Article

SPRING, an effective and reliable framework for image reconstruction in single-particle Coherent Diffraction Imaging

Check for updates

Alessandro Colombo[1] ✉, Mario Sauppe[1], Andre Al Haddad[2], Kartik Ayyer[3,4], Morsal Babayan[5], Rebecca Boll[6], Ritika Dagar[7,8], Simon Dold[6], Thomas Fennel[9], Linos Hecht[1], Gregor Knopp[2], Katharina Kolatzki[1], Bruno Langbehn[10], Filipe R. N. C. Maia[11], Abhishek Mall[3], Parichita Mazumder[3,21], Tommaso Mazza[6], Yevheniy Ovcharenko[6], Ihsan Caner Polat[1], Dirk Raiser[6], Julian C. Schäfer-Zimmermann[1,12,13,14], Kirsten Schnorr[2], Marie Louise Schubert[1,15], Arezu Sehati[11,16], Jonas A. Sellberg[16], Björn Senfftleben[6], Zhou Shen[3], Zhibin Sun[2], Pamela H. W. Svensson[17], Paul Tümmler[9], Sergey Usenko[6], Carl Frederic Ussling[1], Onni Veteläinen[5], Simon Wächter[1], Noelle Walsh[18], Alex V. Weitnauer[1], Tong You[11], Maha Zuod[1], Michael Meyer[6], Christoph Bostedt[2,19], Davide E. Galli[20], Minna Patanen[5] & Daniela Rupp[1]

Coherent Diffraction Imaging (CDI) is an experimental technique to image isolated structures by recording the scattered light. The sample density can be recovered from the scattered field through a Fourier Transform operation. However, the phase of the field is lost during the measurement and has to be algorithmically retrieved. Here we present SPRING, an analysis framework tailored to X-ray Free Electron Laser (XFEL) single-shot single-particle diffraction data that implements the Memetic Phase Retrieval method to mitigate the shortcomings of conventional algorithms. We benchmark the approach on data acquired in two experimental campaigns at SwissFEL and European XFEL. Results reveal unprecedented stability and resilience of the algorithm's behavior on the input parameters, and the capability of identifying the solution in conditions hardly treatable with conventional methods. A user-friendly implementation of SPRING is released as open-source software, aiming at being a reference tool for the CDI community at XFEL and synchrotron facilities.

The advent of Free-Electron Lasers (FELs)[1–7] represented a game-changer for the study of matter and its interaction with light. Short-wavelength FELs are capable of producing extremely bright pulses of coherent and monochromatic light, ranging from the extreme ultraviolet (XUV) spectrum to hard X-rays. A unique feature of those pulses is their time duration, often on the order of a few tens of femtoseconds and nowadays approaching the attosecond time scale[8–10]. Such exceptionally short pulses, in combination with their high brightness, unlocked unprecedented opportunities for time-resolved investigations of ultra-fast processes of molecules[11] and even electrons[12–14], as well as structural studies on especially fragile systems[15–18]. Several experimental techniques can be exploited at XFEL (XUV and X-ray FEL) facilities, gaining immense benefits from the properties of their light pulses. This is particularly true for imaging methods, which intrinsically require a large amount of photons.

Imaging techniques aim at retrieving structural properties of the sample, and, specifically, the spatial distribution of their electron density, to which photons are sensitive in this energy range. Due to the intensity of the light and the high photon energy direct imaging methods like microscopy are hardly viable due to the lack of suitable optics. For this reason, imaging at XFELs is performed in an *indirect* manner, by recording the coherent light scattered off the sample and then retrieving its spatial distribution by sophisticated data analysis approaches. This *lensless* experimental scheme is called Coherent Diffraction Imaging (CDI)[19–24].

While various approaches can be linked to the definition of CDI, like Bragg-CDI[25] or Ptychography[26], the most common technique employed

A full list of affiliations appears at the end of the paper. ✉e-mail: alcolombo@phys.ethz.ch

1

at XFELs is Single Particle CDI (SP-CDI) in the small-angle scattering regime[20,27,28]. Here, the light scattered by an isolated sample is acquired by a detector, which is placed sufficiently far from the interaction region to satisfy the *far-field* condition[29]. The diffraction signal is collected for a scattering angle up to a few degrees, such that the portion of the Ewald's sphere defined by the corresponding momentum transfer accessible by the detector can be well approximated with a flat surface. Under Born's approximation, the field scattered at small angles is then proportional to the Fourier Transform (FT) of the sample's electronic density projected along the axis of the XFEL beam.

Despite the straightforward mathematical relationship, which in principle enables the restoration of the sample's 2D projection by an inverse FT operation on the scattered field, the latter cannot be fully accessed by photon detectors. Detectors are, in fact, two-dimensional arrays of photon counters. The acquired signal is then only sensitive to the intensity of the diffracted field, while its phase is completely lost in the measurement process[30–33].

The field's phases are however retrievable if the sample is confined in space and "small enough", i.e., if the so-called *oversampling condition* is met[18,30]. In such a case, it can be mathematically shown that the information contained in the field's intensity alone is sufficiently redundant to also carry information on its phases[34]. This possibility is exploited by phase retrieval algorithms[31,35,36] to successfully retrieve the lost phases and, consequently, the corresponding image of the sample's electronic density.

The core idea behind *phase retrieval algorithms* was originally conceived in the 1980' by Fienup[37] and then further developed in the following years[35]. These algorithms are based on the iterative application of two constraints. Given a starting guess for the sample's electronic density, they go back and forth from the Fourier representation to the real representation of the spatial density. In the reciprocal space, the signal's amplitudes are replaced by those experimentally measured by the detector. In the real space, the oversampling condition is enforced by constraining the limited spatial extension of the electronic density. With proceeding iterations, the similarity between the FT amplitude of the retrieved density and the experimental data is optimized.

A known issue of phase retrieval algorithms is their tendency to stagnate in local optima and/or being unstable[35]. On top of that, XFEL diffraction data present additional challenges. First, the central scattering region is not recorded, due to a hole in the detector to avoid damages from the FEL transmitted beam. Second, the brightness of the patterns, i.e. the total number of photons recorded by the detector, fluctuates on a shot-by-shot basis, with little or no possibility to be tuned. Third, samples often have unpredictable shapes, giving little or no a-priori information to facilitate the reconstruction process. As a result, the identification of the correct reconstruction by phase retrieval algorithms is not ensured, it is possible only in a restricted range of conditions and it requires a significant case-by-case tuning of the algorithm parameters by an experienced person.

These issues and limitations are well known by the CDI community, and a significant effort has been invested into developing more reliable algorithmic methods. The use of several phase retrieval processes with different starting conditions on the same diffraction data has proved to be particularly effective. On the one hand, the ensemble of final reconstruction outcomes can be compared and statistically analyzed to identify those reconstruction procedures that reached the correct solution and exclude the failed attempts[22,38–40]. On the other hand, the simultaneous execution of several reconstruction processes with different starting conditions allows for more sophisticated optimization strategies based on the sharing of information between the ensemble of ongoing reconstructions[40–42].

Recently, we developed a sophisticated approach to the *phase retrieval problem* in CDI that conceptually belongs to this ensemble-based class. The method is called Memetic Phase Retrieval (MPR)[43,44]. It is based on a natural computing scheme and it had been originally designed to deal with Electron Diffraction Imaging data[45–47]. In this work, we present the SPRING framework, which implements an improved version of the MPR method tailored on the specific features of single-particle single-shot experiments at FELs. Together with this publication, we release its software implementation as an open-source Python3 module, named *spring*, conceived to exploit the

full capability of multi-CPU and multi-GPU computing systems. The pre-release of the SPRING software is currently being used by different research groups recently involved in the analysis of CDI experiments. The *spring* module, its documentation and code can be accessed at https://share.phys.ethz.ch/~nux/software/spring/main/docs/.

This article is a comprehensive report of SPRING, its implementation of the MPR method and its performance on experimental XFEL diffraction data. After a brief introduction to the *phase retrieval problem* in section "The phase problem and conventional approaches", section "Memetic phase retrieval" is dedicated to an intuitive description of the method. In the following section "Dealing with experimental data" we challenge the algorithm on experimental data. Its performance is compared with conventional approaches, and its behavior, capabilities and limitations on images at the limits of the capabilities of standard *phase retrieval* algorithms are probed. The experimental diffraction patterns used as a benchmark have been acquired with the two main CDI detector types of today's XFEL imaging instruments: the pnCCD detector[48] of the SQS instrument at the European XFEL[49,50] and the Jungfrau detector[51,52] installed at the Maloja endstation of SwissFEL[53]. Further details of the SPRING implementation are reported and discussed in Appendices IV B and IV M, aided by an extensive use of pseudo-code.

The quality and reliability of the imaging results, the superior performance and the highly optimized and handy software implementation, make SPRING a strong candidate for becoming a reference analysis tool, fulfilling the currently high demand for a reliable and easy-to-use analysis software tailored on single-particle single-shot CDI data at XFELs.

Results
The phase problem and conventional approaches
The *phase problem* arises when a function $\rho(\vec{x})$ has to be retrieved, but only its Fourier modulus $M(\vec{q}) = |\mathcal{F}[\rho](\vec{q})|$ is known. In single-shot CDI experiments, under the assumptions of *small-angle* and *far-field* scattering, and validity of *Born's approximation*, the recorded diffraction intensities $I(\vec{q})$ acquired by a detector correspond to $M^2(\vec{q})$, while $\rho(\vec{x})$ is the electronic density of the sample under study[30] projected along the beam direction. The phases of the scattered field, lost in the measurement process, have to be retrieved by solving a two-dimensional phase retrieval problem.

The phase problem is known to have a unique solution if the density ρ that produces the scattering signal is limited in space, i.e. it is of limited size and surrounded by an environment of known scattering density (usually vacuum, with value 0). This condition is enforced by a so-called support function $S(\vec{x})$, which assumes values equal to 1 only where the scattering density is known to have non-zero values, or 0 otherwise[19,54]. The spatial extension of the support function and, thus, of the object's density, has to be sufficiently restricted to satisfy the so-called *oversampling condition*[30]. If such a condition is met, it can be mathematically demonstrated that the phase retrieval problem has a unique solution[34] (up to some straightforward ambiguities, see ref. [18]), i.e. there exists only one density ρ fully "contained" in the support function and whose Fourier amplitude corresponds to the measured diffraction data.

Iterative phase retrieval algorithms[35,54] cyclically apply the *support constraint* and the *amplitude constraint* to a starting density ρ. The *support constraint* is enforced by suppressing the density values in real-space that are located outside the support function. The *amplitude constraint* is enforced by replacing the Fourier modulus with the experimentally recorded one, $M(\vec{q})$.

The error $E[\rho]$ of a density ρ is defined as the difference between the Fourier amplitude of the density $|\mathcal{F}[\rho]|$ and the experimental data M. The scope of a reconstruction process is to minimize the error E (see section "The phase problem as an optimization problem" for a more formal description).

The simplest phase retrieval algorithm is the Error Reduction (ER)[54], which takes its name from its characteristics of *always* reducing the error value iteration after iteration. This feature makes the ER algorithm very *stable*, but also prone to *stagnation* in local minima of the error function E.

To circumvent this limitation, more advanced iterative phase retrieval algorithms have been conceived[35,36], like the Hybrid Input-Output (HIO)[37] and the Relaxed Averaged Alternating Reflections (RAAR)[55], which make a more sophisticated use of the support and amplitude constraints. The great advantage of those algorithms is that they suffer less from stagnation in local optima, thus being more *ergodic*, i.e. they can better explore the parameter space. However, they are often unstable, meaning that they fail to stably converge towards an optimum of the error function. From now on, we will refer to those iterative algorithms as IAs.

As a trade-off, a common approach is to alternate a number J of *ergodic* IAs and a number J_{ER} of ER algorithm[35]. This means that a sequence of algorithms is used, and can be formalized as:

$$\rho' \leftarrow \text{Sequence}(\rho) \qquad (1)$$

At the beginning of the reconstruction process, when the density guess ρ is far from the solution, a high number of IAs iterations J is set to better explore the space. This number is reduced during the reconstruction down to J = 0 when the reconstruction is supposed to be close to the solution, to reduce instabilities and let the process converge to an optimum via the ER algorithm.

The support function $S(\vec{x})$ defines the "shape" of the sample density, and it is necessary to ensure the existence of a unique solution. In most experiments, and in particular single-shot single-particle CDI, the spatial extension of the sample is not known a-priori, which in turn means that the support S cannot be correctly defined before the reconstruction process.

This aspect was an open problem until the Shrink-wrap (SW) algorithm[19], which improves the shape of the support function S along with the reconstruction process of the density ρ. This operation is performed by first smoothing the density with a gaussian kernel of standard deviation σ, and then defining the new support function S' as those coordinates where the values of the smoothed density ρ_σ are greater than a fraction τ of the maximum value of ρ_σ. It is possible to indicate the SW algorithm as a function UpdateSupport. The UpdateSupport function calculates a new support function S' for the input density ρ, and depends on the parameters σ and τ:

$$S' \leftarrow \text{UpdateSupport}(\rho) \qquad (2)$$

The execution of the SW algorithm for updating the support function is typically alternated with the execution of an algorithm sequence in Eq. (1)[19].

As a reconstruction process involves not only the sample density ρ, but also its support S, it is convenient to indicate a reconstruction as $\mathcal{R}$, with its spatial density $\mathcal{R} \triangleright \rho$ and its support $\mathcal{R} \triangleright S$. Typically, a number $P \gg 1$ of reconstruction processes are executed with different random initializations, forming a set of reconstructions $\{\mathcal{R}^p\}$.

Algorithm 1.

1: $\{\mathcal{R}^p\} \leftarrow$ Initialize
2: **for** g = 1, . . . , G **do**
3: $\quad\quad \{\mathcal{R}^p\} \leftarrow$ Improve $\{\mathcal{R}^p\}$
4: **end for**
5: $\rho_{out} \leftarrow$ Best $\{\mathcal{R}^p \triangleright \rho\}$

A full phase retrieval procedure is summarized in Alg. 1. The INITI-ALIZE function takes care of randomly initializing a set of P reconstructions $\{\mathcal{R}^p\}$, each with its density $\mathcal{R}^p \triangleright \rho$ and support $\mathcal{R}^p \triangleright S$. Then, the reconstructions are improved G times by the function IMPROVE, defined in Alg. 2. It internally executes a sequence of iterative algorithms (see Eq. (1)) and the support update function (see Eq. (2)) on all the P reconstructions in the set $\{\mathcal{R}^p\}$. At the end of the process, the best results in $\{\mathcal{R}^p\}$ are kept as final reconstruction output.

Algorithm 2.

1: **function** Improve($\{\mathcal{R}^p\}$)
2: $\quad$ **for** p = 1, . . . , P **do**
3: $\quad\quad \mathcal{R}^p \triangleright \rho \leftarrow$ Sequence ($\mathcal{R}^p \triangleright \rho$)

4: $\quad\quad \mathcal{R}^p \triangleright S \leftarrow$ UpdateSupport ($\mathcal{R}^p \triangleright \rho$)
5: $\quad$ **end for**
6: $\quad$ **return** $\{\mathcal{R}^p\}$
7: **end function**

The Memetic Phase Retrieval (MPR) approach[43] implemented in SPRING reshapes this typical workflow to make a more efficient use of the information provided by the several reconstruction processes, as discussed in the next section.

Memetic phase retrieval

The structure of the Memetic Phase Retrieval (MPR) is inspired by Memetic Algorithms, which merge together local optimization methods with Genetic Algorithms[56]. In fact, MPR alternates the execution of conventional iterative phase retrieval algorithms with additional operations, presented in this section, to better minimize the error of the CDI reconstruction. The overall workflow of the MPR approach implemented in SPRING retains its original main structure[43]. It is reported in Alg. 3 and accompanied by a qualitative but intuitive sketch in Fig. 1. The MPR method is based on the simultaneous execution of P reconstructions, identified by the set $\{\mathcal{R}^p\}$, on which the action of conventional iterative phase retrieval algorithms is intertwined with operations inherited from Genetic Algorithms.

Algorithm 3. The MPR main structure
1: $\{\mathcal{R}^p\} \leftarrow$ Initialize
2: **for** g = 1, . . . , G **do**
3: $\quad \{\mathcal{R}^p_{new}\} \leftarrow$ Crossover ($\{\mathcal{R}^p\}$)
4: $\quad \{\mathcal{R}^p\} \leftarrow$ Improve ($\{\mathcal{R}^p\}$)
5: $\quad \{\mathcal{R}^p_{new}\} \leftarrow$ Improve ($\{\mathcal{R}^p_{new}\}$)
6: $\quad \{\mathcal{R}^p\} \leftarrow$ Select ($\{\mathcal{R}^p\}$, $\{\mathcal{R}^p_{new}\}$)
7: **end for**
8: $\rho_{out} \leftarrow$ Average ($\{\mathcal{R}^p \triangleright \rho\}$)

The first step of the algorithm (leftmost side of Fig. 1) consists of the creation of a set of initial guesses for the density. This operation is performed by the INITIALIZE function (line 1 of Algorithm 3), which takes care of filling with starting random values the density matrices $\mathcal{R}^p \triangleright \rho$ and the respective support matrices $\mathcal{R}^p \triangleright S$ for each reconstruction $\mathcal{R}^p$, similarly to the standard approach in Algorithm 1. The set $\{\mathcal{R}^p\}$ will be referred to as a *population*, borrowing the terminology from Evolutionary Algorithms (EAs). This initialization step is not strictly connected to the MPR method, and its implementation, discussed in section "Initialization of the starting densities", has been specifically designed for SPRING.

After the initialization step, the main body of the algorithm is composed by the loop at line 2 of Algorithm 3. We will refer to each iteration of such a loop as *generation*, again borrowing the term from EAs. For each *generation*, the core functions CROSSOVER, IMPROVE and SELECT are called.

The CROSSOVER routine creates a new population of P reconstructions $\{\mathcal{R}^p_{new}\}$ by combining the information provided by the current population $\{\mathcal{R}^p\}$, as sketched in Fig. 1. Each new reconstruction $\mathcal{R}^p_{new}$ is obtained by mixing the density and support values of four different reconstructions in the original population $\{\mathcal{R}^p\}$. Details on the crossover implementation in SPRING are given in section "Crossover". How information is mixed with the crossover operation has been significantly improved over the original MPR method[43], to better deal with intrinsic correlation in the diffraction data (see the tiling operation in section "Crossover") and with the missing intensities in the center (see section Missing intensities at low scattering angles).

The IMPROVE function is analogue to the one defined for the conventional approach in Alg. 2 and improves the density estimates via iterative algorithms and the support functions via the Shrink-wrap algorithm. Here, the operation is performed on the original population $\{\mathcal{R}^p\}$ as well as on the new population $\{\mathcal{R}^p_{new}\}$. SPRING implements an enhanced version of iterative algorithms that allows for a mitigation of high-resolution artifacts produced by missing data in the gaps between the scattering detector tiles
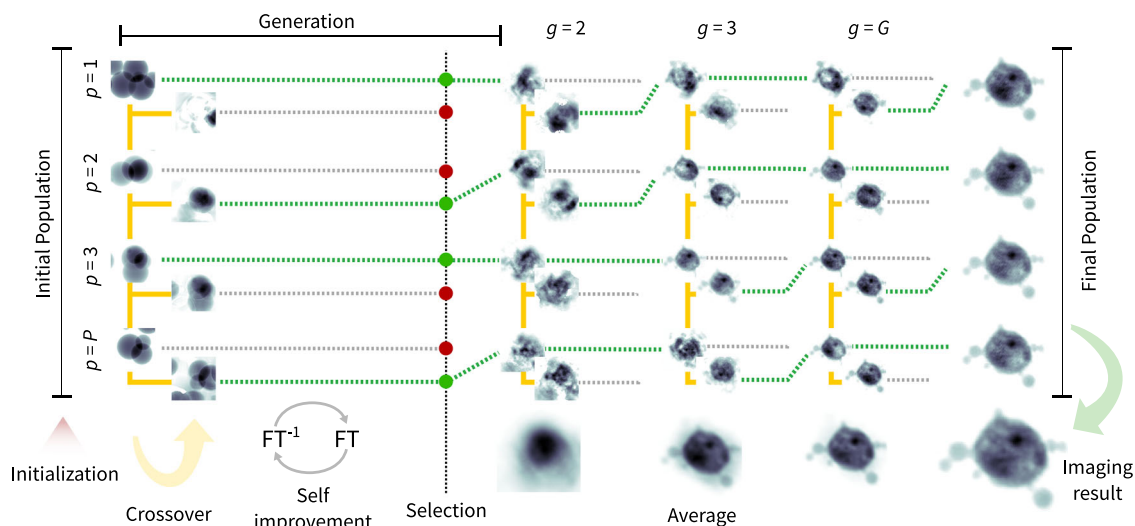
Fig. 1 | Scheme of the MPR method implemented into SPRING. On the left of the figure, a population of reconstructions of size P is initialized. For each *generation g* of the algorithm, three operations are performed. First, a new population of the same size is created by combining density values from individuals of the original population via the *crossover* operator. In the following *self-improvement* step the reconstructions of both populations are locally optimized by alternating sequences of iterative phase retrieval algorithms, to update the density values, and the *Shrink-wrap* algorithm, to update the corresponding support functions. At the end of each generation, reconstructions of the two populations are compared for each index p, and only those that reached the lowest error value survive to the next generation. With the proceeding generations g, the population of reconstructions converges towards a single solution. The convergence can be identified by observing the average reconstruction for each generation, shown in the bottom. The average density at the final generation $g = G$ is kept as reconstruction result of the whole imaging process.

(see section "Constraining intensities at large scattering angles") and a more refined execution flow of iterative phase retrieval algorithms with respect to the original MPR implementation. Additional information on this step, known as *self-improvement* in the context of Memetic Algorithms (and indicated as such in Fig. 1), is given in section "The self-improvement".

The SELECT procedure, whose details are discussed in section "The selection step", compares each individual of the original population, $\mathcal{R}^p$, with the corresponding reconstruction in the new one $\mathcal{R}^p_{new}$ right after the *self-improvement* step. Only those reconstructions that performed "better" than their counterpart (i.e. reached a lower error value) survive to the next generation while others are discarded, as schematically reported in Fig. 1.

The *crossover*, *self-improvement* and *selection* operations are repeated for a given number of generations G, during which the population of reconstructions converges towards the solution. The progress of the process can be monitored by calculating the average of the reconstructions in the population at each generation g, reported on the lower side of Fig. 1. The average reconstruction is a valuable indicator of the performance of the retrieval process, because it presents only the spatial features of the sample that are common to most of the retrieved densities in the population. For the same reason, the average density calculated at the last generation is kept as the final output of the MPR algorithm, i.e. the solution to the imaging problem.

The detailed description and discussion on how the three operations CROSSOVER, IMPROVE and SELECT are implemented in SPRING, along with details on how the starting population is initialized, are deferred to section "Memetic operators". The following part of the manuscript focuses on characterizing and understanding the behavior of SPRING on experimental data acquired at XFEL facilities, and comparing its output and performance with those provided by a conventional use of iterative phase retrieval algorithms.

Dealing with experimental data

The data treated in this work were generated from two CDI experiments conducted at SwissFEL[57] and European XFEL[5]. Sample diffraction patterns, produced by single XFEL pulses, and corresponding CDI reconstructions performed with SPRING are reported in Fig. 2. Technical details on the datasets are given below, while further information on the data selection is provided in Supplementary section 1.

Dataset A – Pure argon and mixed argon/xenon nanoclusters produced by a pulsed cluster source equipped with an Even-Lavie valve[58] are imaged at 700 eV and 1 keV photon energy, equivalent to 1.77 nm and 1.24 nm wavelength, respectively. The XFEL delivered single pulses of 50 fs duration and 2.5 mJ total energy, focused on a sub-3 μm spot. The experiment was performed at the Maloja endstation of SwissFEL[53]. The beamline is equipped with a 4 Mpixel Jungfrau detector optimized for soft x-ray photon detection[51,52]. The detector pixels, with physical size of 75 μm × 75 μm, are organized in separated tiles of 256 × 512 pixel and arranged for the detection of light up to 13.5° scattering angle. An example of a scattering pattern belonging to this dataset is shown in Fig. 2a.

Dataset B – Nanoparticles produced from an aerosolised aqueous solution of calcium chloride and sodium chloride mixture are dried and injected into the experimental chamber in form of aerosol, and are imaged at around 1.1 keV photon energy, equivalent to 1.13 nm wavelength. A constant output atomiser (model 3076, TSI Inc., MI, USA) was used to create a stream of polydisperse aerosol focused to the interaction region using an aerodynamic lens. The experiment was performed at the Small Quantum Systems (SQS) instrument of the European XFEL[49,50] using the Nano-sized Quantum Systems (NQS) end station. Typically, the XFEL was providing soft X-ray pulses with pulse energies of up to 5 mJ and the beam was focused to a diameter of about 5 μm in the interaction region. The scattering light is recorded by a 1 Mpixel pnCCD detector[48]. The detector is composed of two modules of 512 × 1024 pixel, with size 75 μm × 75 μm each, resulting in a total linear dimension of 76.8 mm. Its placement allows the scattered light to be recorded up to 6.3° at the edge of the detector. An example of a scattering pattern taken from this dataset is shown in Fig. 2d.

The *phase retrieval problem* is known to be particularly challenging[59,60] and there are different aspects in which the standard use of iterative phase retrieval algorithms present some shortcomings and limitations[35,36,61]. Among those, the following are of particular relevance when dealing with experimental data:

- The convergence of a reconstruction to the correct solution is not ensured. It is furthermore challenging to identify a correct reconstruction in an automatized manner.
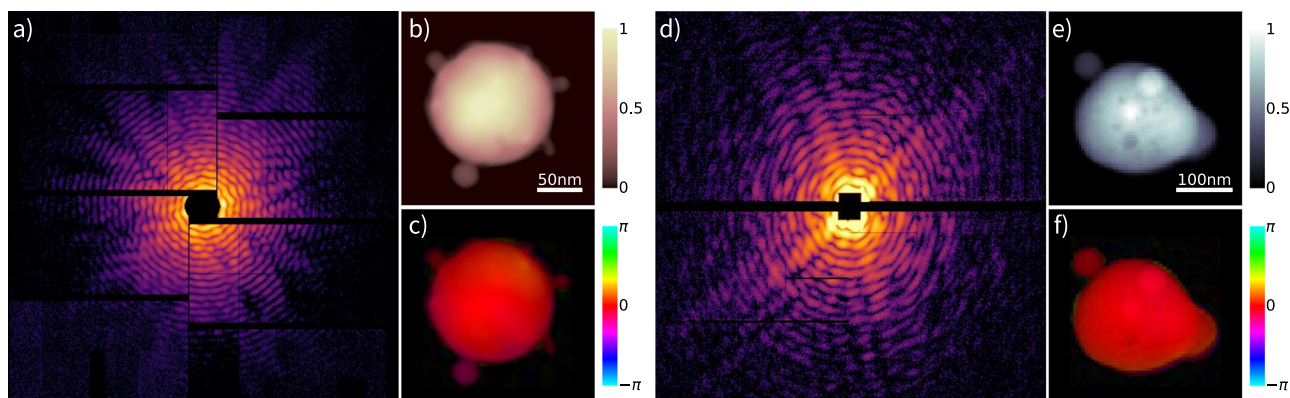
Fig. 2 | Examples of experimental diffraction patterns. In (**a**), a diffraction pattern extracted from dataset A. The corresponding reconstruction, performed with SPRING, is reported in (**b**) for the amplitude, and (**c**) for the phases. In (**d**), an example extracted from dataset B is shown, with reconstructed amplitude and phase reported in (**e**), (**f**).

Fig. 3 | Behavior of a typical reconstruction procedure with SPRING on a diffraction pattern extracted from Dataset A. In (**a**), the value of different performance indicators are reported as a function of the MPR generations. Their characteristics allow the identification of three distinct stages of the reconstruction process, I, II and III, highlighted by the gray dashed lines. A direct insight into the population of reconstructions at generations 5, 20, 40 and 100 is given in (**b**). The first row reports the average reconstruction, indicated as ρ^{avg}, while the second row shows the reconstruction with the lowest error ρ^{best}. The third row is the reconstruction at the 50$^{\text{th}}$ percentile of the errors distribution, while the last row corresponds to the reconstruction with the highest error ρ^{worst}. Please refer to the main text for details.
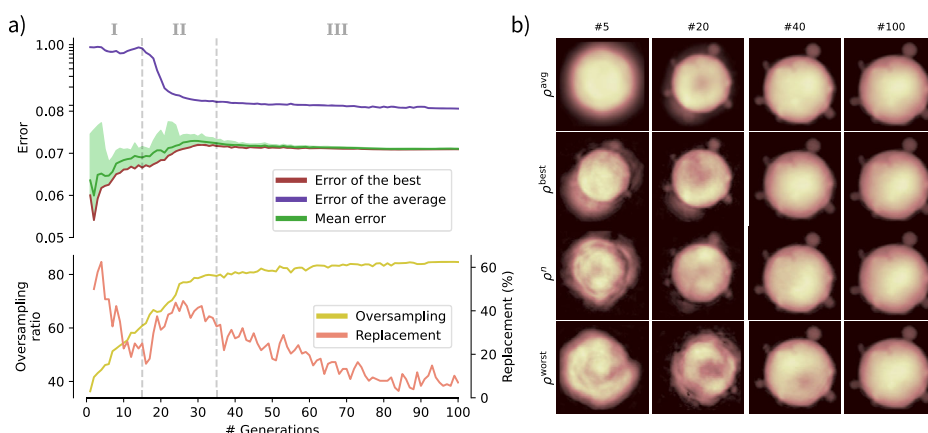
- The algorithm's performance is dependent on the settings and the initialization of the densities. Among those, the ones that define the starting support size and how the support is upgraded during the process are of particular relevance for single-shot XFEL data, where thousands of samples of unknown size and shape are imaged with different brightness and noise.
- Scattering information is missing in some regions of the detector, in particular in the central part due to the presence of a hole to avoid damage from the transmitted XFEL beam (see the patterns in Fig. 2). This makes algorithms particularly unstable, it renders reconstructions more prone to artifacts for increasing sample size and it limits in practice the field of view of the CDI method.

After a general description of SPRING's behavior in a typical reconstruction procedure, the following subsections challenge the SPRING framework on these aspects and compare its performance to conventional approaches.

Monitoring the reconstruction process

This section is dedicated to an overall description of a typical reconstruction process with SPRING, summarized in Fig. 3. The behavior of these indicators as a function of the algorithm generations for a typical phase retrieval routine are shown in Fig. 3a. Figure 3b reports, instead, insights into the population of reconstructions at the different steps. The reconstruction has been performed by using the HIO algorithm in the *self-improvement* step of MPR (see section "The self-improvement"), with a starting number of iterations $\jmath = 20$ (see section "Changing values over generations") and a starting support size $\Sigma^{\text{init}} = 60$ pixel (see section "Initialization of the starting

densities"). The other parameters are kept to their default value indicated in Table 1.

The upper side of Fig. 3a reports the error value. For a given reconstruction $\mathcal{R}$, its value is calculated as it follows:

$$E[\mathcal{R}] = \sqrt{\frac{\sum_{ij}\Xi_{ij} \cdot \left(\left|\mathcal{F}\left[\mathcal{R} \triangleright S \cdot \mathcal{R} \triangleright \rho\right]_{ij}\right| - M_{ij}\right)^2}{\sum_{ij}\Xi_{ij} \cdot M_{ij}^2}} \qquad (3)$$

where Ξ_{ij} is the mask function, whose value is $\Xi_{ij} = 1$ where the intensity values of M_{ij} are known, and $\Xi_{ij} = 0$ otherwise. The operation $\mathcal{R} \triangleright S \cdot \mathcal{R} \triangleright \rho$ operation is the element-wise multiplication between the support matrix $\mathcal{R} \triangleright S_{ij}$ and the density matrix $\mathcal{R} \triangleright \rho_{ij}$. The result of this operation is a density perfectly compatible with the *support constraint*, i.e., with zero values outside the support function (see also section "The phase problem as an optimization problem" for further discussion).

Equation (3) calculates the difference between the retrieved Fourier amplitude and the experimental one, $M_{ij} = \sqrt{T_{ij}}$. It is worth noting that this value is normalized with the L^2 norm of the experimental modulus M_{ij}. On the one hand, this does not influence the reconstruction process, as this normalization factor is a constant value for a given diffraction pattern. On the other hand, this normalization allows to obtain error values that can be interpreted as *relative deviations* and that have similar order of magnitude (often between 10^{-1} and 10^{-2}) across different experimental patterns.

Different error values are reported in Fig. 3a. The red line represents the error of the best reconstruction $\mathcal{R}^{\text{best}}$, i.e. the one with the lowest error value. The region indicated in light green covers the region between the best

reconstruction and the worst one, i.e., the one with highest error value. The line marked in dark green is, instead, the mean value of the error across the whole population. A further error value is reported in blue, corresponding to that of the average reconstruction. The average reconstruction $\mathcal{R}^{avg}$ is computed as it follows:

$$\mathcal{R}^{avg} \triangleright \rho_{ij} = \frac{\sum_{p=0}^{P} \mathcal{R}^p \triangleright \rho_{ij}}{P}$$
$$\mathcal{R}^{avg} \triangleright S_{ij} = \frac{\vee_{p=0}^{P} \mathcal{R}^p \triangleright S_{ij}}{P}$$
(4)

where the operation $\vee_{p=0}^{P}$ indicates the logical OR operation. In practice, the support function $\mathcal{R}^{avg} \triangleright S$ of the average reconstruction is given by the union of all supports of the population of reconstructions $\{\mathcal{R}^p\}$.

The lower plot in Fig. 3a reports two additional quantities. One is termed *oversampling*, and gives information about the size of the support function of the best reconstruction. It is calculated via the formula:

$$O_d = \frac{N_p^2}{\sum_{ij} \mathcal{R}^{best} \triangleright S_{ij}}$$
(5)

which expresses the ratio between the total number of entries in the matrix (of shape $N_p \times N_p$) and the actual extension of the support function. For example, $O_d = 60$ means that the extension of the support function is a 60th of the overall matrix. Higher values of O_d correspond to smaller supports, and vice versa.

The last quantity shown in Fig. 3a is the *replacement factor*, and gives indications about the effectiveness of the selection step (see section "The selection step"). For each generation, the *selection* operation compares each individual of the original population of candidate solutions $\{\mathcal{R}^p\}$ with the corresponding one in the new population $\{\mathcal{R}_{new}^p\}$, with the latter replacing the former if it reached a lower error value. The replacement factor reports how many new reconstructions have performed better than the corresponding ones in the old population and replaced them. A value of 0 % means that no replacement happened, i.e. the new reconstructions $\{\mathcal{R}_{new}^p\}$ performed systematically worse than the parents $\{\mathcal{R}^p\}$. Instead, a value of 100% implies that all the new reconstructions performed better and replaced the original ones.

From the behavior of the aforementioned performance indicators it is convenient to identify three different stages, highlighted as I, II and III in the upper part of Fig. 3a.

Stage I: Shape identification. In this first part of the reconstruction procedure (up to generation 15 in the example reported in 3a) the algorithm tries to identify the correct spatial extension of the sample. Typically, the starting support function is larger than the final correct one, and its size is reduced via the action of the SW algorithm, as indicated by the increasing value of the oversampling ratio reported as a yellow line in Fig. 3a. The shrinking of the support is strictly correlated to an increase in the reconstructions errors, highlighted by the red and green lines in Fig. 3a. This behavior highlights well how the size of the support function influences the error value, and clarifies why a comparison of the error values is only meaningful between reconstructions with the same support size (see section "The modified Shrink-wrap"). In this phase, when the correct sample shape is not yet identified, the error of the average reconstruction (blue line in Fig. 3a) is close to 1 and constant, as the average is calculated on widely differing spatial distributions. This aspect is directly visible in the first column of Fig. 3b, which reports a snapshot of the reconstructions at generation 5. There, the best reconstruction (second row) is already outlining a density distribution resembling the correct one, while the rest of the population still have very differing shapes. As a consequence, the average reconstruction in the first row has no clear features. In this first stage, the replacement factor (orange line in Fig. 3a) drops from a high starting value of around 50%. At the beginning,

the reconstructions are so far away from the solution that the two populations $\{\mathcal{R}^p\}$ and $\{\mathcal{R}_{new}^p\}$ are substantially equivalent, carrying to equal probability of surviving to the next generation. As the support is shrinking it becomes less likely that $\mathcal{R}_{new}^p$ replaces $\mathcal{R}^p$. The condition where the replacement factor is below 50 % but significantly above 0% indicates that the *genetic operators* are effective.

Stage II: Shape definition. This second phase of the reconstruction process (from generation 15 to generation 35 in Fig. 3a) is characterized by the convergence of the population towards the correct sample shape. While the support function is still shrinking (see the oversampling behavior in Fig. 3a) and, consequently, the error values of the reconstructions are still growing (see the error of the best) similarly to Stage I, there are clear signatures of a different behavior. First, the mean error is getting closer to the error of the best and the overall error distribution (light green area in Fig. 3a) is shrinking. This is accompanied by a rapid increase in the replacement factor. However, the clearest signature that identifies this stage comes from the error of the average (blue line in Fig. 3a), characterized by a sudden drop. These are signs that the algorithm has identified the correct sample shape and this information is quickly propagating across the population via the crossover and selection operators. It is possible to directly recognize this dynamics in the second column of Fig. 3b, which reports the status of the population at generation number 20. Here, most of the reconstructions are resembling the correct sample shape (second and third row) and, as a consequence, the same happens to the average reconstruction (first row).

Stage III: Density refinement. Once the sample shape (i.e. the support function) has been identified, the algorithm focuses on the correct retrieval of the density values (from generation 35 in the example reported in Fig. 3a). Here, the oversampling is still slowly rising, thanks to both a further refinement of the support function and the decrease of the σ parameter of the SW algorithm over the generations. However, in this phase, the reduction of the support size is correlated to a slow decrease of the error value (differently from the previous stages I and II) both for the individual reconstructions and for the average. This behavior is due to both the action of the crossover and selection operations that promote the reconstructions with lower error, as well as the decreasing number of IAs iterations (HIO) and the corresponding increase of ER iterations (which naturally tends to reach lower errors). The dynamics of the reconstructions population is recognizable in Fig. 3b, which gives an overview on the reconstructions at generation 40 (third column) and 100 (last generation, fourth column), corresponding to the beginning and the end of Stage III, respectively.

Comparison with the conventional approach
The convergence properties of standard algorithms, i.e., their ability to identify the solution, have a relevant dependence on the input parameters. Those that involve the support function, like its initial size and the threshold value τ for the SW algorithm, are often problematic and strongly influence the convergence to a solution.

This aspect is often a problem when dealing with experimental data because, even within the same dataset, samples typically have different sizes and diffraction patterns are characterized by different brightness (which, in turn, implies a different amount of noise). Human input is frequently required to adjust these parameters almost on an shot-to-shot basis.

Figure 4 reports a comparison between the outcome of a conventional use of iterative algorithms (IAs) and the MPR method implemented in SPRING. In this section we refer to the conventional approach with the term IPR. The conventional IPR approach is obtained by "switching off" the *crossover* and *selection* operations of the MPR framework. In this way, each reconstruction is completely independent and only undergoes the alternated execution of iterative algorithms and the support update with the standard SW algorithm[19]. The population size is set $P = 128$ for MPR, while it is doubled to $P = 256$ for the conventional IPR to make the comparison more fair in terms of computational cost and time to solution. In fact, MPR executes standard IAs on two sets of reconstructions, the original $\{\mathcal{R}^p\}$ and
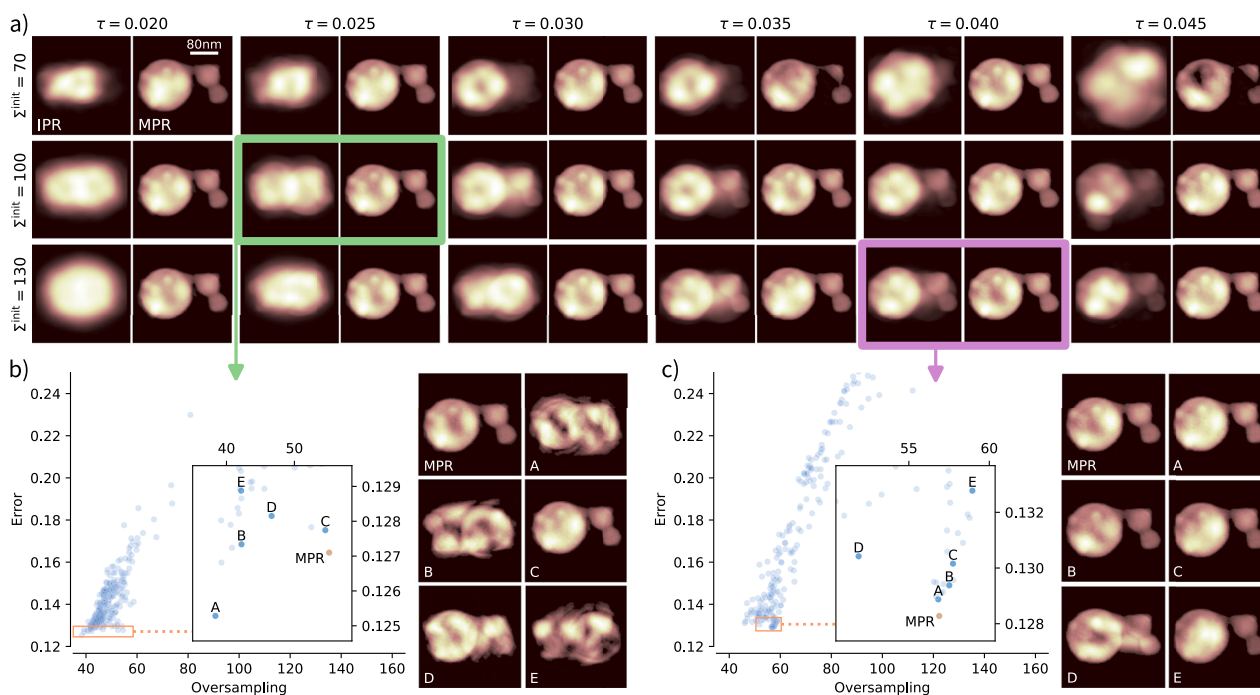
Fig. 4 | Comparison between the conventional use of iterative algorithms, here indicated as IPR, and the SPRING implementation of the MPR method. The diffraction data belongs to Dataset A. In (**a**), the average reconstruction at the end of the retrieval process is shown for both IPR and MPR, as a function of the threshold value τ for the support update via the Shrink-wrap algorithm and the size of the starting support function Σ^{init}. Individual reconstructions are inspected in detail for the cases $(\tau, \Sigma^{init}) = (0.025, 100)$ and $(0.04, 130)$ in sub-figures (**b, c**), respectively. The plots in (**b**) report the oversampling degree (on the x axis, inversely proportional to the retrieved support size) and the error value (y axis) for all 256 individual reconstructions obtained with the IPR method. The inset graph shows a zoom of the main plot in the region around the IPR reconstructions with the lowest error value. In addition, the red dot indicates the error and oversampling value for the best reconstruction in the MPR population. On the right side of the plots in b) and c), six different reconstructions are shown. The best of the MPR population is compared with the best of the IPR population (marked as A) and other four reconstructions (B to E) in the vicinity of A.

the new $\{\mathcal{R}^p_{new}\}$ created by the *crossover* operation, doubling in practice the number of optimized reconstructions. The initial number of iterations of IAs is set to 80, alternated with 20 iterations of Error Reduction algorithm.

Both IPR and MPR are tested against different values of the initial support size Σ^{init} (expressed in pixels, see section "Summary of the parameters and further considerations") and the SW algorithm threshold τ^{19} for its update. The retrieved densities reported in Fig. 4a are calculated by averaging the densities of the whole population $\{\mathcal{R}^p\}$. While the average is a meaningful reconstruction output for MPR, this is not the case for the IPR, for which only a fraction of the reconstruction processes actually converge to the right solution. Therefore, a visual evaluation of the IPR average is a qualitative indicator on how many reconstructions in the population have reached convergence. Figure 4a shows how MPR can reliably retrieve the sample density across most of the parameters range, with incorrect reconstructions only appearing for higher τ values (0.035 and 0.045) and $\Sigma^{init} = 70$ (i.e., a starting support size 30% lower than the actual maximum extension of the sample of around 100 pixel). This example well underlines the resilience of MPR with diverse parameter settings.

While no convergence is reached for the conventional IPR at the lowest threshold value $\tau = 0.02$, the correct sample shape can be glimpsed at $\tau = 0.025$ for $\Sigma^{init} = 100$. The IPR convergence improves for higher threshold values, up to an optimum between $\tau = 0.03$ and $\tau = 0.035$ for $\Sigma^{init} = 100$, and $\tau = 0.035$ to $\tau = 0.04$ for $\Sigma^{init} = 130$. IPR doesn't appear to reach a significant amount of converged reconstructions for the smallest starting support size $\Sigma^{init} = 70$ at any τ value.

Figure 4b is a detailed inspection of the individual IPR outcomes for the parameter combinations $(\tau, \Sigma^{init}) = (0.025, 100)$, i.e. the earliest occurrence of sub-optimal values where IPR reaches the correct solution for some individual reconstructions. Here, the final error of the reconstructions obtained with the IPR method (see Eq. (3)) is reported against the oversampling degree given by the reconstructed support function (see Eq. (5)). Each blue point corresponds to a different reconstruction $\mathcal{R}^p$ in the population of size $P = 256$. The overall distribution of the points clearly shows the strong correlation between the size of the reconstructed support and the error value. The inset plot is a zoom of the low-error region. Some points are marked with letters A-E, and the corresponding retrieved densities are reported on the right side. The reconstruction marked as "A" correspond to the one with the lowest error, while the others are chosen in its neighborhood. Additionally, the best result in the MPR population is reported as a red dot in the inset graph, with the corresponding density shown on the right side of the sub-figure. A first important consideration concerns the best IPR reconstruction "A". Despite having reached the lowest error, the corresponding density is clearly far off from the correct solution. This is caused by a larger retrieved support (i.e., lower oversampling value), and highlights why, in MPR, reconstruction errors are only compared by constraining the same oversampling value for the whole reconstruction population. See section "The modified Shrink-wrap" for further details and considerations. Furthermore, it appears that the best density is the one marked as "C", unsurprisingly the closest to the best MPR density (red dot) but only the fifth in terms of error ranking. This well underlines why, in general, identifying reconstructions that have reached convergence in an automated manner is extremely challenging. Reconstruction "C" and the best of MPR have very similar oversampling degree (i.e., their error can be compared) and the error of the MPR best is lower. This is an indication that, once that the correct oversampling value is identified, MPR can reach a deeper minimum of the error landscape. The statistical evaluation of the final outcomes as function of the oversampling degree and the error value presented in Fig. 4b is provided to give an intuitive understanding. A less intuitive but more refined metric for the identification of the converged reconstructions is proposed and discussed in ref. 39.

A similar consideration applies to Fig. 4c, where the same overview is reported instead for τ and Σ^{init} in the optimal configuration for the

Fig. 5 | Dependence of the reconstruction result on the density initialization. Six reconstruction procedures, with population size $P = 256$ for the conventional IPR and $P = 128$ for the SPRING implementation of MPR, are performed on a diffraction pattern selected from dataset B. The random starting densities are created by setting a different *seed* for the random number generator for each of the six cases, to make the procedures statistically independent between each others. The error and oversampling value reached by the best reconstruction in the population for each of the six attempts is reported in (**a**), for both the conventional IPR (blue dots) and MPR (red dots). The corresponding densities are shown in (**b**) for IPR and (**c**) for MPR.
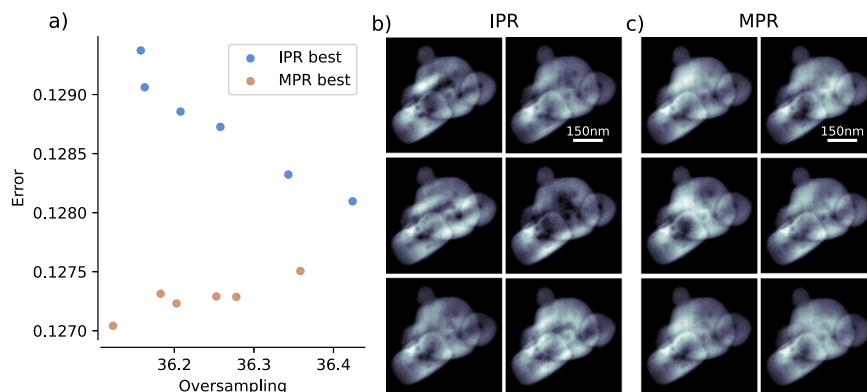
conventional IPR method, corresponding to 0.04 and 130. The higher number of converged reconstructions for the conventional IPR method is evident by visually inspecting the the densities reported on the right side of Fig. 4c. Differently from Fig. 4b, the best IPR reconstruction here corresponds to the one with the lowest error value ("A"), as the correct oversampling value has been identified. All of the other reconstructions right above "A" in the inset graph, i.e., those with lower error values and similar oversampling, have reached a proper convergence. Still, some outliers with low error but too low oversampling appear, as exemplified by the reconstruction "D". Again, the best MPR reconstruction, reported as a red dot in the plot, has a lower error value than "A", meaning that a better optimization is done by MPR when the correct support is retrieved. Despite the optimal τ and Σ^{init} for the standard IPR have been used, still only around 10 % of the 256 reconstructions have reached a neighborhood of the correct solution to the phase problem, and their automatized identification remains challenging. Further tests on the stability of SPRING with different algorithm parameters are given in Supplementary section 2.

In both of the reconstruction attempts investigated in Fig. 4b, c the error of the best MPR density is lower than the one of the best reconstruction for IPR. This property is analyzed in more details in Fig. 5. Here, six full reconstruction procedures, each involving a population of size $P = 128$ reconstructions for MPR and size $P = 256$ for IPR, have been performed. In SPRING, densities are initialized by filling the real-space matrix with a number of spherical density profiles, randomly defined in size and position, as described in section "Initialization of the starting densities". In this test, a different *seed* of the random number generator has been set for each attempt to create different, statistically independent, sets of starting conditions for each reconstruction attempt.

The algorithms parameters have been optimized to ensure the convergence of the conventional IPR method, corresponding to an initial support size $\Sigma^{init} = 130$, a threshold value for the support update $\tau = 0.035$ and an initial number of IAs iterations equal to 80 (alternated with 20 iterations of ER algorithm, see also section "Summary of the parameters and further considerations").

Figure 5a reports the error value and oversampling reached by the best reconstruction in the population $\{\mathcal{R}^p\}$ for each of the six reconstruction attempts. Blue dots indicate the values for the conventional IPR, while red dots correspond to the MPR outcomes. The independent MPR results appear to be systematically better than the ones of the conventional IPR, i.e., the best MPR reconstruction systematically reaches a lower error than the best IPR reconstruction.

The six best densities from the independent reconstruction processes are visually reported in Fig. 5b for the conventional IPR and in Fig. 5c for the SPRING implementation of MPR. It can be observed that the higher values of the IPR errors are connected with a higher variability in the reconstructed densities in Fig. 5b. The MPR results in Fig. 5c are, instead, much more similar despite being statistically independent reconstructions. This observation is a further sign that the MPR method is capable of reaching a spatial density closer to the *global optimum* than the conventional IPR approach.

The tests presented in Figs. 4 and 5 are not exhaustive and cannot be considered a thorough evaluation of the performances of the two approaches, the conventional IPR and the SPRING implementation of MPR. Smaller samples with higher oversampling degree, or brighter diffraction patterns, or different number of IAs iterations may shrink or widen the performance gap between the two methods. On the other hand, no fine-tuning of the MPR parameters have been performed for this test, and all parameters not explicitly mentioned in this section have been kept to their default value (see section "Summary of the parameters and further considerations"). We encourage the reader to perform further, more specific, tests by directly using the open-source *spring* Python package.

Imaging extended samples

The large spatial extension of the sample, corresponding to particularly low oversampling values O_d (see Eq. (5)) renders the phase retrieval problem more challenging as the number of unknowns to be retrieved (i.e. the pixel values contained in the support function) is higher. This is particularly true when scattering signal is missing in the center of the pattern (as visible in Fig. 2), which is an unavoidable feature of XFEL data.

The lack of data in the center, corresponding to low momentum transfer, causes the arising of *loosely constrained modes*[61], i.e., density fluctuations in the reconstructions at low resolution. Those fluctuations have a larger and larger impact on the reconstruction quality with increasing sample size, up to the point where, for particularly low oversampling degrees, the reconstruction attempts are unsuccessful. The SPRING implementation of MPR includes a particular strategy to deal with these *loosely constrained modes*, whose details are given in section "Missing intensities at low scattering angles".

Among those modes, the Gaussian mode is typically the most prominent (see ref. 61), and its strongest effect is the uncertainty that it introduces in the overall density normalization. Thus, it becomes more challenging to correctly retrieve for samples with larger spatial extension and the success of a reconstruction depends more strongly on the initial density normalization. This fact limits in practice the field of view of the CDI technique. Here we show how the MPR method is resilient to different normalization values for the starting densities.

SPRING creates starting guesses by filling the space with randomly distributed spheres. The density profile of the spheres can be tuned with a parameter γ^{init}, which acts similarly to the *gamma value* in image processing: the closer γ^{init} to 0, the flatter the density profiles. The starting densities are then normalized by comparing their L^2 norm in Fourier space with the experimental data, calculated only where intensities are experimentally recorded. Different values for γ^{init} thus have a strong influence on the initial normalization of the densities. See section "Initialization of the starting densities" for a detailed description of the densities initialization implemented in SPRING.

The implications of different values of γ^{init} on the final outcome of the SPRING reconstructions are reported in Fig. 6. A cutout taken from the middle of the diffraction pattern treated in this example is shown in Fig. 6a.
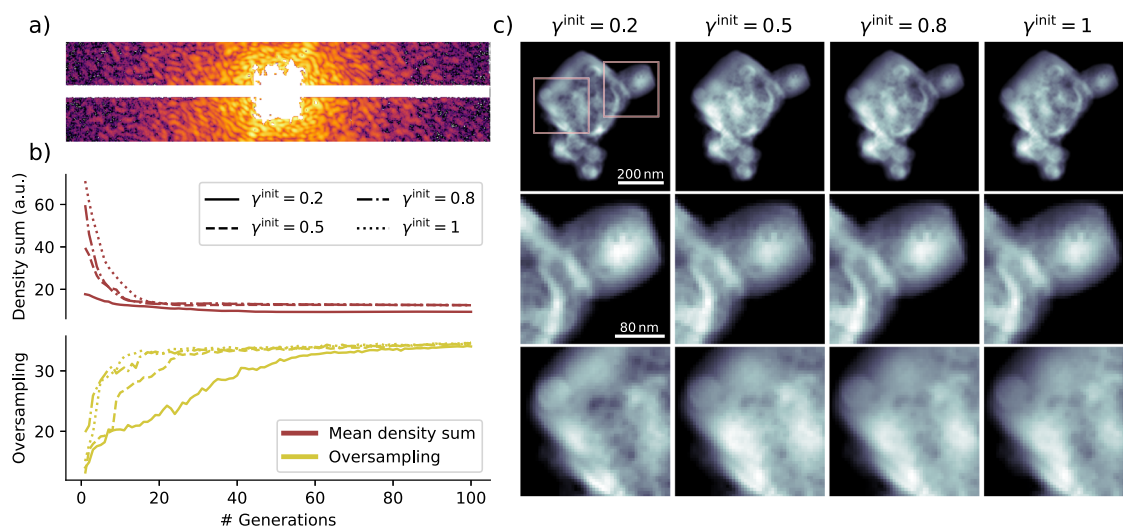
Fig. 6 | Effects of the gamma value of the starting guesses γ^{init} on the phase retrieval outcome. This test is performed on a diffraction image from Dataset B. In (**a**), a cutout of the diffraction data is reported. Missing intensities are now present also in those regions where the detector's pixels reached saturation. In (**b**), the sum of the reconstructed density and the oversampling are reported as a function of the generation for different values of γ^{init}. The reconstruction outcomes are compared in (**c**), where the full reconstructed densities are shown in the first row and two closeups of different regions are reported in the second and third rows.

This pattern shows strong scattering, with photons recorded up to the edge of the image. Due to the limited dynamic range of the pnCCD detector, there is a saturated region towards the center. The dependence of the initial densities normalization on the γ^{init} value can be recognized in the initial value (generation number 0) of the mean density sum reported as a red line in Fig. 6b. This initial value ranges from 18 for the lowest value $\gamma^{init} = 0.2$ to 70 for the highest value $\gamma^{init} = 1$.

The outcomes of the reconstruction procedures are reported in Fig. 6c. High-resolution details are coherently reconstructed among the three cases, as highlighted by the enlarged details in the second and third row. However, the overall reconstructions at the first row reveal differences in the low-resolution brightness profile. While these differences are barely perceivable for γ^{init} equal to 0.5, 0.8 and 1, they are clearly visible in the case of $\gamma^{init} = 0.2$. This is confirmed by observing the behavior of the mean density sum (red line in Fig. 6b), whose final value settles around 10 % lower than the other three cases.

In addition to the intrinsic uncertainty on the density profile that derives from the presence of loosely constrained modes, the lower value for $\gamma^{init} = 0.2$ could be partially explained as being due to the algorithm struggling during the *shape identification* stage (see section "Monitoring the reconstruction process"). This behavior can be discerned by monitoring the value of the oversampling parameter along the reconstruction process, reported as a yellow curve in Fig. 6b. For the higher values of γ^{init} the correct shape is quickly identified and, as a consequence, the oversampling value reaches its final value of 35 within the first 20 generations. In the lowest gamma value case (solid line) the convergence towards the correct shape requires around 60 generations. The long-lasting larger support for $\gamma^{init} = 0.2$ implies the presence of additional and stronger minima of the error function that the algorithm had to overcome during the retrieval (see Fig. 9). In addition, the algorithm had less "time" to optimize the reconstruction once that the correct shape is finally identified. This slow convergence is mostly caused by a threshold value $\tau = 0.03$ of the SW algorithm (see Table 1) set too low for the case of $\gamma^{init} = 0.2$. However, the capability to converge in all the reported cases, with strongly different density normalizations, by using the same set of algorithm's parameters (and especially those concerning the support update) is a further proof of the resilience of SPRING.

A further demanding task for imaging algorithms is dealing with low-brightness diffraction data. In such cases, iterative algorithms tend to become unstable, especially when dealing with large samples, and the extraction of high-resolution features of the sample becomes particularly challenging. In Supplementary section 3.1 SPRING is tested and compared with conventional methods on this kind of diffraction data.

Complex-valued density

The refractive index of materials in the X-ray regime, and in particular its real part, can be fairly approximated to 1[62]. The deviations from 1 are often expressed in terms of $\delta = n - 1$ where $\delta \lesssim 10^{-2}$.

However, the situation changes when the sample is composed of different materials with substantially different δ values at the probing photon energy. In this case, different phase shifts are induced in the light depending on the position in the sample, creating a phase contrast in the scattered field[18,63]. This phase contrast affects not only the phases of the far field, but reshapes also the intensity distribution. CDI is, thus, inherently sensitive to phase contrast in the sample. There are particular wavelengths where the small δ approximation does not hold, namely at photon energies in the vicinity of the core-excitation energies of atoms (a resonant condition)[13].

In this regime, the δ coefficient for a specific element has a sharp fluctuation, thus increasing its phase contrast with other non-resonant materials in the sample. This is the case for mixed xenon-argon clusters imaged at 700eV, close to the 3d electronic resonance of xenon. An example of a diffraction pattern acquired in these conditions is reported in Fig. 7a. In this close-up view of the central part of the pattern it is already possible to recognize non-centrosymmetric features, which are a sign of different phase shifts induced in the field by the sample. Those phase shifts are then reconstructed in real space as complex-valued densities.

When small phase shifts ($\ll \pi$) are expected in the reconstructed data, the positivity of the real part of the density is constrained along with the support function (see section "Real-space constraints"). This is achieved in SPRING by setting the χ parameter to 0.5 (see Table 1). Furthermore, it is convenient to initialize the densities as real-valued distributions. This is achieved by setting the SPRING parameter $\phi^{init} = 0$ (see section "Initialization of the starting densities").

Instead, when relevant phase shifts are expected in the reconstruction, the phase retrieval problem becomes more challenging. First of all, the *positivity* constraint of the real part of the reconstruction cannot be enforced[35]. This means that the χ SPRING's parameter has to be set to 0. Furthermore, fully real-valued starting densities, initialized in SPRING by setting $\phi^{init} = 0$, are sub-optimal starting conditions.

Figure 7 reports the outcome of three MPR reconstructions with $\phi^{init} = 0$ (completely positive real-valued initialization), $\phi^{init} = 0.25$ (real-space

Fig. 7 | Effects of different phase values assigned to the starting guesses on the reconstruction outcomes for different values of ϕ^{init}. In (**a**) a cutout of the experimental data, extracted from Dataset A, shows asymmetries in the intensity distribution, which is a sign of phase contrast induced by the sample density. In (**b**) the complexity indicator Γ is plotted as function of the number of generations, both reporting its mean value within the population of reconstructions (green) and its value calculated on the average reconstruction (purple). In (**c**) the final reconstruction outcome is reported for the three cases, with the phase information encoded in the color map shown on the side.
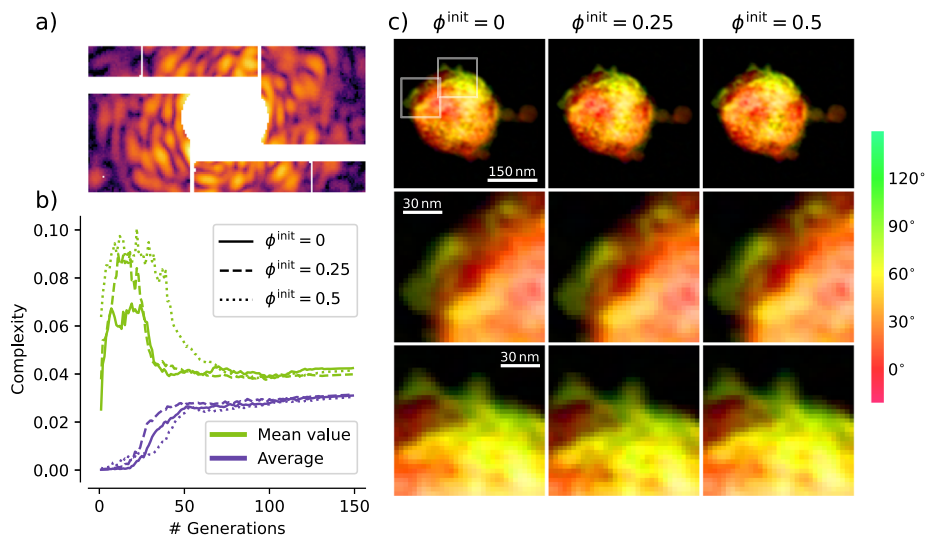
starting phases range from $-\frac{\pi}{4}$ to $\frac{\pi}{4}$) and $\phi^{init} = 0.5$ (real-space starting phases range from $-\frac{\pi}{2}$ to $\frac{\pi}{2}$). In all three reconstructions, *unconstrained modes* are controlled by setting $C_a = 0.3$ (see section "Missing intensities at low scattering angles").

The amount of phase shift in the reconstruction as function of the generations is tracked by the *complexity* indicator Γ (see Eq. (20)), with values $0 \le \Gamma \le 1$. The value $\Gamma = 0$ indicates a completely real-valued density. Its behavior as function of the generations is reported in Fig. 7b. See section "Dealing with missing intensities" for further details and discussion.

Considering the mean value of the complexity in the population (green line), its initial value can be directly correlated to the range of the starting phases ϕ^{init}, with higher values of the latter corresponding to greater values of the former. However, after an initial rise in the complexity value for all the three cases during the first generations of the algorithm, its value quickly drops and settles during the *shape definition* stage at around 0.045, consistently for the three different settings of ϕ^{init}. The same consistency can be observed for the complexity value calculated on the average reconstruction (blue line in Fig. 7b). Here, however, the complexity indicator stays close to 0 as long as the *shape definition* stage is reached, as widely differing densities in the population, despite being potentially strongly complex-valued, cancel out their phase deviations in the averaging operation.

This consistency is directly reflected into the three reconstruction outcomes shown in Fig. 7c. The overall density distribution and the respective phase values are fully matching. This is also true for the high resolution details highlighted by the zoomed-in cutouts of the reconstructions shown in the second and third row. It is worth mentioning that, for this diffraction pattern, the conventional IPR method couldn't reach proper convergence, as discussed in Supplementary Fig. 5.

Thus, in most use-cases, starting densities can be initialized as completely real-valued ($\phi^{init} = 0$), as different values of ϕ^{init} don't carry significant improvements on the final reconstruction outcome. This is a further demonstration of the stability of the MPR approach, which is capable of converging to the same solution with the same algorithm settings, but starting from very different, theoretically sub-optimal, conditions.

Discussion

We have presented an algorithmic framework to deal with the phase retrieval problem in single-particle single-shot Coherent Diffraction Imaging, called SPRING. SPRING implements the Memetic Phase Retrieval (MPR)[43] method, which combines the widespread iterative phase retrieval algorithms with operations inherited from Genetic Algorithms tailored for XUV and X-ray Free-Electron Laser (XFEL) diffraction data.

The performance of SPRING has been investigated and compared with standard methods on data acquired in different experimental contexts with the two most common types of scattering detectors at XFELs. For diffraction patterns acquired in optimal conditions, the quality of the imaging result is consistent for a wide range of algorithm settings, virtually removing the case-by-case tuning required by conventional approaches. Furthermore, its limits are tested for situations that are typically challenging or untreatable for standard data analysis workflows, including large sample sizes, large regions of missing scattering intensities and starting conditions significantly far from the ideal ones. Overall, we show how SPRING is remarkably robust against variations of the settings, consistent in terms of quality of the imaging result, and flexible with respect to the type and characteristics of the diffraction data.

Those features are a game-changer for single-particle experiments at XFEL facilities, where the acquisition conditions, like the brightness of the diffraction data and the sample size, are often sub-optimal, not completely under control and vary on a shot-to-shot basis. In fact, the low sensitivity of SPRING to the algorithm parameters makes it possible to reconstruct large fractions of a typical dataset with the same algorithm settings. This is a pivotal aspect for CDI at XFELs, where analyzable diffraction images are typically recorded during an experiment at a rate from one to a few patterns per second (depending on the hitrate, the repetition rate of the XFEL and the frame-rate of the detector), leaving hundreds of thousands of bright patterns to be analyzed at the end of the campaign.

While the core structure of SPRING is established, further developments and enhanced performances of the approach are expected in the future. Among the foreseen improvements, the main ones involve the implementation of additional and more performant iterative phase retrieval algorithms and more sophisticated strategies to deal with noisy and missing diffraction data. Furthermore, the MPR algorithm is fully compatible with an extension to three-dimensional diffraction datasets[44]: its implementation in SPRING and the corresponding performance will be investigated in detail in future works.

A Python module, called *spring*, is released as open-source software in the context of this publication. The *spring* package implements a highly optimized version of the algorithm, as well as additional helper functions to aid scientists in dealing with the preparation of experimental data and the imaging results.

Currently, CDI experiments at XFELs are successfully accomplished mainly with the involvement of highly experienced research groups, due to the technical challenges to address in the experiment as well as due to the intricate and complex procedures required for data analysis and imaging. Thanks to its enhanced reliability and the ease of use of its software implementation, SPRING will play a main role in easing the data analysis process and making imaging experiments at XFELs more accessible to a wider, less specialized and more applied scientific community.

Methods

The phase problem as an optimization problem

Starting from any spatial density $\rho(\vec{x})$, a new density $\rho_S(\vec{x})$ that is fully contained in the support function $S(x, y)$ can be created by the following operator P_S:

$$\rho_S = P_S\rho = S \cdot \rho \qquad (6)$$

where $\cdot$ indicates the element-wise matrix multiplication, and the dependence on the coordinates x and y has been omitted for the sake of brevity.

Similarly, it is possible to produce a density ρ_M, whose Fourier modulus is equal to the square root of the experimental data, by applying the following operator P_M to any given density ρ:

$$\rho_M = P_M\rho = \mathcal{F}^{-1}\left[M \cdot \frac{\tilde{\rho}}{|\tilde{\rho}|}\right] \qquad (7)$$

where $\mathcal{F}^{-1}$ is the inverse FT operation, M is the square root of the experimental intensities I, and $\tilde{\rho} = \mathcal{F}[\rho]$. The two operators in Eq. (6) and Eq. (7) can be easily demonstrated to be projectors, as $P_M{}^2 = P_M$ and $P_S{}^2 = P_S$, and they are usually addressed as *modulus projector* and *support projector*[18,30,35].

The solution to the phase problem can be written in terms of P_M and P_S as the density function ρ_{sol} that satisfies both constraints (i.e. $P_M\rho_{sol} = \rho_{sol}$ and $P_S\rho_{sol} = \rho_{sol}$). It is convenient to define the error function:

$$E_{S,M}[\rho] = \| M - \mathcal{F}[P_S\rho] \| \qquad (8)$$

where $\| \cdot \|$ denotes the L^2 norm. Eq. (8) indicates how *far* a density ρ, projected on the support function, is from the experimental Fourier amplitudes M. In theory, the solution to the phase problem ρ_{sol} has error $E_{S,M}[\rho_{sol}] = 0$. In practice, due to intrinsic noise in the experimental data, the error E in Eq. (8) cannot reach 0, and the solution ρ_{sol} has to be defined as the one that minimizes the error, i.e.:

$$\rho_{sol} := \left\{ \rho : E_{S,M}[\rho] = \min_\rho E_{S,M} \right\} \qquad (9)$$

Equation (9) expresses the phase retrieval problem as an *optimization problem*. Phase retrieval algorithms are, thus, *optimization algorithms*, which aim at minimizing the *optimization target*, i.e. the error value, defined in Eq. (8).

The conceptually simplest phase retrieval algorithm, Error Reduction (ER)[37], acts by applying in an alternate manner the two projection operators in Eq. (6) and Eq. (7). A single *iteration* of the ER algorithm that constrains a given support function S and experimental moduli M can be thus defined as $\mathrm{ER} = P_M P_S$. Its action on a density ρ gives as a result a new density $\rho' = P_M P_S\rho$.

The ER algorithm can be demonstrated to be a *steepest-descent* method for the error E because the error value is always reduced with iterations, i.e., $E[\mathrm{ER}^{i+1}\rho] \leq E[\mathrm{ER}^i\rho]$. This renders the ER algorithm particularly *stable*, because it is guaranteed to converge to an optimum of the error value. It has however the drawback of getting stuck in local minima.

Memetic operators

This section describes in a detailed way the SPRING implementation of the different operations performed by the MPR approach (see the overview in Fig. 1). The core of MPR[43] is built around three operators inherited from *Memetic algorithms*[56]. The *crossover*, responsible for the creation of a new population of candidate solutions from the existing one, is discussed in section "Crossover". The *self-improvement*, where conventional iterative phase retrieval algorithms are executed for each reconstruction in the population, is presented in section "The self-improvement". The *Selection*, where only the reconstructions with the lowest error value are selected to survive to the next algorithm generation, is described in section "The selection step". Details on the initialization of the starting reconstructions is given in section "Initialization of the starting densities". An overview of the parameters that can be tuned in MPR is given in section "Summary of the parameters and further considerations", followed by further considerations and implementation details on less central but still relevant operations implemented in the method.

In the next sections the following formalism is used. A reconstruction is indicated as $\mathcal{R}$, which comprises the complex-valued matrix $\mathcal{R} \triangleright \rho_{ij}$ containing the density values and the boolean-valued matrix $\mathcal{R} \triangleright S_{ij}$ that defines the corresponding support function. The Fourier Transformation (FT) of the density matrix is indicated as $\mathcal{R} \triangleright \tilde{\rho}_{ij}$. The density ρ and the support S have dimension $N_p \times N_p$, equal to the size of the matrix containing the experimentally measured intensities I_{ij} and the corresponding experimental modulus $M_{ij} = \sqrt{I_{ij}}$.

A population of P reconstructions is reported as $\{\mathcal{R}^p\}$, with $p = 1, \ldots, P$. The density values of the reconstruction with index p in the population is thus indicated as $\mathcal{R}^p \triangleright \rho_{ij}$.

Algorithm 4. The crossover routine

1: **function** CROSSOVER($\{\mathcal{R}^p\}$)
2:   $\{\mathcal{R}^p_{new}\} \leftarrow$ COPY($\{\mathcal{R}^p\}$)   ▷ Initialize the new population of reconstructions $\{\mathcal{R}^p_{new}\}$ as a copy of the old one.

3:   **for** $p = 1, \ldots, P$ **do**   ▷ Loop on the single individuals of the new population
4:     $\mathcal{R}^a, \mathcal{R}^b, \mathcal{R}^c \leftarrow$ RANDOMCHOICE ($\{\mathcal{R}^p\}$)   ▷ Extract three random individuals from the parent population
5:     $\mathcal{R}^a, \mathcal{R}^b, \mathcal{R}^c \leftarrow$ RESHIFT ($\{\mathcal{R}^a, \mathcal{R}^b, \mathcal{R}^c\}, \mathcal{R}^p$)   ▷ Transform the three extracted *parents* such that they overlap in real space with the reconstruction $\mathcal{R}^p$

6:     $C^{map} \leftarrow$ CROSSOVERMAP (t_{min}, t_{max}, C_p)   ▷ Create the binary crossover map C^{map}
7:     **for all** i, j **do**   ▷ Loop over the matrix coordinates i, j
8:       **if** $C^{map}_{ij} = 1$ **then**
9:         $\mathcal{R}^p_{new} \triangleright \tilde{\rho}_{ij} \leftarrow \mathcal{R}^a \triangleright \tilde{\rho}_{ij} + C_w \times \left(\mathcal{R}^b \triangleright \tilde{\rho}_{ij} - \mathcal{R}^c \triangleright \tilde{\rho}_{ij}\right)$ ▷ Combine the densities of reconstructions $\mathcal{R}^a$, $\mathcal{R}^b$ and $\mathcal{R}^c$ where $C^{map}_{ij} = 1$

10:      **end If**
11:      $\mathcal{R}^p_{new} \triangleright S_{ij} \leftarrow \mathcal{R}^a \triangleright S_{ij}$ **OR** $\left(\mathcal{R}^b \triangleright S_{ij}$ **AND** $\mathcal{R}^c \triangleright S_{ij}$ **AND** $\mathcal{R}^p \triangleright S_{ij}\right)$ ▷ Combine the supports of reconstructions $\mathcal{R}^a$, $\mathcal{R}^b$, $\mathcal{R}^c$ and $\mathcal{R}^p$ with logical operators

12:    **end for**
13: **end for**
14: **return** $\{\mathcal{R}^p_{new}\}$ ▷ The newly created population is returned as output of the CROSSOVER function

15: **end Function**

Crossover

The crossover operation is a central step of an evolutionary algorithm. It is responsible for creating a new population from the pre-existing one, enabling the exploration of the parameters space. In the crossover step, reported in Algorithm 4, a new population of reconstructions $\{\mathcal{R}_{new}^p\}$ is formed by combining individuals in the original population $\{\mathcal{R}^p\}$.

The new density values $\mathcal{R}_{new}^p \triangleright \rho_{ij}$ are filled by following the crossover scheme employed in Differential Evolutionary algorithms[64]. Some values of the new density matrix $\mathcal{R}_{new}^p \triangleright \rho_{ij}$ are directly copied from the corresponding element in the *parent* population $\mathcal{R}^p \triangleright \rho_{ij}$. The others are instead created by combining three randomly-chosen individuals in the original population $\{\mathcal{R}^p\}$. This operation is performed by the routine RANDOMCHOICE at line 4, which returns three randomly-selected reconstructions from the *parent* population, $\mathcal{R}^a, \mathcal{R}^b, \mathcal{R}^c$. The three extracted *individuals* are different from each other and from $\mathcal{R}^p$. The values of the matrix of $\mathcal{R}_{new}^p \triangleright \rho_{ij}$ are set as a combination of the three "parent" densities $\mathcal{R}^a \triangleright \rho_{ij}, \mathcal{R}^b \triangleright \rho_{ij}, \mathcal{R}^c \triangleright \rho_{ij}$ following the formula at line 9 of Algorithm 4.

There are two important preparatory steps for the actual mixing of the density values. The first one is accomplished by the function RESHIFT at line 5, which deals with the ambiguities intrinsic of the phase retrieval problem. In fact, there exists a set of transformations of a density ρ that only affect the phases of its Fourier representation, $\arg(\tilde{\rho})$. These transformations are:

- Spatial translation: $\rho(\vec{x}) \rightarrow \rho(\vec{x} + \Delta \vec{x})$
- Complex conjugation and coordinates inversion: $\rho(\vec{x}) \rightarrow \rho^*(-\vec{x})$
- Addition of a global phase factor: $\rho(\vec{x}) \rightarrow e^{i\phi}\rho(\vec{x})$

And any combination of those. These transformations have no effect on the modulus in Fourier representation, which is the only accessible experimental information on the sample. The direct implication is that a density ρ and all its transformations listed above are equivalent candidate solutions to the phase retrieval problem. These ambiguities must be resolved before any mathematical operation that involves multiple densities, like the crossover mixing (line 9 in Algorithm 4) and the calculation of the average density.

The operation is performed by the routine RESHIFT at line 5 of Algorithm 4. It requires a set of reconstructions to be transformed, in this case $\{\mathcal{R}^a, \mathcal{R}^b, \mathcal{R}^c\}$, and a reference reconstruction, in this case $\mathcal{R}^p$. The function then applies a combination of the transformations listed above to the input densities and supports such that they are as similar as possible to the reference density. This operation can be done efficiently by making use of cross-correlation functions, as detailed in section 2.2.7 of ref. 18, and as further discussed in section "Aligning the reconstructions".

A second preparatory step, accomplished by the CROSSOVERMAP function at line 6 of Algorithm 4, is to define which elements in the matrix $\mathcal{R}_{new}^p \triangleright \tilde{\rho}_{ij}$, i.e., the Fourier representation of the density matrix $\mathcal{R}_{new}^p \triangleright \rho_{ij}$, are modified by the mixing operation. As exemplified in Fig. 8, the Fourier domain is divided into rectangular tiles of size t_x and t_y. The values of t_x and t_y are randomly chosen in the range $[t_{min}, t_{max}]$, and given as parameters to the function CROSSOVERMAP. The value of t_{min} is typically on the order of a few pixels (e.g. $t_{min} = 4$), while t_{max} can assume values corresponding to a relatively large fraction of the linear dimension N_p of the total matrix, for example $t_{max} = \frac{N_p}{8}$. The tiles domain is then randomly shifted and rotated, as visible in Fig. 8. Afterwards, tiles are elected as "crossover regions" with a probability C_p, given as a parameter. The output of the CROSSOVERMAP function is then a binary matrix C^{map} of size $N_p \times N_p$, whose values are set to 1 only at those coordinates i, j belonging to tiles that have been selected for the crossover operation. The tiling of Fourier domain enables mixing of the information during the crossover operation without destroying the local correlations existing in the FT of the density. These correlations derive from the limited spatial extension of the sample and are a consequence of the *oversampling* method. For further information, please refer to refs. 18,65.

It is finally possible to calculate the new values for the new density $\mathcal{R}_{new}^p \triangleright \tilde{\rho}_{ij}$. The operation is performed at line 9 only for those coordinates i, j where the crossover map $C_{ij}^{map} = 1$. The mathematical operation depends on a parameter C_w, typical of Differential Evolution, called "crossover weight"[64].

Additionally, a new population of support functions has to be created. The mixing operation is performed here for all the entries in the new support matrix $\mathcal{R}_{new}^p \triangleright S_{ij}$, without any tiling operation, by mixing the support information of the three selected densities, i.e. $\mathcal{R}^a \triangleright S_{ij}, \mathcal{R}^b \triangleright S_{ij}$ and $\mathcal{R}^c \triangleright S_{ij}$. As long as the support matrix is a binary function, the mixing operation at line 11 of Algorithm 4 is performed via binary operator. In practical terms, the new support function is inherited from $\mathcal{R}^a$, with the addition of those regions included in the support function of all the other parents $\mathcal{R}^b, \mathcal{R}^c$ and $\mathcal{R}^p$.

Once the new densities and supports are created for all reconstructions in the new population $\{\mathcal{R}_{new}^p\}$, the real-space representation of the new densities $\mathcal{R}_{new}^p \triangleright \rho$ is calculated via an inverse FT of $\mathcal{R}_{new}^p \triangleright \tilde{\rho}$ and the new population is returned as output.

The self-improvement

Algorithm 5. The self-improvement routine

1: **function** IMPROVE($\{\mathcal{R}^p\}, \{\mathcal{R}_{new}^p\}$)
2: $\quad \alpha^{best} \leftarrow \sum_{i,j} \mathcal{R}_P^{best} \triangleright S_{ij}$ ▷ Calculate the area α^{best} of the reconstruction with the lowest error in $\{\mathcal{R}^p\}$.
3: $\quad \alpha^{all} \leftarrow \sum_{i,j} \bigcup_{p=1} \mathcal{R}^p \triangleright S_{ij}$ ▷ Calculate the area α^{all} of the union of all the supports in $\{\mathcal{R}^p\}$
4: $\quad \alpha^{eval} \leftarrow \frac{g}{G}\alpha^{best} + (1 - \frac{g}{G})\alpha^{all}$ ▷ Calculate the area α^{eval} for the error evaluation of the reconstruction as a linear combination of α^{best} and α^{all}, with a coefficient dependent of the current generation g
5: $\quad$ **for all** $\mathcal{R} \in \{\mathcal{R}^p\} \cup \{\mathcal{R}_{new}^p\}$ **do**   ▷ For all the reconstructions in the two populations
6: $\quad\quad$ **repeat** $R - 2$
7: $\quad\quad\quad \mathcal{R} \triangleright \rho \leftarrow$ SEQUENCE($\mathcal{R} \triangleright \rho, \mathcal{R} \triangleright S, M$)   ▷ Improve the density with a sequence of IAs
8: $\quad\quad\quad \mathcal{R} \triangleright S \leftarrow$ UPDATESUPPORT($\mathcal{R} \triangleright \rho, \sigma, \tau$)   ▷ Shrink the support function via the Shrink-wrap algorithm
9: $\quad\quad$ **end repeat**
10: $\quad\quad \mathcal{R} \triangleright \rho \leftarrow$ SEQUENCE($\mathcal{R} \triangleright \rho, \mathcal{R} \triangleright S, M$)
11: $\quad\quad \mathcal{R} \triangleright S \leftarrow$ UPDATESUPPORTSIZE($\mathcal{R} \triangleright \rho, \sigma, \alpha^{best}$)   ▷ Update the support to match a size equal to α^{best}.
12: $\quad\quad \mathcal{R} \triangleright \rho \leftarrow$ SEQUENCE($\mathcal{R} \triangleright \rho, \mathcal{R} \triangleright S, M$)
13: $\quad\quad \mathcal{R} \triangleright S \leftarrow$ UPDATESUPPORTSIZE($\mathcal{R} \triangleright \rho, \sigma, \alpha^{eval}$)   ▷ Update the support to match a size equal to α^{eval}.
14: $\quad\quad \mathcal{R} \triangleright \rho \leftarrow$ EVALSEQUENCE($\mathcal{R} \triangleright \rho, \mathcal{R} \triangleright S, M$)   ▷ Improve the density using only the Error Reduction algorithm
15: $\quad\quad \mathcal{R} \triangleright \mathcal{E} \leftarrow$ GETERROR($\mathcal{R} \triangleright \rho, \mathcal{R} \triangleright S, M$)   ▷ Calculate the reconstruction error
16: $\quad\quad \mathcal{R} \triangleright S \leftarrow$ UPDATESUPPORT($\mathcal{R} \triangleright \rho, \sigma, \tau$)   ▷ Shrink the support function via the Shrink-wrap algorithm
17: $\quad$ **end for**
18: $\quad$ **return** $\{\mathcal{R}^p\}, \{\mathcal{R}_{new}^p\}$ ▷ Return the populations of optimized reconstructions
19: **end function**
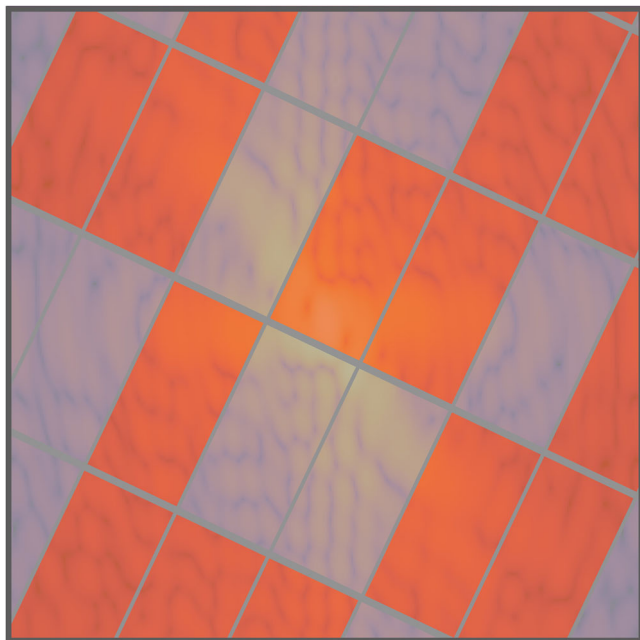
Fig. 8 | Example of a crossover map. The domain is divided into tiles of random height and width. A random shift in space, as well as a random rotation, are then applied to the tiles. Tiles are then elected for the crossover operation with a probability C_p. Coordinates of the matrix that belongs to the elected tiles are then subjected to the crossover mixing.

This step, commonly defined as *self-improvement* in the context of memetic algorithms[66], is responsible for the local optimization of the candidate solutions. In the framework of phase retrieval, this local optimization is accomplished by iterative phase retrieval algorithms (IAs)[35] in order to improve the densities $\mathcal{R} \triangleright \rho$ and the Shrink-wrap algorithm (SW)[19] as regards their respective support functions $\mathcal{R} \triangleright S$. The structure of the *self-improvement* step is reported in Algorithm 5. The alternation of IAs and SW is repeated R times.

For the first $R - 2$ repetitions (see line 6), the densities are improved via IAs through the function SEQUENCE, and the support functions are updated via the SW method through the function UPDATESUPPORT. For the last two repetitions, at line 11 and line 13, the update of the support is instead performed by the function UPDATESUPPORTAREA in place of the standard UPDATESUPPORT. The UPDATESUPPORTAREA function ensures that the new support function has a total area (i.e. number of coordinates where its value is 1) corresponding to the input value α.

The first call of UPDATESUPPORTAREA (line 11) updates all the supports to a target area α^{best}, i.e. the area of the support corresponding to the best reconstruction (i.e. the one with the lowest error value), calculated at line 2 of Alg. 5. The role of this function call is to push away from stagnation those reconstructions that are stuck in a local minimum where the support area is significantly larger than the target.

The second call of UPDATESUPPORTAREA (line 13) sets the support area to a value α^{eval}. The quantity α^{eval} is calculated as a linear combination of α^{best} and of the area of the union of all supports α^{all} (see line 3). The coefficient of the linear combination $\frac{g}{G}$ depends on the current algorithm generation $g = 1, \ldots, G$, tuning α^{eval} closer to α^{all} at the beginning of the reconstruction and closer to α^{best} towards the last generations (see line 4). This second call ensures that all the reconstructions in the two populations have the same support area before the execution of the last algorithms sequence EVALSEQUENCE (line 14, where only the Error Reduction algorithm is executed) and the following error evaluation (line 15). The importance of this aspect and its central role in the MPR method is described in the following section "The modified Shrink-wrap".

Once that the error value is calculated via the GETERROR function, which calculates the quantity introduced in Eq. (3), the support of the reconstructions are upgraded via the UPDATESUPPORT call (standard Shrink-wrap algorithm) and the two *improved* populations $\{\mathcal{R}^p\}$ and $\{\mathcal{R}^p_{new}\}$ are returned.

The modified Shrink-wrap
The functions UPDATESUPPORTSIZE in Algorithm 5 indicate a modified version of the Shrink-wrap algorithm for the update of the support function. This algorithm inherits the first step from the Shrink-wrap, i.e. it performs a gaussian smoothing, with standard deviation σ given as parameter, on the input density $\mathcal{R} \triangleright \rho$. The standard Shrink-wrap then calculates the new support by thresholding the smoothed density ρ_σ at a value $\tau \times \max[\rho_\sigma]$.

Conversely, this modified version ensures that the overall extension of the support, i.e. the number of coordinates where the density is allowed to be non-zero, equals to the quantity α. The threshold τ is then automatically computed within the UPDATESUPPORTSIZE based on the value α and the values of the smoothed density ρ_σ.

This peculiar way of upgrading the support matrix ensures that each density $\mathcal{R} \triangleright \rho$ is then optimized by using a support constraint $\mathcal{R} \triangleright S$ with equivalent "strength". This aspect is of fundamental importance for the following SELECTION step (see line 6 of Algorithm 3), where reconstructions in $\{\mathcal{R}^p\}$ and $\{\mathcal{R}^p_{new}\}$ are compared based on their error value, as discussed in the following section.

The selection step

Algorithm 6. The selection step
1: **function** Select($\{\mathcal{R}^p\}$, $\{\mathcal{R}^p_{new}\}$)
2:     **for** $p = 1, \ldots, P$ **do** ▷ Loop over all indexes p in the reconstructions populations of size P
3:         **if** $\mathcal{R}^p_{new} \triangleright \mathcal{E} < \mathcal{R}^p \triangleright \mathcal{E}$ **then** ▷ For each index p, the error of the reconstruction in the parent population $\mathcal{R}^p$ and in the new population $\mathcal{R}^p_{new}$ are compared
4:             $\mathcal{R}^p \leftarrow \mathcal{R}^p_{new}$ ▷ If the error of the reconstruction belonging to the new population is lower than the one of the parent population, replace the parent individual with the new one
5:         **end if**
6:     **end for**
7:   **return** $\{\mathcal{R}^p\}$
8: **end function**

The implementation of the SELECT function at line 6 of Algorithm 3 is reported in Algorithm 6. It has the important role of comparing the performance of the reconstructions in $\{\mathcal{R}^p\}$ and $\{\mathcal{R}^p_{new}\}$ and upgrading the current population $\{\mathcal{R}^p\}$ accordingly. This operation is based on the evaluation of the error value, calculated at line 15 of Alg. 5, for each reconstruction $\mathcal{R}$ in the two populations. For each index p, the error values of the reconstruction in the original population $\mathcal{R}^p$ and in the new one $\mathcal{R}^p_{new}$ are compared (line 3 of Alg. 6). The reconstruction $\mathcal{R}^p$ is then replaced by $\mathcal{R}^p_{new}$ if the error of the latter $\mathcal{R}^p_{new} \triangleright \mathcal{E}$ has a lower value, otherwise $\mathcal{R}^p_{new}$ is discarded. The upgraded population $\{\mathcal{R}^p\}$ is then returned by the function and a new generation of the MPR algorithm starts.

Despite its conceptual simplicity, the selection step has a peculiar aspect deriving from the nature of the phase retrieval problem, and has to be carefully handled as discussed in the following. The evaluation of the error involves the use of the two constraints required to ensure the existence of an unique solution to the phase problem, which are enforced by the two projector operators, P_M and P_S, introduced by Eq. (7) and Eq. (6), respectively.

The action of the two projections on a generic density ρ is graphically represented in Fig. 9a, along with the error value E which can

Fig. 9 | Effect of different support sizes on the evaluation of the error value. In (**a**), graphical representation of the action of the two projectors, P_M and P_S, on a generic density ρ. The error value, indicated as a dashed gray line, is the distance between the two projections. For further discussion about the interpretation of the phase retrieval problem in terms of sets, please refer to refs. 18,30,35. In (**b**) supports with decreasing sizes, from A to D, correspond to decreasing volumes of the sets S^A to S^D in (**a**). The dependency of the error value E on different support sizes is also shown in (**a**), with larger supports yielding smaller error values for the same density ρ.
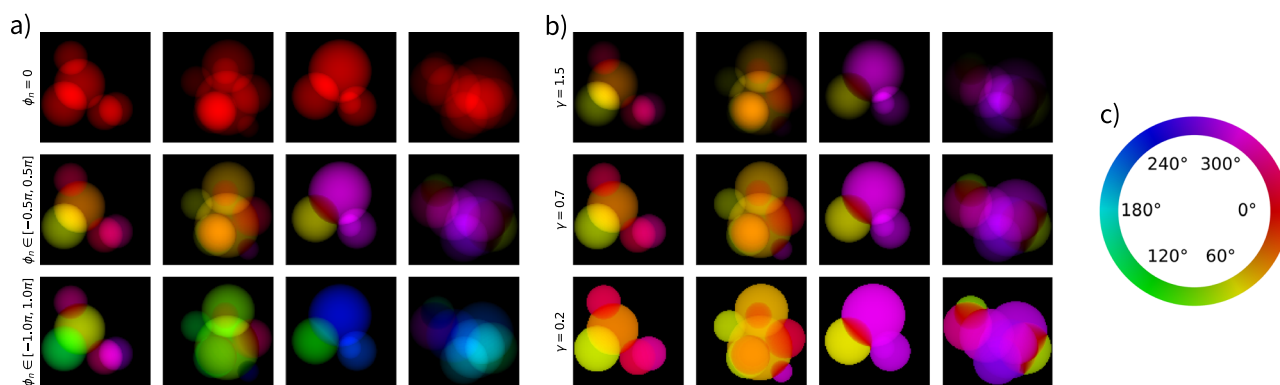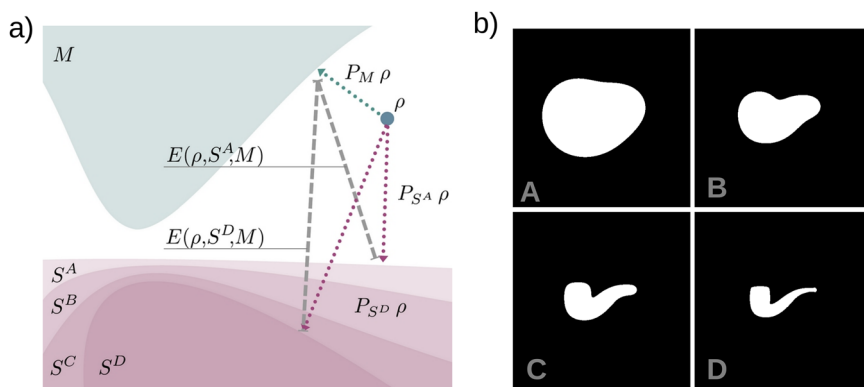
Fig. 10 | Examples of densities initialization. Densities are randomly initialized by filling a square area of the complex-valued matrix of spherical profiles with random sizes and positions. In (**a**), for each row, four different densities are initialized with increasing values for the phases ϕ_n^{init}. In (**b**), different gamma values are applied to the densities. The phase of the complex-valued matrix are encoded following the color scheme shown in (**c**).

be interpreted as the *distance* between the two projections. Finding the solution to the phase problem can be also interpreted in an intuitive manner as the density ρ for which the distance between the projections on the two sets is the smallest.

The P_M operator acts equivalently for all reconstructions, because the set M is the same for all the densities (the experimental pattern I_{ij} is known a-priori and does not change during the reconstruction procedure). The situation is substantially different for the case of the P_S operator, which projects a given density ρ onto the set S. Each reconstruction in a population has its own support function which differs from the others. As a consequence, the projection operator P_S acts differently for each reconstruction $\mathcal{R}$.

This feature poses a problem when evaluating the error value of the reconstruction, which can be interpreted as a distance between the two sets as depicted in Fig. 9. In principle, each of the supports shown in Fig. 9b is enough to define a unique solution to the phase problem. However, a looser support function corresponds to a set S in Fig. 9a with higher volume. Thus, reconstructions evaluated on a larger support function inherently have a smaller error value, as the distance between the sets M and S reduces. It is in general incorrect to compare the errors of two reconstructions with different support functions.

However, the comparability between the reconstructions can be restored by enforcing all the reconstructions $\{\mathcal{R}^p\}$ and $\{\mathcal{R}^p_{\text{new}}\}$ to have a set S with equal volume, i.e. their support functions $\mathcal{R}^p \triangleright S$ to be composed by the same number of pixels. This fundamental step is achieved by the UPDATESUPPORTSIZE function described in section IV E, executed right before the last sequence of iterative algorithms

EVALSEQUENCE in the IMPROVE routine (at line 14 of Alg. 5) and the error evaluation (line 15 of Alg. 5).

Initialization of the starting densities

The initialization of the starting guesses at the beginning of the reconstruction process, so far indicated by the INITIALIZE function, plays a crucial role. On the one hand, it is desirable to create starting guesses as close as possible to the solution of the phase problem. On the other hand, the process has to be sufficiently randomized to enable a reasonably wide exploration of the parameters space. Despite that the initialization itself does not conceptually belong to the MPR method, the way in which densities are initialized may affect the final outcome of the reconstruction, and its description is useful to understand and evaluate the tests conducted on experimental data presented in section "Dealing with experimental data".

The initialization step employed here is based on filling a region of the real space with a number of spherical densities. The real space region is a square of size $\Sigma^{\text{init}} \times \Sigma^{\text{init}}$ pixels. This area is then filled by summing up a number N^{init} of densities of projected spheres with different radii R_n^{init}. To each sphere, a global phase value ϕ_n^{init} can be assigned.

An example of this procedure is shown in Fig. 10a, where four densities are stochastically initialized. In the first row, all the densities are real-valued, i.e. they all have a fixed global phase, in this case $\phi_n^{\text{init}} = 0$ (see color map in Fig. 10c). Here, densities are initialized in a square of edge $\Sigma^{\text{init}} = 100$ pixel. The number of spherical profiles is randomly chosen in the range $N^{\text{init}} \in [2, 8]$. Their diameter is extracted in the range $D_n^{\text{init}} \in [20, 60]$ pixel. The second and third rows of Fig. 10a

show, instead, the same initialization but with different ranges for the spheres phases, which are $\phi_n^{init} \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ and $\phi_n^{init} \in [\pi, \pi]$ respectively.

The densities shown in Fig. 10a correspond to the optical depth of spherical particles. These profiles can additionally be tuned by applying an exponentiation operation to the modulus of the overall density with exponent γ^{init}, i.e. $|\rho|^{\gamma^{init}}$. All the densities in Fig. 10a are produced with $\gamma^{init} = 1$. Figure 10b shows instead the effect of different values for the initialization exponent γ^{init}, i.e., 1.5, 0.7 and 0.2, applied to the densities of the second row in Fig. 10a. The effect of γ^{init} is equivalent to the *gamma* value in image processing, where contrast can be increased ($\gamma^{init} > 1$) or decreased ($\gamma^{init} < 1$).

The last step performed in the initialization step concerns the normalization of the density. The Parseval's theorem states that the Fourier transform operation preserves the L^2 norm. Thus, the L^2 norm of the density ρ is fixed by the knowledge of the diffraction pattern I. In numerical terms, it means that $\sum_{ij}|\rho_{ij}|^2 = \sum_{ij}|FT[\rho]_{ij}|^2 = \sum_{ij} I_{ij}$. So, if the full diffraction image I_{ij} is known, the normalization is automatically constrained by the application of the modulus projector P_M in Eq. (7).

However, one unavoidable limiting factor of real experimental data is that the inner part of the diffraction image is not recorded, due to the presence of a beam stopper or a hole in the detector to prevent from damages created by the highly intense transmitted light beam. This feature represents the main obstacle when attempting CDI on experimental data, as the phase retrieval algorithms have to also retrieve the missing intensities where the data is not available[18]. This aspect is discussed in detail in section "Dealing with experimental data".

To aid phase retrieval algorithms in their optimization task, it is convenient attempt a normalization of the starting densities that is as close as possible to the correct one. Known values in the scattering image can be identified by a *mask* matrix Ξ, with $\Xi_{ij} = 1$ where the scattering values I_{ij} are known, and $\Xi_{ij} = 0$ otherwise. Each density ρ is then multiplied by a normalization factor:

$$\nu = \frac{\sum_{ij}\Xi_{ij}I_{ij}}{\sum_{ij}\Xi_{ij}|FT[\rho]_{ij}|^2} \tag{10}$$

In practical terms, the full L^2 norm is inferred by comparing the *partial L^2* norms calculated only on the known pixels in the scattering data, where $\Xi_{ij} = 1$.

Summary of the parameters and further considerations
The parameters and terms introduced so far are summarized in Table 1. This overview allows to further discuss some aspects that were only briefly mentioned in the previous section and require more discussion.

Changing values over generations
As the MPR generations progress it is useful to modify some parameters that affect the update of the support function and the execution of iterative algorithms. This is particularly true for the number of iterations of iterative algorithms (HIO and RAAR). In fact, a large number of iterations of these iterative algorithms (IAs) allows better exploration of the parameters space, suffering less stagnation issues compared to the ER algorithm. Their drawback is, however, that they often fail to keep the solution once they are

Table 1 | Glossary of the most important parameters and keywords of the SPRING framework

Name	Type	Allowed values	Default	Description
Global (see section "Memetic phase retrieval")				
P	int	≥4	128	Population size
G	int	> 1	100	Number of generations
N_p	int	–	–	Linear dimension of the diffraction data
Initialization (see section "Initialization of the starting densities")				
Σ^{init}	int	–	–	Initial linear dimension of the support function
N^{init}	int	≥1	5	Amount of spherical profiles that fills the initial support
D_n^{init}	rand (**float**)	[> 0, ≤Σ^{init}]	[0.2 · Σ^{init}, 0.9 · Σ^{init}]	Range of diameters for the spherical profiles
ϕ_n^{init}	rand (**float**)	[≥ − π, ≤π]	[0, 0]	Range of phase values for the spherical profiles
γ^{init}	**float**	> 0	1	Gamma value applied to the modulus of the initial density
Crossover				
C_p	float	0 < · ≤1	0.6	Crossover probability (see section "Crossover")
C_w	float	0≤ · ≤2	0.4	Differential weight (see section "Crossover")
C_a	float	0≤ · ≤1	0	Crossover averaging (see section "Missing intensities at low scattering angles")
Support update				
τ, τ_{end}	float	0 < · ≤1	–, $\frac{2}{3}\tau$	Threshold for the support update via the shrink-wrap algorithm
σ, σ_{end}	float	≥0	2, 0.5	Smoothing amount for the support update via the shrink-wrap algorithm
Iterative algorithms				
R	int	≥2	3	Number of repetitions of the main algorithms sequence (see section "The self-improvement")
χ	float	0 < · ≤1	0.5	Fraction of the complex plane allowed for density values (see section "Real-space constraints")
J	int	> 0	40	Starting number of iterations of IA (HIO or RAAR) (see section "Changing values over generations")
β	float	0 < · ≤1	0.9	Beta parameter for HIO or RAAR algorithms
J_{ER}	int	> 0	40	Starting number of iterations of Error Reduction (see section "Changing values over generations")
J_{eval}	int	> 0	40	Number of iterations of Error Reduction for the evaluation sequence (see section "The algorithms sequence")

close by[35]. A common practice is then to reduce their number of iterations over the reconstruction process. For these reasons, their number of iterations J is set at the beginning, and it is linearly reduced to J_{end} over the generations. By default, $J_{end} = 0$ for HIO and RAAR.

A similar consideration applies to the update of the support function via the Shrink-wrap algorithm[19]. There, both the threshold τ and the smoothing about σ can be reduced along the reconstruction process to improve the quality of the reconstruction result. In this framework, the starting values of τ and σ are then linearly changed as function of the generations to the final values τ_{end} and σ_{end}. By default, these values are set to $\tau_{end} = \frac{2}{3}\tau$ and $\sigma_{end} = 0.5$ px.

The algorithms sequence

Iterative algorithms are executed as a SEQUENCE, introduced in Eq. (1), which is repeated a number R of times for each reconstruction $\mathcal{R}$ at each generation $g = 1, \ldots, G$, and alternated with the routine UPDATESUPPORT at each repetition (see Algorithm 5). While, in principle, several different combinations of algorithms can be used, the structure of the algorithms sequence is kept as simple as possible in this work. As highlighted in section "The self-improvement", two algorithms sequences are used:

Main sequence. It is indicated as SEQUENCE, and it is composed of some iterations of the HIO or RAAR algorithm (with the given value of the β parameter), followed by some iterations of the Error Reduction algorithm. The starting number of IAs iterations is set by the parameter J and the starting number of ER iterations is set by J_{ER} (see Table 1). Their number is linearly modified as the generations progress. Their final value is 0 for IAs and $J + J_{ER}$ for the ER algorithm, i.e. only the Error Reduction algorithm is executed towards the end of the reconstruction procedure.

Evaluation sequence. The evaluation sequence is designed to steer the current reconstruction towards the closest local minimum of the error function, immediately preceding the error evaluation. For this reason, in the current implementation of MPR, it is composed only of a number of iterations J_{eval} of the Error Reduction algorithm that is kept constant during the reconstruction process.

Real-space constraints

In addition to the support function, a further constraint can be added in real space. In literature, this constraint is indicated often as *positivity* constraint. When applied, it restricts the real part of the density values ρ_{ij} to be non negative[35]. In general, the values ρ_{ij} are allowed to cover the complete complex set, i.e., their phases $\arg(\rho_{ij}) \in [-\pi, \pi]$, while $\arg(\rho_{ij}) \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ if positivity is constrained. Within MPR, the positivity constraint is generalized by the parameter χ, such that phases are, in general, constrained to the set $[-\chi\pi, \chi\pi]$. The default value is $\chi = 0.5$, which is equivalent to the application of the typical positivity constraint. In the examples shown in this manuscript, χ is set to its default value if not explicitly indicated.

Aligning the reconstructions

The RESHIFT operation, discussed in section "Crossover", is necessary to minimize the intrinsic ambiguities of the phase retrieval problem before performing operations that combine different densities, like the *crossover* or the averaging. A generic density ρ can be transformed to match a *reference density* ρ^{ref} by calculating the cross correlation function between the two, i.e. $\rho \star \rho^{ref}$. The coordinates of $\max\left[|\rho \star \rho^{ref}|\right]$ (or, to be more precise, the coordinates relative to the center of the autocorrelation matrix) indicate the spatial shift that has to be applied to ρ to spatially overlap with ρ^{ref}. The global phase that has to be added to ρ is, instead, the phase of this maximum value. The ambiguity of complex conjugation and inversion, i.e. $\rho(\vec{x}) \rightarrow \rho^*(-\vec{x})$, is resolved by performing the same operations on the transformed version $\rho^*(-\vec{x})$. Then, the one that gives the highest maximum value for the autocorrelation is kept.

This operation turns out to be safe and stable when the population of reconstructions is already converging towards a solution, while it may present some issues at the beginning of the reconstruction process. This is particularly true for large samples, where the relative amount of information lost in the central hole of the detector is relevant and renders the reconstruction process challenging. In fact, the dominant information that determines the outcome of the cross-correlation operation is the most intense one in Fourier space. Due to the isolated nature of the samples (necessary to satisfy the oversampling condition) the brightest region in the Fourier domain is the one at the center, corresponding to low momentum transfer, where experimental data is usually missing. The misestimate of these values by phase retrieval algorithms may result in misalignment of the reconstructions, and further amplifications of the corresponding artifacts (see also section "Dealing with missing intensities").

To mitigate this issue, the coordinates of the intensity matrix at which the experimental values are unknown are excluded from the calculation of the cross-correlation function. In particular, the cross-correlation between ρ and ρ^{ref} is calculated via the *convolution theorem* as follows:

$$\rho \star \rho^{ref} = \mathcal{F}^{-1}\left[\Xi \cdot \overline{\mathcal{F}[\rho]} \cdot \mathcal{F}[\rho^{ref}]\right] \tag{11}$$

where Ξ_{ij}, called mask, has values 1 where the intensities are known and 0 otherwise.

Dealing with missing intensities

The unavoidable feature of CDI diffraction patterns at FELs is the lack of information in some regions of the reciprocal space. Those regions can be categorized into three types:

1. Regions in-between the detector modules. CDI detectors are constructed as an assembly of tiles, with some space between them to leave room for electronics. Those areas can be identified in the dark stripes of the patterns shown in Fig. 2.
2. Central region of the scattering image. A center hole allows the unscattered beam to pass through the detector without damaging it. This appears in the diffraction patterns as a large area of unknown intensities at the center of the detector with circular (Fig. 2a) or rectangular (Fig. 2d) shape.
3. Saturated pixels. Detector pixels are capable of recording radiation up to a maximum threshold value. Above this value, the pixel sensors no longer respond in a linear manner to the incoming radiation. Thus, those pixels have to be excluded from the retrieval procedure, as they do not carry correct information. Due to the decreasing scattering intensity with increasing momentum transfer, which is an intrinsic feature of diffraction images from isolated particles, saturated pixels are typically close to the detector center, further reducing the information recorded at low momentum transfer.

While 1) and 2) are predetermined by design, 3) varies on a shot-to-shot basis. Some strategies are implemented in MPR to deal with these issues.

With regards to point 1), iterative algorithms typically over-estimate the scattering signal in those regions, causing the appearance of artifacts in the reconstruction. A strategy to extract an upper bound for the missing intensities from the known ones has been developed. This is accompanied by the introduction of a modified version of the modulus projector P_M that constrains this upper bound during the execution of iterative algorithms. This method greatly improves the quality of reconstructions belonging to Dataset B (Fig. 2d). It is used in all of the tests that involve this dataset. It is worth underlining that this strategy is not strictly connected to the use of the MPR algorithm and can be directly implemented into standard imaging workflows. The implementation details and technical aspects of this strategy are reported in the following section "Constraining intensities at large scattering angles".

The way in which MPR deals with the missing intensities in the central region of the detector corresponding to low momentum transfer, deriving
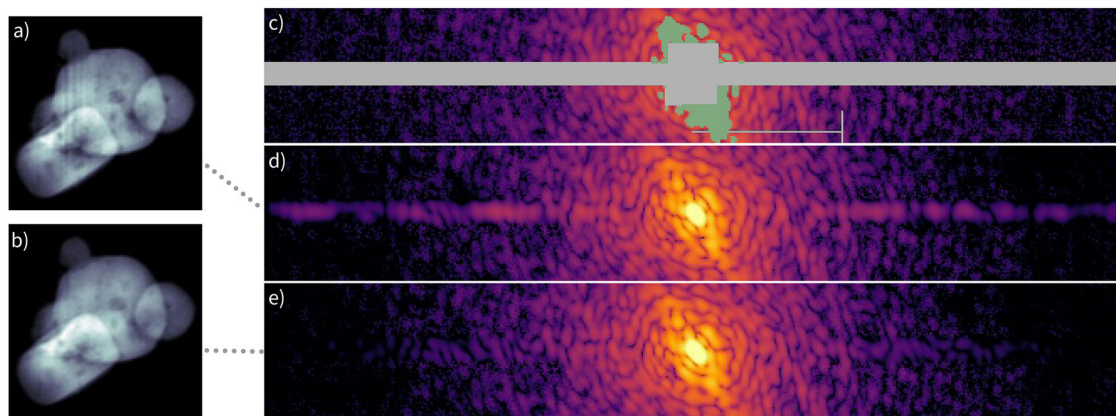
Fig. 11 | Effects of missing intensities in the detector's gap on the reconstructions. In (**c**), a cutout of the experimental data, taken from Dataset B, is shown. The gray region identifies the missing values due to the detector's geometry, while the green one highlights the intensities missing due to pixels saturation. In (**a**), the reconstruction is performed without constraints on the missing values shown in (**c**). The reconstruction shows strong artifacts in the form of vertical lines in the reconstructed scattering density. Those artifacts come from the overestimation of the intensities in the detector's gap. This is highlighted in (**d**), where the amplitude of the Fourier Transform of (**a**) is shown. Subfigure (**b**) shows the reconstruction performed by placing an upper bound to the missing intensities in the gap, as described in the main text, where the strong artifacts are now missing. The respective Fourier Transform in (**e**) shows how, in this case, the overall amplitude of the intensities in the detector's gap are now fitting the overall intensity behavior.

from points 2) and 3), is instead strictly connected to the *crossover* operation. This topic is discussed in the following section.

Constraining intensities at large scattering angles

The problem of missing intensities due to the detector's tiles arrangement is particularly impacting the case of Dataset B (see Fig. 2d), where the data has been acquired via the pnCCD detector of the SQS beamline at European XFEL[48]. The detector is composed of two modules, upper and lower, which can be independently moved. This arrangement is such that, in the horizontal direction, there is a region of unknown values at all $\vec{q}$ magnitudes, highlighted in gray color in Fig. 11c. Such a large, spatially correlated, region of unknown values in Fourier space is challenging for reconstruction algorithms, which often tend to misestimate those values during the reconstruction process. This is well exemplified by the reconstruction shown in Fig. 11a, where strong artifacts in the form of vertical lines arise from an overestimation of the reconstructed intensities in the horizontal gap (see Fig. 11d).

A workaround to this problem, implemented in MPR, is to set upper bounds for those intensity values, based on the generic behavior of the scattering pattern. Those upper bounds can be calculated by considering the behavior of the scattering intensity as function of the distance from the detector's center. The scattering intensity $I(\vec{q})$ can be rewritten in terms of polar coordinates of the momentum transfer $\vec{q} = (q, \phi)$, i.e., $I(q, \phi)$. For a given magnitude of the momentum transfer q (corresponding to a fixed scattering angle, i.e., to a fixed distance from the detector's center), the average scattering intensity $\mu_I(q)$ can be calculated as:

$$\mu_I(q) = \langle I(q, \phi) \rangle_\phi = \frac{\int_0^{2\pi} d\phi\, I(q, \phi) \Xi(q, \phi)}{\int_0^{2\pi} d\phi\, \Xi(q, \phi)} \tag{12}$$

where $\Xi(q, \phi)$ is the mask function described in section "Initialization of the starting densities", which assumes a value of $\Xi(q, \phi) = 0$ where pixels are unknown, or $\Xi(q, \phi) = 1$ otherwise. The function $\langle I(q, \phi) \rangle_\phi$ is commonly addressed as the *radial profile* of the scattering. Still, Eq. (12) does not take into account the fluctuations of the values that happen along the integration circumference. It is then necessary to take into account those fluctuations by calculating the standard deviation of the average intensity at a given magnitude of the momentum transfer, i.e.:

$$\sigma_I(q) = \sqrt{\langle I^2(q, \phi) \rangle_\phi - \langle I(q, \phi) \rangle_\phi^2} \tag{13}$$

The upper bound for the intensities $U(q)$, which is assumed then to be dependent only on the magnitude of q, is defined as:

$$U(q) = \mu_I(q) + \eta \cdot \sigma_I(q) \tag{14}$$

where $\eta > 0$ is a constant value. In practical terms, the upper bound for the missing intensity is set to be the average value of the known intensities at that scattering angle increased by η times their standard deviation. The exact value for the parameter η is a free parameter. In the context of this paper, this value is kept to $\eta = 1.5$.

The upper bounds in Eq. (14) work upon the assumption that the missing intensities follow the same statistics as the known ones. This is a fair assumption when, for a given momentum q, the amount of known intensities is substantially larger than the unknown ones. As the gap between the detector's modules is fixed, while the length of the circumference where the integration is calculated increases with q, this condition is certainly met far away from the image center. Conversely, for low q values close to the center, the relative amount of unknown intensities increases, up to the point where a statistical extrapolation from the known intensities becomes inadequate. In the context of this paper, the upper bound in Eq. (14) is only used for those scattering angles where the relative amount of unknown intensities is within 20% of the total. A further consideration is necessary for regions where the detector is saturated. First, the upper bound is not applied here. Second, if at a given magnitude of the momentum q more than 5% of the pixels are saturated, then the upper bound U_I is not applied.

Once the upper bound is defined based on Eq. (14) with the subsequent considerations, the next discussion point is to define how this upper bound is enforced during the reconstruction process. Such an operation is performed by a modification of the intensity projector, introduced in Eq. (7) in a simplified form. In fact, the projector P_M replaces the current estimate of the Fourier amplitudes $|\tilde{\rho}_{ij}|$ with the experimental ones, $M_{ij} = \sqrt{I_{ij}}$ only where those values are known, i.e., where $\Xi_{ij} > 0$. The action of P_M, which creates a new density ρ^{new}, can then be defined as:

$$\tilde{\rho}_{ij}^{\text{new}} = \begin{cases} M_{ij} \dfrac{\tilde{\rho}_{ij}}{|\tilde{\rho}_{ij}|} & \text{if } \Xi_{ij} = 1 \\ \tilde{\rho}_{ij} & \text{otherwise} \end{cases} \tag{15}$$

The enforcement of the upper bounds for the unknown intensity values can be done via a modification of the intensity projector P_M, which

now acts as:

$$\tilde{\rho}_{ij}^{\text{new}} = \begin{cases} M_{ij}\, \dfrac{\tilde{\rho}_{ij}}{|\tilde{\rho}_{ij}|} & \text{if } \Xi_{ij} = 1 \\[2mm] U_{ij}\, \dfrac{\tilde{\rho}_{ij}}{|\tilde{\rho}_{ij}|} & \text{if } \Xi_{ij} = 0 \\ & \text{and } U_{ij} \text{ is valid} \\ & \text{and } |\tilde{\rho}_{ij}| > U_{ij} \\[2mm] \tilde{\rho}_{ij} & \text{otherwise} \end{cases} \qquad (16)$$

where U_{ij} is the matrix that numerically encodes the upper bounds for each pixel coordinate calculated via Eq. (14).

The improvements in terms of reconstruction quality derived from the use of the upper bounds for the unknown experimental data in the detector's gap are clearly observable in the reconstruction shown in Fig. 11b (to be compared to the unconstrained version in Fig. 11a). The corresponding Fourier intensities are shown in Fig. 11e, where signs deriving from a misestimate of the unknown values are no longer visible (to be compared with the unconstrained result in Fig. 11d).

It is worth noting that the presented method to mitigate this type of artifact is not strictly dependent on the MPR approach, and can be directly transferred to any imaging routine that makes use of standard iterative algorithms for phase retrieval.

The constraint of upper bounds for missing data results in a clear improvement in the reconstruction quality for diffraction patterns that belong to Dataset B. In regards to Dataset A (see Fig. 2a), the arrangement of the detector modules is such that, away from the central part, most of the intensities are recorded at all scattering directions. In this arrangement, the gap between the detector's components is not a main issue in terms of the quality of the reconstructions, which are naturally less prone to such artifacts.

Still, in both datasets, information at low momentum transfer q, i.e. close to the detector's center, is missing due to the presence of a hole or notch in the detector's modules (to allow for the transmission of the main FEL beam in order to avoid damage) and/or due to saturation of the detector's pixels (see the green region in Fig. 11c). How the MPR algorithm deals with this aspect is discussed in the next section.

Missing intensities at low scattering angles

In general, the lack of diffraction data in the central part of the detector is a strong limiting factor for CDI. In fact, the central region of the detector corresponds to low momentum transfer (i.e. low scattering angles), which encode the low resolution information on the sample, and in particular:

1. The overall shape of the sample. It is harder to retrieve the correct support function, pivotal for a successful reconstruction and for the correct identification of the structural properties of the sample.
2. The density profile at large length scales. This prevents iterative algorithms from correctly identifying the amplitude of the scattering density in real space.

This missing data has to be retrieved by iterative algorithms using the known intensities. This is theoretically possible because incorrectly retrieved intensities in the unknown central region of the detector cause an increase in density values outside the support function in real space (i.e., they are incompatible with the support constraint).

However, there are some specific intensity distributions in the central part that are still highly compatible with the support constraint. Those are known as "weakly constrained modes"[61] and can be added to a density ρ^{sol}, i.e., a solution to a specific phase retrieval problem, without significantly changing its error value. The possibility of constraining those modes becomes weaker and weaker with (i) larger samples (as the support function is wider and the problem less constrained) and (ii) larger region of missing intensities (as their effect on the density outside the support function is lower). For further details on the weakly constrained modes please refer to ref. 61.

During a reconstruction process, those *unconstrained modes* tend to arise, especially due to the action of more *ergodic* algorithms like HIO and RAAR. The crossover operation implemented in MPR (see section "Crossover") aims to augment the explorative power of standard IAs. As a side effect, the crossover operation further amplifies those unwanted modes.

Within the MPR method implemented in SPRING, the occurrence of *unconstrained modes* due to missing data in the central part of the detector can be mitigated by injecting information from the average reconstruction into the crossover operation. The average reconstructed intensities, calculated over the population $\{\mathcal{R}^p\}$, can be extracted in different ways. One possibility is to employ the amplitude of the Fourier Transform of the average reconstruction ρ^{avg}, i.e.:

$$\dot{M}_{ij}^{\text{avg}} = |\mathcal{F}[\rho^{\text{avg}}]_{ij}| = \left|\mathcal{F}\left[\frac{\sum_p \mathcal{R}^p \triangleright \rho}{P}\right]_{ij}\right| \qquad (17)$$

The second possibility is to instead directly calculate the average of the retrieved Fourier moduli, i.e.

$$\ddot{M}_{ij}^{\text{avg}} = \frac{\sum_p |\mathcal{F}[\mathcal{R}^p \triangleright \rho]_{ij}|}{P} \qquad (18)$$

It can be easily verified that $\dot{M}_{ij}^{\text{avg}} \leq \ddot{M}_{ij}^{\text{avg}}$, where the equal relationship only holds when all the densities ρ^p have the same phase values in Fourier space. Due to the strongly differing phase values in particular in the first stages of the reconstruction process, $\dot{M}_{ij}^{\text{avg}}$ defined in Eq. (17) tends to be an underestimate of the correct Fourier moduli. On the other hand, $\ddot{M}_{ij}^{\text{avg}}$ tends to give an overestimate of the moduli, due to the strictly positive nature of the quantity $|\mathcal{F}[\rho^p]_{ij}|$. These qualitative considerations lead us to the use of the average between the quantities in Eq. (17) and Eq. (18) as an estimate for the missing Fourier moduli.

The information gained via Eq. (17) and Eq. (18) is then propagated into the population of reconstructions via the crossover operator (section "Crossover"). There, the density values of three different *parent* reconstructions, $\mathcal{R}^a$, $\mathcal{R}^b$ and $\mathcal{R}^c$, are combined in reciprocal space following the operation described at line 9 of Alg. 4. The parent reconstruction $\mathcal{R}^a$ is then replaced with a modified reconstruction $\bar{\mathcal{R}}^a$. The densities in Fourier representation $\mathcal{R}^a \triangleright \tilde{\rho}$ and $\bar{\mathcal{R}}^a \triangleright \tilde{\rho}$ have the same phase values. The modulus of $\bar{\mathcal{R}}^a \triangleright \tilde{\rho}$ is instead defined as:

$$|\bar{\mathcal{R}}^a \triangleright \tilde{\rho}_{ij}| = C_a \cdot \frac{\dot{M}_{ij} + \ddot{M}_{ij}}{2} + (1 - C_a) \cdot |\mathcal{R}^a \triangleright \tilde{\rho}_{ij}| \qquad (19)$$

The reconstruction $\bar{\mathcal{R}}^a$ is then used in place of $\mathcal{R}^a$ as a parent density for the crossover. In practical terms, Eq. (19) sets the amplitude of $\bar{\mathcal{R}}^a \triangleright \tilde{\rho}$ as a linear combination of the average moduli and the amplitude of the original reconstruction $\mathcal{R}^a \triangleright \tilde{\rho}$, depending on the parameter C_a. When $C_a = 1$, $\bar{\mathcal{R}}^a \triangleright \tilde{\rho}_{ij}$ is obtained by completely replacing the moduli of $\mathcal{R}^a \triangleright \tilde{\rho}_{ij}$ with the average ones. When $C_a = 0$, $\bar{\mathcal{R}}^a \triangleright \tilde{\rho}_{ij}$ is just a replica of the original density $\mathcal{R}^a \triangleright \tilde{\rho}_{ij}$.

The behavior of SPRING for a spatially large sample, a particle agglomerate, is shown in Fig. 12 for three different values of the C_a parameter. Figure 12b is divided into three columns, corresponding to $C_a = 0$, $C_a = 0.2$ and $C_a = 0.5$. For each case, the average reconstruction is reported along with three representative individuals, similarly to what is reported in Fig. 3. Here, the color scale encodes the phase component of the complex-valued reconstructed densities, following the color scheme in Fig. 10. In the individual reconstructions, large-scale fluctuations of the density phases are recognizable, which come from the introduction of the unconstrained modes (apart from the Gaussian one, see ref. 61).

It is possible to inspect the behavior of these unconstrained modes by tracking two quantities. First, the unconstrained modes affect the overall sum of the reconstructed density, i.e. $|\sum_{ij}\rho_{ij}|$. This value corresponds to the central peak of the Fourier Transform and is theoretically encoded at the
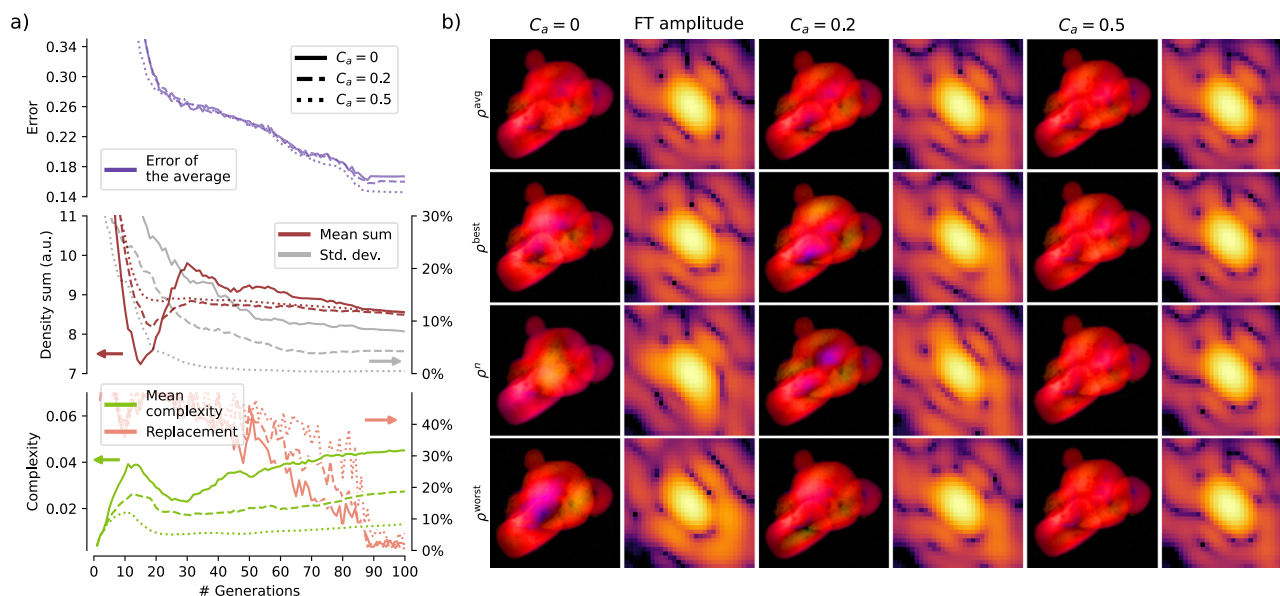
Fig. 12 | Effects of the presence of unconstrained modes on the reconstructions, performed on a diffraction image taken from Dataset B. In (**a**), different performance indicators are reported as function of the algorithm's generations. Solid lines report the value of the indicators for the *standard* crossover operation described in section "Crossover", which corresponds to $C_a = 0$. Dashed and dotted lines describe instead the values for $C_a = 0.2$ and $C_a = 0.5$, respectively. The relative standard deviation (gray line) and the replacement factor (orange line) are expressed in %, with the scale reported on the right axis of the respective plots. In (**b**), insights into the status of population of reconstructions at the last generation of the algorithm (100) are reported for the three different values of C_a. In the first row, the average reconstruction is shown, indicated as ρ^{avg}. The second row shows the reconstructions with the lowest error, ρ^{best}. The third row reports the density in the middle of the errors distribution, ρ^n. The last row depicts the density ρ^{worst} with the highest error. For each density, a cut-out of the modulus of its Fourier transform is also reported in logarithmic color scale, to highlight how the missing data in the central part of the detector affects the reconstructions.

center of the diffraction pattern, i.e. $|\sum_{ij}\rho_{ij}| = \sqrt{I(0,0)}$. As the central part of the pattern is missing, this value is not fixed and tends to fluctuate during the reconstruction process. It is well highlighted by tracking the mean value of the densities sum in the population of reconstructions, shown as a solid red line in Fig. 12a for the case $C_a = 0$ (no average information is injected into the crossover). The behavior shows that the density sum quickly drops in the first 15 generations, during the *shape identification* phase, undershooting its final value (at generation 100) by around 20%. In the following *shape definition* stage, the density sum quickly recovers, now overshooting by around 10% the final value, which is slowly reached during the *density refinement* step. It is worth looking also at the behavior of the standard deviation of the density sum (expressed in % relative to the mean value), represented as a gray solid line in the same plot of Fig. 12a, which slowly decreases towards a final value of around 9%. Thus, the width of the sums distribution within the population at the final generation 100 is around the 20 % of its value.

A second way of tracking the presence and the strength of unconstrained modes in the population is to define a *complexity indicator* that estimates the fluctuations of the phase values in real space. For a density ρ, the quantity Γ can be defined as:

$$\Gamma = 1 - \frac{|\sum_{ij}\rho_{ij}|}{\sum_{ij}|\rho_{ij}|} \tag{20}$$

The value of Γ can assume values between 0 and 1. A value of $\Gamma = 0$ implies that all the entries of the matrix ρ_{ij} have a constant phase value, i.e. there is no phase contrast in the reconstruction. As the refractive index of materials in the X-ray region of the electromagnetic spectrum is typically very close to 1, no (or very small) phase shifts are expected (there are conditions where this approximation is not valid, as discussed in section II G). Instead, the higher the value of Γ, the stronger are the density phases fluctuations in real space. The behavior of the *complexity* indicator Γ for the case $C_a = 0$ is revealed in Fig. 12a as a green solid line, which reports the mean value within the population of reconstructions. The value of Γ quickly raises from 0 (as

densities are initialized as fully real-valued, see Table 1 and section "Initialization of the starting densities") up to 0.04 and then stabilizes right above 0.02 in the first two stages of the reconstruction. During the following *shape refinement* stage, its value starts to increase again in an apparently divergent behavior, reaching its maximum value in the last generations.

The high values of Γ within the populations are a direct implication of the fluctuations in the density values visible in the first column of Fig. 12b, especially in rows 2-4, where three individual reconstructions of the population $\{\mathcal{R}^p\}$ are shown. Those fluctuations, whose shape and strength widely differ among the single individuals, can be related to the variations of the reconstructed missing intensities in the central part of the diffraction pattern, shown on the right side. First of all, the distribution of the reconstructed intensities varies strongly among the reconstructions. Second, they present strong asymmetries with respect to the center of the diffraction image, which is the sign of a remarkable deviation from a real-valued density distribution in real space.

The impact of different values of the $C_a > 0$, i.e. $C_a = 0.2$ and $C_a = 0.5$, on the reconstruction process are also reported in Fig. 12a. For what concerns the mean value of the density sum (red line in 12a) the large fluctuations of the value described before in the case of $C_a = 0$ are increasingly dampened for $C_a = 0.2$ (dashed red line) and $C_a = 0.5$ (dotted red line). In particular, the sum value quickly stabilizes within the first 15 generations for $C_a = 0.5$, with only slight adjustments towards its final value over the remaining iterations. The stronger "constraint" on the missing intensities is also directly visible in the relative standard deviation within the reconstructions populations of the sum values (gray line in 12a), which drops much quicker with the proceeding of the iterations, reaching a final value below 5 % in the case of $C_a = 0.2$ and close to 0 % for $C_a = 0.5$. It is worth noting here that, despite the substantially different behavior of the sum, the final value reached at generation 100 is almost identical for the three cases.

Similar considerations apply for the complexity indicator, reported as a green line in Fig. 12a. Its average value within the population is increasingly lower for higher values of C_a. This feature is directly visible by observing the individual reconstructions in Fig. 12b. In particular, the large-scale

Fig. 13 | Time to solution and speedup for a full reconstruction process with MPR, with different combinations of CPU threads and GPUs in use. In (**a**), a single GPU is employed with increasing number of CPU threads. The speedup is relative to the case where the number of CPU threads is set to 4. The time to solution is reported in seconds above the markers. In (**b**), The speedup is reported as function of the number of GPUs in use, with a constant number of CPU threads per GPU. The speedup value is relative to the case where only one GPU is employed.
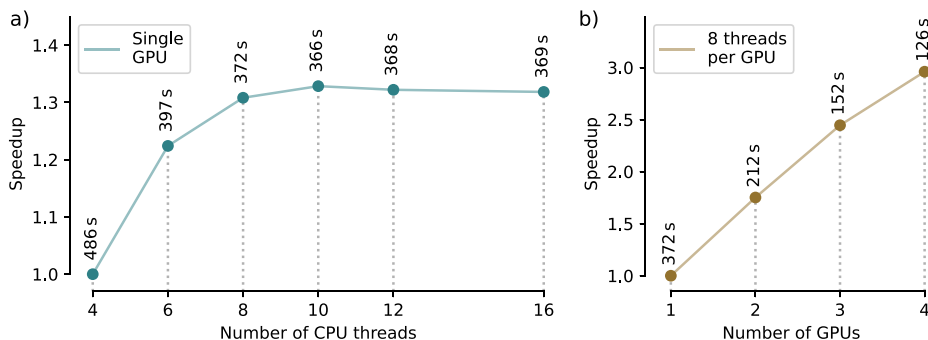
fluctuations of the phase values in real space (and the corresponding asymmetry in the central cutout of the Fourier amplitudes) are weaker, but still recognizable, for $C_a = 0.2$ (see third and fourth columns of Fig. 12b). It is instead hard to spot differences, both in real space and in Fourier amplitudes, in the case of $C_a = 0.5$.

Despite the large fluctuations observed for $C_a = 0$ (and still present in $C_a = 0.2$) the final average density presents very similar features in all the three cases (first row of 12b), both in terms of phase fluctuations in real space and in terms of retrieved missing amplitudes in Fourier space. This observation hints towards the fact that, even in the case of $C_a = 0$, despite the prominent arising of the unconstrained modes in the individual reconstructions, those modes are randomly distributed and tend to cancel out in the averaging operation.

However, the use of the average Fourier modulus via the C_a parameter unveils its importance by comparing the *replacement factor* of the crossover operation (see section "Monitoring the reconstruction process"), shown as orange line in Fig. 12a. During the *density refining* stage (and in particular from generation 50), the replacement indicator is systematically higher for higher values of C_a, meaning that the crossover operator is more effective in feeding the population with new reconstructions that perform better than the previous generation. The direct implication is that the error of the average reconstruction is lower, as reported by the upper plot in Fig. 12a, i.e. the error value is better optimized.

Computational cost and performance

The parallel execution of several individual reconstructions is a key ingredient of the MPR approach and it comes at a computational cost. In this section we briefly discuss this aspect and give a reference for the time to solution that a user may expect. The presented timing tests, summarized by Fig. 13, are performed with the MPR implementation provided by the *spring* python module, which can exploit multiple CPU cores as well as multiple Graphical Processing Units (GPUs) in a shared-memory environment. The part of the software executed on the GPU is written in CUDA® programming language[67], which is designed only for GPUs produced by NVIDIA®.

Among the three operations cyclically performed by MPR, the computational cost of the *crossover* and *selection* is, at a first approximation, negligible compared to the computational effort required in the *self-improvement*. There, in fact, thousands of FT have to be executed for each reconstruction in the populations $\{\mathcal{R}^p\}$ and $\{\mathcal{R}^p_{new}\}$.

In theory, the time to solution of a full reconstruction process with MPR is linearly dependent on the number of iterations of iterative phase retrieval algorithms (J and J_{ER}), the size of the population P and the number of generations G (see Table 1). It is furthermore higher than linearly dependent on the number of entries in the matrix containing the diffraction data, N_p^2, which is at least $N_p^2 \log(N_p^2)$, thanks to the properties of the Fast Fourier Transform employed for the FT calculation[68].

Timing and scaling tests for different combinations of number of CPU threads and GPUs are reported in Fig. 13. These tests are performed on a server equipped with two AMD Epyc™ 7302 CPUs with 16 cores each, for a total of 32 physical computing cores. The system is accelerated by four

NVIDIA GeForce RTX™ 3090 high-end consumer-level GPUs. The values of the MPR parameters that influence the time to solution have been set to their default values (see Table 1), i.e., $J = J_{ER} = 40$, $P = 128$, $G = 100$ and $N_p = 512$.

Figure 13 a reports the timing and scaling performance with a single GPU in use and increasing number of CPU threads. We decided to use 4 threads as a starting value, as this is typically the minimum number of CPU cores provided nowadays even on a lower-end consumer-level computer accelerated by a GPU. As most of the calculations are offloaded to the GPU, a strong improvement in performance should not be expected with the increasing number of CPU cores in use. Still, NVIDIA GPUs can handle communications with CPU threads in parallel, overlapping communication and computation up to some degree. In this specific hardware configuration, an appreciable gain in performance is visible for up to 8 CPU threads. Figure 13b shows, instead, the timing and scaling of a single reconstruction process for increasing number of GPUs, from 1 to 4. The number of CPU threads per GPU is kept constant and equal to 8.

The tests reported in Fig. 13 are only a rough estimate of the performance that can be expected for a given hardware configuration. They are anyway enough to conclude that, depending on the number and performance of the available GPUs, the time to solution can vary from a few to some minutes. The use of GPU accelerators designed for data centers, as well as the development expected in the upcoming years in terms of GPU performance, boosted by Artificial Intelligence, will allow a sub-minute time to solution for this MPR configuration.

Relevant speedups in the time to solution can be furthermore obtained by tuning the binning and rescaling of the experimental data. All the tests in this manuscript have been performed by rescaling the data via binning operation to a size of 512×512 (i.e., $N_p = 512$), which ensures a sufficient degree of *oversampling* to handle any practically tractable diffraction pattern. Still, in many cases, the sample is sufficiently small to rescale the diffraction data to $N_p = 256$, thus providing at least a four-fold speedup in the time to solution of MPR, reaching the sub-minute time scale.

Data availability

The raw data recorded for the experiment at the European XFEL are available at https://doi.org/10.22003/XFEL.EU-DATA-004396-00. Diffraction patterns treated in this work can be found in the SPRING software repository at https://gitlab.ethz.ch/nux/spring.

Code availability

The *spring* module is released as open source software at https://gitlab.ethz.ch/nux/spring. The documentation is available at https://share.phys.ethz.ch/~nux/software/spring/main/docs/. A version of the code and its documentation is archived at https://doi.org/10.5905/ethz-1007-832.

References

1. Schneider, J. R. Flash-from accelerator test facility to the first single-pass soft x-ray free-electron laser. *J. Phys. B: At., Mol. Optical Phys.* **43**, 194001 (2010).
2. Emma, P. et al. First lasing and operation of an ångstrom-wavelength free-electron laser. *Nat. photonics* **4**, 641–647 (2010).
3. Allaria, E. et al. Highly coherent and stable pulses from the Fermi-seeded free-electron laser in the extreme ultraviolet. *Nat. Photonics* **6**, 699–704 (2012).
4. Ishikawa, T. et al. A compact x-ray free-electron laser emitting in the sub-ångström region. *Nat. photonics* **6**, 540–544 (2012).
5. Decking, W. et al. A MHz-repetition-rate hard x-ray free-electron laser driven by a superconducting linear accelerator. *Nat. photonics* **14**, 391–397 (2020).
6. Kang, H.-S. et al. Hard x-ray free-electron laser with femtosecond-scale timing jitter. *Nat. Photonics* **11**, 708–713 (2017).
7. Milne, C. J. et al. Swissfel: the Swiss X-ray free electron laser. *Appl. Sci.* **7**, 720 (2017).
8. Duris, J. et al. Tunable isolated attosecond x-ray pulses with gigawatt peak power from a free-electron laser. *Nat. Photonics* **14**, 30–36 (2020).
9. Yan, J. et al. Terawatt-attosecond hard X-ray free-electron laser at high repetition rate. *Nat. Photonics* **18**, 1293–1298 (2024).
10. Franz, P. et al. Terawatt-scale attosecond x-ray pulses from a cascaded superradiant free-electron laser. *Nat. Photonics* **18** (2024).
11. Dold, S. et al. Melting, bubble-like expansion, and explosion of superheated plasmonic nanoparticles. *Phys. Rev. Lett.* **134**, 136101 (2025).
12. Chapman, H. N. X-ray free-electron lasers for the structure and dynamics of macromolecules. *Annu. Rev. Biochem.* **88**, 35–58 (2019).
13. Rupp, D. et al. Imaging plasma formation in isolated nanoparticles with ultrafast resonant scattering. *Struct. Dyn.* **7**, 034303 (2020).
14. Nishiyama, T. et al. Ultrafast structural dynamics of nanoparticles in intense laser fields. *Phys. Rev. Lett.* **123**, 123201 (2019).
15. Feinberg, A. J. et al. X-ray diffractive imaging of highly ionized helium nanodroplets. *Phys. Rev. Res.* **4**, L022063 (2022).
16. Niozu, A. et al. Characterizing crystalline defects in single nanoparticles from angular correlations of single-shot diffracted x-rays. *IUCr. J.* **7**, 276–286 (2020).
17. Niozu, A. et al. Crystallization kinetics of atomic crystals revealed by a single-shot and single-particle x-ray diffraction experiment. *Proc. Natl Acad. Sci.* **118**, e2111747118 (2021).
18. Colombo, A. & Rupp, D. Imaging Clusters and Their Dynamics with Single-shot Coherent Diffraction. In *Structural Dynamics with X-ray and Electron Scattering.* https://doi.org/10.1039/BK9781837671564-00172 (Royal Society of Chemistry, 2023).
19. Marchesini, S. et al. X-ray image reconstruction from a diffraction pattern alone. *Phys. Rev. B* **68**, 140101 (2003).
20. Seibert, M. M. et al. Single mimivirus particles intercepted and imaged with an X-ray laser. *Nature* **470**, 78–81 (2011).
21. Hantke, M. F. et al. High-throughput imaging of heterogeneous cell organelles with an x-ray laser. *Nat. Photonics* **8**, 943–949 (2014).
22. Van Der Schot, G. et al. Imaging single cells in a beam of live cyanobacteria with an x-ray laser. *Nat. Commun.* **6**, 5704 (2015).
23. Ekeberg, T. et al. Observation of a single protein by ultrafast x-ray diffraction. *Light Sci. Appl.* **13**, 15 (2024).
24. Colombo, A. et al. Three-dimensional femtosecond snapshots of isolated faceted nanostructures. *Sci. Adv.* **9**, eade5839 (2023).
25. Pfeifer, M. A., Williams, G. J., Vartanyants, I. A., Harder, R. & Robinson, I. K. Three-dimensional mapping of a deformation field inside a nanocrystal. *Nature* **442**, 63–66 (2006).
26. Rodenburg, J. M. et al. Hard-x-ray lensless imaging of extended objects. *Phys. Rev. Lett.* **98**, 034801 (2007).
27. Chapman, H. N. & Nugent, K. A. Coherent lensless x-ray imaging. *Nat. photonics* **4**, 833–839 (2010).
28. Miao, J., Sandberg, R. L. & Song, C. Coherent x-ray diffraction imaging. *IEEE J. Sel. Top. Quantum Electron.* **18**, 399–410 (2011).
29. Chapman, H. N. et al. High-resolution ab initio three-dimensional x-ray diffraction microscopy. *JOSA A* **23**, 1179–1200 (2006).
30. Kirian, R. A. & Chapman, H. N. Imaging of objects by coherent diffraction of x-ray free-electron laser pulses. *Synchrotron light sources and free-electron lasers: Accelerator physics, instrumentation and science applications* 1337–1397 (2020).
31. Fannjiang, A. & Strohmer, T. The numerics of phase retrieval. *Acta Numerica* **29**, 125–228 (2020).
32. Nugent, K. A. Coherent methods in the X-ray sciences. *Adv. Phys.* **59**, 1–99 (2010).
33. Thibault, P. & Elser, V. X-ray diffraction microscopy. *Annu. Rev. Condens. Matter Phys.* **1**, 237–255 (2010).
34. Grohs, P., Koppensteiner, S. & Rathmair, M. Phase retrieval: uniqueness and stability. *SIAM Rev.* **62**, 301–350 (2020).
35. Marchesini, S. Invited article: A unified evaluation of iterative projection algorithms for phase retrieval. *Rev. Sci. Instrum.* **78**, 011301 (2007).
36. Shechtman, Y. et al. Phase retrieval with application to optical imaging: a contemporary overview. *IEEE Signal Process. Mag.* **32**, 87–109 (2015).
37. Fienup, J. R. Phase retrieval algorithms: a comparison. *Appl. Opt.* **21**, 2758–2769 (1982).
38. Sekiguchi, Y., Oroguchi, T. & Nakasako, M. Classification and assessment of retrieved electron density maps in coherent x-ray diffraction imaging using multivariate analysis. *Synchrotron Radiat.* **23**, 312–323 (2016).
39. Favre-Nicolin, V., Leake, S. & Chushkin, Y. Free log-likelihood as an unbiased metric for coherent diffraction imaging. *Sci. Rep.* **10**, 2664 (2020).
40. Takayama, Y. & Nakasako, M. Similarity score for screening phase-retrieved maps in x-ray diffraction imaging–characterization in reciprocal space. *Synchrotron Radiat.* **31**, 95–112 (2024).
41. Chen, C.-C., Miao, J., Wang, C. & Lee, T. Application of optimization technique to noncrystalline x-ray diffraction microscopy: Guided hybrid input-output method. *Phys. Rev. B-Condens. Matter Mater. Phys.* **76**, 064113 (2007).
42. Yoshida, S., Harada, K., Uezu, S., Takayama, Y. & Nakasako, M. Protocol using similarity score and improved shrink-wrap algorithm for better convergence of phase-retrieval calculation in x-ray diffraction imaging. *Synchrotron Radiat.* **31**, 113–128 (2024).
43. Colombo, A., Galli, D. E., De Caro, L., Scattarella, F. & Carlino, E. Facing the phase problem in coherent diffractive imaging via memetic algorithms. *Sci. Rep.* **7**, 1–12 (2017).
44. Colombo, A. *High performance computational intelligence for coherent diffraction data analysis and imaging.* Ph.D. thesis, Università degli Studi di Milano (2018).
45. Abbey, B. et al. Keyhole coherent diffractive imaging. *Nat. Phys.* **4**, 394–398 (2008).
46. De Caro, L., Carlino, E., Vittoria, F. A., Siliqi, D. & Giannini, C. Keyhole electron diffractive imaging (kedi). *Found. Crystallogr.* **68**, 687–702 (2012).
47. Carlino, E. et al. Coherent diffraction imaging in transmission electron microscopy for atomic resolution quantitative studies of the matter. *Materials* **11**, 2323 (2018).
48. Kuster, M. et al. The 1-megapixel pnccd detector for the small quantum systems instrument at the European XFEL: system and operation aspects. *J. Synchrotron Radiat.* **28**, 576–587 (2021).
49. Mazza, T. et al. The beam transport system for the small quantum systems instrument at the European XFEL: optical layout and first commissioning results. *J. Synchrotron Radiat.* **30**, 457–467 (2023).
50. Tschentscher, T. et al. Photon beam transport and scientific instruments at the European XFEL. *Appl. Sci.* **7**, 592 (2017).
51. Mozzanica, A. et al. The Jungfrau detector for applications at synchrotron light sources and XFELs. *Synchrotron Radiat. N.* **31**, 16–20 (2018).

52. Hinger, V. et al. Advancing the Jungfrau detector toward low-energy X-ray applications. *J. Instrum.* **17**, C09027 (2022).

53. Sun, Z. et al. Ultrafast single-particle imaging with intense X-ray pulses. *Chimia* **76**, 529–529 (2022).

54. Fienup, J. R. Reconstruction of an object from the modulus of its Fourier transform. *Opt. Lett.* **3**, 27–29 (1978).

55. Luke, D. R. Relaxed averaged alternating reflections for diffraction imaging. *Inverse Probl.* **21**, 37 (2004).

56. Moscato, P., Cotta, C. & Mendes, A. et al. Memetic algorithms. *Handbook of approximation algorithms and metaheuristics* (2002).

57. Prat, E. et al. A compact and cost-effective hard X-ray free-electron laser driven by a high-brightness and low-energy electron beam. *Nat. Photonics* **14**, 748–754 (2020).

58. Pentlehner, D. et al. Rapidly pulsed helium droplet source. *Rev. Sci. Instrum* **80**, 043302 (2009).

59. Chen, Y., Chi, Y., Fan, J. & Ma, C. Gradient descent with random initialization: Fast global convergence for nonconvex phase retrieval. *Math. Program.* **176**, 5–37 (2019).

60. Sun, J., Qu, Q. & Wright, J. A geometric analysis of phase retrieval. *Found. Computat. Math.* **18**, 1131–1198 (2018).

61. Thibault, P., Elser, V., Jacobsen, C., Shapiro, D. & Sayre, D. Reconstruction of a yeast cell from x-ray diffraction data. *Acta Crystallogr. Sect. A: Found. Crystallogr.* **62**, 248–261 (2006).

62. Paganin, D. et al. *Coherent X-ray optics*. 6 (Oxford University Press on Demand, 2006).

63. Colombo, A. et al. The scatman: an approximate method for fast wide-angle scattering simulations. *J. Appl. Crystallogr.* **55**, 1232–1246 (2022).

64. Storn, R. & Price, K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**, 341–359 (1997).

65. Miao, J., Ishikawa, T., Anderson, E. H. & Hodgson, K. O. Phase retrieval of diffraction patterns from noncrystalline samples using the oversampling method. *Phys. Rev. B* **67**, 174104 (2003).

66. Moscato, P. & Cotta, C. A gentle introduction to memetic algorithms. In *Handbook of metaheuristics*, 105–144 (Springer, 2003).

67. Cuda documentation. https://developer.nvidia.com/cuda-zone (2024).

68. Cooley, J. W. & Tukey, J. W. An algorithm for the machine calculation of complex fourier series. *Math. Comput.* **19**, 297–301 (1965).

Author contributions

A.C., M.S., A.A., T.F., L.H., G.K., K.K., B.L., I.C.P., J.C.S.Z., K.S., M.L.S., Z.Sun, P.T., C.F.U., S.W., A.V.W., M.Z., C.B. and D.Rupp actively contributed in the experimental campaign at the Maloja endstation of SwissFEL (proposal 20211710). The experiment was led by M.S. and D.Rupp. A.C., M.S., K.A., M.B., R.B., R.D., S.D., F.R.N.C.M., A.M., P.M., T.M., Y.O., D.Raiser, A.S., J.A.S., B.S., S.U., Z. Shen, P.H.W.S., O.V., N.W., T.Y., M.M., D.Rupp and M.P. actively contributed in the experimental campaign at the SQS instrument of the European XFEL (proposal 4396). The experiment was led by M.P. A.C. and D.E.G. designed the original scheme of the MPR method. A.C. led the following development of SPRING with the input and support of D.Rupp. A.C. designed, implemented and optimized the SPRING software and wrote its documentation. A.C., M.S. and D.Rupp worked out the performance assessment of SPRING. A.C. and D.Rupp wrote the main body of the manuscript, with contribution and assistance from all coauthors. All authors have read and approved the manuscript.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41524-025-01661-y.

Correspondence and requests for materials should be addressed to Alessandro Colombo.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

[1]Laboratory for Solid State Physics, ETH Zurich, Zürich, Switzerland. [2]Paul Scherrer Institut, Villigen, Switzerland. [3]Max Planck Institute for the Structure and Dynamics of Matter, Hamburg, Germany. [4]The Hamburg Center for Ultrafast Imaging, Hamburg, Germany. [5]University of Oulu, Oulu, Finland. [6]European XFEL, Schenefeld, Germany. [7]Ludwig Maximilian University Munich, Garching, Germany. [8]Max Planck Institute of Quantum Optics, Garching, Germany. [9]University of Rostock, Rostock, Germany. [10]Technische Universitat Berlin, Institut fur Optik und Atomare Physik, Berlin, Germany. [11]Department of Cell and Molecular Biology, Uppsala University, Uppsala, Sweden. [12]Department for the Ecology of Animal Societies, Max Planck Institute of Animal Behavior, Konstanz, Germany. [13]Department of Biology, University of Konstanz, Konstanz, Germany. [14]Centre for the Advanced Study of Collective Behaviour, University of Konstanz, Konstanz, Germany. [15]Karlsruhe Institute of Technology (KIT), Institute of Nanotechnology, Karlsruhe, Germany. [16]Department of Applied Physics, AlbaNova University Center, KTH Royal Institute of Technology, Stockholm, Sweden. [17]Department of Physics and Astronomy, Uppsala University, Uppsala, Sweden. [18]MAX IV, Lund, Sweden. [19]Laboratory for Ultrafast X-ray Sciences, Institute of Chemical Sciences and Engineering, Ecole Polytechnique Federale de Lausanne (EPFL), Lausanne, Switzerland. [20]Physics Department, Università degli Studi di Milano, Milano, Italy. [21]Present address: The Hamburg Center for Ultrafast Imaging, Hamburg, Germany. ✉e-mail: alcolombo@phys.ethz.ch