neural network, single particle imaging, coherent diffractive imaging, phase retrieval, flash x-ray imaging

Abstract

Phase retrieval is an important optimization problem that occurs, for example, in the analysis of coherent diffraction patterns from isolated proteins. All iterative algorithms employed for phase retrieval in this context require some *a priori* knowledge of the object, usually in the form of a support that describes the extent of the particle. Phase retrieval is a time-consuming task that can often fail, particularly if the support is too loose or of bad quality. This paper presents a neural network that can produce low-resolution estimates of the phased object in a fraction of the time it takes for a full phase retrieval. It can also successfully be used as support for further analysis. Our network is trained on simulated data from biological macromolecules and is thus tailored to the type of data seen in a typical CDI experiment. Other approaches to support finding require very accurate data without missing regions or the full phase-retrieval algorithm to be run for a long time. Our network could speed up offline analysis and provide real-time feedback during data collection.

1. Introduction

Coherent x-ray diffraction imaging (CDI) enables high-resolution structure determination for nanoscopic objects [1]. However, the strength of the scattering signal depends on the sample's cross-section, which means that imaging small objects, such as a protein, requires highly brilliant sources. The emergence of free-electron lasers enabled the application of CDI to image biological particles [2–5]. The radiation damage provoked by the high brilliance causes the sample to explode, turning into hot, high-pressured plasma [6, 7]. Still, with a short enough pulse, the x-rays leave the sample within the first few femtoseconds of the photon-atom interactions when the structure remains intact. CDI can provide structural information on nanoparticles and, in particular, aims to provide new structural information on biomolecules, such as proteins, and their dynamics on ultrashort timescales.

Iterative phase retrieval algorithms [8, 9] enable the electron density reconstruction from the scattered signal alone. In general, the measured amplitudes are kept constant throughout the iterations of the algorithms while the phases are iteratively updated. Supports are used as constraints in real space and provide a region where the object can have non-zero density while constraining the rest of the real space to zero. This constraint is what makes phase retrieval possible in CDI. Multiple solutions are possible for each set of amplitudes as phase reconstructions are invariant under translation, centrosymmetric flip, and global phase shift. In practice, the support is usually set to some initial guess, usually given by the autocorrelation function or a low-resolution map obtained from prior knowledge. The support can then be refined throughout the reconstruction using the shrinkwrap algorithm [10], which reduces the support size by applying a low-pass thresholding filter. However, performing phase retrieval can be time-consuming and computationally expensive; moreover, the convergence is still very sensitive to some parameters, such as initial guesses, the shrinkwrap threshold, and the choice of algorithm.

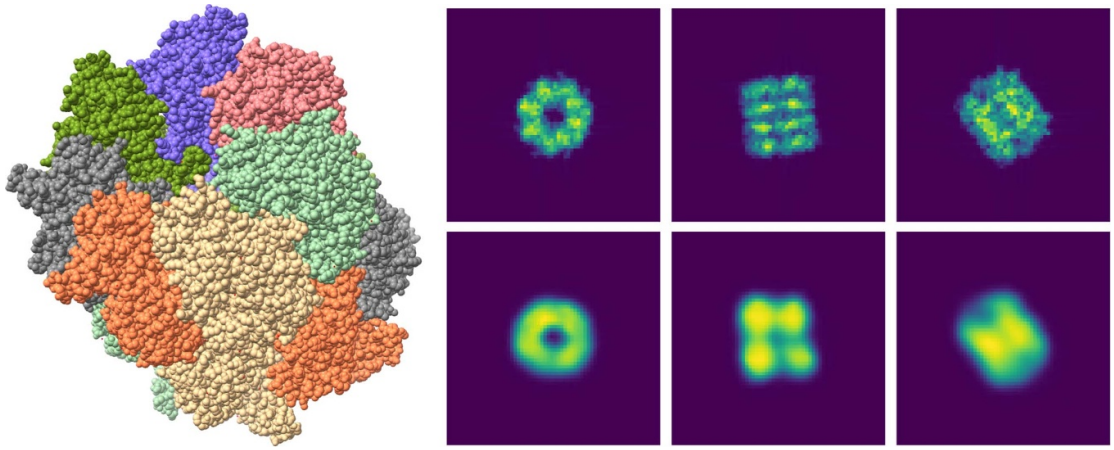

Figure 1. Left: 3D electron density of GroEL from PDB entry 1SS8 [11]. Right: 2D projections of GroEL in different orientations. In the bottom row, are the corresponding electron density maps retrieved using our DL model.

Phase retrieval algorithms generally depend on a good quality support with boundaries that tightly constrain the object. The triple-intersection method tries to tackle the problem of finding tighter boundaries geometrically [12]. This algorithm is fast but requires a good approximation of the autocorrelation, which is often not available for experimental data. Yet, when this technique can be applied, it returns a reasonable estimate of the optimal object support and possibly the best *a priori* support without relying on previous low-resolution masks or, more recently, machine learning [13, 14].

Several studies have demonstrated the potential of utilizing machine learning approaches, both on their own and in conjunction with traditional iterative phasing methods, to address the phase problem in 2-dimensional [13, 15] and 3-dimensional Bragg CDI [14, 16–18]. Though these algorithms have been applied to experimental data, they are naturally biased towards their training data, mostly convex nanostructures. Machine learning provides great advantages in terms of speed and is also robust to noise and missing data. Here, we introduce a deep-learning model that can estimate the real-space structure and the support directly from CDI data. Figure 1 shows a few examples of reconstructions from different orientations of the same protein. The model is specifically trained for CDI of proteins and other biological structures.

2. Phasing and support finding using deep-learning

We use a generative encoder–decoder network with 14 trainable layers to predict the 2D electron densities of proteins. This architecture processes diffraction data with convolutions while enabling the dense layers to reinterpret the data and generate a real-space density processed by the decoder. The model is reported in figure 2. The convolution kernels have dimensions 3×3 for every layer and stride 3. For every two convolutional layers, we apply 2×2 max-pooling, starting from an input of shape 128×128 and then reaching the bottleneck with an overall dimension of 16×16 with 256 feature maps. The bottleneck consists of 2 stacked dense layers, where the first has 128 nodes and the second 65 536. The decoder architecture is symmetrical to the encoder and feeds the dense layer into a $16 \times 16 \times 256$ convolutional layer, followed by a second one with the same dimensions, kernel size, and stride. The output layer has 128×128 pixels. Every layer, including the fully connected bottleneck, uses rectified linear unit (ReLU) activation functions, while the output layer employs a hyperbolic tangent function (Tanh). We used a dataset of 9900 paired diffraction and real-space images during training, with a batch of 64 patterns and target values, optimizing a mean-squared loss function. The network was implemented in Keras [19] and Tensorflow [20] and trained on an Nvidia RTX 2080 Ti graphics card. 5% of the 9900 frames were used for validation and 1000 patterns were used for testing the network performance.

In addition to this network, we trained a U-Net, similar to what was done in [21]. Still, we achieved better phasing performance using the encoder–decoder architecture. To ensure a more comprehensive and challenging performance analysis, the test predictions were studied on diffraction patterns from PDB files that were not part of the training dataset. The test dataset is not only composed of different data, but is simulated from different PDB files. We tested feeding the network with either diffraction patterns or autocorrelation functions calculated from those patterns. We also tested generating either supports or density maps as outputs. Furthermore, we evaluated three different loss functions: (1) binary cross-entropy,

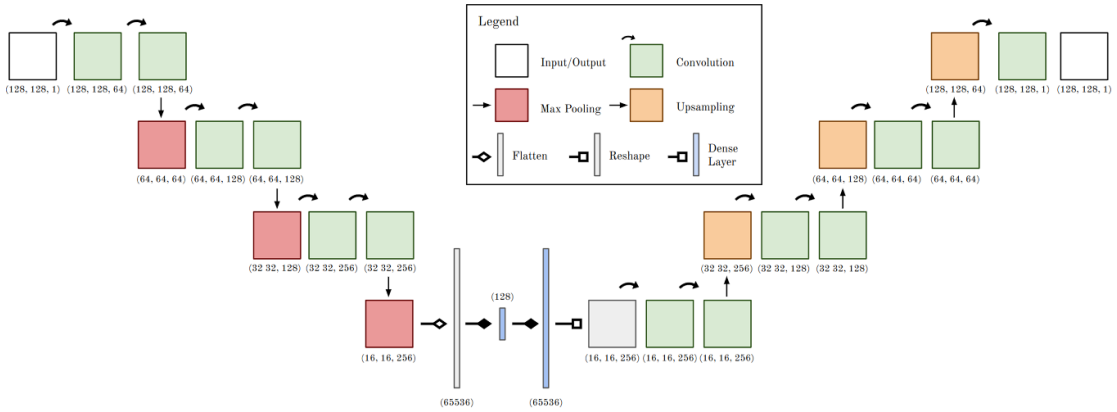

Figure 2. Architecture of the encoder–decoder network.

(2) Mean squared error (MSE), and (3) a custom loss function designed to avoid too small supports that would immediately break phase retrieval.

The custom loss function was defined as a weighted sum of the false positives and false negatives, i.e. regions where the guess is too loose or too small respectively,

$$L_3 = \frac{1}{\sum_{i\in\mathbf{S}}\hat{y}_i}\left(\sum_{j\in S-\hat{\mathbf{S}}}\hat{y}_j + \gamma\cdot\sum_{k\in\hat{\mathbf{S}}-S} y_k\right) \tag{1}$$

where γ is a weighting parameter used to punish small supports, S is the true support, $\hat{S}$ is the estimated support, and the sums correspond to the pixel sum over the set difference. The most reliable model was trained on intensities as inputs and using the mean-squared error loss function. The output densities were then thresholded to 10% of the maximum of the output to calculate supports for phasing. Over 90% of the supports generated from the network introduced less than 2 false negative pixels, while on average having 230 false positives. Figure 6 displays the input diffraction patterns utilized in our neural network and their corresponding outputs. Overall, the encoder–decoder model showed a better performance in support finding compared to the U-Net while also providing a density guess, which can be used in phasing as shown in section 6.

3. The triple-intersection method

The triple-intersection method estimates an *ab initio* support from measurements by employing the intersection of multiple copies of the autocorrelation function to find a tight support. It works by calculating the autocorrelation function of the signal. From the autocorrelation area, we calculate a binary mask and the mask's extremal points. The extremal points are estimated along the same direction to guarantee that the resulting intersection remains larger than or equal to the ideal support size. Convex autocorrelations lead to loose support, while concave autocorrelations can lead to accurately identified minimal supports. Several methods can be used to find tighter supports in 2D, according to the object geometry (concavity, convexity, number of global extremal points, etcetera) as introduced by Crimmins *et al* [12]. These ideas can be extended in 3D [23, 24]. A simple example of the effectiveness of this approach is reported in figure 3. While interesting, this method requires a very high-quality autocorrelation, which is usually not available in the presence of noise or missing data.

4. Estimating support masks

Neural networks can be trained to retrieve the densities. Still, if the outputs have low resolution, we can imagine using them as support constraints by thresholding the estimates and creating binary masks instead. We benchmarked the encoder–decoder model against the triple-intersection method on the test dataset. If the support is centrosymmetric, phase retrieval algorithms often need help breaking this invariance and might thus not reach the correct solution. Both the deep learning (DL) method and the triple-intersection method can, in many cases, break this symmetry, which significantly aids the reconstruction.

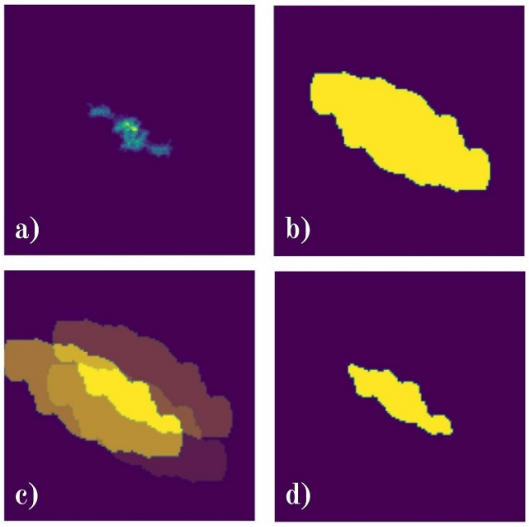

Figure 3. Example of how the triple-intersection method works. (a) Electron density (PDB-ID 1K3E [22]). (b) Autocorrelation. (c) Three copies of the autocorrelation are displayed in transparency to highlight the intersection. (d) Estimated support.

To quantify the performance of these two procedures for support estimation, we calculated the area difference from the ideal support of the protein's outline. We estimated a support mask from the neural network's guesses, and using the triple-intersection method, we estimated a support mask from each autocorrelation. We then overlapped the retrieved supports to match the ideal masks and calculated the area difference. As shown in figure 4, the neural network demonstrated a remarkable performance, with very few false negatives. However, it does have slightly more false positives on average.

For phasing, more false negatives can be considered a worse outcome than false positives, as HIO and ER usually fail to converge if the support is smaller than the particle. The triple-intersection method should, in principle, lead to looser supports than the ideal case. Still, it can be difficult to correctly estimate the optimal shifting vectors, leading to errors, which accounts for the observed false negatives, especially for experimental autocorrelations. The DL approach is, therefore, particularly promising.

5. Phasing noisy data

The triple intersection method has not been used for experimental CDI data since it requires a high-quality autocorrelation and cannot handle the noise and missing regions in such experiments. In a previous paper, [21], we presented a U-Net that could reduce the noise and retrieve missing regions. An open question was if the autocorrelation retrieved from such an improved pattern would be good enough to make the triple-intersection method relevant for CDI. We, therefore, tried to use denoised patterns as input for both strategies. Unfortunately, while denoising improved the quality of diffraction patterns for phasing, even small masks covering low-angle data still affected the autocorrelation, posing challenges for automating the triple-intersection method support-finding pipeline. On the other hand, after data healing with U-Net, having the encoder–decoder network on denoised and demasked data was successful. The reliability of the neural network's output as an accurate reconstruction of an object is not always guaranteed. Therefore, exploring whether the thresholded output can serve as a viable support mask for phasing becomes important, as finding increasingly tighter supports is crucial to determining the minimum amount of information required for successful phasing. We compared phase retrieval using different support boundaries. First, we used the autocorrelation function mask, then a circular support of maximal diameter equivalent to the maximum elongation of the protein. Finally, we performed phase retrieval using the DL model guess as a restraint. We run HIO for 10 000 iterations while applying 5 iterations of error reduction (ER) every 5th iteration of HIO. We also tested this routine with Shrinkwrap using a circular support as a starting guess. When convolving the reconstructed densities in the shrinkwrap step, we applied a Gaussian kernel with a standard deviation of 1 and a threshold of 0.1. The PRTF in figure 5 shows that the highest resolution was achieved when using Shrinkwrap, closely followed by the mask estimated from the neural network output without Shrinkwrap. In our experience, a failing Shrinkwrap is the main cause of failed

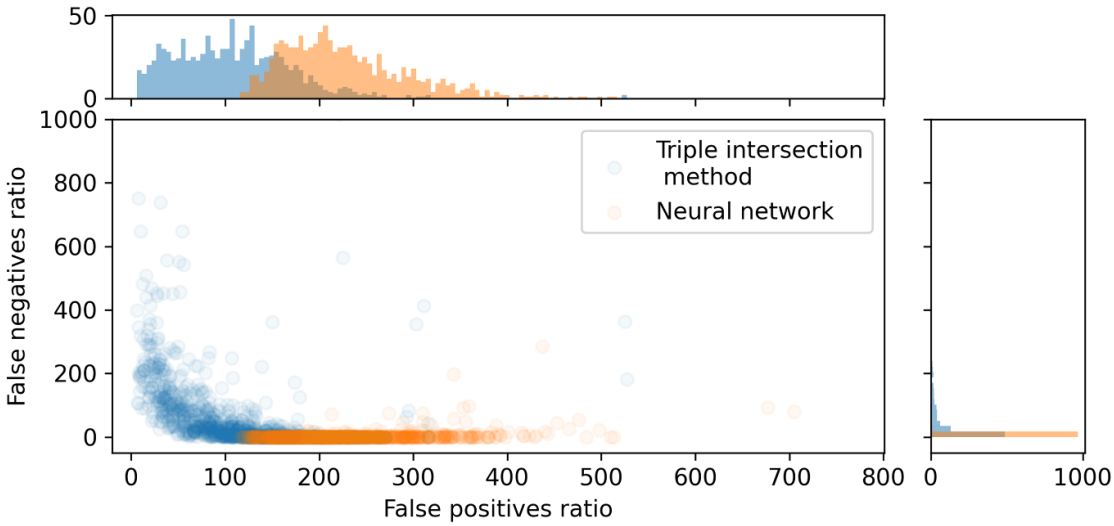

Figure 4. The scatter plot of how the two methods perform to identify the pixels contributing to the support. The top histogram shows the false positives, which are the pixels included in the support but not part of the ideal support, while the right histogram shows the false negatives. It is worth noting that, on average, the number of false negatives for the neural network is very low, indicating the method's effectiveness.

reconstructions for challenging phase-retrieval problems. The PRTF behavior in the case of Shrinkwrap with autocorrelation and HIO+ER with tighter support estimate is similar. This indicates the potential of neural networks for phasing even just for initialization purposes, as they serve as a valuable starting point for breaking centrosymmetry, outperforming circular supports when the particle size is known. The shrinkwrap algorithm often struggles to handle low signals, background signals, and missing regions. A good support estimate could help obtain higher resolutions in these more challenging cases.

6. Phasing speed

The estimates from the neural network are often quite accurate, as reported in figure 6. It is possible to see some key features of the proteins, such as shape or non-uniform densities. Yet, we can see that they are still low-resolution estimates. Instead of using them only as a support for phasing, we can pass these estimates as an initialization of real space to improve the convergence of phase-retrieval with Shrinkwrap. The accuracy and quality of this initial model affect the success rate and efficiency of the phasing process, but in the absence of prior knowledge, the densities are usually randomly initialized in the first iteration of phase retrieval. Here, we compare phasing results using initial guesses generated by the neural network with those obtained through standard random initialization of real space. As in the previous section, we run phasing 10 times with random initialization, running HIO for 10 000 iterations, with 5 iterations of error reduction every 5th iteration of HIO. The Fourier error is calculated at the end of every ER section for a total of 2000 values. The starting support was a circular mask with a diameter of the size of the protein. We started in both cases from circular support, and when convolving, in the shrinkwrap step, we applied a Gaussian kernel with a standard deviation of 1 and a threshold of 0.15. Remarkably, our model demonstrated a notable speed improvement, reducing the convergence as in figure 7. We also notice a large variance in the error when we start with a random initialization of the algorithm. In some cases, random densities lead to a local minimum, which the phasing procedure does not manage to escape within the total number of iterations used here. While HIO can sometimes escape these local minima, it poses the question of how to handle poor reconstructions, especially with experimental data when the correct phases are unknown. Including poor reconstructions would generally lead to a lower resolution, while filtering bad reconstructions would require prior knowledge of the densities. On the other hand, using an initial guess significantly speeds up the convergence and stability of the algorithm. It is, however, worth noting that without the random start, we can no longer use the PRTF to estimate the resolution.

7. Conclusions

In this work, we provided a pipeline based on deep learning for estimating particle densities from CDI data from biological nanoparticles. We have demonstrated the use of these estimates for support finding and as starting guesses for phase retrieval. The neural network model has been trained and tested on simulated

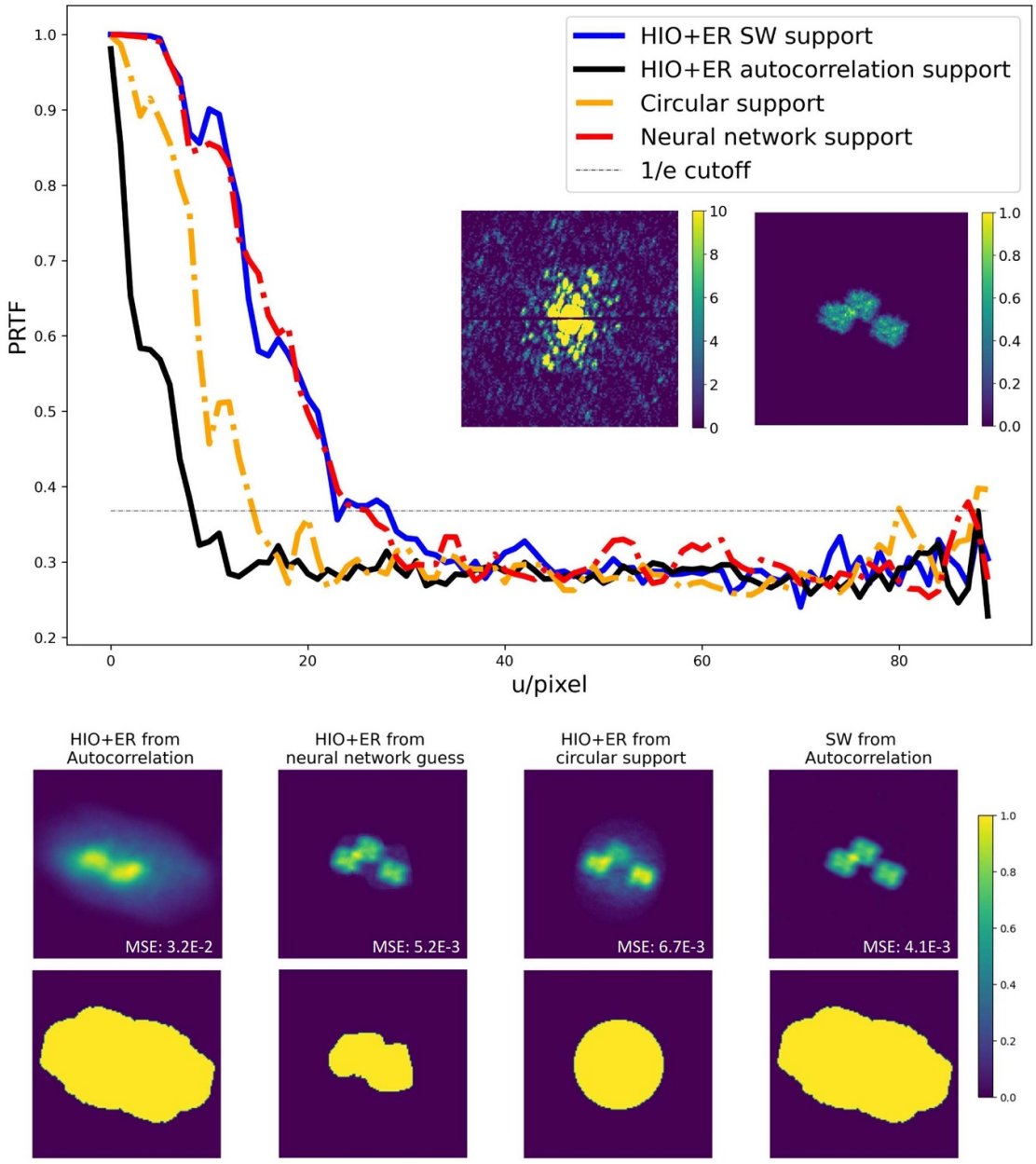

Figure 5. PRTF for reconstructions from PDB entry 2ABI [25]. The plot also displays the diffraction intensities and the ground truth. Below the plot, in the top row, from left to right, we show an average reconstruction when using the autocorrelation as a starting support, when using the neural network guess as a starting support, when using a circular support, and when using a circular support with Shrinkwrap. In the bottom row, we show the respective support masks.

images. In particular, we see a strong speedup for the reconstructions when using both the support and the starting guess to seed the phase retrieval algorithm. We also note that the returned supports usually break centrosymmetry, which is often challenging for phase retrieval algorithms. We also applied the so-called triple intersection method to obtain tight supports. This algorithm only yields reasonable outcomes when artifacts do not distort the autocorrelation function and are thus unsuitable for data, which includes noise and missing regions. The method remains unusable under realistic experimental conditions even after denoising the data.

CDI experiments require extensive processing of raw data to characterize the camera response to experimental conditions and phase retrieval is performed only after long data processing. Generative neural networks offer a promising solution for addressing the phase retrieval problem, introducing an alternative to iterative phasing algorithms. They can potentially enable real-time screening of CDI experiments and

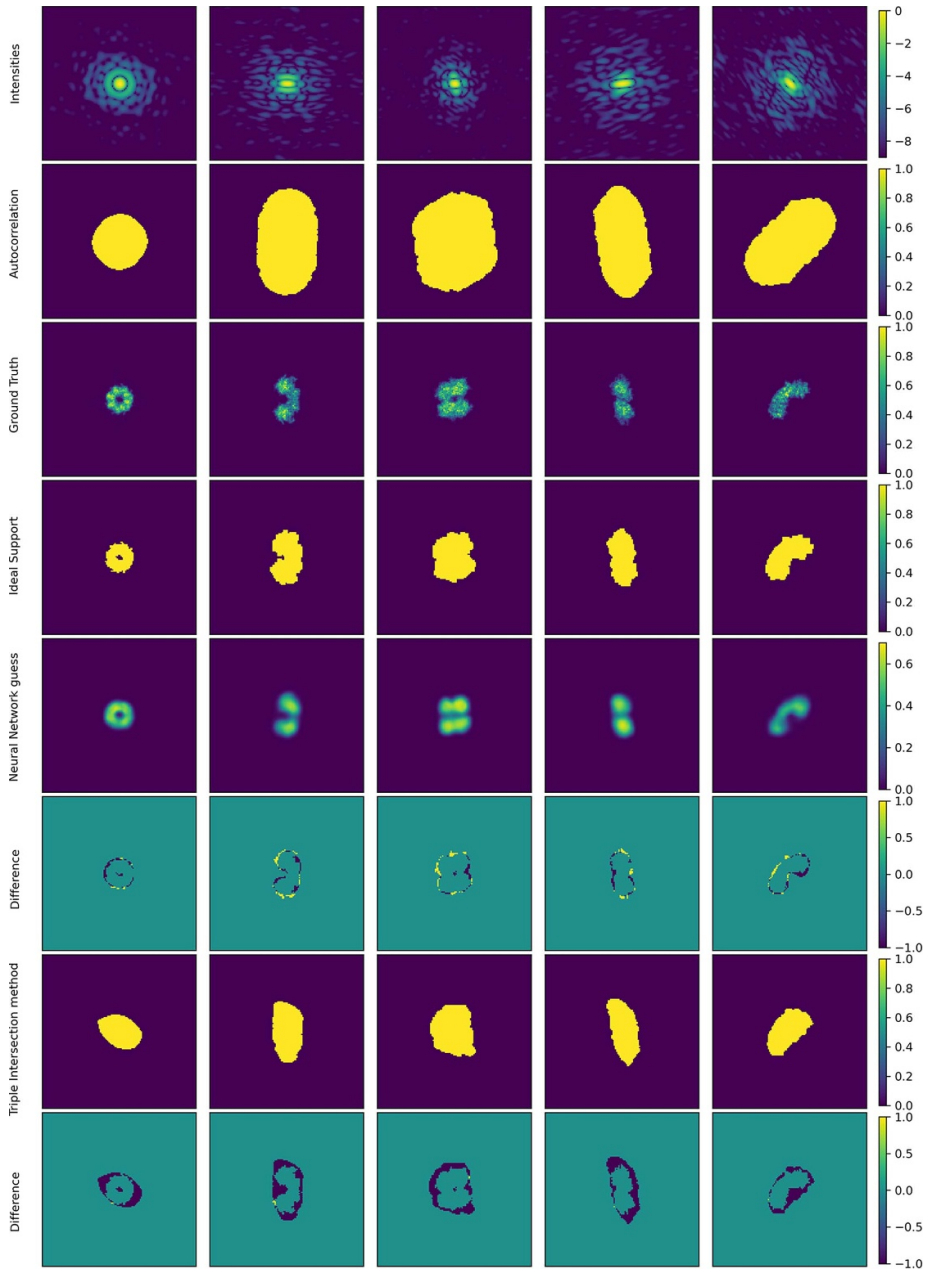

Figure 6. Examples from our test dataset and one using a GroEL particle (PDB-ID 1SS8 [11]), which has a molecular weight outside the range considered in our dataset. First row: diffraction patterns. Second row: autocorrelation masks. Third row: ground-truth. Fourth row: ideal support estimated by thresholding the ground-truth. Fifth row: neural network estimates. Sixth row: the difference between the ideal support and the support estimated by thresholding the neural network's guesses. Seventh row: support estimate obtained from the triple-intersection method. Eight row: difference between the ideal supports and supports from the seventh row. Each column is relative to a different biomacromolecule, PDB-ID in order from left to right: 1SS8 [11], 3ETA [26], 5IRC [27], 3EPP [28], 5KLT [29].

provide immediate feedback on the sample. Offline analysis would also benefit from low-resolution initial guesses, as discussed in this paper, enhancing phasing performance and simplifying existing algorithm routines, which often rely on trial and error for parameter selection. While deep learning and AI methods show promise, they are inherently black boxes. The lack of output explainability is a critical limitation for their applications and further development and testing are necessary on real data. Another challenge, more specific to imaging techniques such as CDI, includes finding proper methods to evaluate the resolution limit of the network's output and reliably validate them.

The success of this network is primarily dependent on the large dataset covering diffraction from a wide set of proteins. We thus provide our datasets to the community to facilitate reproducibility and help future

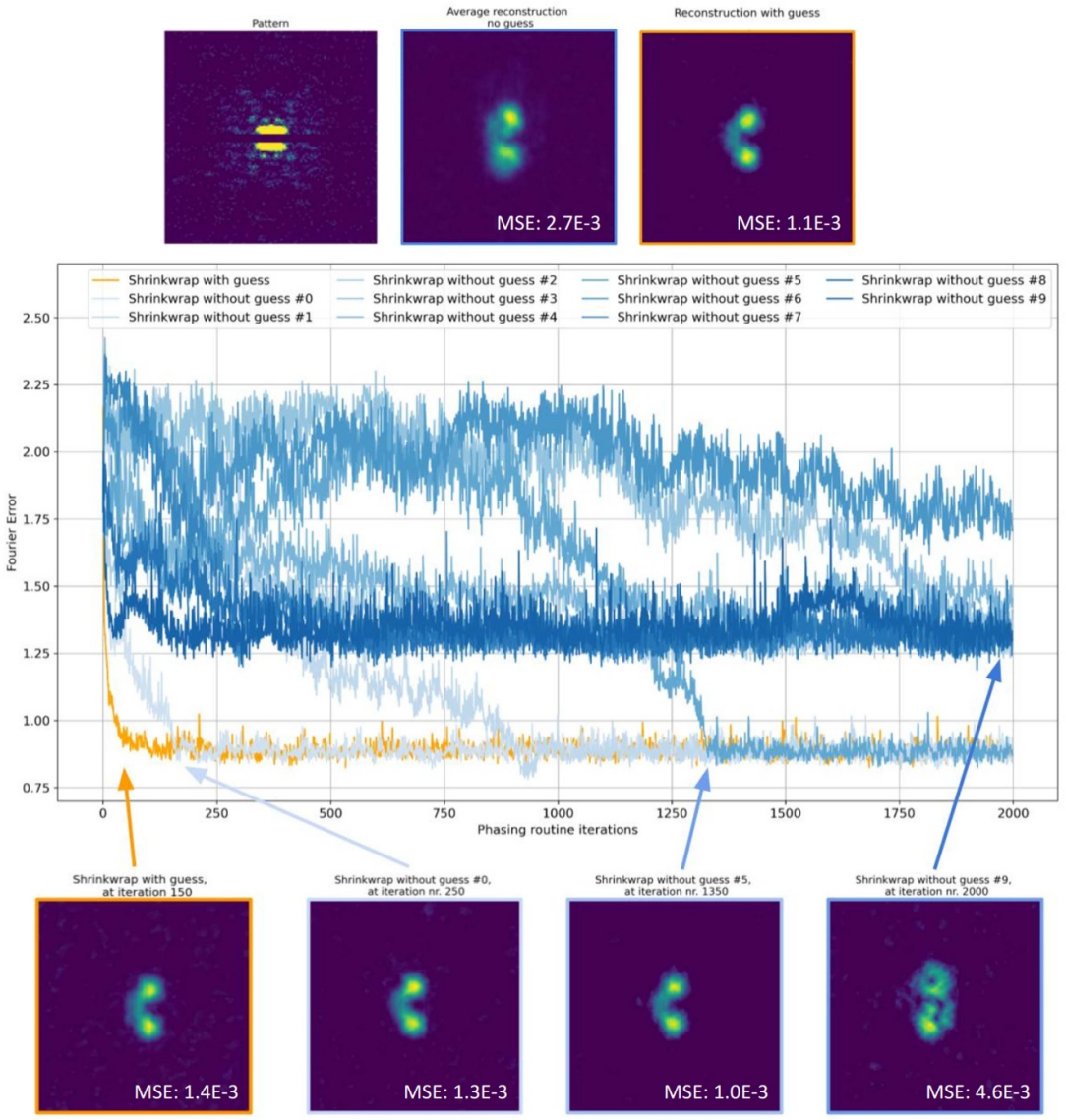

Figure 7. Fourier error for 10 reconstructions of PDB-ID 3ETA [26] obtained with Shrinkwrap and random initialization of densities, in blue. An example of reconstruction with an initial guess from the encoder-decor model is reported in orange. We applied a centrosymmetric flip to the real-space densities to align with the majority of phased outcomes achieved without an initial guess.

developments in machine-learning applications in CDI. We believe an open, standardized, and common dataset would greatly benefit the scientific community.

Data availability statement

The code for the neural network model that supports the findings of this study is available at the following URL/DOI: https://github.com/AlfredoBellisario/SPInn. The dataset is also available on the Coherent x-ray Imaging Data Bank at the following URL/DOI: https://cxidb.org/id-232.html. The training dataset consists of 10,900 atomic coordinate files sourced from the Protein Data Bank. To ensure higher generalization of the network model, no repetitions of patterns and proteins were included in the dataset—for each of these files, we simulated a single diffraction pattern, followed by calculating their respective support, density, and autocorrelation. In the simulation process, we utilized Chimera to project the PDB files and subsequently computed the intensities as the squared modulus of the Fourier transforms of the electron densities.

Acknowledgments

We thank Filipe Maia and August Wollter for fruitful discussions. This work was funded by the Swedish Research Council (Grant Number 2017-05336) and the Swedish Foundation for Strategic Research (Grant Number ITM17-0455).

ORCID iDs

Alfredo Bellisario https://orcid.org/0000-0001-5155-4431
Tomas Ekeberg https://orcid.org/0000-0003-0092-2104

References

[1] Marchesini S *et al* 2003 Coherent x-ray diffractive imaging: applications and limitations *Opt. Express* **11** 2344–53
[2] Seibert M M *et al* 2011 Single mimivirus particles intercepted and imaged with an x-ray laser *Nature* **470** 78–81
[3] Van Der Schot G *et al* 2015 Imaging single cells in a beam of live cyanobacteria with an x-ray laser *Nat. Commun.* **6** 1–9
[4] Hantke M F *et al* 2014 High-throughput imaging of heterogeneous cell organelles with an x-ray laser *Nat. Photon.* **8** 943–9
[5] Ekeberg T *et al* 2015 Three-dimensional reconstruction of the giant mimivirus particle with an x-ray free-electron laser *Phys. Rev. Lett.* **114** 098102
[6] Neutze R, Wouts R, Van der Spoel D, Weckert E and Hajdu J 2000 Potential for biomolecular imaging with femtosecond x-ray pulses *Nature* **406** 752–7
[7] Chapman H N *et al* 2006 Femtosecond diffractive imaging with a soft-x-ray free-electron laser *Nat. Phys.* **2** 839–43
[8] Gerchberg R W and Saxton W O 1972 Practical algorithm for determination of phase from image and diffraction plane pictures *Optik* **35** 237
[9] Fienup J R 1978 Reconstruction of an object from the modulus of its Fourier transform *Opt. Lett.* **3** 27–29
[10] Marchesini S, He H, Chapman H N, Hau-Riege S P, Noy A, Howells M R, Weierstall U and Spence J C 2003 X-ray image reconstruction from a diffraction pattern alone *Phys. Rev.* B **68** 140101
[11] Chaudhry C, Horwich A L, Brunger A T and Adams P D 2004 Exploring the structural dynamics of the E. coli Chaperonin GroEL using translation-libration-screw crystallographic refinement of intermediate states *J. Mol. Biol.* **342** 229–45
[12] Crimmins T, Fienup J and Thelen B 1990 Improved bounds on object support from autocorrelation support and application to phase retrieval *J. Opt. Soc. Am.* A **7** 3–13
[13] Cherukara M J, Nashed Y S and Harder R J 2018 Real-time coherent diffraction inversion using deep generative networks *Sci. Rep.* **8** 1–8
[14] Wu L, Yoo S, Suzana A F, Assefa T A, Diao J, Harder R J, Cha W and Robinson I K 2021 Three-dimensional coherent x-ray diffraction imaging via deep convolutional neural networks *npj Comput. Mater.* **7** 1–8
[15] Wu L, Juhas P, Yoo S and Robinson I 2021 Complex imaging of phase domains by deep neural networks *IUCrJ* **8** 12–21
[16] Scheinker A and Pokharel R 2020 Adaptive 3D convolutional neural network-based reconstruction method for 3D coherent diffraction imaging *J. Appl. Phys.* **128** 184901
[17] Stielow T and Scheel S 2021 Reconstruction of nanoscale particles from single-shot wide-angle free-electron-laser diffraction patterns with physics-informed neural networks *Phys. Rev.* E **103** 053312
[18] Yao Y, Chan H, Sankaranarayanan S, Balaprakash P, Harder R J and Cherukara M J 2021 AutoPhaseNN: unsupervised physics-aware deep learning of 3D nanoscale coherent imaging (https://doi.org/10.1038/s41524-022-00803-w)
[19] Chollet F *et al* 2015 Keras (available at: https://keras.io)
[20] Abadi M *et al* 2015 TensorFlow: large-scale machine learning on heterogeneous systems Software available from tensorflow.org
[21] Bellisario A, Maia F R N C and Ekeberg T 2022 Noise reduction and mask removal neural network for x-ray single-particle imaging *J. Appl. Crystallogr.* **55** 122–32
[22] Luo Y *et al* 2001 Structural and biochemical characterization of the type III secretion chaperones CesT and SigE *Nat. Struct. Biol.* **8** 1031–6
[23] Fienup J R, Thelen B J, Reiley M F and Paxman R G 1997 3D locator sets of opaque objects for phase retrieval *Proc. SPIE* **3170** 88–96
[24] Reiley M F, Paxman R G, Fienup J R, Gleichman K W and Marron J C 1997 3D reconstruction of opaque objects from Fourier intensity data *Proc. SPIE* **3170** 76–87
[25] Hudson W H, Youn C and Ortlund E A 2014 Crystal structure of the mineralocorticoid receptor DNA binding domain in complex with DNA *PLoS One* **9** e107000
[26] Patnaik S *et al* 2009 Discovery of 3, 5-disubstituted-1H-pyrrolo [2, 3-b] pyridines as potent inhibitors of the insulin-like growth factor-1 receptor (IGF-1R) tyrosine kinase *Bioorg. Med. Chem. Lett.* **19** 3136–40
[27] Amin E, Jaiswal M, Derewenda U, Reis K, Nouri K, Koessmeier K T, Aspenström P, Somlyo A V, Dvorsky R and Ahmadian M R 2016 Deciphering the molecular and functional basis of RHOGAP family proteins: a systematic approach toward selective inactivation of rho family proteins *J. Biol. Chem.* **291** 20353–71
[28] Wang D-P *et al* 2023 Crystal structure of mRNA cap (guanine-N7) methyltransferase E12 subunit from monkeypox virus and discovery of its inhibitors *Int. J. Biol. Macromol.* **253** 127565
[29] Smith K M *et al* 2018 Contribution of the residue at position 4 within classical nuclear localization signals to modulating interaction with importins and nuclear targeting *Biochim. Biophys. Acta, Mol. Cell Res.* **1865** 1114–29