Using automatic differentiation as a general framework for ptychographic reconstruction

Saugat Kandel, S. Maddali, Marc Allain, Stephan Hruszkewycz, Chris Jacobsen, Y. S. G. Nashed

Using automatic differentiation as a general framework for ptychographic reconstruction

Saugat Kandel,[1] **S. Maddali,**[2] **Marc Allain,**[3] **Stephan O. Hruszkewycz,**[2] **Chris Jacobsen,**[4,5,6] **and Youssef S. G. Nashed**[7,*]

[1]*Applied Physics, Northwestern University, Evanston, Illinois 60208, USA*
[2]*Materials Science Division, Argonne National Laboratory, Lemont, IL 60439, USA*
[3]*Aix Marseille Univ, CNRS, Centrale Marseille, Institut Fresnel, Marseille, France*
[4]*Advanced Photon Source, Argonne National Laboratory, Lemont, Illinois 60439, USA*
[5]*Department of Physics & Astronomy, Northwestern University, Evanston, Illinois 60208, USA*
[6]*Chemistry of Life Processes Institute, Northwestern University, Evanston, Illinois 60208, USA*
[7]*Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, Illinois 60439, USA*
[*]*ynashed@anl.gov*

Abstract: Coherent diffraction imaging methods enable imaging beyond lens-imposed resolution limits. In these methods, the object can be recovered by minimizing an error metric that quantifies the difference between diffraction patterns as observed, and those calculated from a present guess of the object. Efficient minimization methods require analytical calculation of the derivatives of the error metric, which is not always straightforward. This limits our ability to explore variations of basic imaging approaches. In this paper, we propose to substitute analytical derivative expressions with the automatic differentiation method, whereby we can achieve object reconstruction by specifying only the physics-based experimental forward model. We demonstrate the generality of the proposed method through straightforward object reconstruction for a variety of complex ptychographic experimental models.

1. Introduction

Ptychography is a coherent diffraction imaging (CDI) technique that acquires a series of intensity diffraction patterns through spatial shifts of the illumination (probe) across the sample (object) in a set of overlapping beam positions. Given a large number of overlapping beam positions, the ptychography experiment yields sufficient redundant information with which we can reconstruct the object structure to sub-beam-size spatial resolution, and even determine additional experimental parameters such as the structure of the probe itself. First proposed by Hoppe in 1969 [1], the ptychographic technique was realized experimentally and rapidly developed algorithmically in the 2000s [2–6]. By removing the typical CDI limitation that the probe size has to be larger than the sample, ptychography has enabled high resolution imaging of extended objects, making it a powerful imaging technique. As such, the ptychographic technique has found application not only as a 2D far-field diffraction imaging method but also as 2D near-field diffraction imaging method [7], a 3D Bragg imaging method [8], a 3D multislice imaging method [9], and a part of the 3D tomographic imaging method [10] including for objects beyond the depth of focus limit [11].

Imaging with ptychography involves solving the challenging phase retrieval problem, where one attempts to reconstruct an object from only the magnitude of its Fourier transform. In general, the phase retrieval problem is ill-posed [12]; solving it requires the use of oversampling and support constraints. These are typically used in an iterative projection framework that updates the object guess by applying a Fourier magnitude projection and a real-space constraint projection [13–16]. Alternatively, we can also frame phase retrieval as a nonlinear minimization problem, where

we minimize an error metric using a gradient-based approach. The gradient-based approach is flexible and can include in the forward model a large variety of the physical phenomena related to the probing light (such as partial coherence [17], source fluctuations [18], and errors in positions [19, 20]), or the detection process (such as the measurement noise [21, 22] and the finite size of the pixel [23]). As such, this method has been the focus of much recent literature, leading to the development of steepest descent methods [16, 24, 25], conjugate gradient methods [4, 16, 26], Gauss-Newton methods [27], and quasi-Newton methods [28]. These algorithms have found application in the far-field ptychographic problem not only to solve for the object alone [24, 25, 29–32] but also to additionally solve for the probe [32, 33] as well.

Gradient-based phase retrieval methods in the literature tend to rely on the availability of a closed-form expression for the gradient calculation. This closed-form expression is typically obtained by writing down an explicit expression for the error metric to minimize, then symbolically differentiating the error metric with respect to the individual input parameters [23]. Calculating the gradient in this fashion is laborious; a slight modification of the forward model usually requires a complete rederivation and algorithmic reimplementation of the gradient expressions. This becomes especially limiting if we desire to explore variations of, or introduce new capabilities to, our basic experimental methodology. As such, it is more than desirable to have an approach beyond symbolic differentiation in order to easily explore a variety of algorithms and approaches.

Automatic differentiation [34], or *algorithmic differentiation*, provides such an alternative to symbolic differentiation. This approach is based upon the observation that vector-valued functions can, in general, be interpreted as composites of basic arithmetic operations and a limited number of elementary functions (including exponentials and trigonometric operations). Differentiation of functions can then be understood as a recursive application of the chain rule of differentiation, wherein we repeatedly differentiate the same elementary functions (with known derivatives), only with different input parameters. This is a mechanistic process and can hence be performed entirely in software. Given a set of input numeric parameters, the automatic differentiation method computes the exact derivative by accumulating the numerical values of the elementary functions and their derivatives, without ever calculating the closed form derivative expression (Section 2).

While the field of automatic differentiation has a long and storied history [34], it is only recently that the emergence of deep neural network methods has driven its widespread adoption in the optimization and machine learning communities. Specifically, there has now arisen a need to perform gradient-based minimization to optimize state-of-the-art neural networks, which can be compositions of thousands or even millions of individual elementary functions. Since calculating closed-form derivatives for these is not feasible, automatic differentiation has become the tool of choice, thus leading to the recent rapid adoption and advancement of such software. In 2014, when Jurling and Fienup first proposed an automatic differentiation framework for phase retrieval [23], they commented on the lack of suitable existing software packages. Since then, we have seen the rise of multiple powerful, easy-to-use, and computationally efficient automatic differentiation frameworks such as TensorFlow [35], PyTorch [36], and Autograd [37]. More recently, we have even seen proof-of-concept demonstrations [38, 39] that successfully adapt these software packages to solve the phase retrieval problem.

In this paper, we first provide an overview of the *reverse-mode* automatic differentiation algorithm (also referred to as the *backpropagation* algorithm) for gradient calculations (Section 2), and then mathematically justify the application of this algorithm as a general framework for ptychographic phase retrieval. We demonstrate the numerical correctness of the reverse-mode automatic differentiation framework through a comparison with the popular symbolic-gradient-based ePIE method (Section 3). Finally, we demonstrate the generalizability of this framework through successful phase retrieval for increasingly complex ptychographic forward models–near-field ptychography and 3D Bragg projection ptychography–emphasizing the flexibility and

potential capacity of the approach for non-standard ptychography experiments (Section 4). This shows that the reverse-mode automatic differentiation framework allows the practitioner to update and change the forward model as necessary to better reflect the physics of the problem, without prior consideration towards how to symbolically differentiate the error metric.

2. Overview of automatic differentiation

In this section, we provide a limited overview of the automatic differentiation procedure, focusing singularly on the *reverse-mode automatic differentiation* (or *reverse-mode AD*) framework, finally motivating the application of this framework to the phase retrieval problem. Detailed and rigorous examinations of the automatic differentiation procedure–the various modes and the algorithmic frameworks–are available [34], as is a detailed exposition of the reverse-mode AD procedure in application to the phase retrieval problem [23].

To demonstrate the idea of automatic differentiation, we first consider differentiable functions $f, \phi_1, \phi_2, \phi_3 : \mathbb{R} \to \mathbb{R}$ with the function composition $f = \phi_3 \circ \phi_2 \circ \phi_1$, with the assumption that ϕ_1, ϕ_2, and ϕ_3 are elementary functions with priorly available individual function derivatives $\frac{d\phi_1}{dx}$, $\frac{d\phi_2}{dx}$, and $\frac{d\phi_3}{dx}$ (for any $x \in \mathbb{R}$). At a given point $x = c$, we can compute the value of f through the sequence of successive evaluations shown in Table 1. We refer to this sequence of computations as the *forward pass*.

Table 1. Schema for reverse-mode automatic differentiation.

Forward Pass	Backward Pass	Computational graph
$v_0 = c$	$\bar{v}_3 = 1$	
$v_1 = \phi_1(v_0)$	$\bar{v}_2 = \bar{v}_3 \dfrac{dv_3}{dv_2}$	
$v_2 = \phi_2(v_1)$		
$v_3 = \phi_3(v_2)$	$\bar{v}_1 = \bar{v}_2 \dfrac{dv_2}{dv_1}$	
$f(c) = v_3$		
	$\bar{v}_0 = \bar{v}_1 \dfrac{dv_1}{dv_0}$	
	$\bar{c} = \bar{v}_0$	

The gradient is evaluated through an accumulation of intermediate values calculated for the individual elementary functions and their derivatives.

In the evaluation trace shown in Table 1, we follow the notation in the literature [34, 40, 41] and index the variables stored in the memory as v_i, with $i \leq 0$ for the input variables, and $i > 0$ for the intermediate computed variables. To calculate the derivative of f at $x = c$, we use the chain rule of differentiation:

$$\frac{df}{dx}\bigg|_{x=c} = \frac{d\phi_3}{dx}\bigg|_{x=\phi_2(\phi_1(c))} \cdot \frac{d\phi_2}{dx}\bigg|_{x=\phi_1(c)} \cdot \frac{d\phi_1}{dx}\bigg|_{x=c}.$$

To evaluate this derivative, we perform a backward pass (*i.e.*, a reversed sequence of computations) during which we associate each intermediate variable from the forward pass, v_i, with a new adjoint variable $\bar{v}_i = \frac{df}{dv_i}$; we then evaluate $\bar{v}_i$ from $i = 3$ to $i = 0$. This is shown in Table 1. Noting that

$$\frac{dv_{i+1}}{dv_i} = \frac{d\phi_{i+1}}{dx}\bigg|_{x=v_i}, \tag{1}$$

we can see that each step in the backward pass requires as input not only the derivative values calculated in the previous steps, but also the intermediate variables calculated during the forward pass. This is illustrated in the computational graph in Table 1. Thus, to calculate the final derivative value, we need to:

1. identify the derivatives of the elementary functions ϕ_1, ϕ_2, and ϕ_3,

2. perform a forward pass evaluation of the function and store the intermediate values calculated, and

3. perform a backward pass to accumulate the final derivative.

This reverse-mode automatic differentiation scheme calculates the exact derivative value (up to floating point errors) without relying on the closed form expression of the derivative; instead, it relies only on the structure of the computational graph.

From Table 1, we can see that for the reverse-mode AD gradient calculation, the values of the intermediate variables calculated during the forward pass are shared with the backward pass; each elementary expression is only computed once but reused multiple times. This ensures that the gradient calculation is very efficient computationally. In fact, for functions of the form $f : \mathbb{R}^n \rightarrow \mathbb{R}$, if the function evaluation (forward pass) requires

$$\text{ops}(f) = N$$

floating point operations, then the number of floating point operations required for the gradient evaluation (backward pass) is always given by

$$\text{ops}(\nabla f) = k \cdot \text{ops}(f) = kN, \quad \text{with } 0 < k < 6 \text{ a constant,}$$

such that k typically satisfies $k \in [2, 3]$, no matter the value of n [34, 40]. In other words, for reverse-mode AD, barring memory limitations, the time required to calculate the gradient ∇f is *always* within an order of magnitude of the time required to calculate the function value itself. This is known as the *cheap gradient principle*.

As such, the reverse-mode AD procedure is ideally suited to provide gradient-based iterations aiming at solving of optimization problems. A quintessential case in point is machine learning with neural networks–the 'training step' in large neural networks boils down to the numerical optimization of an error metric involving a large number of functional compositions and up to millions of input parameters. In such a situation, the derivation of the closed-form gradient with pen and paper is simply out of reach. Thus, the implementation of gradient descent procedures that use the AD framework has been key to the recent meteoric rise in machine learning applications.

In contrast to the neural network case, a typical error metric for the phase retrieval problem only involves a limited number of functional compositions, and the closed form gradients can be calculated manually. This has been demonstrated in prior phase retrieval literature: the Wirtinger flow method and its variants [24, 25, 42], the PIE family of methods [32], and numerous other methods proposed in the literature [14, 16, 22, 29, 43] are examples of gradient-based descent procedures that rely on such closed-form gradient expressions. However, the impressively flexible AD framework not only simplifies these selfsame gradient calculations, but also allows us to modify the forward model; this empowers us to address the full range of problems that are related to phase retrieval. Consequently, we expect that the AD framework should also greatly benefit the phase retrieval community.

As we demonstrate in Section 3 for the ptychographic problem, the error metric for the phase retrieval problem is a scalar-valued multivariate objective function of complex variables, $f : \mathbb{C}^n \rightarrow \mathbb{R}$ [16, 44]. To minimize such a function using a gradient descent approach, we adopt

the Wirtinger gradient descent formalism [45–47]. For some $z \in \mathbb{C}^n$, the Wirtinger gradient operator is defined as

$$\boldsymbol{\nabla} f(z) = \frac{\partial f}{\partial z^*} = \frac{1}{2} \left(\frac{\partial f}{\partial \mathfrak{R}[z]} + i \frac{\partial f}{\partial \mathfrak{I}[z]} \right), \qquad (2)$$

where $i = \sqrt{-1}$, z^* is the element-wise complex conjugate of z, $\mathfrak{R}[z]$ and $\mathfrak{I}[z]$ are respectively the real and imaginary parts of z, and $\frac{\partial f}{\partial \mathfrak{R}[z]}$ and $\frac{\partial f}{\partial \mathfrak{I}[z]}$ the componentwise partial derivatives with respect to the real and imaginary parts of z. Since the partial derivatives $\frac{\partial f}{\partial \mathfrak{R}[z]}$ and $\frac{\partial f}{\partial \mathfrak{I}[z]}$ are both individually real-valued, we can calculate them by separately using the reverse-mode AD framework in the same fashion as shown in Table 1. This ensures that the gradient calculation procedure is very efficient, with the time cost once again comparable to that for the objective function itself.

3. Validation: Far-field transmission ptychography

In this section, we first establish a forward model for far-field transmission ptychography. We then use reverse-mode AD to set up a gradient descent procedure that is, by construction, equivalent to the ePIE reconstruction method [32] which uses a closed-form gradient expression. We compare these frameworks numerically and establish that the automatic differentiation framework calculates gradient values that are identical to those calculated via the closed-form gradient expressions. This comparison to the well-known ePIE method serves to establish the validity of the AD framework for phase retrieval. The ePIE method, however, cannot be used out-of-the-box for object reconstruction once we modify the ptychographic forward model–as such, it is not well-suited for use with the AD framework. Instead, we demonstrate that we can use the flexible and easy-to-use state-of-the-art accelerated adaptive gradient descent algorithms (like the Adaptive Moment Estimation or *Adam* algorithm [48]) that are commonly available out-of-the-box with AD software to efficiently solve the ptychographic reconstruction problem.

3.1. Forward model for far-field transmission ptychography

In far-field transmission ptychography, an unknown object is illuminated with a coherent beam, called the probe, which is localized to a small area on the object. The intensity of the wavefront transmitted through the object is then measured at the far field by a pixel-array detector. The beam is used to raster scan the object in a grid of J spatially overlapping illumination spots, generating a set of J diffraction patterns at the detector plane. The forward model for far-field ptychography consists of the following steps:

1. The complex valued object transmission function O is approximated by a total of N pixels in the object plane. For the illumination position $\boldsymbol{r}_j$, a shift operator $\boldsymbol{S}_j$, with an $M \times N$ matrix representation, extracts the M object pixels illuminated by the probe beam $\mathcal{P}$ containing M pixels in the object plane. The thus transmitted wave function is represented by the exit wave $\psi_j \in \mathbb{C}^M$:

$$\psi_j = \mathcal{P} \odot (\boldsymbol{S}_j O) \quad \text{for } j = 1, 2, ..., J; \qquad (3)$$

where $\odot$ is the element-wise Hadamard product operator.

2. Each transmitted exit wave ψ_j propagates to the far field detector plane, where the detector then records a real-valued intensity pattern containing M pixels. If there is no noise fluctuation, the expected intensity of this jth wave-field at the detector plane reads

$$h_j = |\mathcal{F}\psi_j|^2 + \nu_j \qquad (4)$$

where v_j is the expected level of background events, $\mathcal{F}$ is the matrix representation of the two-dimensional digital Fourier transform, and $|\cdot|$ extracts the modulus element-wise for a complex-valued vector.

3. The model in Eq. (4) describes the behavior of the detected intensity in average. During an experiment, each diffraction pattern is subject to random fluctuations produced by the instrumental and the Poisson counting fluctuations (shot-noise). If one further assumes that the instrumental (thermal) noise is negligible, a Gaussian additive perturbation is usually accurate enough to describe how the counting fluctuation plagues the square-root of the measured intensity (see, for instance, [22] and references therein):

$$y_j^{1/2} = h_j^{1/2} + \varepsilon_j \tag{5}$$

where ε_j is a centered Gaussian random vector.

The above relations are the forward (observation) model that predicts how the observations $\{y_j^{1/2}\}_{j=1}^J$ behave when the sample and the probe are jointly given. The inversion/reconstruction step simply aims at retrieving both these quantities from the observations. For that purpose, the maximum likelihood estimator defines the solution of this reconstruction step via a minimization problem that can be easily derived from the fluctuation model of Eq. (5). In our case it leads to

$$(O^\star, \mathcal{P}^\star) \in \mathrm{argmin}_{O,\mathcal{P}} g(O, \mathcal{P}) \tag{6}$$

where g is a separable fitting-function that reads

$$g = \sum_{j=1}^J g_j \quad \text{with} \quad g_j(O, \mathcal{P}) := ||y_j^{1/2} - h_j^{1/2}(O, \mathcal{P})||^2 \tag{7}$$

where $||\cdot||$ denotes the usual Euclidian norm in $\mathbb{C}^N$, and $h_j^{1/2}$ and $y_j^{1/2}$ denote the componentwise square roots of the vectors h_j and y_j respectively. While this fitting-function has been used in the phase retrieval literature for decades, there exist some alternative noise models that would in turn provide alternative functionals to minimize [4, 22]. In any case, the optimization step derived from the noise model involves a large-scale phase retrieval problem that is rather challenging. Phase retrieval problems are NP-hard problems because of their inherent non-convex structure and the search for good and computationally efficient heuristics is still an open problem. The overlapping between successive scanning probe is however recognized as a key factor that helps in preventing stagnation for standard, derivative-based iterations. Thanks to AD, these derivatives can be efficiently computed and further used by one of the iterative solvers that will be discussed in the next section.

3.2. Iterative ptychographic reconstruction with reverse-mode AD: ePIE and Adam

We first consider the *extended Ptychographical Iterative Engine* or ePIE algorithm [5], which can be summarized in a generalized form in Algorithm 1.

With $b = 1$, Algorithm 1 exhibits one iteration of the ePIE procedure, during which it uses the information in single diffraction patterns to cyclically update the current probe and object estimates. The core (or "Engine") of this update is the correction (Eq. (10)) computed from the considered probe position $j \in \mathfrak{J}'$ using the derivative of g with respect to the sample O and the probe $\mathcal{P}$. In its original formulation, the ePIE procedure uses the well-known closed form expression of these derivatives (see Eqs. (6) and (7) in [5]) to compute these updates.

Alternatively, we can use the reverse-mode AD procedure for the numerical computation of these derivatives by relying on the Wirtinger formalism (Eq. (2)) for complex derivatives. The

---

Algorithm 1 Generalized ePIE gradient descent iteration

Require: Probe and object initial guesses $\mathcal{P}^0$ and O^0.

Require: Minibatch size b. For the original ePIE procedure, $b = 1$.

1:   Randomly shuffle the probe positions $\mathfrak{J} = \{1, 2, ..., J\} \xrightarrow{\text{shuffle}} \mathfrak{J}'$.

2:   **for** $k = 1$ to J/b **do**

3:      With the probe positions indexed as $j \in \mathfrak{J}'$, calculate the partial derivatives:

$$\partial_O g^k := \sum_{j=b(k-1)+1}^{bk} \partial_O g(O_j^{k-1}) \quad \text{and} \quad \partial_{\mathcal{P}} g^k := \sum_{j=b(k-1)+1}^{bk} \partial_{\mathcal{P}} g(\mathcal{P}_j^{k-1}) \tag{8}$$

4:      Calculate the step sizes

$$\alpha_O^k := 1/\left\| \left\| \mathcal{P}^{k-1} \right\|^2 \right\|_{\max} \quad \text{and} \quad \alpha_{\mathcal{P}}^k := 1/\left\| \sum_{j=b(k-1)+1}^{bk} \left\| \mathcal{S}_j^* O^{k-1} \right\|^2 \right\|_{\max}. \tag{9}$$

     where $\|\cdot\|_{\max}$ denotes the maximum element of the enclosed vector.

5:      Set

$$O^k = O^{k-1} - \alpha_O^k \partial_O g^k \quad \text{and} \quad \mathcal{P}^k = \mathcal{P}^{k-1} - \alpha_{\mathcal{P}}^k \partial_{\mathcal{P}} g^k \tag{10}$$

6:   **end for**

---

derivative with respect to the real and imaginary parts of the complex variable are evaluated separately, and they are then accumulated according to the equations

$$\partial_O g^k := \frac{1}{2} \left(\frac{\partial}{\partial g} \Re[O] \Big|_{O^{k-1}} + i \frac{\partial g}{\partial \Im[O]} \Big|_{O=O^{k-1}} \right) \tag{11}$$

and

$$\partial_{\mathcal{P}} g^k := \frac{1}{2} \left(\frac{\partial g}{\partial \Re[\mathcal{P}]} \Big|_{\mathcal{P}=\mathcal{P}^{k-1}} + i \frac{\partial g}{\partial \Im[\mathcal{P}]} \Big|_{\mathcal{P}=\mathcal{P}^{k-1}} \right). \tag{12}$$

The AD version of ePIE is mathematically equivalent to the original ePIE [5]; thus, both the algorithms should generate the same sequence of iterates. We refer to the AD version of the ePIE solver as AD-ePIE.

In Fig. 1 we present a simplified representation of the computational graph structure that generates the AD-ePIE iterates (see also Table 1). We can see from the figure that the modular reverse-mode AD framework is agnostic as to the choice of the error metric, the update scheme, and even to the forward model itself; it generalizes straightforwardly to any variations in these. We demonstrate this in Section 4, where we vary both the forward model and the update scheme.

A natural extension to the ePIE/AD-ePIE algorithm is to use several probe positions to compute a single update: that is, to increase the minibatch size $b \geq 1$ in Algorithm 1. In this perspective, Algorithm 1 belongs to a larger family of iterates that takes advantage of the natural partitioning of the dataset [22]. Similar or identical optimization strategies are indeed well known under different names in several communities, such as 'ordered-subset' in image processing [49], 'incremental gradient' [50, Chapter 1] in the optimization literature, or 'stochastic gradient' for neural-network learning algorithms [51].

A salient feature of ePIE/AD-ePIE is that the chosen step sizes α_O^k and $\alpha_{\mathcal{P}}^k$ (Eq. (9)) for the object and probe updates respectively equate to the inverse of the *Lipschitz constant* of the partial
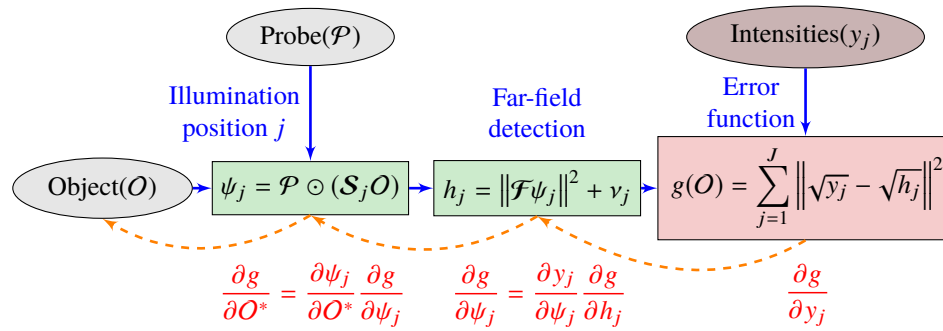
Fig. 1. Simplified representation of the forward and backward passes for the object update for far-field ptychography. The solid blue arrows indicate the forward pass direction. The dashed orange arrows indicate the backward pass direction.

gradients $\partial_O g^k$ and $\partial_P g^k$ [33]. This choice of step sizes is well-known in the optimization literature, and is particularly useful in a batch gradient descent setting where the derivatives in the update (Eq. (10)) are calculated using *all* available probe positions at once (setting $b = J$). As an example, with a steepest-descent iteration algorithm [50] these step sizes would then ensure global convergence (toward a local minimizer). However, the Lipschitz constants are not usually known *a priori*; they generally have to be carefully derived using the closed-form gradient expressions [33]. Changes to the forward model, or the inclusion of additional terms in the error metric (e.g. regularizers), can thus require a re-deriving of both the closed form gradient expression and the Lipschitz constant. As such, using the Lipschitz constants to calculate the step sizes would negate the flexibility that is the hallmark of the AD procedure.

To circumvent this limitation and enable phase retrieval within the AD framework, we can substitute the ePIE/AD-ePIE method with a choice of state-of-the-art adaptive gradient descent algorithms that are widely used in the machine learning literature [52], such as Momentum, Nesterov Momentum, Adagrad, Adadelta, RMSProp, or Adam. The performance of these exotic methods specifically in phase retrieval applications is a new but promising area of research, with the recent literature [32, 53, 54] demonstrating that momentum-based accelerated gradient descent methods converge to a solution faster than standard gradient descent methods. In this work, we use the Adam (Adaptive Moment Estimation) gradient descent procedure [48] that uses a momentum-like acceleration and, crucially, does not rely on the Lipschitz constant for the gradient descent. As such, the Adam method is robust to changes in the error metric and therefore well-suited to phase retrieval applications.

The Adam optimization method is available out-of-the-box in the commonly used AD toolsets TensorFlow [35] and PyTorch [36], so that it can be used directly without first implementing the underlying algorithm. Nevertheless, for the sake of clarity, we present in Appendix A the parameter initialization step and the variable update step that make up the Adam optimizer for the object variable (the probe updates are similarly calculated). To solve the ptychography problem, we incorporate the Adam optimizer in a fashion identical to Algorithm 1, but with steps 4 and 5 substituted with the update computed from the Adam optimizer. In Section 3.3, we provide a minimal comparison of performance of the Adam method to the ePIE/AD-ePIE method by iterating through the individual probe positions and calculating the object and probe updates separately per probe position. In Section 3.4, we present the search strategy used for the choice of the Adam *hyperparameters* (viz., the initial probe update step size, the initial object update step size, and the minibatch size) adopted in this work, and provide heuristic guidelines for the choice

of hyperparameters to achieve computationally efficient ptychographic reconstruction. Beyond our heuristic approach, a more detailed evaluation of the application of the Adam algorithm for phase retrieval is an important research question but is beyond the scope of this paper.

3.3. Numerical results

For a numerical validation of the reverse-mode AD procedure, we simulated a far-field transmission ptychography experiment with an incident x-ray of energy 8.7 keV. We used the 'Mandrill' and 'Cameraman' images, each 128×128 pixels in size, as the test object magnitude and phase respectively, and embedded the test object at the center of a simulation box of size 190×190 pixels (with tight support). We illuminated the test object with a complex-valued probe function approximated using an array of size 64×64 pixels and with a total integrated intensity of 10^6 photons at the object plane. The probe function was obtained by propagating the exit wave from a square aperture of width $7\,\mu m$ to a distance of $\zeta = 15$ cm so that it contained diffraction fringes characterized by $\sqrt{\lambda\zeta} = 4.6\,\mu m$. The raster grid for the object scan was obtained by translating the probe latitudinally and longitudinally in steps of $3.5\,\mu m$ (6 pixels). The real-space pixel grid was obtained by assuming exact Nyquist sampling with respect to a detector with a pixel pitch of $55\,\mu m$ placed at a distance of 14.6 m from the object. We thereby generated 484 far-field diffraction patterns at the detector position, then incorporated Poisson noise into these diffraction patterns to get the simulated far-field data points.
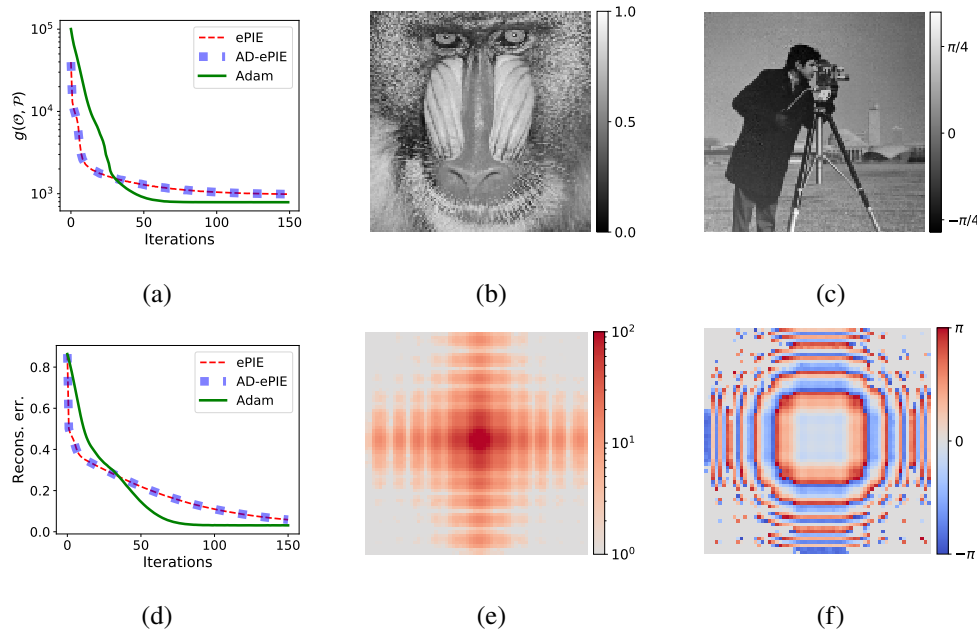

Fig. 2. The value of the (a) error metric $g(O, \mathcal{P})$ and (d) the normalized reconstruction error for the object for the ePIE method, the AD-ePIE method, and the Adam method for the far-field ptychography experiment. Adam reconstructions for the (b) object magnitude, (c) object phase, (e) probe magnitude, and (f) probe phase. The normalized root-mean-squared reconstruction error (NRMSE) was 0.03 for the object and 0.02 for the probe.

Our implementation of the AD-based ptychographic reconstruction framework follows previous work [38, 39] and uses the TensorFlow [35] machine learning library for the gradient calculation.

For the reconstruction, we used a $7\,\mu$m (12 pixels) square aperture as the initial probe guess, and a random complex array as the initial object guess. Using the same starting parameters, we performed 150 iterations through the data set (with the probe value held constant for the first iteration) for the ePIE, AD-ePIE, and the Adam methods, with the same randomized sequence of diffraction patterns for all three algorithms. For the Adam method, we used initial update step sizes 0.1 and 0.001 for the probe and object respectively.

As we demonstrate in Fig. 2, the AD-ePIE method followed the same reconstruction path as the ePIE method for both the error metric $g(O, \mathcal{P})$, and the normalized object reconstruction error calculated per pixel from the ground truth [55]. This demonstrates that the reverse-mode AD framework described in this paper calculates gradient values numerically identical to those calculated using the closed form symbolic derivatives [5]. Additionally, reconstruction using the Adam algorithm also converges to the same final probe and object structures as the ePIE algorithm: the final Adam object and probe structures only differ by 5% and 4% respectively from the corresponding final ePIE structures.

The reconstruction algorithms implemented in this work do not use any advanced domain decomposition techniques to parallelize the ptychographic reconstruction [38, 56]; they iterate sequentially through minibatches of diffraction patterns. When these algorithms were implemented to run on a single 3.00 GHz Intel Xeon processor with a minibatch $b = 1$, the runtime for each minibatch update for the forward model and the ePIE algorithm (implemented with the *numpy* library in Python) were found to be ≈ 1 ms, and ≈ 1.2 ms respectively. The corresponding runtime for the AD-ePIE and Adam algorithms (implemented with TensorFlow) were found to be ≈ 4 ms, and ≈ 5.8 ms respectively. Disregarding backend discrepancies, this is in agreement with the expected computational costs described in Section 2—the ePIE algorithm implements the symbolic gradient expressions directly and has a runtime comparable to the forward model itself. The AD-ePIE method has a runtime that is within a small scaling factor of that for the forward model, and the Adam algorithm requires some additional computation for the necessary moment updates (Algorithm 2). In practice, the Adam algorithm converges to the final solution in 80 iterations (≈ 225 s) while the ePIE algorithm requires 150 iterations (≈ 87 s) to achieve convergence. Once we use larger minibatch sizes, however, as we demonstrate in Section 3.4, the Adam algorithm can converge to a solution much faster than the basic ePIE approach.

3.4. Choosing Adam hyperparameters for efficient ptychographic reconstruction

In this work, for reconstructions with the Adam algorithm (Algorithms 1 and 2), we manually supply three key hyperparameters: the minibatch size b, and the initial Adam object and probe update step sizes α_O^A and $\alpha_{\mathcal{P}}^A$. For optimal performance of these reconstruction algorithms, we need to choose hyperparameter values that simultaneously optimize the hardware utilization, the time cost per minibatch update, and the total number of minibatch updates required to converge to a solution—this makes for a difficult research problem [57]. As such, in this work, we take a primarily heuristic two-step approach: 1) we pick a minibatch size that optimizes the hardware utilization and time cost per minibatch update, and then, with this number size fixed, 2) we perform a gridsearch to identify the values for α_O^A and $\alpha_{\mathcal{P}}^A$ that reduce the number of iterations required for the reconstruction.

Ptychographic reconstruction with $b = 1$, as used in Section 3.3, has optimally low time cost per minibatch update, but is not suitable for implementation on parallel computing hardware. In fact, when implemented for use in a nVIDIA K40 GPU, this reconstruction procedure has minimal GPU utilization ($< 10\%$), and shows no noticeable improvement over the CPU version in the time cost per minibatch update. This suggests the use of larger minibatches for optimal GPU utilization. As we increase the minibatch size, however, the number of intermediate values stored in the memory also increases proportionally. As such, we need to choose a minibatch size that ensures that the resulting computational graph, including all the intermediate values calculated in

the forward and backward passes, fits within the GPU memory. For the ptychographic dataset simulated in Section 3.3, we can set $b = 100$; this choice combines much greater GPU utilization ($> 50\%$) with minimal time cost per minibatch update (≈ 5.2 ms), and also maintains stochasticity in the gradient directions computed per update.

With the minibatch size fixed, we can then perform a gridsearch to identify the α_O^A and $\alpha_{\mathcal{P}}^A$ values that lead to fast ptychographic reconstruction. While there exist sophisticated methods to modify these parameters within the descent procedure [48], our algorithms use the more common approach that uses fixed step sizes. Similarly, while there exist *early stopping* methods to identify the convergence of the reconstruction procedure [58], we simply monitor the error metric $g(O, \mathcal{P})$ [Eq. (7)] for a fixed, large, number of iterations and choose the initial step sizes that give us the lowest values for the error metric.

To examine how the choice of these hyperparameters affects the reconstruction process at different experimental conditions, we followed the procedure in Section 3.3, but with the integrated probe intensity set to 10^3 and 10^9 photons respectively to generate two additional ptychographic datasets. To reconstruct the object and probe from each dataset, we used the Adam algorithm with a minibatch $b = 100$, and performed a gridsearch on a logarithmic grid of initial probe update step sizes $[100, 10, 1, 0.1, 0.01]$ and initial object update step sizes of $[10, 1, 0.1, 0.01, 0.001]$.

In Fig. 3, we show the object NRMSE obtained for each of these hyperparameter sets after 1500 iterations (a runtime of ≈ 30s). We can see that $\alpha_O^A = 0.01$ remains an optimal object update step size for all three experiments. The probe update step size requires more careful tuning, with $\alpha_{\mathcal{P}}^A = 1.0$ a good starting choice for the tuning procedure. For $(\alpha_O^A, \alpha_{\mathcal{P}}^A) = (0.01, 1.0)$, after 1500 iterations, we obtained object and probe $[(O, \mathcal{P})]$ reconstruction errors of $(0.28, 0.51)$, $(0.03, 0.02)$, and $(0.004, 0.0004)$ respectively for integrated probe intensities of 10^3, 10^6, and 10^9. In comparison, when we stop the ePIE algorithm ($b = 1$) after 150 iterations (≈ 87 s) without ensuring convergence, we get the corresponding reconstructions errors of $(0.28, 0.50)$, $(0.06, 0.04)$, and $(0.05, 0.008)$. This demonstrates that the Adam algorithm, with the provided hyperparameter search strategy, is a fast and robust choice for ptychographic reconstruction within the AD framework.

4. Applications to other ptychographic forward models

In this section, we apply the reverse-mode AD framework developed in Section 3 for the far-field transmission ptychography experiment to the more complex experimental regimes of near-field ptychography and multi-angle Bragg ptychography.

4.1. Near-field ptychography

Just as in the far-field case, the near-field ptychography experiment uses a probe beam to raster scan the object in a grid of spatially overlapping illumination spots, generating a set of J exit waves after the probe-object interaction. The detector, however, is placed at a distance z from the object so that the Fresnel number satisfies the condition

$$W^2/\lambda z \gg 1,$$

where λ is the wavelength of the incident probe, and W is the lateral extent of the illuminated area [7] at any given probe position. This is the high Fresnel number condition for the near-field regime. In this regime, given a probe $\mathcal{P}$, an object O, a lateral shift operator $\mathcal{S}_j$, and an exit wave $\psi_j = \mathcal{P} \odot (\mathcal{S}_j O)$, the expected intensity at the detector plane is given by

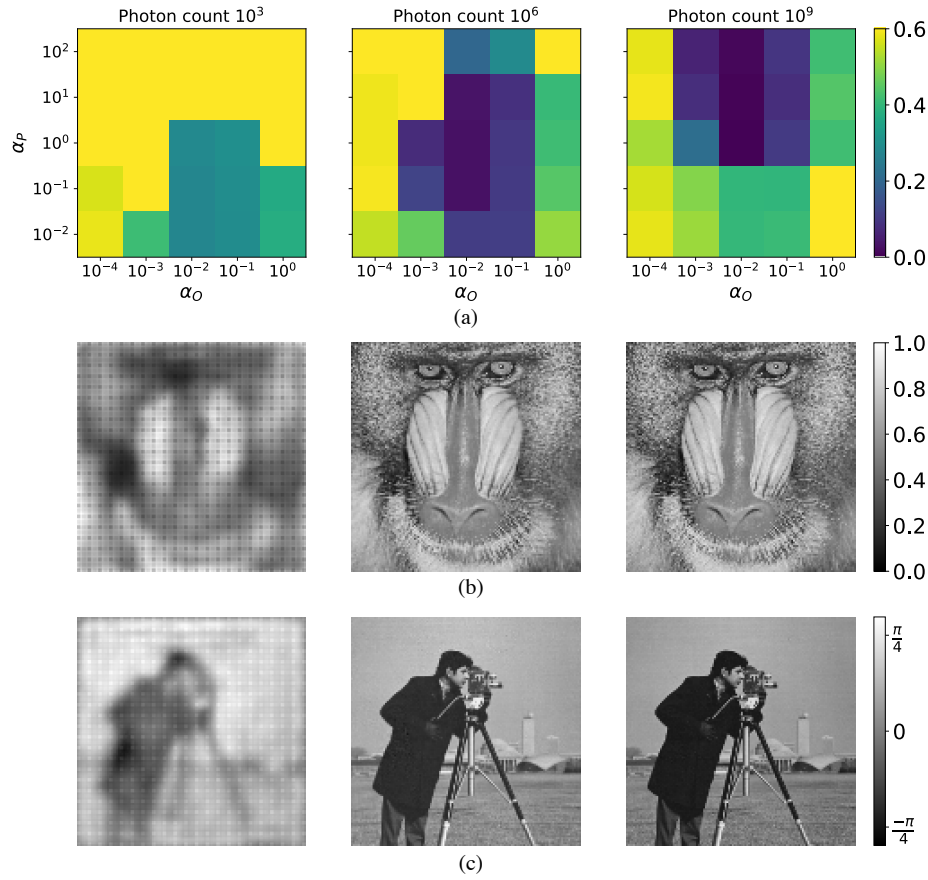$$h_j = \left\| D_z\{\psi_j\} \right\|^2 + \nu_j \tag{13}$$

Fig. 3. (a) Reconstruction errors (NRMSE) obtained after 1500 iterations of the Adam algorithm (with $b = 100$) as a function of initial object and probe update step sizes, for incident probe integrated intensities of 10^3 (left), 10^6 (mid), and 10^9 (right) respectively. For clarity, the NRMSEs plotted are capped at 0.6. (b) Final object magnitudes and (c) phases obtained for $\alpha_O^A = 0.01$ and $\alpha_\varphi = 1.0$. For the low photon count of 10^3 (left), the reconstructed structures are deteriorated due to the raster grid artifact.

where ν_j is the expected background, and D_z is the Fresnel free-space propagator for the distance z defined by the expression

$$D_z\{\psi_j\} = \mathcal{F}^{-1}\left[(\mathcal{F}\psi_j) \odot \exp\left\{\frac{-iz\lambda}{4\pi}\left(q_x^2 + q_y^2\right)\right\}\right] \tag{14}$$

where (q_x, q_y) the reciprocal space coordinates [59]. The experimentally measured intensity patterns are again denoted as y_j. Detailed characterizations of this near-field ptychography experiment can be found in [7], [60], [61], and [62].

We simulated a near-field ptychography experiment using an incident 8.7 keV probe beam approximated using an array of size 512×512 pixels at the object plane, with the pixel pitch set to $0.6\,\mu\text{m}$. The probe was initialized as a Gaussian with a FWHM of $19\,\mu\text{m}$ (50 pixels), then passed through a diffuser and modeled as a speckle pattern with an average flux of 10^4 photons/pixel. The simulated object consists of the 'Mandrill' and 'Cameraman' images of size

192×192 pixels, modeled as the object magnitude and phase respectively. The raster grid was obtained by translating the object in the horizontal and vertical in steps of $10\,\mu$m (44 pixels), generating a total of 25 diffraction patterns at a detector placed 4.7 cm from the object plane. We added Poisson noise to the diffraction patterns to obtain our simulated dataset.

(a)   (b)

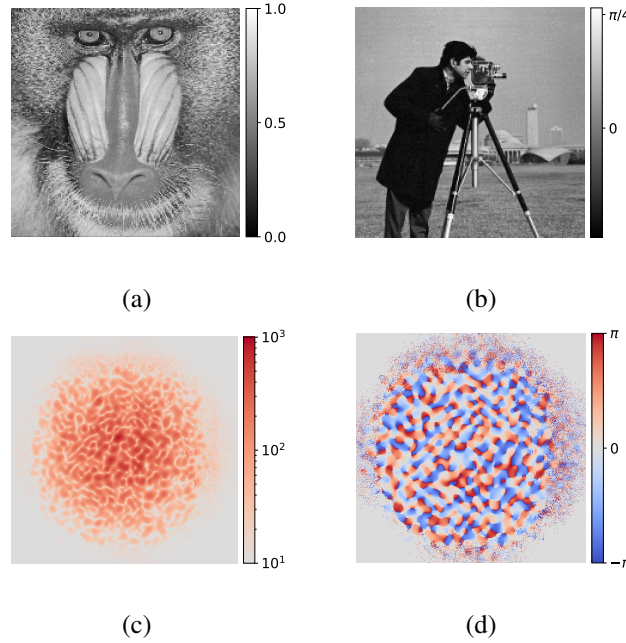(c)   (d)

Fig. 4. Near-field ptychographic reconstruction with reverse-mode AD. Shown here are the successfully reconstructed (a) object magnitude, (b) object phase, (c) probe magnitude, and (d) probe phase. The object and probe were reconstructed with overall NRMSEs of 0.01 and 0.12 respectively.

For the near-field reconstruction, we obtained an initial probe estimate by backpropagating the average measured intensity, $y_{\text{avg}} = (\sum_{j=1}^{J} y_j)/J$, to give

$$\mathcal{P}^0 = \mathcal{F}^{-1}\left[(\mathcal{F}\sqrt{y_{\text{avg}}}) \odot \exp\left\{\frac{iz\lambda}{4\pi}(q_x^2 + q_y^2)\right\}\right].\tag{15}$$

The object was initialized as a 192×192 array of random complex numbers. To solve for the probe and object structures, we again used the Adam optimizer in the approach outlined in Algorithm 1, but considering 5 probe positions per iteration (*i.e.* with a minibatch size of 5). The initial Adam update step sizes were set to $\alpha_O^A = 0.01$ and $\alpha_{\mathcal{P}}^A = 10$ for the object and probe respectively. The minibatch size and initial step size were chosen as described in Section 3.4.

After 10,000 iterations of Adam gradient descent, we obtained the object and probe reconstructions shown in Fig. 4. The object was reconstructed with an NRMSE of 0.01, and the probe with an NRMSE of 0.12. The discrepancies in the reconstructed probe were contained at the edges of the full-field probe, in regions which did not interact with the object and were thus unconstrained. These reconstructions demonstrate that the straightforward generalization of the reverse-mode AD reconstruction framework from the far-field ptychography case (Fig. 1) to the near-field ptychography case successfully solves the near-field ptychography phase retrieval problem.

4.2. Multi-angle Bragg ptychography

Multi-angle Bragg projection ptychography (maBPP) [63] is a ptychographic experiment that allows for two degrees of freedom in the scan parameters: 1) the choice of the planar scan positions for the usual two-dimensional ptychographic scan, and 2) the choice of angular scan positions corresponding to small object rotations of a crystalline object oriented to satisfy a Bragg diffraction condition. The maBPP experiment uses the far-field ptychography setup, but with the detector placed to measure a crystalline Bragg peak, typically displaced from the direct beam by tens of degrees. The detector records a set of two-dimensional (2D) coherent diffraction patterns at overlapping scan positions at one or more angles near the Bragg diffraction condition. The diffraction patterns are then used to reconstruct a three-dimensional (3D) strain-sensitive image of an extended crystalline sample [64]. An example experimental geometry with the exit beam at a 60° angle to the incident beam direction is shown in Fig. 5.
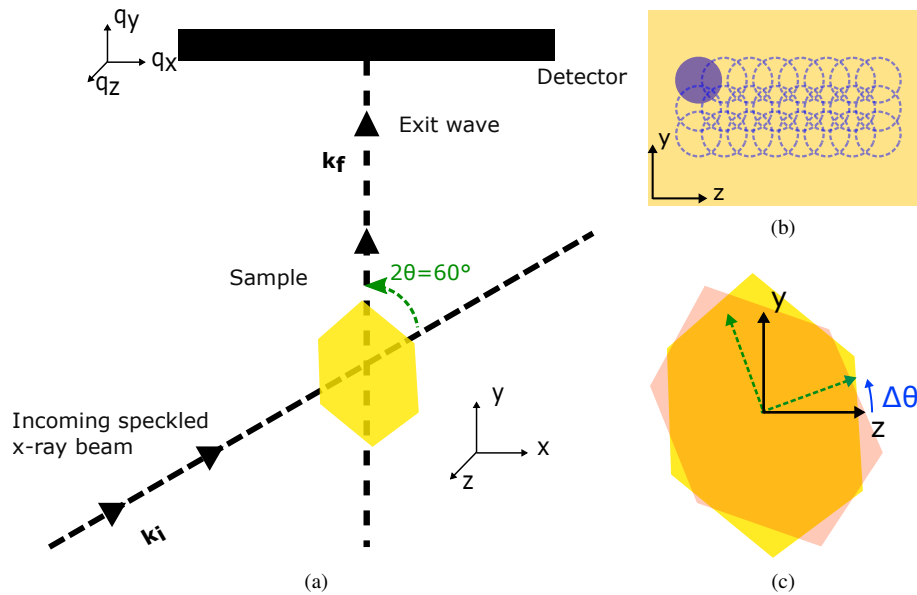

Fig. 5. (a) Simulated experimental geometry for multi-angle Bragg ptychography. The incident ($\mathbf{k}_i$) and exit ($\mathbf{k}_f$) beams define a scanning angle. (b) At each scanning angle, the incident beam is shifted along an overlapping raster grid in the yz plane to generate a set of 2D coherent diffraction patterns. (c) For angular sampling around the Bragg peak (say with $2\theta = 60°$), the sample is rotated through small angles $\Delta\theta$, and the incident and exit beams are modulated correspondingly.

To develop the maBPP forward model, we consider a 3D diffracting crystal volume O and an illumination volume $\mathcal{P}$. The incident probe direction $\mathbf{k}_i$ and the exit beam direction $\mathbf{k}_f$ satisfy the Bragg diffraction condition when the scattering vector $\mathbf{q} = \mathbf{k}_f - \mathbf{k}_i$ coincides with some reciprocal lattice vector $\mathbf{G}_{hkl}$ for the crystal. At the Bragg condition, if the probe volume interacts with the slice of the object indicated by the lateral shift operator $\mathcal{S}_j$, then the 2D exit wave ψ_j is calculated as [64]

$$\psi_j = \mathcal{R}\mathcal{P} \odot (\mathcal{S}_j O) \qquad (16)$$

where $\mathcal{R}$ is the matrix operator performing the projection along the exit beam direction.

When the object is rotated by a small angle $\Delta\theta_j$, the scattering vector deviates from the Bragg condition by $\mathbf{Q}_j = \mathbf{q}_j - \mathbf{G}_{hkl}$. The corresponding change in the object-probe interaction can be encoded in terms of a phase shift operator defined as $\mathfrak{Q}_j = \exp\left(i\mathbf{r} \cdot \mathbf{Q}_j\right)$. The 2D exit wave is then given by [63]

$$\psi_j = \mathcal{R}\mathfrak{Q}_j \odot \left(\mathcal{P} \odot (\mathcal{S}_j O)\right), \tag{17}$$

and the expected intensity at the detector plane is then

$$h_j = \left\|\mathcal{F}\psi_j\right\|^2 + v_j = \left\|\mathcal{F}\mathcal{R}\mathfrak{Q}_j \odot \left(\mathcal{P} \odot (\mathcal{S}_j O)\right)\right\|^2 + v_j, \tag{18}$$

where v_j accounts for the background events. The experimentally measured intensities are denoted by y_j.

We can again adapt the forward model of the far-field ptychography experiment to the multi-angle Bragg experiment by generalizing the probe and object models to 3D, and applying the angle-dependent phase shifts $\mathfrak{Q}_j$ along with the projection operator $\mathcal{R}$. This 3D generalization comes at the cost of a dramatic increase in the parameter space, and the reconstruction becomes correspondingly more difficult. With this in mind, for ease of reconstruction we use a speckle pattern, which was found to help in solving the phase retrieval problem for 2D CDI [65, 66], as a highly diverse structured probe illumination. We also impose the commonly applied [64] additional constraints on the reconstruction problem: we assume that the probe structure and the object thickness (along the x−direction) are known in advance.

For the multi-angle Bragg ptychography experiments, we simulated the object as a compact crystal represented by set of grid points inside a faceted volume. We set the interior points to have magnitude 0.5 and a spatially varying phase structure emulating a strain field. We defined the orthonormal real space axes (x, y, z) such that each object voxel is of size $66 \times 66 \times 66$ nm^3, and such that the polyhedral crystal is situated obliquely within (with $\approx 32\%$ of the volume of) a cuboid of size $26 \times 56 \times 50$ voxels ($\approx 25 \ \mu m^3$). The cuboid was itself placed at the center of a simulation box of total size $64 \times 162 \times 112$ voxels ($\approx 334 \ \mu m^3$). A 64×64 pixel detector with a pixel pitch of 55 μm was placed at a distance of 1.5 m from the object, normal to the real space y-axis. The probe was initialized as Gaussian beam of energy 8.7 keV with an FWHM of 396 nm (6 pixels) contained entirely within a 64×64 pixel array, then passed through a diffuser and modeled as a speckle pattern. The probe was then interpolated to approximate the incident beam at a Bragg condition of $2\theta = 60°$ (Fig. 5) such that $\mathbf{k}_i \perp z$ and $\mathbf{k}_f \parallel y$. The beam profile was assumed to be constant along the propagation direction during its propagation through the simulation box. To obtain the ptychographic data set, we applied phase modulations corresponding to an angular shift between $\pm 0.14°$ around the Bragg condition using an angular step size of $\Delta\theta = 0.02°$, for a total of 15 angles. At each such angle, we translated the probe along the y and z directions with a step size of 132 nm (2 pixels) in a raster grid of 41×24 scan positions, *i.e.* with an overlap of $\approx 86\%$ per step in the y direction and $\approx 83\%$ per step in the z direction. This generated a total of 14760 diffraction patterns, to each of which we added Poisson noise, which gave us a data set with an overall intensity maximum of 13560 photons/pixel.

During the reconstruction, we solved for a volume of size $30 \times 78 \times 82$ voxels, initialized as a random complex array, and placed centrally in the simulation box. The object support along the x−direction is set to be slightly ($\approx 15\%$) larger than the actual object thickness. This allows for the expected spreading of the object induced by resolution effects due to the low photon count [67]. To solve for the object structure, we again adapted the approach outlined in Algorithm 1 to use the Adam optimizer with the initial update step size 0.01, and with minibatches each containing 400 diffraction patterns selected in a stochastic fashion irrespective of either probe position or object rotation. The minibatch size and initial update step size were chosen as described in Section 3.4. After 70 iterations of Adam gradient descent, we obtained a reconstruction of the crystal (and the
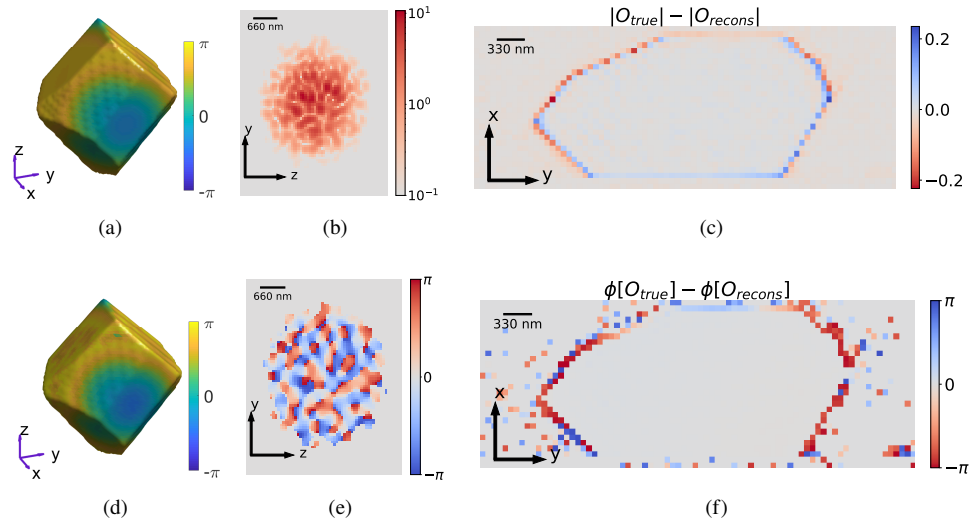
Fig. 6. Multi-angle Bragg ptychographic reconstruction with reverse-mode AD. (a) True and (d) reconstructed object structure with surface phase variation. YZ cross-section of the (b) probe magnitude and (e) phase at $x = 32$. (c) XY cross-section of the error in the reconstructions of the object (c) magnitude and (f) phase at $z = 41$. For the sake of clarity, the error in the phase is set to 0 wherever $\|O_{recons}\| < 0.01$. The normalized error (NRMSE) for the 3D reconstruction is 0.09.

loose support) shown in Fig. 6. The overall normalized reconstruction error, calculated using a 3D adaptation of the sub-pixel registration algorithm presented in [55], was found to be 0.09, which is much larger than the NRMSE observed for the 2D objects in Sections 3.3 and 4.1. As we can see in Fig. 6, the error in the reconstruction is primarily due to discrepancies at the object edges—this is a physical effect that can be attributed to the presence of shot noise that corrupts the diffraction data [67]. This results demonstrates that the reconstruction accurately captures the physics of the experiment.

5. Conclusion

Our results in this paper demonstrate that the automatic differentiation technique can be used to construct a general gradient-based inversion framework for ptychographic phase retrieval. Specifically, we have shown that we can minimize a ptychographic error metric by first using reverse-mode AD to numerically calculate the gradients of the error metric, then using these within a state-of-the-art adaptive gradient descent algorithm. We can thereby solve the ptychographic problem without ever performing an analytical derivation for any closed-form gradient expression. This inversion framework is robust to changes in the forward model and can be used identically for phase retrieval in such varying experimental configurations as far-field transmission ptychography, near-field ptychography, and multi-angle Bragg ptychography.

The inversion framework showcased in this work does not rely on any particular choice of the error metric or the experimental configuration. We can change the error metric either to accommodate alternate noise models, or to incorporate *a priori* knowledge about the experiment through the choice of a regularizer. We can tailor the experimental setup and introduce new degrees of freedom when necessary, then simply change the forward model correspondingly.

Additionally, by using the state-of-the-art TensorFlow library for the gradient calculations, we gain in-built scalability and parallelization (multi-CPU/GPU architectures), thus allowing for convenient phase retrieval for experimental models large and small. In the future, we aim to leverage these flexibilities to incorporate a variety of physical phenomena–such as probe fluctuations, positional inaccuracies, and limitations in the detection process–into the phase retrieval framework. This would enable high-resolution phase retrieval from near-field/far-field 3D (or 2D) Bragg and non-Bragg experiments, all within a single consistent platform.

The flexible AD-based inversion framework is expected to provide a unified approach to phase retrieval for general (ptychographic and non-ptychographic) CDI experiments with x-rays, electrons, or optical waves, even as we move towards increasingly complex imaging regimes. This would potentially give researchers the ability to explore variations of basic CDI methodologies in a convenient and straightforward manner, and could thereby prove a powerful addition to the phase retreiver's toolbox.

A. Parameter update with the Adam optimizer

To examine how the Adam optimizer performs the parameter updates, we consider the far-field transmission ptychography setting from Section 3.1, with the error metric $g(O, \mathcal{P})$ which is to be minimized with respect to the object parameter O. The optimization is performed in the minibatch setting described in Algorithm 1, wherein the partial derivative at the kth step, $\partial_O g^k$, is calculated using Eq. (8).

The Adam optimization step consists of a parameter initialization step and a variable update step. The initialization step, which precedes the gradient descent iterations, is outlined in Algorithm 2a. The parameter update step in Algorithm 2b replaces Steps 4 and 5 in the iterative descent setting outlined in Algorithm 1. Typical out-of-the-box implementations of the Adam optimizer only support gradient descent for real-valued variables, in which case the real and imaginary components of the object O are separately updated. To formalize this update step, we can separate the real and imaginary components of $\partial_O g^k$ as

$$\partial_{\Re[O]} g^k = \frac{1}{2} \frac{\partial g^k}{\partial \Re[O]} \quad \text{and} \quad \partial_{\Im[O]} g^k = \frac{1}{2} \frac{\partial g^k}{\partial \Im[O]} \tag{19}$$

such that $\partial_O g^k = \partial_{\Re[O]} g^k + i \partial_{\Im[O]} g^k$. We can now see that the first and second moment parameters created during the initialization step for $\Re[O]$ (and similarly for $\Im[O]$) are subsequently updated alongside the object variable throughout the optimization process. These updates depend on both the current gradient value and an exponentially weighted accumulation of all past gradient values, thereby adapting the magnitude of direction of the parameter update at every step. For an appropriately chosen initial step size α_o^A, updating the optimization parameter in this fashion leads to fast and stable convergence [48].

Algorithm 2a Adam object initialization

Require: Initial Adam step size α_o^A.

1: Initialize the moment vectors

$$
\begin{aligned}
\left.
\begin{aligned}
m_{\Re[O]}^0 &\longleftarrow 0 \\
m_{\Im[O]}^0 &\longleftarrow 0
\end{aligned}
\right\} \quad &\text{First moments,} \\
&\quad m_{\Re[O]}, m_{\Im[O]} \in \mathbb{R}^N \\[6pt]
\left.
\begin{aligned}
v_{\Re[O]}^0 &\longleftarrow 0 \\
v_{\Im[O]}^0 &\longleftarrow 0
\end{aligned}
\right\} \quad &\text{Second moments,} \\
&\quad v_{\Re[O]}, v_{\Im[O]} \in \mathbb{R}^N
\end{aligned}
$$

2: Initialize the exponential decay rates with their recommended default values [48]:

$$\beta_1 = 0.9 \qquad \text{(Decay rate for } m_{\Re[O]}, m_{\Im[O]})$$

$$\beta_2 = 0.999 \qquad \text{(Decay rate for } v_{\Re[O]}, v_{\Im[O]})$$

$$\epsilon = 10^{-8} \qquad \text{(Constant used to avoid division by zero)}$$

Algorithm 2b Adam object update (at step k)

Require: Object estimate O^{k-1} and the partial derivative $\partial_o g^k$.

1: Update the first and second moments:

$$
m_{\Re[O]}^k = \frac{\beta_1 m_{\Re[O]}^{k-1} + (1-\beta_1)\partial_{\Re[O]} g^k}{1 - (\beta_1)^k}
$$

$$
v_{\Re[O]}^k = \frac{\beta_2 v_{\Re[O]}^{k-1} + (1-\beta_2)\left\|\partial_{\Re[O]} g^k\right\|^2}{1 - (\beta_2)^k}
$$

(and similarly for $m_{\Im[O]}^k$ and $v_{\Im[O]}^k$).

2: Update the object estimate:

$$
\Re[O]^k = \Re[O]^{k-1} - \alpha_o^A \frac{m_{\Re[O]}^k}{\sqrt{v_{\Re[O]}^k} + \epsilon}
$$

$$
\Im[O]^k = \Im[O]^{k-1} - \alpha_o^A \frac{m_{\Im[O]}^k}{\sqrt{v_{\Im[O]}^k} + \epsilon}
$$

References

1. W. Hoppe, "Beugung im inhomogenen Primarstrahlwellenfeld. III. Amplituden-und Phasenbestimmung bei unperiodischen Objekten," Acta Crystallogr. A **25**, 508–514 (1969).
2. J. M. Rodenburg and H. M. Faulkner, "A phase retrieval algorithm for shifting illumination," Appl. Phys. Lett. **85**, 4795–4797 (2004).
3. H. M. L. Faulkner and J. Rodenburg, "Movable aperture lensless transmission microscopy: A novel phase retrieval algorithm," Phys. Rev. Lett. **93**, 023903 (2004).
4. M. Guizar-Sicairos and J. Fienup, "Phase retrieval with transverse translation diversity: a nonlinear optimization approach," Opt. Express **16**, 7264–7278 (2008).
5. A. M. Maiden and J. M. Rodenburg, "An improved ptychographical phase retrieval algorithm for diffractive imaging," Ultramicroscopy **109**, 1256–1262 (2009).
6. P. Thibault, M. Dierolf, O. Bunk, A. Menzel, and F. Pfeiffer, "Probe retrieval in ptychographic coherent diffractive imaging," Ultramicroscopy **109**, 338–343 (2009).
7. M. Stockmar, P. Cloetens, I. Zanette, B. Enders, M. Dierolf, F. Pfeiffer, and P. Thibault, "Near-field ptychography: phase retrieval for inline holography using a structured illumination," Sci. Rep. **3**, 1927 (2013).
8. P. Godard, G. Carbone, M. Allain, F. Mastropietro, G. Chen, L. Capello, A. Diaz, T. Metzger, J. Stangl, and V. Chamard, "Three-dimensional high-resolution quantitative microscopy of extended crystals," Nat. Commun. **2**, 568 (2011).
9. A. M. Maiden, M. J. Humphry, and J. M. Rodenburg, "Ptychographic transmission microscopy in three dimensions using a multi-slice approach," J. Opt. Soc. Am. A **29**, 1606–1614 (2012).
10. M. Dierolf, A. Menzel, P. Thibault, P. Schneider, C. M. Kewish, R. Wepf, O. Bunk, and F. Pfeiffer, "Ptychographic x-ray computed tomography at the nanoscale," Nature **467**, 436–439 (2010).
11. M. A. Gilles, Y. S. G. Nashed, M. Du, C. Jacobsen, and S. M. Wild, "3D x-ray imaging of continuous objects beyond the depth of focus limit," Optica **5**, 1078–1085 (2018).

12. R. H. T. Bates, "Fourier phase problems are uniquely solvable in more than one dimension. I. Underlying theory," Optik **61**, 247–262 (1982).
13. R. W. Gerchberg and W. O. Saxton, "A practical algorithm for the determination of phase from image and diffraction plane pictures," Optik **35**, 237–246 (1972).
14. J. R. Fienup, "Phase retrieval algorithms: a comparison," Appl. Opt. **21**, 2758–2769 (1982).
15. V. Elser, "Phase retrieval by iterated projections," J. Opt. Soc. Am. A **20**, 40–55 (2003).
16. S. Marchesini, "A unified evaluation of iterative projection algorithms for phase retrieval," Rev. Sci. Instruments **78**, 011301 (2007).
17. J. Clark, X. Huang, R. Harder, and I. Robinson, "High-resolution three-dimensional partially coherent diffraction imaging," Nat. Commun. **3**, 993 (2012).
18. P. Thibault and A. Menzel, "Reconstructing state mixtures from diffraction measurements," Nature **494**, 68–71 (2013).
19. A. Tripathi, I. McNulty, and O. G. Shpyrko, "Ptychographic overlap constraint errors and the limits of their numerical recovery using conjugate gradient descent methods," Opt. Express **22**, 1452–1466 (2014).
20. P. Dwivedi, A. Konijnenberg, S. Pereira, and H. Urbach, "Lateral position correction in ptychography using the gradient of intensity patterns," Ultramicroscopy **192**, 29–36 (2018).
21. P. Thibault and M. Guizar-Sicairos, "Maximum-likelihood refinement for coherent diffractive imaging," New J. Phys. **14**, 063004 (2012).
22. P. Godard, M. Allain, V. Chamard, and J. Rodenburg, "Noise models for low counting rate coherent diffraction imaging," Opt. Express **20**, 25914–25934 (2012).
23. A. S. Jurling and J. R. Fienup, "Applications of algorithmic differentiation to phase retrieval algorithms," J. Opt. Soc. Am. A **31**, 1348–1359 (2014).
24. E. J. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via Wirtinger flow: theory and algorithms," IEEE Trans. Inf. Theory **61**, 1985–2007 (2015).
25. H. Zhang and Y. Liang, "Reshaped wirtinger flow for solving quadratic system of equations," Adv. Neural. Inf. Process. Syst. **29**, 2622–2630 (2016).
26. Z. Wei, W. Chen, C.-W. Qiu, and X. Chen, "Conjugate gradient method for phase retrieval based on the Wirtinger derivative," J. Opt. Soc. Am. A **34**, 708 (2017).
27. J. Zhong, L. Tian, P. Varma, and L. Waller, "Nonlinear optimization algorithm for partially coherent phase retrieval and source recovery," IEEE Trans. Comput. Imaging **2**, 310–322 (2016).
28. J. Li and T. Zhou, "Numerical optimization algorithms for wavefront phase retrieval from multiple measurements," Inverse Probl. & Imaging **11**, 721–743 (2017).
29. J. Qian, C. Yang, A. Schirotzek, F. Maia, and S. Marchesini, "Efficient algorithms for ptychographic phase retrieval," Inverse Probl. Appl. Contemp. Math **615**, 261–280 (2014).
30. A. W. Yan, A. J. D'Alfonso, A. J. Morgan, C. T. Putkunz, and L. J. Allen, "Fast deterministic ptychographic imaging using x-rays," Microsc. Microanal. **20**, 1090–1099 (2014).
31. L.-H. Yeh, "Analysis and comparison of fourier ptychographic phase retrieval algorithms," Tech. rep., Technical Report No. UCB/EECS-2016-86 (University of California) (2016).
32. A. M. Maiden, D. Johnson, and P. Li, "Further improvements to the ptychographical iterative engine," Optica **4**, 736–745 (2017).
33. R. Hesse, D. R. Luke, S. Sabach, and M. K. Tam, "Proximal heterogeneous block implicit-explicit method and application to blind ptychographic diffraction imaging," SIAM J. Imaging Sci. **8**, 426–457 (2015).
34. A. Griewank and A. Walther, *Evaluating derivatives: principles and techniques of algorithmic differentiation*, vol. 105 (SIAM, 2008).
35. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. J. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Józefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. G. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. A. Tucker, V. Vanhoucke, V. Vasudevan, F. B. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," arXiv preprint arXiv:1603:04467 (2016).
36. A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *31st Conference on Neural Information Processing Systems,* (NIPS, 2017).
37. D. Maclaurin, "Modeling, inference and optimization with composable differentiable procedures," Ph.D. thesis, Harvard University (2016).
38. Y. S. G. Nashed, T. Peterka, J. Deng, and C. Jacobsen, "Distributed automatic differentiation for ptychography," Procedia Comput. Sci. **108**, 404–414 (2017).
39. S. Ghosh, Y. S. Nashed, O. Cossairt, and A. Katsaggelos, "ADP : Automatic differentiation ptychography," in *IEEE International Conference on Computational Photography (ICCP),* (IEEE, 2018), pp. 1–10.
40. A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: a survey," J. Mach. Learn. Res. **18**, 1–43 (2015).
41. P. H. Hoffmann, "A hitchhiker's guide to automatic differentiation," Numer. Algorithms **72**, 775–811 (2016).
42. L. Bian, J. Suo, G. Zheng, K. Guo, F. Chen, and Q. Dai, "Fourier ptychographic reconstruction using wirtinger flow optimization," Opt. Express **23**, 4856–4866 (2015).

43. Z. Wen, C. Yang, X. Liu, and S. Marchesini, "Alternating direction methods for classical and ptychographic phase retrieval," Inverse Probl. **28**, 115010 (2012).

44. J. R. Fienup, T. R. Crimmins, and W. Holsztynski, "Reconstruction of the support of an object from the support of its autocorrelation," J. Opt. Soc. Am. **72**, 610–624 (1982).

45. D. Brandwood, "A complex gradient operator and its application in adaptive array theory," IEE Proc. F-Communications, Radar Signal Process. **130**, 11–16 (1983).

46. K. Kreutz-Delgado, "The complex gradient operator and the cr-calculus," arXiv preprint arXiv:0906.4835 (2009).

47. L. Sorber, M. V. Barel, and L. D. Lathauwer, "Unconstrained optimization of real functions in complex variables," SIAM J. Optim. **22**, 879–898 (2012).

48. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980 (2014).

49. S. Ahn and J. A. Fessler, "Globally convergent image reconstruction for emission tomography using relaxed ordered subsets algorithms," IEEE Trans. Med. Imaging **22**, 613–626 (2003).

50. D. P. Bertsekas, *Nonlinear programming* (Athena Scientific, 1999).

51. Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade,* (Springer, 2012), pp. 9–48.

52. S. Ruder, "An overview of gradient descent optimization algorithms," arXiv preprint arXiv:1609.04747 (2016).

53. E. J. R. Pauwels, A. Beck, Y. C. Eldar, and S. Sabach, "On fienup methods for sparse phase retrieval," IEEE Trans. Signal Process. **66**, 982–991 (2018).

54. R. Xu, M. Soltanolkotabi, J. P. Haldar, W. Unglaub, J. Zusman, A. F. Levi, and R. M. Leahy, "Accelerated wirtinger flow: A fast algorithm for ptychography," arXiv preprint arXiv:1806.05546 (2018).

55. M. Guizar-Sicairos, S. T. Thurman, and J. R. Fienup, "Efficient subpixel image registration algorithms," Opt. Lett. **33**, 156–158 (2008).

56. Y. S. Nashed, D. J. Vine, T. Peterka, J. Deng, R. Ross, and C. Jacobsen, "Parallel ptychographic reconstruction," Opt. Express **22**, 223015 (2014).

57. J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," J. Mach. Learn. Res. **13**, 281–305 (2012).

58. Y. Yao, L. Rosasco, and A. Caponnetto, "On early stopping in gradient descent learning," Constr. Approx. **26**, 289–315 (2007).

59. J. W. Goodman, *Introduction to Fourier Optics* (W.H. Freeman, 2017).

60. M. Stockmar, I. Zanette, M. Dierolf, B. Enders, R. Clare, F. Pfeiffer, P. Cloetens, A. Bonnin, and P. Thibault, "X-ray near-field ptychography for optically thick specimens," Phys. Rev. Appl. **3**, 1–6 (2015).

61. R. M. Clare, M. Stockmar, M. Dierolf, I. Zanette, and F. Pfeiffer, "Characterization of near-field ptychography," Opt. Express **23**, 19728 (2015).

62. A. L. Robisch, K. Kröger, A. Rack, and T. Salditt, "Near-field ptychography using lateral and longitudinal shifts," New J. Phys. **17**, 073033 (2015).

63. M. O. Hill, I. Calvo-Almazan, M. Allain, M. V. Holt, A. Ulvestad, J. Treu, G. Koblmuller, C. Huang, X. Huang, H. Yan, E. Nazaretski, Y. S. Chu, G. B. Stephenson, V. Chamard, L. J. Lauhon, and S. O. Hruszkewycz, "Measuring three-dimensional strain and structural defects in a single InGaAs nanowire using coherent x-ray multiangle Bragg projection ptychography," Nano Lett. **18**, 811–819 (2018).

64. S. O. Hruszkewycz, M. Allain, M. V. Holt, C. E. Murray, J. R. Holt, P. H. Fuoss, and V. Chamard, "High-resolution three-dimensional structural microscopy by single-angle Bragg ptychography," Nat. Mater. **16**, 244–251 (2017).

65. A. Fannjiang, "Absolute uniqueness of phase retrieval with random illumination," Inverse Probl. **28**, 075008 (2012).

66. T. B. Edo, D. J. Batey, A. M. Maiden, C. Rau, U. Wagner, Z. D. Pešić, T. A. Waigh, and J. M. Rodenburg, "Sampling in x-ray ptychography," Phys. Rev. A **87**, 053850 (2013).

67. V. Chamard, M. Allain, P. Godard, A. Talneau, G. Patriarche, and M. Burghammer, "Strain in a silicon-on-insulator nanostructure revealed by 3D x-ray Bragg ptychography," Sci. Rep. **5**, 9827 (2015).